Integrating the Parcel and Mail Network

ERASMUS UNIVERSITEIT ROTTERDAM

Integrating the Parcel and Mail Network

by

M.G. van Doorn

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

Student number:	449573	
Project duration:	April 1, 2022 - October 1, 2022	
Company supervisor:	T. Jonker,	PostNL
Thesis committee:	dr. P.C. Bouman,	Erasmus University, supervisor
	prof. dr. A.P.M. Wagelmans	Erasmus University, second assessor

The views stated in this report are those of the author and not necessarily those of the supervisor, Erasmus School of Economics or Erasmus University Rotterdam.

ERASMUS
UNIVERSITEIT
ROTTERDAM

Abstract

Within the postal and parcel market, the tight labour market, decline in mail volumes, and the strivings to reduce emissions call for innovative supply chain models to maintain an affordable, high-quality network for last-mile delivery. Especially car-delivery areas within mail distribution, pressurise the sustainability of the current supply chain networks. This research explores a new supply chain model for PostNL, a large Dutch postal and parcel company, whereby we integrate mail from car-delivery areas into the parcel network. We build a model to optimise the new delivery areas to test the benefits of the integration. We create a column generation-based approach with two RMP methods to find the best integer solution to the Set Covering constraints and two Pricing methods to construct the new delivery areas. We show that it is best to use an exact RMP model to find the integer solution and a Randomised Constructive heuristic to create the delivery areas. With this combination, we develop new delivery areas. A central finding of the experiments is that at most 0.35 of the total delivery time is spent delivering mail in a combined delivery area. If the fraction increases, it is better to have the mail in a separate mail delivery area. We find substantial savings in our case study of €560 per day on quiet days for a hybrid delivery model in which both integrated and separate delivery is possible. However, on busy days it remains best to deliver both mail and parcel separate. Above all, our Randomised Constructive heuristic creates a basis allowing the generation of all possible delivery areas. We use the delivery areas as input to a commercial solver, which we reason is more effective than a column generation approach.

Preface

I want to use the preface to share my personal development and learnings while writing my thesis. I hope that writing it at the beginning of my thesis can help some of you at the start of writing yours.

My first important lesson is to think before you read. I highly value the written word, especially in the context of academics. Immediately reading about what others did makes your focus shift to the best solution. My advice is first to think for yourself. What are the requirements of the solution? Is there a basic, straightforward approach to get a first answer to your research question? If so, please implement this approach. It shows you what data you need and gives a first answer to your research question. Reading literature afterwards allows you to be more critical about the limitations of the proposed models since you know what you need. Moreover, as companies do not always work with these complicated models, you can show the added value of using advanced models instead of simple calculations.

My second important lesson is that you cannot change the world in half a year and are neither guaranteed to make a significant impact with your outcomes. I think it is better to write your thesis with a development goal instead of a result-oriented goal. Think of what you want to learn and try if you can implement that within the process. I liked to get to know PostNL and understand their view on integrated delivery. The part of my thesis I most enjoyed was talking with different people, from operations to directors, to understand their opinions. Their opinions hardly influenced this thesis, but they taught me how organisations work and what drives people, a valuable insight I will take with me during my career. Another consequence of a development goal is to let easier go of the urge to remove unfavourable experiments. Your approach might not be beneficial, or not in all situations, but that is alright. I believe the actual results are irrelevant to the quality of a thesis.

I did not necessarily enjoy every week of writing this thesis, but I am happy with what I have learned over the past half a year. I hope you will learn something from my thesis as well!

Contents

1 Introduction

PostNL is a large Dutch postal and parcel company which the government has appointed to deliver mail to every resident in the Netherlands. For PostNL, it is essential to have an accessible, reliable and affordable network to deliver mail and parcels. The tight labour market, decline in mail volumes, and the striving of PostNL to reduce emissions pressurise these goals. To maintain its accessible, reliable and affordable network, PostNL is innovative to increase the efficiency of the network. An opportunity to boost efficiency is to keep the mail delivery and parcel delivery no longer separate but to integrate the delivery of mail and parcels. If combining the networks leads to a more efficient network, it resolves some of the challenges brought by the tight labour market and helps reduce the number of daily kilometres, which will reduce delivery costs, thus alleviating the costs of an accessible and reliable network.

To explore this possibility, we need to understand the current delivery structure. We first discuss the current delivery situation in Section 1.1, regarding the mail and parcel network's characteristics and volume developments. We conclude this subsection by identifying the current challenges within the networks. Based on our understanding of the current delivery network and its challenges, we present a new convenient supply chain model for PostNL in Section 1.2, which includes mail and parcel delivery integration. In Section 1.3, we define our main research question to discover whether the new supply chain model leads to an increase in efficiency. To investigate the effects of the new supply chain model, we need to know the division of the new delivery areas. We formulate subquestions for our research, focusing on making the best-suited model to create new delivery areas. Moreover, we define our assessment criteria for measuring the benefits of the new supply chain model.

1.1 State of the art

We first present the current supply chain network for the mail and parcel network. After that, we discuss the volume developments in both business units. We conclude this subsection by identifying the current challenges within the delivery.

1.1.1 Mail delivery network

In the Netherlands, everyone must be able to receive mail, as was agreed in the universal service obligation. PostNL is responsible for carrying out this service obligation in the Netherlands. To do so, PostNL delivers mail five days a week: from Tuesday to Saturday. Another agreement is that everyone must have a letterbox nearby, and 95% of the mail received in these letterboxes by PostNL must be delivered the next day (PostNL).

We now discuss the supply chain that enables PostNL to satisfy the agreements of the universal service obligation. Figure 1 gives a graphical overview of the sorting and distribution process, which we now consider in more detail. The first step to mail delivery is the collection phase, in which PostNL collects all letters and cards from retailers, mailboxes on the street, or specific clients.

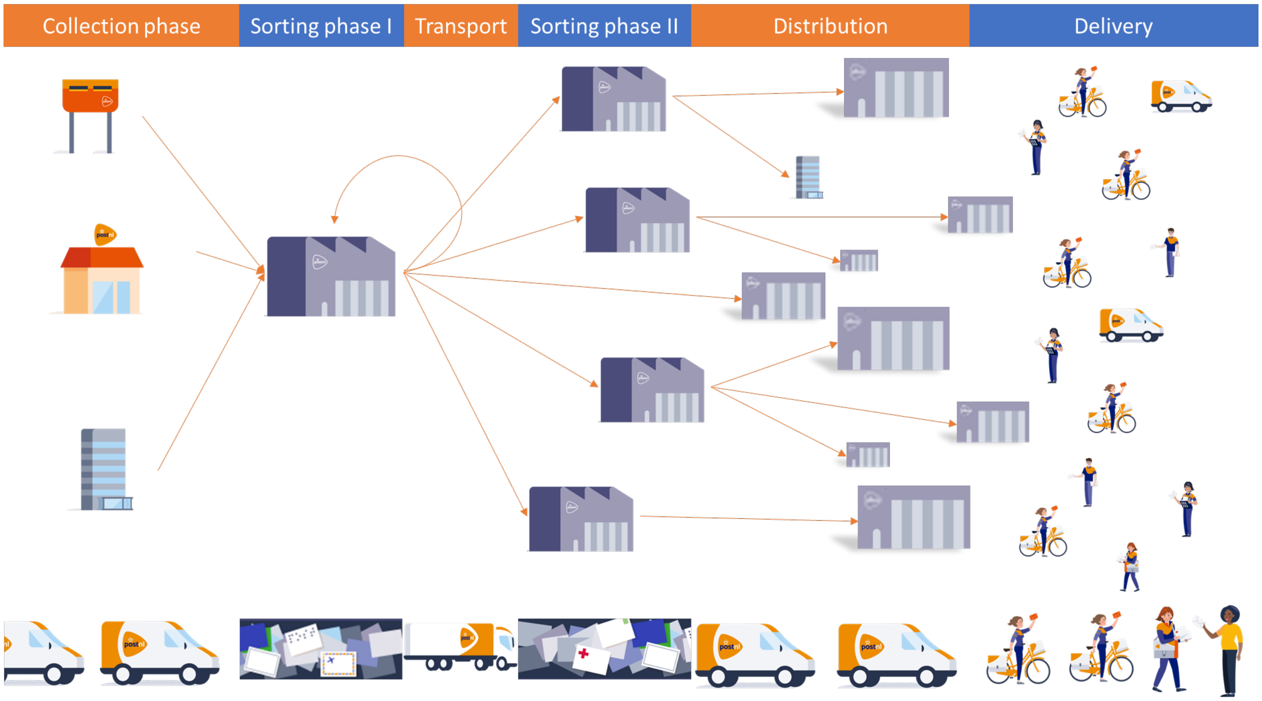

Figure 1: A graphical overview of the mail sorting and distribution process. From left to right, it shows the chronological steps of the mail supply chain. The vehicles indicate the size of the different flows.

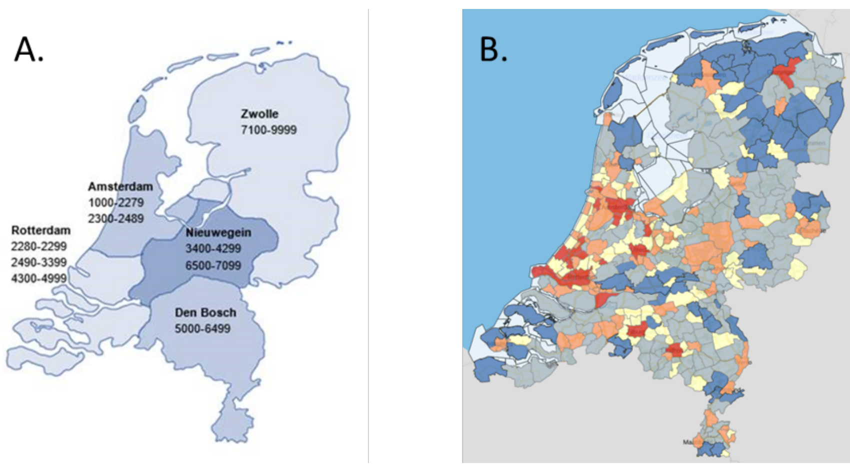

Figure 2: **A.** Overview of which sorting centres deliver which zip codes. **B.** Density of addresses in the Netherlands per municipality. A dark blue colour indicates an average of less than 500 $addresses/km^2$, grey 500-1000 $addresses/km^2$, yellow 1000-1500 $addresses/km^2$, orange 1500-2500 $addresses/km^2$ and red 2500 $addresses/km^2$ or more (CBS (2021)).

The collectors bring the mail to the nearest sorting centre for mail, of which there are five in the Netherlands. In two sorting rounds, the mail is in the correct order for the delivery tour of every mail deliverer. The first shift sorts the letters and cards into five groups, one for each sorting centre. Each sorting centre services a different range of zip codes; see Figure 2A. After PostNL transports the mail to the destination sorting centre, the second round of sorting starts. Within two iterations of sorting, the mail is in the correct order for every mail deliverer. Part of the mail sent is not suitable for the sorting machines and is sorted (partly) by hand and added later. The next step is distribution, where some mail is delivered to post office boxes. Most mail goes to one of the approximately 2500 small depots, often garage boxes, where deliverers start their delivery.

A characteristic of mail delivery is its high delivery density. The mail delivery network contains many depots, on average one depot per village or city, to fit the network to that characteristic. Due to this structure, distances between deliveries and haul distances, the distance to get to the delivery area, are short. In densely populated areas, these distances are even shorter. Mail is therefore often delivered by foot or bike. Mail deliverers provide 92% of the addresses by bike or foot, 5% by mopeds, and the final 3% by car.

PostNL decides the best vehicle for the delivery area based on the address density. As can be seen in Figure 2B from CBS (2021), the address density in the Netherlands differs, ranging from less than 500 addresses per square kilometre to more than 2500 addresses per square kilometre. The north and northeast of the Netherlands contain mostly less than 500 addresses per square kilometre. These sparsely distributed areas are often delivered by car as it takes relatively much time to get to the following delivery location. Despite the use of cars, it takes more time and is more expensive to travel to the next location than for bike, foot and moped delivery areas. The car delivery areas are, therefore, the least efficient delivery areas.

1.1.2 Parcel delivery network

The structure of the parcel delivery network is similar to the mail delivery network, but both networks are distinct. Parcel delivery starts with collecting parcels from retailers, specific clients, the international stream and via the fulfilment centre of PostNL; this is an automatic location that prepares the order for the customer.

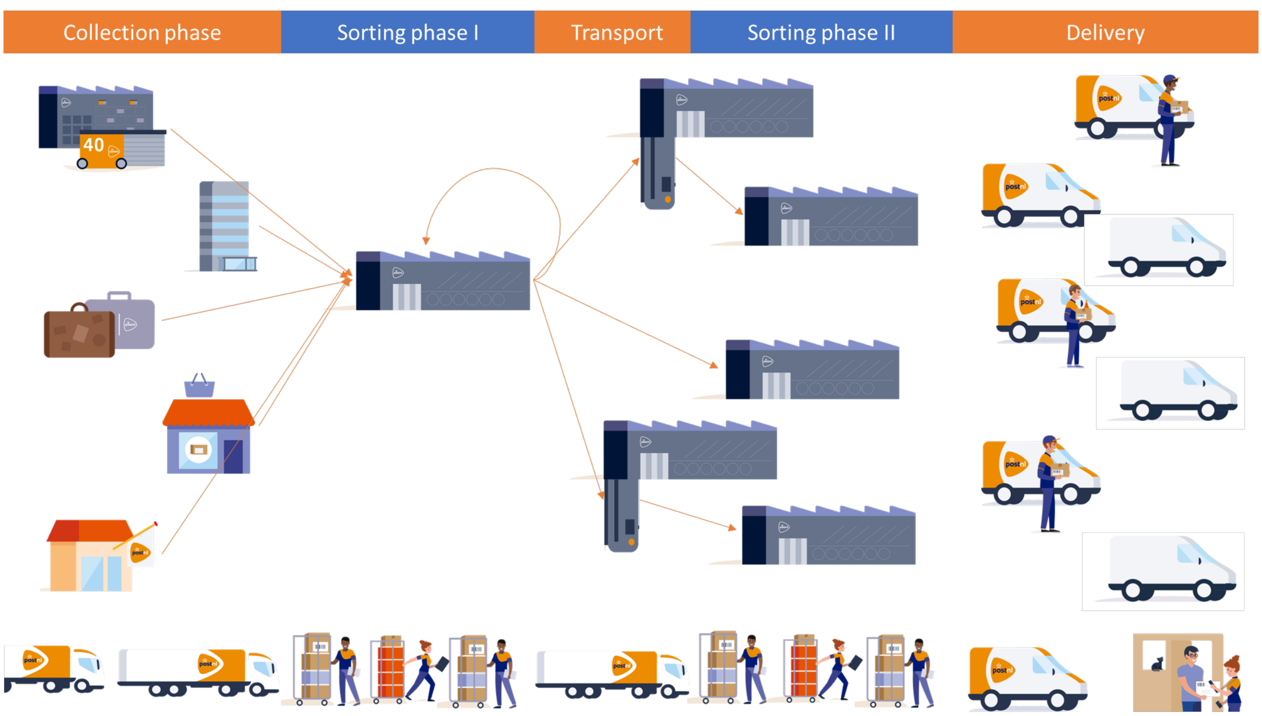

Figure 3: A graphical overview of the sorting and distribution process of the parcels. From left to right, it shows the chronological steps of the parcel supply chain. The vehicles indicate the size of the different flows.

PostNL brings all these parcels to the nearest depot, sorting them based on their destination. Trucks bring the roll containers with boxes to the destination depot, sometimes with an extra stop at the cross-dock to merge smaller parcel streams with the same destination. At the destination depot, PostNL sorts the parcels per route. The parcel deliverer places the sorted packages himself inside the van and starts his delivery route after that. Figure 3 summarises the sorting and distribution process by a graphical overview. Another characteristic of the parcel delivery network is that subcontractors deliver part of the parcels. These are people or small companies hired by PostNL to take care of the delivery of parcels. As PostNL does not employ these deliverers, PostNL has no right to have complete control over their work activities. Therefore PostNL is not allowed to create routes for these deliverers but only to assign delivery regions to them. This freedom, together with the widespread working times, make that the parcel deliverers get a higher wage than mail deliverers.

Parcels often include large volumes, and the network is about eight times as sparse as the mail delivery network. Due to these characteristics, only about 30 parcel depots, large buildings near the highway, are currently used. The most significant difference compared to the mail network is that the distribution starts directly at these depots without an extra distribution step. With only these few depots, the haul distances in the parcel delivery network are significantly higher than in the mail delivery network. Therefore, PostNL always delivers parcels by vans. The van's volume and deliverer's working hours can limit the number of packages assigned to each van.

1.1.3 Volume developments

We know the main characteristics of the supply chain model for both mail and parcel. We now describe the volume developments for both networks. We first discuss the developments for the volumes of parcel, after that for mail.

E-commerce had been growing steadily over the last couple of years but increased drastically with the start of the Covid-19 pandemic. Online ordering became the sole option as all non-essential shops had to close consecutively for at least a couple of months. The closure forced smaller stores to establish or extend their webshop, enabling more online shopping opportunities. PostNL expects that this change in the behaviour of customers is not only temporary but will have a lasting impact. 85% of the growth increase is estimated to be due to recurring orders. The increase in online shopping further enhanced the growth of parcel volumes over the last two years, as seen in Figure 4A. The normalised earnings before income and taxes of parcels increased by 10.0%, from €209 million in 2020 to €230 million in 2021 (PostNL (2022)).

Covid-19 also had an impact on the mail delivery by PostNL. With the limitations on physical contact, people sent more cards to show attentiveness or support. The corona pandemic, combined with the elections, led to a non-yearly recurrent increase in government mail, like vaccination invitations, voting passes and many self-tests. These effects mainly opposed the overall decline in mail volume, and the volume stayed approximately constant. PostNL assumes it cannot account on 70 million of these items for next year and expects the decline in volume to continue in the coming year. In 2019 PostNL took over Sandd, which led to complete coverage of the postal market by PostNL and an increase in volume. Figure 4B shows the evolution of the volume of mail. The normalised earnings before income and taxes of mail in the Netherlands increased from €96 million to €160 million in 2021, a growth of 66.7% (PostNL (2022)).

We observe two trends, leaving the effects of incidental changes out of consideration, the number of parcels delivered increases, and the number of mail delivered decreases. Although it is unclear for what time frame these trends will continue, PostNL expects them to persevere in the coming years.

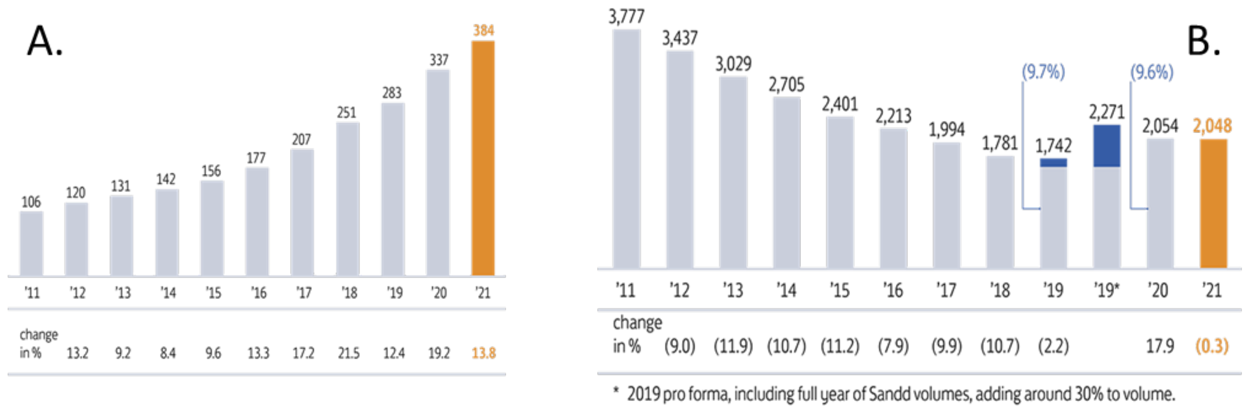

Figure 4: A. Volume development of parcels in the Netherlands in millions. **B.** Volume development of mail in the Netherlands in millions.

1.1.4 Challenges within the networks

In Section 1.1.1 and Section 1.1.2, we discuss how PostNL optimises both networks concerning the characteristics of the items that need to be delivered. Mail has a compact network with depots in almost every village as volumes are low and the delivery density is very high. The number of parcels is lower than mail, and the volume it comprises is much higher. Therefore, faster vehicles with more storage space are used, making the haul distance less critical. Combining this with the scarcity of space in urban areas leads to the conclusion that having a few large depots is better than a dense network for parcel delivery.

We identify a decrease in mail quantities in Section 1.1.3, which indicates that the current network of PostNL will be less suitable in the future to ensure efficient mail delivery. Several technical improvements are made to enhance tracking and make the delivery routes more efficient. Despite these efforts, having an affordable network remains challenging, especially in car delivery areas of the mail network.

1.2 Integrating the mail and parcel network at supply chain level

One promising solution for the pressurised efficiency within the car delivery areas of mail is to merge the mail delivered by car into the parcel network and have only one deliverer for both mail and parcels within a region. With the integration realised, PostNL visits each street with only one vehicle. Integrating the parcel and mail delivery networks requires a new network that is optimal for the characteristics of the combined delivery items. For a new optimal network, the logistic infrastructure has to change considerably.

This research is pioneering in exploring the effects of combining both networks. PostNL needs to validate the outcomes in real life and test for unseen problems before they can adjust the networks on a large scale. We consider all long-term infrastructure, like depots, fixed to facilitate small-scale implementation. Moreover, we assume the collection and sorting processes to remain independent. However, we assume it is possible to restructure them to have the mail and parcel sorted when needed for integration. If combining the networks delivers the promise to improve the efficiency of PostNL, new research is required to discover the best ways to realise this assumption.

As we assume a fixed long-term infrastructure, we have to decide the best integration approach within the current infrastructure. The current size of most mail depots is insufficient to serve as a storage space for parcels. As the parcel network handles high volumes, it can also store at least parts of the mail volumes. We use the parcel network as a primary network based on these volume limitations, which is in line with the increase in parcel volumes and decline in mail volumes. Mail is sorted per parcel deliverer and delivered at the parcel depot. As the parcel deliverer already has to load his van himself, he can easily load the mail as well. The parcel deliverer can place the mail on the front seat so he can quickly get the mail during delivery without opening the back door. Figure 5 schematically shows the assumed integration whereby PostNL brings the mail sorted per route to the parcel depot, where the parcel deliverers start their combined delivery. Within this research, we investigate this newly proposed supply chain model. In this new model, we connect the sorting phase of the mail delivery network to the distribution phase of the parcel delivery network. In Section 1.3, we explain how we analyse and asses this new supply chain model. It is out of scope of this work to deal with the issue that PostNL cannot give subcontractors an exact route to follow, whereas they can provide it to mail deliverers.

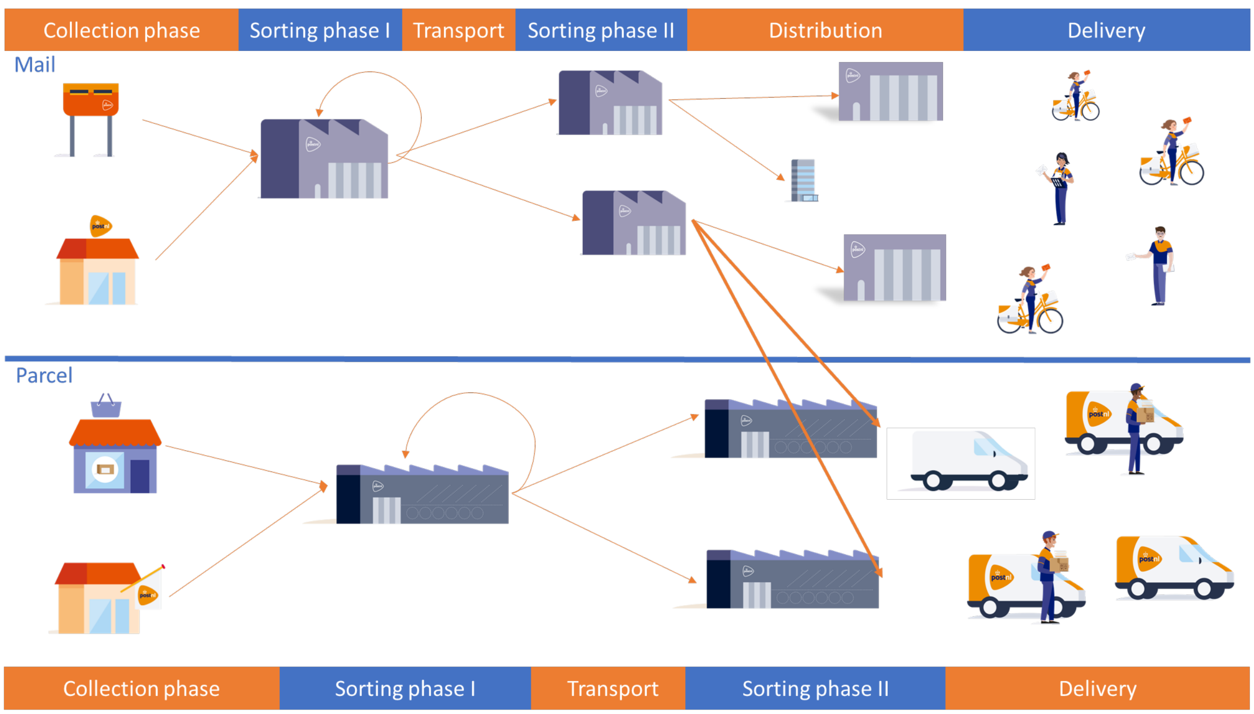

Figure 5: A schematic overview of the level at which we effectuate the integration of parcel and mail. After completing Sorting phase II of the supply chain of mail, PostNL distributes the mail to the parcel depot, where it is integrated in Sorting phase II such that the parcel deliverer can take it with him.

1.3 Research question

We have just explained the level at which we realise the integration in the supply chain, and we can now formulate our main research question:

Is our new supply chain model for mail from car-delivery areas beneficial for PostNL in the last-mile delivery?

To measure the benefits, we need to know how the delivery areas for the parcel deliverers change. As it is part of the freedom of a parcel deliverer to determine his delivery round, PostNL only influences the delivery area. When we investigate the effects of the integration of both networks, we explore a hypothetical change. To answer our main research question, we build a model to create new delivery areas, which we can then analyse. We get to this model by answering the following subquestions of our research question:

1. **Which methods exist in the literature to create new optimal delivery areas?** We investigate the state-of-the-art literature in Section 2 regarding this topic; Set Covering models, Capacitated Clustering models, Continuous Approximation and conclude how we can use them.

2. **How do we formally define the problem of creating new last-mile delivery areas?** Based on the literature, we know how to make the new delivery areas, but they also need to be feasible for the situation of PostNL. In Section 3, we explicitly define our problem of creating new delivery areas. We combine constraints by PostNL, like that a day's work must fit in one vehicle and conditions to enhance the implementation as all delivery areas must be continuous.

3. **How do we match the methods from the literature with our problem?** With inspiration from the literature and our precise problem formulation in mind, we define in Section 4 which methods we use to create delivery areas. As we find several promising models, we implement a column generation approach that can use either an exact model or a heuristic approach to get an integer solution (RMP methods) and create new delivery regions (Pricing methods).

4. **What model best suits the situation of PostNL?** The size and structure of the data and the time available to solve the model can highly influence the quality of the model; we need to test which model is best suited for the situation of PostNL. In Section 5, we investigate the four different combinations of models and conclude that the exact RMP model is best for solving the RMP and the Randomised Constructive heuristic to solve the Pricing Problem. After tuning the parameters, we use this model to develop the new delivery areas, which we analyse to answer our main research question.

5. **What is the quality of the model?** Before we answer our main research question, we discuss the quality and reliability of our model in Section 6. We mention several limitations, considering the approximations and assumptions we make earlier. We also place our work within the theoretical framework.

Once we have an answer to all questions stated above, we finally answer our main research question in Section 7. We measure the possible benefits at three different levels (1) human resources, (2) environmental benefits and (3) financial benefits, stated at reciprocal importance. We explain what we mean by each of these terms below.

1. **Human resources:** The changes in employment measure the human resource impact. We quantify this by changes in full-time equivalents of mail and parcel deliverers. More efficient use of human resources will resolve some of the challenges encountered by the tight labour market.

2. **Environmental benefits:** The change in expected, driven kilometres for delivery measures the environmental efficiency. PostNL strives to be more sustainable and constantly looks for opportunities to operate more efficiently and not clock up unnecessary kilometres. Moreover,

reducing emissions is important for reaching sustainable development goals 9 and 15, Industry, Innovation and Infrastructure and Climate Action, respectively.

3. **Financial benefits:** The financial benefits include a combination of the monetised environmental benefits and human resource impact. As financial benefits are an integral measure of benefits, it is the most important.

Based on our findings, we show PostNL that there is potential in integrating both delivery networks. A substantial saving can be realised on quiet days like Saturdays, whereas we calculate a loss for integration on crowded days like Tuesdays. Promising is our rule-of-thumb that a deliverer should spend no more than 35% of delivery time in a combined delivery area on delivering mail. With this rule in mind, we advise PostNL to check if parcel deliverers currently have time left on Saturdays, which they can use for mail delivery.

Moreover, we discuss in Section 6 that our work can be seen as a critical review of the work of Bard and Jarrah (2013). In comparison to Bard and Jarrah (2013), we can generate a lower bound to compare our algorithm's quality and test one of our, and theirs, assumptions' impact. Both show the algorithm's quality can and must be improved to be useful as a delivery area creating model.

2 Literature review

In our search to create new optimal delivery areas, we start with the investigation of relevant literature. This section discusses relevant research based on the topic or approach to creating these delivery regions. We first discuss the paper of Winkenbach et al. (2016) in Section 2.1, who, to the extent of our knowledge, are the only ones who have already investigated the merging of mail and parcel networks. Because of this significant similarity, we discuss their work separately.

Afterwards, we discuss the Capacitated Clustering Problem (CCP) in Section 2.2. The CCP is a problem whereby the target is to create clusters such that they satisfy a capacity constraint. The CCP relates closely to our problem; our model should create new delivery areas satisfying capacity constraints regarding the allowed volume and working time for deliverers.

Besides the CCP, we also investigate the Set Covering Problem (SCP) in Section 2.3. The goal of the SCP is to find a set of clusters that cover all locations. The SCP is applied to various contexts like line balancing production, crew scheduling and service installation, according to Crawford et al. (2018). The SCP can easily be applied to our problem, as we need to be able to deliver mail and parcels at all locations. Both the CCP and the SCP are NP-complete (Öncan (2007), Chvatal (1979)). As many data points are involved in our problem, we expect that it is computationally intensive to solve the problem exactly. To reduce computation time, we dive into the existing heuristic approaches.

Besides the heuristic approaches, we also investigate Continuous Approximation (CA) in Section 2.4 as another approach to efficiently deal with the vast amount of data. We can use CA to describe the properties of the data used within our model and the delivery areas afterwards.

Finally, we discuss the literature's relevance for our problem in Section 2.5. We conclude that the work of Bard and Jarrah (2013) is most relevant for us from the literature regarding the CCP and Lan et al. (2007) concerning the SCP. We use CA to evaluate the properties of the delivery areas after creation for comparison.

2.1 Integration of the mail and parcel network

The research of Winkenbach et al. (2016) focuses on the situation of the French national postal operator, La Poste. At La Poste, mail and parcel are already delivered together within rural areas, and they investigate if it is cost-efficient to merge both networks for medium-sized cities. They first create clusters of 500 x 500 m^2 and capture the delivery characteristics of the locations included through CA. Afterwards, they create a Mixed Integer Linear Program to solve a location routing problem. A location routing problem determines both the optimal delivery route and the best location of the depots. Their model sees the 500 x 500 m^2 clusters as delivery points, not the individual addresses. Based on the city of Nantes, the results show that it leads to a cost saving of three per cent if both networks are combined entirely.

2.2 Capacitated Clustering Problem (CCP)

The CCP was first introduced by Fisher and Jaikumar (1981) to solve a problem similar to our problem: to create the delivery areas and not the specific delivery routes. Fisher and Jaikumar (1981) want to solve the exact routing problem but create a two-step approach to reduce computation time. As a first step, they make clusters of demand locations whereby the vehicles' capacity is sufficient to deliver to all locations, later known as the Capacitated Clustering Problem. The second step is constructing the optimal delivery route for each cluster. Although clustering is only half of the work in this heuristic, it is an NP-hard problem to create optimal capacitated clusters and finding a feasible assignment is already NP-complete (Öncan (2007)).

Another problem similar to the CCP exists in the literature: the Generalised Assignment Problem. Within this problem, the aim is to assign all locations to a specific agent (in our case, a vehicle). According to Mahmoodi Darani et al. (2016), the main difference is how the locations used as a reference for the assignment, the seed locations, are determined. A researcher predetermines the seed locations for a GAP, whereas the algorithm chooses the seed locations for a CCP. However, this distinction does not conform with the reviewed literature. We treat both problems with predetermined seed locations or freely chosen seed locations as CCPs if it is named a CCP in the corresponding research.

Mulvey and Beck (1984), Bard and Jarrah (2013) and Negreiros and Palhano (2006) each implement a different method of how to determine the seed location. Mulvey and Beck (1984) create clusters of data points that are as homogeneous as possible while certain capacity constraints are satisfied. Their model decides which point functions as a median and thus as a reference for clustering. Their seed locations do not have to be part of the data set. Bard and Jarrah (2013) choose the seed location beforehand, and Negreiros and Palhano (2006) take the centroid as a reference instead of the median. However, having the centroid as a reference does result in a non-linear objective function. Deng and Bard (2011) develop a new formulation of the CCP to deal with centroids as a reference efficiently. This formulation is written as a quadratic binary integer program by Brimberg et al. (2019). Within this formulation, the data points are nodes with corresponding connecting edges. The goal is to maximise the sum of the edge weight within each cluster, therefore no longer requiring a reference. Lewis et al. (2014) solve the formulation created by Deng and Bard (2011) in an exact manner. They solve some of the instances of 30, 40 or 50 locations to optimality within two hours and find a good solution for each. A nice achievement, but limited to an unrealistic small data set for us.

To still solve a CCP to optimality, but with faster computation times such that larger instances can be solved, researchers often use column generation (Mehrotra and Trick (1998), Lorena and Senne (2004) and Bard and Jarrah (2013)). Mehrotra and Trick (1998) speed up the column generation process by applying a special branch-and-bound algorithm and can solve instances with up to 60 locations to optimality. Lorena and Senne (2004) use a combination of column generation and Lagrangean/surrogate relaxation, which accelerates the computational process significantly—allowing them to solve instances with up to 400 locations to near optimality. Bard and Jarrah (2013) use a CCP together with column generation. The model they create can solve instances of up to 10,000 locations. As their instances vary from 5,000 to 50,000 locations, they implement a preprocessing step in which they merge the closest locations until about 8,000 locations exist. Finally, they use a variable-fixing heuristic to get an integer solution. Their fixing heuristics iteratively fixes four new delivery areas, one in every quarter of the total delivery region. For the work of Bard and Jarrah (2013), it is unclear what the optimality gap is. We notice that these three pieces of research show it is inevitable to give in on solution quality to gain computation speed, as these three methods could solve instances about ten times as large as the previous one. However, they can account for less and less optimality.

Another approach to speed up the computation time is to solve the CCP using heuristics entirely, which has the disadvantage that the optimality gap is often unclear. Over the years, the developed heuristics can solve increasingly larger data sets. Osman and Christofides (1994) combine a Tabu Search and Simulated Annealing for the first time and find better results than previously noted on randomly generated instances of 50 or 100 locations. Cano et al. (2002) use a Greedy Random Adaptive Search Procedure (GRASP). GRASP consists of two phases: a construction phase, which finds a feasible solution, and a local search phase which improves the current result, for which Cano et al. (2002) use K-means clustering. It turns out that this method improved the results on several existing benchmarks, ranging from 75 to 2310 locations. Brimberg et al. (2019) implement two different Variable Neighbourhood Descent methods on standard instances differing in size up to 2000 locations. Zhou et al. (2019) use a combination of Tabu Search and Memetic algorithm. The Tabu Search is also allowed to explore the infeasible solution space and therefore implemented as a local optimisation mechanism. The Memetic algorithm is applied to have a cluster-based crossover of favourable properties. The results show that the algorithm outperforms the state-of-the-art algorithms on several benchmark instances ranging from 20 to 2000 locations, including the ones that Brimberg et al. (2019) use.

2.3 Set Covering Problem (SCP)

The SCP is a fundamental problem, and many researchers have investigated it. The input is a matrix containing the information of which set covers which locations and a vector representing the costs of choosing the corresponding set. The output is a combination of sets covering all locations at least once for the least amount of costs (Lessing et al. (2004)). As mentioned, the SCP is NP-complete (Chvatal (1979)).

Lagrangian relaxation is often applied to solve the SCP (Caprara et al. (2000)). Both heuristic and exact approaches make use of these techniques. At first the focus was on exact approaches (Beasley (1987), Fisher and Kedia (1990), Balas and Carrera (1996)). The exact approaches can solve instances up to 400 rows and 4000 columns within hours of computation time (Caprara et al. (2000)). Translated to our problem, the number of rows indicates the number of possible delivery areas, and the number of columns resembles the number of locations.

Later, the focus shifted towards the use of heuristics. Sometimes exact approaches are enhanced by the use of heuristics. An example is the work of Balas and Carrera (1996), who developed DYNSGARD, a dynamic subgradient optimisation combined with Branch-and-Bound. The proposed method creates a dual feasible LP solution at every iteration. Moreover, instead of relaxing the constraints of all rows, they first define a set of non-overlapping rows and only relax the constraints of the rows not covered by this set. They combine this with primal and dual heuristics to tighten the upper and lower bounds.

As our instances will be considerably larger than the exact approach can solve within reasonable computation time, we now consider literature that focuses on large-size instances. Caprara et al. (1999) develop a heuristic specially tailored for large-size instances. With up to 5000 rows and 1,000,000 columns. The strength of the algorithm is the combination of several approaches. Their algorithm combines a dynamic pricing scheme for the variables with subgradient optimisation and greedy algorithms. Moreover, column fixing is applied in a systematic way to improve the values of the solution. They developed the algorithm as part of a competition, which it won. The combination of dynamic pricing and subgradient optimisation drastically reduced computation time and became the main ingredient for success.

A newer set of heuristics is called meta-heuristics. Within these heuristics, the objective can deteriorate if that leads to an improvement later. The deterioration enables the algorithms to escape from local minima (Laporte (2009)). Lan et al. (2007) create a meta-heuristic to solve the SCP. Their approach is called Meta-heuristic for Randomized Priority Search, or Meta-RaPS. Within this algorithm, they first create a feasible solution by a construction method, including random components. Afterwards, the intensification phase starts with an improvement heuristic. The algorithm can solve instances with up to 1000 rows $\times$ 10,000 columns for non-uniform cost SCPs and 28,160 rows $\times$ 11,264 columns for uniform cost problems within reasonable computation time.

2.4 Continuous Approximation (CA)

According to Langevin et al. (1996) C.F. Daganzo, one of the significant contributors to the field of continuous approximation, describes the use of CA as "a main goal of this approach is to obtain reasonable solutions with as little information as possible, and to gain a clear understanding of the trade-offs." This method groups discrete data points within a cluster, for which continuous properties are calculated based on the enclosed data points. Shortly after Daganzo introduced the concept of CA, research showed that CA provides a high-quality approximation (Eilon et al. (1974), and Newell (1973)). CA is ideally suited to make decisions at a strategic level or when the actual demand is still unknown. For the operational level, discrete optimisation is indispensable. Within our research, we have two possible uses for CA: to evaluate the properties of the delivery areas after we create them or to describe the characteristics of clusters we make during the preprocessing, similar to Winkenbach et al. (2016). We consider both approaches in more detail below.

As we only create the delivery areas and no explicit delivery routes, we explore if CA is a good approach to estimate the number of driven kilometres afterwards. A basic implementation of CA, first designed by Beardwood et al. (1959), is to approximate the optimal tour length of a travelling salesman. They can capture this within a simple formula only involving a constant (k), the number of points (n) and the size of the area (A); $k\sqrt{nA}$. They show that the created formula is asymptotically optimal. Later Larson and Odoni (1981) show that this converges rapidly for reasonably compact and convex regions. Stein (1978) and Jaillet (1988) estimate the constant k for Euclidian metric and Manhattan metric respectively. Eilon et al. (1971) develop an empirical formula to calculate the distance to visit n points uniformly scattered with a centrally located depot within the square zone. Daganzo (1984a) extends this implementation for a formula which applies to irregular shapes and non-uniform distributed data points. He builds on his previous research (Daganzo (1984b)), which shows the best strategy to divide the delivery region. The main takeaway of that research is that the solution improves when the optimisation includes the haul distances of delivery areas. Daganzo (1984a) states that a region can always be described by a rectangle as minor deviations only have a small impact on the expected travelling distance. He creates a new formula, especially for rectangular areas. Daganzo et al. (1990) validate the results and refine them. Jabali et al. (2012) combine both aspects as they radially divide the delivery area, creating circular trapezoids, which are estimated to be rectangles.

Within the field of logistics optimization, CA is often implemented to calculate the average unit cost or distance of delivery within a certain area (Smilowitz and Daganzo (2007), Winkenbach et al. (2016), Ghaffarinasab et al. (2018)), building on the work of Daganzo (Daganzo (1984b), Daganzo (1999)). The formula no longer only includes distances but takes, for example, volume and stop time into account as well. Winkenbach et al. (2016) use the formula to accurately describe the properties of preprocessed clusters, which enables them to reduce the problem size.

2.5 Application

Now that we have a clear overview of the state-of-the-art literature regarding the Capacitated Clustering Problem, Set Covering Problem and Continuous Approximation, we discuss how this research relates to our problem and which approaches we use as a basis for our implementation.

Instances for the SCP are larger for both exact methods, 60 to 400 locations, and heuristics, 10,000 to 1,000,000 locations, respectively. The origin of these differences is that the SCP contains a smaller part of our problem than the CCP. The SCP only decides which delivery areas combined lead to efficient full coverage, whereas the CCP also includes the step of making these delivery areas.

Next to this disparity, the number of locations gives not all information on the expected computation time. The relative demand compared to the capacity should also be considered. For example, suppose each location has a demand equal to the capacity. In that case, we can quickly establish the delivery areas and conclude that we need all delivery areas for full coverage. As deliverers of PostNL deliver at least 300 locations on average combined with that our data set consists of thousands of locations, we can safely assume our situation is not trivial to solve; neither each location needs its own delivery area nor can all locations be grouped in one delivery area. Based on this conclusion, it is unrealistic to expect an exact solution to our problem within reasonable computation time, irrespective of whether we use a CCP or SCP. As mentioned, the CCP captures our complete problem, and the SCP captures it only partly. Therefore, we prefer a CCP over an SCP.

The most promising research within the discussed literature regarding CCP is that of Bard and Jarrah (2013). They solve instances containing about 8,000 locations, four times larger than the second largest data set. They created their model within the context of combining commercial and residential pickup and delivery networks. We can easily translate this to the context of integrating the parcel and mail network. The largest data sets contain about 8,000 locations, which is still limited if we want to recreate the delivery areas for both parcel and mail delivery. Therefore, we have to group some locations beforehand. In the literature several options for pre-processing exists; grouping the n-nearest neighbours (Bard and Jarrah (2013)) or using a simple shape like rectangles (Daganzo (1984a)), squares (Winkenbach et al. (2016) or a radial approach (Jabali et al. (2012)). We combine elements of the approaches of Winkenbach et al. (2016) and Bard and Jarrah (2013) into a new approach. We create squares but prevent the creation of too large clustered locations by only allowing the grouping as long as it satisfies a capacity limit.

The approach of Bard and Jarrah (2013) is a column generation-based method, which splits the creation of new delivery areas and calculation of the best integer solution. This splitting enables us to use the algorithms we discuss regarding the SCP. Within this research, the work of Lan et al. (2007) is most interesting, as we expect our situation to be comparable to a uniform cost situation. We only have two different costs: full-day work for either a mail deliverer or a parcel deliverer. For a uniform cost situation, they manage to solve instances with up to about 25,000 possible delivery areas and 10,000 possible locations. Although the algorithm of Caprara et al. (1999) can solve instances containing ten times more locations, it could do so with only 5,000 potential delivery areas and is, therefore, less useful. As we have to preprocess our data for implementing the CCP, being able to solve instances of 10,000 locations is sufficient to apply the SCP to the (preprocessed) data as well.

Finally, we use the basic CA formula of Beardwood et al. (1959) to calculate the expected kilometres within the created delivery areas. We use this formula as it is simple and proven to converge rapidly for reasonable compact and convex regions, which our delivery areas will be. Moreover, we include the haul distance as a variable based on the results of Daganzo (1984a) and enforce our delivery areas to be rectangles based on Daganzo (1984b).

3 Problem Formalisation

We conclude in Section 2 that Bard and Jarrah (2013) and Lan et al. (2007) are the most relevant research for us to create delivery areas. Before we create an approach based on these algorithms, we explicitly define our problem of creating delivery areas. Section 3.1 discusses why we only optimise over the monetised human resources. Next, we formulate the essential constraints we need to take into account. We formulate the service constraints in Section 3.2 and the capacity constraints in Section 3.3. In Section 3.4, we enforce a location to be assigned to a chosen delivery area and specify the domain of each variable. This section introduces a problem formulation which we can see as the master problem for a column generation approach. We explain the details of the column generation approach and the extra constraints needed for implementation in Section 4.

3.1 Objective function

The main goal is to create delivery areas that optimise the benefits. Therefore we ideally optimise over the human resources, the environmental and financial benefits. We create delivery areas and not the explicit routing; thus, we do not know the driven kilometres and can only calculate them afterwards. Not knowing the kilometres beforehand makes it hard to include environmental benefits and vehicle use's variable costs, like fuel costs. We assume that if a delivery area takes less than a full working day to deliver, we still have to compensate the corresponding deliverer for a full working day. We, therefore, decide only to optimise the delivery areas over the number of full working days of the deliverers. This way, we entirely cover the possible human resource benefits. The wages comprise a large part of the delivery costs for PostNL. The objective is given in (1a). Before optimising, we set the maximum number of possible delivery areas p arbitrarily large. Parameter C_k indicates the costs of using delivery area k. Depending on the delivery area, these costs equal a full day's loan for a mail or parcel deliverer. Binary variable a_k is introduced to count the number of actual delivery areas created. a_k is equal to one if delivery area k has at least one assigned location.

$$Minimize \sum_{k=1}^{p} C_k a_k \tag{1a}$$

3.2 Service constraints

The created delivery areas need a network coverage of 100% combined, as all locations in the Netherlands can receive mail and parcels. We implement this requirement by the set covering Constraints (1b). N represents the set of locations. $z_{i,k}$ is a binary variable equal to 1 if location i is assigned to delivery area k.

$$\sum_{k=1}^{p} z_{i,k} \geq 1, \quad \forall i \in N \tag{1b}$$

3.3 Capacity constraints

Two factors can limit complete delivery, either the volume that fits in the van or the working times of the deliverer. It must be possible to deliver a complete delivery area within a working day without reloading the vehicle. To restrain the volume of each delivery area to the vehicle's capacity, we implement Constraints (1c). Parameter D_i indicates the volume of the demand at location i.

Parameter Q expresses the available volume in the vehicle, which is assumed to be equal for both mail and parcel vehicles.

$$\sum_{i=1}^{n} D_i z_{i,k} \leq Q, \quad k = 1, \ldots, p \tag{1c}$$

Next, we formulate the working time constraints. To linearly implement these constraints, we have to make some assumptions. The most significant assumption is that we know the delivery times of each location (τ_i) beforehand, including the travel time to the next neighbour. In reality, the delivery times can vary per location per delivery and the travel time depends on the next location in the route. We create delivery areas that must be feasible under varying delivery characteristics and do not explicitly create the delivery route. To get reliable values for the travel time to the next neighbour, based on the idea of Bard and Jarrah (2009), we look at the h nearest neighbours of location i and combine this with the delivery probabilities of these neighbours. We define a location's delivery probability as the fraction of deliveries made over all delivery days in the data set. We take the weighted average of the time to the h nearest neighbours and the corresponding delivery probability. Afterwards, we add the norm time for delivery, which is distinct for parcels and mail, to get values of $\tau_{i,k}$. The h nearest neighbours are different when combined delivery of mail and parcels is possible instead of only separate delivery. This difference leads to other delivery times based on the type of delivery area, which we indicate with the index k. The extensive calculations can be found in Appendix A.1.

Constraints (1d) implement the working time limitation. Besides the delivery time and travel time to the next location, two other factors also impact the working time. The first one is the haul time, the time it takes to travel from the depot to the nearest location in delivery area k (t_k^{haul}). Based on Daganzo (1984a), we include the haul time as a variable instead of a parameter. The second term is T^{fixed}, which is a parameter to account for the fixed time it takes to load the vehicle and to do the administrative tasks afterwards.

To calculate the values of $\tau_{i,k}$, we assume that one of the h nearest neighbours is the next delivery location. This assumption holds best for continuous delivery areas. We state the definition of continuous delivery areas in Property 1. A cluster in Bard and Jarrah (2013) is equal to a delivery area in this research. Not satisfying Property 1 indicates that several deliverers are active within the same delivery region, which is very unlikely to be the case in the optimal solution. We expect that enforcing Property 1, on top of the already formulated constraints, reduces the solution space without significantly reducing the solution quality. We indicate that our delivery area must satisfy this condition by Constraints (1e). We explicitly formulate it in Section 4.4.1, as we create a structure within Section 4 which allows for a more straightforward explicit formulation.

Property 1: Continuous Delivery Areas

Let C be a delivery area that contains a set of locations L⊆N. C is said to be continuous if location i∈conv(L)⇒ i, where conv(L) is the convex hull of the locations in the set L (Bard and Jarrah (2013).

The h nearest neighbours of location i are in the same delivery area for locations near the centre of the delivery area. However, we cannot assure this for locations towards the boundaries of the delivery areas, despite implementing Property 1. We introduce a second property to limit the relative number of locations that are 'at the boundaries', namely that a delivery area must be *robust*. We formalize *robust*ness in Property 2. By choosing the ratio parameter, we can choose the balance to be more towards a realistic travel time ($ratio = 1$) or towards freedom for the algorithm to decide on the optimal shape for delivery. Constraints (1f) indicate that a delivery area must satisfy this condition, which we again formulate explicitly in Section 4.4.1.

Property 2: Robust Delivery Areas

Let $X^{max} - X^{min}$ be the length of delivery area C and let $Y^{max} - Y^{min}$ be its width, where $X^{max} = max\{X_i : i \in C\}$, $X^{min} = min\{X_i : i \in C\}$, and similarly for $Y^{max} - Y^{min}$. C is said to be robust if its aspect ratio $\rho = (X^{max} - X^{min})/(Y^{max} - Y^{min})$ lies in the interval $[\frac{1}{Ratio}, Ratio]$ (Bard and Jarrah (2013)).

$$\sum_{i=1}^{n} \tau_{i,k} z_{i,k} + t_k^{\text{haul}} \leq T^{\max} - T^{\text{fixed}}, \quad k = 1, \ldots, p \tag{1d}$$

$$\{i : z_{i,k} = 1, \quad i = 1, \ldots, n\}, \quad k = 1, \ldots, p \text{ satisfy } \textit{Property 1} \tag{1e}$$

$$\{i : z_{i,k} = 1, \quad i = 1, \ldots, n\}, \quad k = 1, \ldots, p \text{ satisfy } \textit{Property 2} \tag{1f}$$

3.4 Final constraints

We already mentioned that a_k is equal to one if at least one location is assigned to the delivery area k; Constraints (1g) enforce this. Constraints (1h) indicate that all $z_{i,k}$ and a_k variables are binary. Constraints (1i) ensure that t_k^{haul} are positive.

$$z_{i,k} \leq a_k, \qquad i = 1, \ldots, n; \quad k = 1, \ldots, p \tag{1g}$$

$$z_{i,k}, a_k \in \{0, 1\}, \qquad i = 1, \ldots, n; \quad k = 1, \ldots, p \tag{1h}$$

$$t_k^{haul} \in \mathbb{R}_{\geq 0}, \qquad k = 1, \ldots, p \tag{1i}$$

4 Methods

In Section 3 we formalise our problem. Based on the literature in Section 2, we expect that we cannot solve our problem to optimality within reasonable computation time, as this is limited to instances containing about 400 locations. In reality, we are dealing with more than 100,000 locations. We group our data before applying the algorithm and divide our entire delivery region into smaller subregions, which we explain in Section 4.1, to find good solutions within acceptable computation time. We create a column generation-like approach to solve our problem based on these subregions. In Section 4.2, we give a general overview of the algorithm's workings. We implement two approaches to solve the Restricted Master Problem (RMP) in our algorithm. We formulate one approach as an exact Mixed Integer Linear Programming (MILP) model, and the other as a Set Covering heuristic; Section 4.3 describes both. We also propose two different methods to solve the Pricing Problem, which is the problem of creating new possible delivery areas. In Section 4.4, we explain two different Pricing methods: an exact model as a MILP model and a Randomised Construction heuristic. Finally, we discuss the other general methods part of our algorithm in Section 4.5. These include calculating the initial solution, removing double occurring locations, deciding for which we subregions we solve the Pricing method and how we implement column management. We define the formula for estimating the driven kilometres in Section 4.6.

4.1 General preprocessing

4.1.1 Grouping the data

We group several data points into one location beforehand to limit the computation time of the model. To allow separate delivery to remain possible, we only group locations with the same type of delivery. We group locations within a certain amount of meters of each other, given that their combined volume or delivery time is at most x% of the allowed van's volume or deliverer's working time, respectively. When not placing this restriction on the volume and delivery time, too large locations are created, limiting the model's flexibility. We test for the best value of x in Section 5.3.

We sum the key properties of delivery time and volume of the locations grouped, which becomes the respective property of the grouped location. The centroid of all addresses is the address of the grouped location.

4.1.2 Creating subregions

We divide the entire delivery region into smaller subregions to improve the computation speed of the algorithm. Creating subregions increases the computation speed in two different ways. First, it is a suitable division for a column generation-based approach. Using column generation, we only deal with useful delivery areas instead of all possible options, reducing the problem size. Secondly, as explained in more detail below, we create new potential delivery areas within a subregion, which cuts the number of options for each new delivery area.

The procedure to divide the delivery region into subregions consists of three steps: creating a grid, selecting the corresponding seed location, and determining which locations belong to its subregion. We now explain these steps' main points, based on the approach of Bard and Jarrah (2009).

The most elaborate step in constructing subregions is creating a grid. Figure 6A - C show how we construct the grid line by line. Figure 6A depicts the original situation: we have locations spread out over the entire area. Our goal is to divide the entire region in a grid, whereby each grid box contains at most x% of the limiting capacity, which can either be delivery time or volume. We use

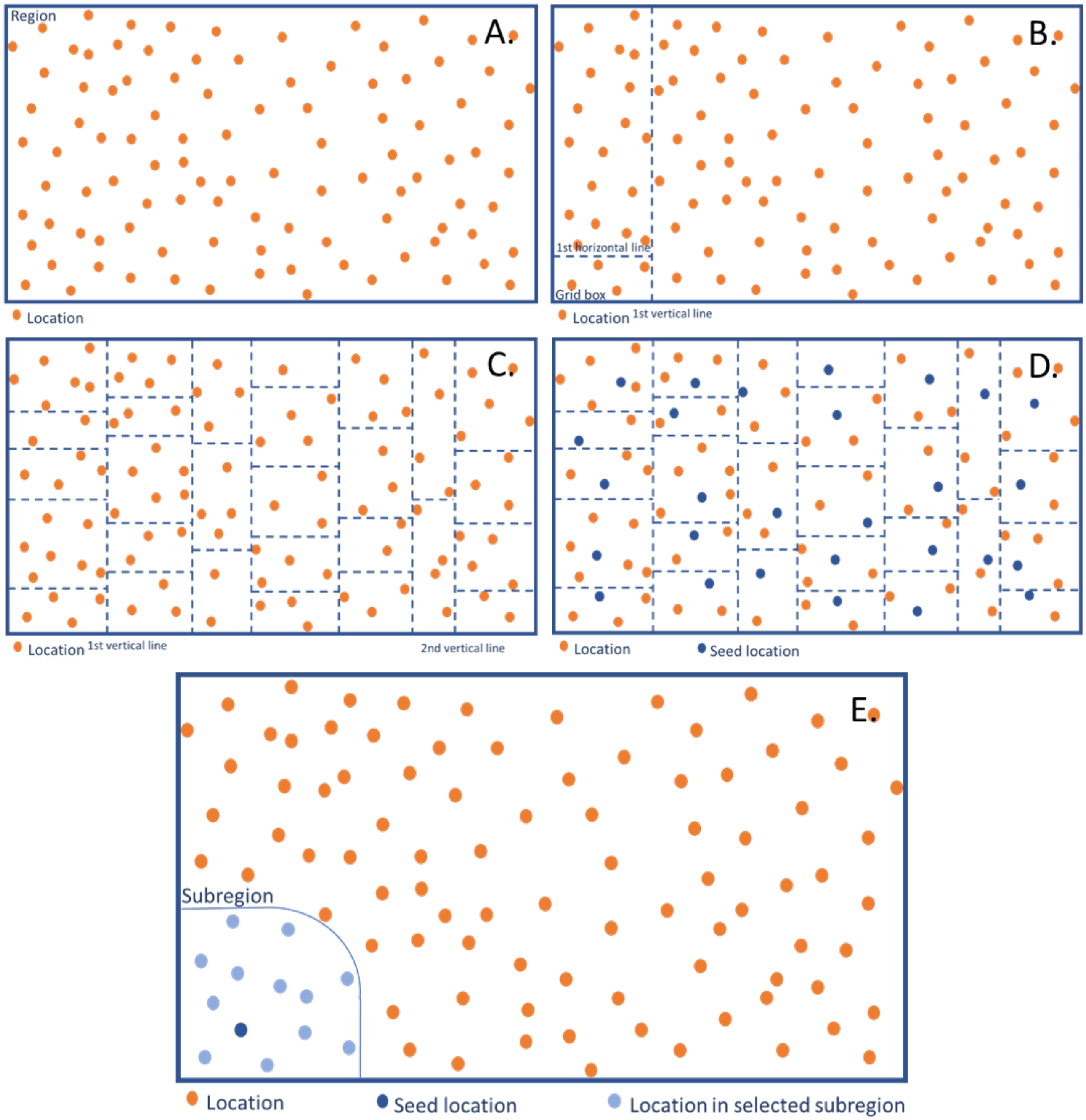

Figure 6: **A.** Start situation with the entire region. **B.** One grid box created bottom left with max 5% capacity. **C.** Situation with the entire grid. **D.** The location closest to the centroid is selected as the seed location for each grid box. **E.** The subregion for one of the seed locations.

the same value of x as in Section 4.1.1 To create the desired grid, we first determine the location of the first vertical line based on the estimated number of grids needed. After that, we add horizontal lines to satisfy each grid box's $x\%$ capacity requirement. Figure 6B shows the situation with one vertical and one horizontal line. In this case, these four locations' volume or delivery times are too close to $x\%$ to add another location. After placing a vertical line at the left of the service area, we place the following line at the region's right. We repeat this procedure until each location is captured within a grid box satisfying the imposed constraints, as shown in Figure 6C. We refer to Appendix A.2 for the full details.

With the grid created, our next step is to establish the seed location of each grid box. We determine the centroid of the locations contained in a grid box. The seed location is the location that is closest to the centroid of that grid box. Figure 6D shows all grid boxes with their corresponding seed location in blue. The only use of the grid is to ensure the seed locations are evenly divided over the region, as the seed locations play a crucial role. We use the determined seed locations throughout the entire model.

Finally, we decide which locations, besides the grid box locations, are part of the subregion of interest as Figure 6E shows. We do so by adding all locations within a 20-kilometre range from the seed location, as long as the new location has the potential to be part of a feasible delivery area. A 20-kilometre range is sufficient for all seed locations, except those at the boundaries. When we increase the range to satisfy the boundary locations, the middle subregions contain many locations, drastically increasing computation time. The procedure to determine if a new location has this potential is similar to a part of the Randomised Construction heuristic, which Section 4.4.2 explains in detail.

We create these subregions with three different settings: subregions that consider only mail locations, only parcel locations, and all locations. Combined, the first two represent separate delivery, the last integrated delivery and all of them together hybrid delivery. Note that as we explained in Section 3.3, the delivery times of a location differ depending on separate or combined delivery, which also causes the subregions to vary with different settings.

4.1.3 Delivery area assumptions

The seed location is the centre of all created delivery areas within the subregion. This way, we can create delivery areas in several subregions and are sure they never completely overlap. We formulate this in Property 3. Property 3 not only introduces the desired centrality but also limits the shape of the delivery area to a rectangle. We pick the rectangular shape as it allows for straightforward implementation of Property 1 and Property 2. For rectangular delivery areas Property 1, the continuity property, is always satisfied and the ratio property, Property 2, is easily checked. Moreover, we know by the results of Daganzo (1984a) that rectangles are a good fit to describe delivery areas since deviations from a rectangle have a limited impact on the expected travelling distance.

We further restrict the properties of each created delivery area by imposing Property 4. We implement Property 4 to conveniently know which locations need be included to satisfy Property 3.

Property 3: Centrality Delivery Areas

Any delivery area generated for a subregion should span a symmetric rectangle centred at the seed location s (Jarrah and Bard (2012)).

Property 4: Parallel Delivery Areas

Any delivery area generated must be parallel to the X-axis or Y-axis.

4.2 Overview general algorithm

The general solution approach is given in Algorithm 1. We depict the algorithm to give a general overview without describing the updating of all mentioned parameters completely. We explain these details during the thorough discussion of the methods, which we do in the indicated subsections.

Right now, we consider the coherence between all methods. We start our algorithm with the calculation of subregions as described above (line 1, Section 4.1), after which we find an initial solution to be able to start the column generation (line 2, Section 4.5.1). We solve the LP relaxation of the exact RMP model formulation (line 4, Section 4.3.1) to obtain the dual values. Based on the duals arising from the solution, we try to improve the solution by creating new delivery areas in a subset of the subregions (lines 8 - 12, Section 4.5.3). We construct the new delivery area using one of the two methods to solve the Pricing Problem (line 9, Section 4.4). We add the new delivery areas to the set of possible delivery areas and again solve the LP relaxation to see if our solution improved and to obtain new dual values. We repeat this process until we reach a certain number of delivery areas (line 5, Section 4.5.4). We then also calculate the integer solution, as we need an integer solution in the end, and save this (line 6, Section 4.3). When the integer solution does not improve over the last five iterations, the algorithm stops and returns the best-found solution. We define *no improvement* as when the current solution improves the best integer solution so far with less than 0.01 or does not improve it at all.

Algorithm 1 General overview of solution approach

```
 1: subregions ← createSubregions(preprocessed_locations)
 2: delivery_areas ← initialSolution(preprocessed_locations)
 3: while iterations_constant_objective ≤ 5 do
 4:     duals, objective ← LPRelaxationRMP(delivery_areas)
 5:     if len(delivery_areas) > delivery_areas_amount then
 6:         int_objective, int_delivery_areas ← RMP(delivery_areas)
 7:     end if
 8:     for highest potential subregions do
 9:         if pricingProblem(subregion_i, duals) ≤ 0 then
10:             delivery_areas ← delivery_areas + pricingProblem(subregion_i, duals)
11:         end if
12:     end for
13: end while
14: final_solution ← removeDoubles(best_int_delivery_areas)
15: return best_int_objective, final_solution
```

4.3 RMP methods

We implement two different methods to find an integer solution of which delivery areas give the best solution from the set of delivery areas, the Restricted Master Problem (RMP). An exact model formulated as a Mixed Integer Linear Program, to which we refer as the exact RMP model and an approximate Set Covering heuristic based on Lan et al. (2007). The exact RMP model is easy to implement, but it can take a long time before it reaches optimality. The Set Covering heuristic is more involved to implement, and we can give no guarantee on the obtained results. However, as the results of Lan et al. (2007) are very promising, it might give better results within a certain time window. To test this, we implement both methods and compare them in Section 5.2. We first describe the exact RMP model in Section 4.3.1. Section 4.3.2 explains the second approach based on the Set Covering heuristic of Lan et al. (2007).

4.3.1 Exact RMP model

The exact RMP model aims to decide from all possible delivery areas which delivery areas cover all locations at least once with the least amount of costs, which is essentially the Set Covering Problem. We explain the formulation of the exact RMP model below. Besides using the model to obtain an integer solution, we solve the LP-relaxation to get the dual variables, which we use to create new delivery areas (line 4 of Algorithm 1).

Exact RMP model Formulation

Sets and indexes

N = set of locations,	i = location index	$i \in N$
K = set of delivery areas,	k = delivery area index	$k \in K$

Decision Variables

$$a_k = \begin{cases} 1 & \text{if delivery area } k \text{ is chosen;} \\ 0 & \text{otherwise.} \end{cases}$$

Parameters

Z_i^k = binary parameter, 1 if delivery area k serves location i and 0 otherwise

C_k = cost of delivery area k

Exact RMP model

$$Minimise \sum_{k \in K} C_k a_k \tag{2a}$$

$$\text{Subject to} \sum_{k \in K} Z_i^k a_k \geq 1, \quad i \in N \tag{2b}$$

$$a_k \in \{0,1\}, \; k \in K \tag{2c}$$

The objective function (2a) of the exact RMP model is similar to the objective of the master problem (1a); they both minimise the costs of the delivery areas created. For our exact RMP model, we do not have an arbitrarily limit of p delivery areas, but we create the delivery areas k beforehand. At every iteration of the total algorithm, the set K will be larger as each iteration adds new delivery areas (lines 8 - 12 of Algorithm 1). Constraints (2b) implement the set covering constraints and make sure each location is covered at least once. An important difference with Constraints (1b) is that we indicate which locations a delivery area covers beforehand with parameters Z_i^k, instead of using decision variables z_i. The Pricing method determines the values of Z_i^k during the creation of the delivery areas. Finally, Constraints (2c) ensure that the a_k variables are in the binary domain.

4.3.2 Set Covering heuristic

As mentioned before, we base the Set Covering heuristic on the work of Lan et al. (2007). The goal of the Set Covering heuristic is equal to that of the exact RMP model: to find a set of delivery areas that cover all locations for the least cost. In general, the heuristic consists of two steps: an initial solution is created, followed by an improvement step in which we try to improve the initial solution by neighbourhood search. We now explain the main workings and refer to the work of Lan et al. (2007) for the complete description.

Before we create our initial solution, we first check if any of the generated delivery areas *dominates* another delivery area. A delivery area *dominates* if it contains at least all locations of the dominated delivery area. If this is the case, the dominated delivery area is never favourable, and we remove it from the possible delivery areas.

The creation of the initial solution is a randomised greedy approach. In most cases, the algorithm iteratively picks the *best* delivery area to add until all locations are covered. We define *best* as the delivery area with the most new locations for the least amount of costs per new location. However, with 10% probability, it is possible to add a delivery area that differs at most 15% from the *best* delivery area. After constructing our initial solution, we check if any *redundant* delivery areas are present and remove them. A delivery area is *redundant* if it only contains locations that are covered by other delivery areas as well.

Now that we have our initial solution, we try to improve it by performing a neighbourhood search. We randomly remove at most 20% of the delivery areas and create a smaller Set Covering subproblem containing the locations we no longer cover and all delivery areas that cover at least one of these locations. We use our initialisation algorithm to find a new solution for this subproblem. We combine the new solution with the retained delivery areas. After we remove any *redundant* delivery areas, we check if our solution is improved. If this is the case, we will update our solution. After the algorithm performs the neighbourhood search 30 times, it returns the best-found solution and its objective.

4.4 Pricing methods

Similar to solving the RMP, we have two approaches to solve the Pricing Problem. We formulate one approach again as an MILP model, which we call the exact Pricing model. The second approach is a Randomised Construction heuristic. To goal of the Pricing methods is to create a new delivery area which has lowest objective. Both approaches are subregion specific, so both methods create a new delivery area in a predetermined subregion r. Each delivery area we create must comply with Constraints (1c) - (1f) as indicated in Section 3 and with Property 3, which we define in Section 4.1.

The exact Pricing model guarantees that the constructed delivery area is optimal regarding the current dual values. As the formulation is vast, we conveniently build the Randomised Construction heuristic that creates feasible delivery areas that satisfy all the constraints. In Section 5.2, we test if a possible increase in computation speed of the heuristic opposes the lack of guarantee of optimality. We explain the formulation of the exact Pricing model in Section 4.4.1. Section 4.4.2 introduces the Randomised Construction heuristic.

4.4.1 Exact Pricing model

We first introduce the exact Pricing model formulation and then explain the formulation of the constraints in more detail. We base the formulation of the exact Pricing model on Bard and Jarrah (2013).

Exact Pricing Model Formulation

Sets and indexes

$N(r)$ = set of locations belonging to subregion r, $\quad i$ = location index $\quad i \in N(r)$

s = index of seed location of subregion r $\quad s \subset N(r)$

A = ordered list of the $|N(r)|$ locations ordered by their increasing absolute X-distance from the seed $\quad a$ = index of location i in list A $\quad i \in N(r)$

B = ordered list of the $|N(r)|$ locations ordered by their increasing absolute Y-distance from the seed $\quad b$ = index of location i in list B $\quad i \in N(r)$

Decision Variables

$$z_i = \begin{cases} 1 & \text{if location } i \text{ is assigned to the created delivery area} \\ 0 & \text{otherwise.} \end{cases}$$

$$t_i^{min} = \begin{cases} 1 & \text{if location } i \text{ has the shortest travel time to the depot within the created delivery area;} \\ 0 & \text{otherwise.} \end{cases}$$

$$x_a = \begin{cases} 1 & \text{if corresponding location } i \text{ of index } a \text{ satisfies convexity and centrality property in X-direction;} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{\{i\}} = \begin{cases} 1 & \text{if corresponding location } i \text{ of index } b \text{ satisfies convexity and centrality property in Y-direction;} \\ 0 & \text{otherwise.} \end{cases}$$

$$\xi_i^{max}(\xi_i^{min}) = \begin{cases} 1 & \text{if location } i \text{ has the largest (smallest) X-coordinate within the created delivery area;} \\ 0 & \text{otherwise.} \end{cases}$$

$$\nu_i^{max}(\nu_i^{min}) = \begin{cases} 1 & \text{if location } i \text{ has the largest (smallest) Y-coordinate within the created delivery area;} \\ 0 & \text{otherwise.} \end{cases}$$

$X^{max}(X^{min})$ = largest (smallest) X-coordinate within the created delivery area

$Y^{max}(Y^{min})$ = largest (smallest) Y-coordinate within the created delivery area

t^{haul} = time to drive from depot to nearest location of created delivery area

Parameters

θ_i = dual variable associated with Constraints (2b) for location i of the LP-relaxation

τ_i = delivery time of location i, including travel time to next location

C_r = costs of creating a delivery area in subregion r

D_i = demand of location i

Q = capacity of the vehicle

$Ratio$ = aspect ratio for a delivery area

$T^{\max}$ = maximal allowed working time for a deliverer

T^{fixed} = fixed time it takes to load the vehicle and to do the administrative tasks afterwards

T_i = two-way travel time between point i and the depot it is served from.

$$T_i = 2\sqrt{(X_i - X_{depot,i}^2) + (Y_i - Y_{depot,i}^2)}/V_{haul,i}$$

V_{haul} = average speed during the distance between the delivery area and nearest depot for location$i$

$X_{depot}(Y_{depot})$ = X (Y)-coordinate of the nearest depot for location i

$X_i(Y_i)$ = X (Y)-coordinate of location i

$X_s(Y_s)$ = X (Y)-coordinate of the seed location s

$\widehat{X}_r^{max}(\widehat{X}_r^{min})$ = largest (smallest) X-coordinate in subregion r

$\widehat{Y}_r^{max}(\widehat{Y}_r^{min})$ = largest (smallest) Y-coordinate in subregion r

$$\text{Min. } C_r - \sum_{i\in N(r)} \theta_i z_i \tag{3a}$$

$$\text{S.t.} \sum_{i\in N(r)} D_i z_i \leq Q \tag{3b}$$

$$\sum_{i\in N(r)} \tau_i z_i + t^{\text{haul}} \leq T^{max} - T^{\text{fixed}} \tag{3c}$$

$$z_i \leq x_i, z_i \leq y_i, z_i \geq x_i + y_i - 1, \forall i \in N(r) \tag{3d}$$

$$x_a \leq x_{a-1}, \quad \forall a \in A \setminus 0 \tag{3e}$$

$$y_b \leq y_{b-1}, \quad \forall b \in B \setminus 0 \tag{3f}$$

$$X^{\max} \geq X_i z_i, \forall i \in N(r) \tag{3g}$$

$$X^{\max} = \sum_{i\in N(r)} X_i \xi_i^{\max}, \sum_{i\in N(r)} \xi_i^{\max} = 1, \xi_i^{\max} \leq z_i, \forall i \in N(r) \tag{3h}$$

$$X^{\min} \leq X_i z_i + X_s\left(1 - z_i\right), \forall i \in N(r) \tag{3i}$$

$$X^{\min} = \sum_{i\in N(r)} X_i \xi_i^{\min}, \sum_{i\in N(r)} \xi_i^{\min} = 1, \xi_i^{\min} \leq z_i, \forall i \in N(r) \tag{3j}$$

$$Y^{\max} \geq Y_i z_i, \forall i \in N(r) \tag{3k}$$

$$Y^{\max} = \sum_{i\in N(r)} Y_i \nu_i^{\max}, \sum_{i\in N(r)} \nu_i^{\max} = 1, \nu_i^{\max} \leq z_i, \forall i \in N(r) \tag{3l}$$

$$Y^{\min} \leq Y_i z_i + Y_s\left(1 - z_i\right), \forall i \in N(r) \tag{3m}$$

$$Y^{\min} = \sum_{i\in N(r)} Y_i \nu_i^{\min}, \sum_{i\in N(r)} \nu_i^{\min} = 1, \nu_i^{\min} \leq z_i, \forall i \in N(r) \tag{3n}$$

$$t^{\text{haul}} \leq T_i z_i + T_s(1 - z_i), \forall i \in N(r) \tag{3o}$$

$$t^{\text{haul}} = \sum_{i\in N(r)} T_i t_i^{min}, \sum_{i\in N(r)} t_i^{min} = 1, t_i^{min} \leq z_i, \forall i \in N(r) \tag{3p}$$

$$1/Ratio \leq \left(X^{\max} - X^{\min}\right) / \left(Y^{\max} - Y^{\min}\right) \leq Ratio \tag{3q}$$

$$x_a \in \{0,1\} \forall a \in A, y_b \in \{0,1\} \forall b \in B \tag{3r}$$

$$z_i, \xi_i^{\max}, \xi_i^{\min}, v_i^{\max}, v_i^{\min} \in [0,1], \forall i \in N(r) \tag{3s}$$

$$X^{\max} \in \left[X_s, \widehat{X}_r^{\max}\right], X^{\min} \in \left[\widehat{X}_r^{\min}, X_s\right], Y^{\max} \in \left[Y_s, \widehat{Y}^{\max}\right], Y^{\min} \in \left[\widehat{Y}_r^{\min}, Y_s\right] \tag{3t}$$

$$t^{\text{haul}} \in \mathbb{R}_{\geq 0} \tag{3u}$$

Solving the exact Pricing model results in a new delivery area, which we add to the list of delivery areas that the overall algorithm can use. The objective (3a) is to minimise the costs of the created delivery area. The costs consist of a standard fee for the subregion r; this is equal to the labour costs of a full-day delivery, which we reduce by the dual value of each assigned location.

Constraint (3b) implements the condition that a delivery area cannot exceed the vehicle capacity. Constraint (3c) ensures that delivering all included locations within the allowed working time is possible. A deliverer spends most of the time delivering the items and driving towards the next location (τ_i). Please note that we give an extensive explanation of the calculation of τ_i in Section 3 and Appendix A.1. We determine the total time delivering items and driving to the next location with the first term of Constraint (3c) ($\sum_{i \in N(r)} \tau_i z_{i,r}$). Besides this, we also consider the time to travel between the delivery area and the depot by the decision variable t^{haul}. Combined, this must be smaller than the total working time of a day (T^{max}) minus the time the fixed operations take (T^{fixed}).

We desire each delivery area to be continuous as formulated in Property 1. In Section 3 Constraints (1e) we only formulate it in a global manner. Now we combine Property 1 with Property 3 and Property 4 and formulate it explicitly for the exact Pricing model. These properties are implemented by Constraints (3d) - (3f). Constraints (3e) make sure that if a location is part of the delivery area, the help variables corresponding to locations with a smaller X-distance to the seed, $x_{a-1}, ..., x_0$, will have to be equal to 1 as well. Constraints (3f) apply similar reasoning regarding the Y-distance of locations to the seed. Constraints (3d) ensure that a location can only be part of the delivery area if both its corresponding help variables are equal to 1. Together, these constraints ensure that every created delivery area will be a continuous rectangle centred at the seed.

Constraints (3g) - (3p) each serve to set the relevant decision variables to the correct value. Constraints (3g) and Constraints (3h) enforce that X^{max} is set to the largest X-coordinate found within the selected locations. To keep notation neat, we state that we implement these constraints for all locations within $N(r)$. However, as we enforce a symmetric rectangle around the seed, we know that locations with a smaller X-coordinate than the seed location will never be the location with the largest X-coordinate. Constraints (3g) and Constraints (3h) are therefore actually only implemented for locations with a higher or equal X-coordinate than the seed location. Constraints (3g) ensure X^{max} is at least as large as the location with the largest X-coordinate. Constraints (3h) enforce that the value X^{max} attains is exactly equal to the largest X-coordinate and not larger. In a similar way the value of X^{min} is set to the smallest X-coordinate by Constraints (3i) - (3j). We include an additional term in Constraints (3i) compared to Constraints (3g) to prevent locations not part of the newly created delivery area from enforcing X^{min} to be less than or equal to zero, which often leads to infeasibility. As the largest value for X^{min} is X_s, which is always included, X_s can be used to ensure positive values Analogous the values of Y^{max} and Y^{min} are set to the correct value by respectively Constraints (3k) - (3l) and Constraints (3m) - (3n). To make sure we use the minimal haul time within the model, we implement Constraints (3o) and Constraints (3p) to set t^{haul} to its correct value. We only implement these constraints for locations with a shorter distance to the depot than the seed location. With similar reasoning as above, we conclude that t^{haul} attains the value of the haul time to the seed location in all other situations. We include the second term in Constraints (3o) again to prevent locations not part of the newly created delivery area from enforcing t^{haul} to be less than or equal to zero.

Constraint (3q) implements Property 2 and thus limits the ratio between the length and width of each created delivery area. Whereas Constraint (3q) implements Property 2 explicitly, Constraints (3g) - (3m) are solely implemented to enforce the correct values for the variables used in Constraint (3q). Property 2 is thus costly to implement in terms of variables and constraints.

Finally Constraints (3r) - (3u) implement the allowed domain for all variables. Constraints (3r) ensure that the x and y variables are binary variables. By Constraints (3d) z_i is also forced to be binary, so we do not need to enforce this by a separate constraint. It is, therefore, sufficient to state that z_i must be continuous between 0 and 1, as done in Constraints (3s). Similar reasoning holds for $\xi_i^{\max}, \xi_i^{\min}, \nu_i^{\max}$ and $\nu_i^{\min}$. Constraints (3t) limit the domain of $X^{\max}, X^{\min}, Y^{\max}$ and $Y^{\min}$ to range between the location of the seed and either the highest or lowest value present in the data set, as each delivery area will be a symmetric rectangle around the seed. Constraint (3u) ensures that t^{haul} attains a non negative value.

4.4.2 Randomized Construction heuristic

As input for our Randomised Construction heuristic, we define the subregion r for which we want to create a new delivery area. As output we have a delivery area satisfying all constraints as mentioned in the Section 3, Property 3 and Property 4.

We start our algorithm by randomly picking a location within subregion r that functions as our boundary location. First, this location is our boundary location in the X-direction and afterwards in the Y-direction. As we know that the seed location must be in the middle of the rectangle by Property 3 and we have a ratio limitation by Property 2, we create the smallest possible delivery area based on our boundary location. Suppose the smallest delivery area already exceeds either the volume or time capacity constraint, similar to Constraints (3b) and (3c), respectively. In that case, we drop this delivery area and continue to the next iteration. We also use this procedure when we create the subregions to test if a location has the potential to be a part of a feasible delivery area, as we mention in Section 4.1.

For clarity of explanation, we now assume our boundary location bounds the delivery area's width and thus X-distance. If the smallest possible delivery area is feasible, we add locations with increasing Y-distance from the seed, given that their X-location is within bounds. We continue to add these locations as long as all our constraints hold. When adding a new location will violate one of our constraints, we will stop and have created our new delivery area.

We do a final feasibility check to make sure the delivery area satisfies all constraints as formulated in Section 3, Property 3 and Property 4. If the delivery area satisfies all constraints, we store it as a potential delivery area. When we have 30 different possible delivery areas or all locations are used as the boundary location, we calculate the objective value for each of them, similar to (3a). We return the delivery area with the lowest objective.

The most significant difference with the exact Pricing model is that we create a random delivery area, which is not directly guided by the dual variables. However, the dual variables are not entirely untouched. We make multiple possible delivery areas and use the dual variables to calculate the quality of the delivery areas afterwards.

4.5 General methods

So far, we have discussed the general outline and two options to find an integer solution (RMP methods) and two to create new delivery areas (Pricing methods). We now discuss the methods part of the general approach. We first clarify how we use the Pricing Problem to develop an initial solution in Section 4.5.1. Section 4.5.2 explains two methods to ensure that each location is covered by exactly one delivery area. Afterwards, Section 4.5.3 explains how we choose the subregions for which we solve the Pricing Problem in the next iteration and how the tabu list is updated. Finally, we discuss our column management approach in Section 4.5.4.

4.5.1 Initial solution

We use a Pricing method combined with artificial dual variables to create a useful and feasible initial solution. We start with all artificial dual variables equal to one and set them to zero if the created delivery area covers the corresponding location. We set the standard costs (C_r) to zero, as we only focus on adding new locations.

First, we solve a Pricing method for every tenth subregion and add the created delivery area to the initial solution if at least one new location is covered. After completing this first iteration, we select the 25 subregions with the highest number of uncovered locations *nearby* for the next iteration. A location is seen as *nearby* if it belongs to either the first half of the locations ordered based on X-distances from the seed location or the first half based on the Y-distances.

When we solve a Pricing method for a subregion, we add the subregion to a tabu list, and the algorithm can no longer choose it in all following iterations. When the final five selected subregions do not lead to a delivery area with a new location, we reset the tabu list. The algorithm can then choose from all subregions in the next iteration.

Once at least one delivery area covers every location, we have an initial solution and return the created delivery areas.

4.5.2 Making locations uniquely covered

The algorithm ensures that the chosen delivery areas serve each location at least once. However, multiple delivery areas can service the location. To ensure unique servicing in our final solution, the double removing method removes multiple times serviced locations from all but one delivery area. We discuss two methods: one with the centroid as reference and one with the delivery time as reference. We compare both methods in Section 5.5 to find which suits our situation the best.

The double removing method removes the locations with a greedy approach. In the first method, we determine the distance to the centroid of each delivery area where the location occurs. The double removing method removes the location from all delivery areas except the one with the smallest distance to the centroid.

The second approach is similar, except that we calculate the delivery time of each delivery area in which the location occurs. We remove the location from all delivery areas except the one with the largest delivery time.

4.5.3 Region determination

At every iteration, we solve a Pricing method for a limited amount of subregions. We solve a Pricing method for 50 subregions or five per cent of the subregions, whichever is larger.

For the first iteration after initialisation, we pick the subregions for which we solve a Pricing method at random. A more complicated method to determine the subregions exists for all other iterations. We use a tabu list to prevent picking subregions with little improvement potential. Figure 7A shows an overview of how the tabu list is updated. The subregions for which the Pricing method results in a delivery area with a positive objective value are tabu for seven iterations. We split the subregions leading to a delivery area with a negative objective value into four quartiles based on the size of the objective value. The most negative quartile is not tabu for any iteration and is available in the next iteration. The second quartile is tabu for one iteration, the third for two iterations and the fourth and final quartile for three iterations.

Figure 7B gives a graphic representation of the structure of the subregions, which we select to explore in the next iteration. Half of the subregions for the next iteration are chosen based on their potential to improve the solution. We examine the subregions with the highest summed dual

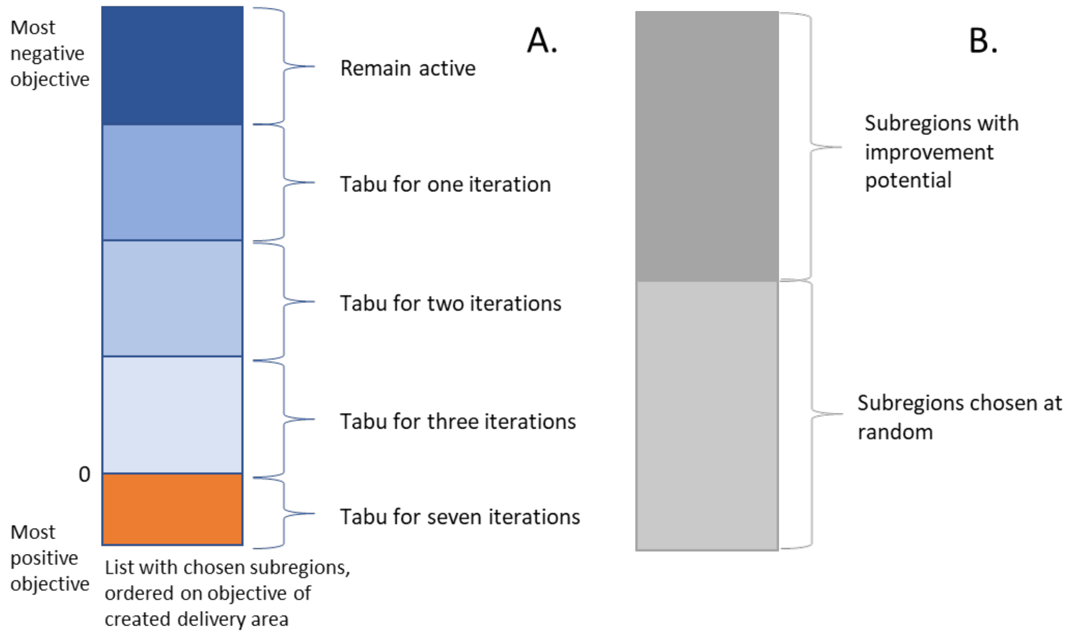

Figure 7: A. Overview of how we determine the number of iterations a subregion is tabu. **B.** Overview of how we choose the subregions for the next iteration.

values in the next iteration. We randomly pick the other half of the subregions out of all possible subregions that are not tabu and not already chosen.

4.5.4 Column management

Besides previously described algorithms, we also implement a column management approach. We implement column management for two reasons. First, we do not want to continuously solve the RMP method as this is more computationally intensive than calculating the LP-relaxation. Secondly, we want to limit the number of possible delivery areas, as this reduces computation time.

We decide to calculate the first integer solution after we obtain 500 delivery areas and, after that, every 400 new delivery areas. As the first columns are generally not used in the final solution, we want to remove these after a while. We delete all delivery areas not used by the first integer solution, which has at least 1500 delivery areas as input.

4.6 Kilometre estimation

For our analysis, after we find a solution by our algorithm, we estimate the delivery distance within each area with the simple formula of Beardwood et al. (1959) ($k\sqrt{nA}$). The constant k is set to be 0.765 as calculated by Stein (1978). n is the number of locations, and A is the area in km^2. The number of preprocessed locations is used for n in this analysis, as the assumption of uniformly distributed points does not hold for the unprocessed locations. Moreover, the distance between the locations represented by the same preprocessed location is minimal. We calculate the area by finding the smallest and largest X- and Y-coordinates of the locations covered by the delivery area. The area of the rectangle with these coordinates as vertices is A. We add the haul distance twice to the formula of Beardwood et al. (1959) to get a complete estimation of the delivery distance travelled by the deliverer within a working day. The formula is limited in its precision, but it provides an indication of the delivery distances within a delivery area.

5 Computational experiments

Within this section, we test the performance of our algorithm. The analysis centres around a case study of the service area of the parcel depot in Kolham, a village in the northeast of The Netherlands. First, we explore in Section 5.1 whether the integration of both networks improves the delivery times within the service area of Kolham. Only if we note an improvement in delivery times the new supply chain model has the potential to be better. Next, we compare the four possible combinations of RMP methods, namely the exact RMP model and Set Covering heuristic, and Pricing methods, i.e. the exact Pricing model and Randomised Construction heuristic, in Section 5.2. We test on a subset of about one-fourth of the case study's data to test on real data, but with reduced computation time. We use the best combination for our case study. Afterwards, we investigate the best settings for the preprocessing phase in Section 5.3. To know whether the outcomes of our algorithm are robust to changes in the cost parameters, we analyse whether the model responds intuitively to changes in the cost parameters in Section 5.4. We expect that our two double removing methods result in different created delivery areas. To test the impact, we apply both our double removing methods to the same data in Section 5.5. With the algorithm and parameters determined, aware of the impact of the double removing method, we perform our case study experiments in Section 5.6. We explore both the optimal delivery method on the quietest and busiest day. Finally, we explore the impact of our defined Properties. In Section 5.7, we dive into the effects of Property 4, that delivery areas must be parallel to either the X- or Y-axis, by rotating the data set.

We execute experiments on a PC with a 2.6 GHz Intel Core i5-1147G7 processor and 16.0 GB RAM. The solution methods are implemented in Pycharm using Python 3.9, and all MILP models are solved using the commercial solver Gurobi 9.5.1 with default settings unless mentioned otherwise. The Ratio parameter is 5 for all experiments.

5.1 Impact of integration on delivery times

We first analyse the effects of switching from separate delivery to our integrated model in which PostNL delivers mail and parcels together. Table 1 shows that there is a slight improvement for the parcel delivery, only visible in the total delivery time. However, for mail delivery, the delivery times for combined delivery decrease on average from 0.48 to 0.36, a decrease of 25%. The decrease of 25% in delivery times is a first indication that integration of the networks is more efficient. One reason is that it takes no time to deliver mail when PostNL hands over a parcel as well. Another argument is that the next delivery location is closer, as parcel locations can also be the next location. However, Table 1 also shows the more efficient delivery comes at the cost of a tenfold increase in haul time for mail, as delivery rounds now always start at the parcel depot.

Moreover, Table 1 shows parcel is on average faster delivered than mail. An unexpected result at first sight, as for parcel delivery, a driver must stop, ring and wait for the person to answer the door, which takes longer than delivering a card in a mailbox. However, τ_i represents the weighted delivery time according to the probability of visiting the location. The visiting probability reduces the calculated delivery time significantly.

5.2 Algorithm comparison

Within Section 4, we describe the main structure of the algorithm and two options for calculating the integer solution (RMP) and two for obtaining new delivery areas (Pricing Problem). We combine these four approaches into four different algorithms. We either use an entire model approach, the exact RMP model and exact Pricing model or a complete heuristic approach with the Set Covering heuristic and Randomised Construction heuristic. We also have two combinations of an exact model

Table 1: Analysis of the values for τ_i, the estimated weighted delivery time of location i and t_i^{haul}, twice the time to travel between location i and the nearest depot, of mail and parcel for both separate and integrated delivery. We perform the analysis before grouping the locations

Delivery method	Average τ_i (min)	Total τ_i (min)	Average t^{haul} (min)
Mail			
Separate delivery	0.48	4492	4.29
Combined delivery	0.36	3366	46.99
Parcel			
Separate delivery	0.34	43009	27.65
Combined delivery	0.34	42420	27.65

and heuristic (exact RMP model and Randomised Construction heuristic or Set Covering heuristic and exact Pricing model). We explore which of these four combinations delivers the best results by applying all four approaches to a relatively small data set of 2260 locations, of which 1705 are parcel locations and 555 mail locations. This small data set is a subset of the data set of the case study in Kolham on Saturdays, obtained by drawing a rectangle on the map such that it contains no more than 2500 locations. We limit the number of locations to 2500 to allow for a comparison to optimality, as we explain below, within reasonable time, but still use real data.

Both RMP methods stop when they reach our stopping criteria, which is for the exact RMP model to achieve an optimality gap of 0.01% and for the Set Covering heuristic when the objective does not improve over the last five iterations. Moreover, we also limit the run time for each iteration to 10 minutes. During our test runs, the run time of the exact RMP model increased significantly with a growing number of delivery areas. If more than 1000 delivery areas are possible, the exact RMP model approach cannot reach optimality within 12 hours. For our test cases with a run time of one hour, the integer objective only decreased during the first 10 minutes. An optimality gap is impossible to implement for the Set Covering heuristic as it does not generate a lower bound; therefore, we did not choose an optimality gap. During all our experiments described, we include this time cap of 10 minutes on the RMP method.

We want to know to what extent our algorithms reach optimality. We generate all delivery areas using only a small data set. We give them to the exact RMP model, which returns a lower bound that functions as an approximation of optimality. We generate the delivery areas by changing our Randomised Construction heuristic to no longer pick a random location as the boundary but to iterate over all locations as boundary locations in both X- and Y-directions. When we iterate over all subregions, we create all possible delivery areas. We generate all, namely 42765, delivery areas in 110 minutes. For the lower bound, we set an optimality gap of 5% and a time limit of 3 hours for the exact RMP model. The best integer solution the model finds is 3339.84. This objective is used as a reference in Table 2 to obtain the gap values. We calculate all gap values by taking the difference between the integer solution and the best-bound relative to the integer solution.

Table 2 shows the performance of our four combinations of algorithms. We observe two trends. The algorithm with the exact RMP model as the integer algorithm outperforms the Set Covering heuristic by about 20% irrespective of the Pricing method. Moreover, the different objectives for both Pricing methods when combined with the Set Covering heuristic show the Set Covering heuristic's randomised nature. The constructive methods do return the same objective when combined with the exact RMP model.

Concerning the Pricing methods, we see that the Randomised Construction heuristic is approximately twice as fast as the exact Pricing model, without loss of quality. Therefore, we take the exact RMP model as the RMP method and the Randomised Construction heuristic as the method

Table 2: The objective, gap and computation time of the four possible combinations of integer and constructive methods on a data set with 2260 locations. The gap is determined based on a lower bound, which originated from solving the exact RMP model with all possible delivery areas.

RMP method & Pricing method	Objective	Gap (%)	Time (min)
Exact RMP model & Exact Pricing model	4053.12	17.6	266
Exact RMP model & Randomised Construction	4053.12	17.6	130
Set Covering & Exact Pricing model	5129.28	34.9	218
Set Covering & Randomised Construction	4970.88	32.8	86

to create new delivery areas. This combination results in the best-found objective and is more than twice as fast as the methods which contain the exact Pricing model.

When we generate all possible solutions, the optimality gap to the best-found integer solution is significant, even for the best algorithm (17.6%). We keep this in mind when discussing the other experiments' results.

5.3 Determining preprocessing parameters

We test which preprocessing parameters for the procedure described in Section 4.1.1 give the best results for our problem. We examine five different combinations of two preprocessing parameters: the parameter indicating within which distance locations can be grouped (*proximity*) and the parameter indicating the maximum allowed capacity of one location (*% capacity*). The results in Table 3 show that enlarging the distance has only a minor impact on the number of subregions, as increasing the distance from 200 to 500 only reduces the subregions by 2. Seen from the most restrictive situation, the number of locations halves as either one of the variables increases. Further expanding the allowed capacity only slightly reduces the number of locations.

Table 3: This table shows the objective and computation times for delivery in Kolham on Saturdays for different values of grouping proximity and allowed capacity in one location.

Parameters *(proximity, % capacity)*	(200, 5)	(200, 7)	(500, 5)	(500, 7)	(500, 10)
# of locations	9980	4860	5424	4491	4417
# of subregions (total)	5218	3736	5216	3741	2613
Separate delivery					
Objective (€)	22417.92	22259.52	21784.32	22101.12	21691.20
Computation time (min)	201	190	366	184	220
Combined delivery					
Objective (€)	21859.20	22017.60	22334.40	21700.30	21859.20
Computation time (min)	404	184	276	179	154
Hybrid delivery					
Objective (€)	22264.32	22450.56	22991.04	21630.72	21789.12
Computation time (min)	597	233	207	304	234

In theory, the objective of hybrid delivery should be the lowest for each situation as it contains all delivery areas from separate and combined delivery. However, this behaviour only occurs for a proximity value of 500 metres and 7% capacity. The model cannot select the subregions leading to a decrease in objective in time for the other approaches, probably due to the large optimality gap as we see in Section 5.2. The hybrid model contains about twice the number of subregions as the others, potentially increasing the optimality gap. To thoroughly investigate the new delivery areas, we use the preprocessing parameter values that enable the model to pick the best type of delivery for each location. We conclude that the preprocessing parameter values of 500 metres and 7% capacity are optimal for our problem.

5.4 Impact of costs parameter

As mentioned in Section 1.1.2, mail deliverers get a lower wage than parcel deliverers. We investigate the model's sensitivity to these cost parameters in this section.

Table 4 shows the results. Our model responds intuitively to the variation in cost ratio. A relative more expensive parcel deliverer leads to an increase in mail deliverers. We also test if having more diverse costs leads to a shorter computation time. Looking at the computation time in Table 4, we see no such pattern; the computation times go up and down when the cost difference increases. Furthermore, we note that the total number of delivery areas is equal for a cost ratio of 1:1 and 1:1.26. We conclude that the current wage difference does not lead to less efficient utilisation of the delivery areas, to enhance the delivery by the cheaper mail deliverer.

The ratio of 1:2 does lead to an unexpected result, an increase in mail areas not accompanied by a decrease in parcel areas. A possible explanation is that our preprocessing parameter values are not optimal for this ratio, and better solutions exist, but the algorithm cannot reach them in time. With this in mind, we can still conclude that our model responds intuitively to the cost parameters.

Table 4: This table shows for different cost ratios the number of delivery areas specific for mail (m), parcel (p) and total (t) and the computation time.

Cost ratio *(mail:parcel)*	# of delivery areas (m/p/t)	Computation time (min)
1:1	3/135/138	273
1:1.26 (reality)	7/131/138	207
1:2	10/131/141	222

5.5 Impact of double removing method

As our model uses set covering constraints and not set partitioning constraints, our algorithm contains a separate double removing method to ensure each location is only covered once. We investigate the impact of our double removing method by comparing the results of our two double removing methods as described in Section 4.5.2, with the centroid and delivery time as a reference, respectively. Table 5 shows that the total number of delivery areas is independent of the double removing method, just like the number of delivery areas entirely focused on mail delivery. Figure 8A - B shows that different methods to remove the double locations result in a considerable difference between total delivery time per delivery area. With the longest time as a reference, mail delivery areas contain more locations, meaning that less mail is part of a combined delivery area. Almost half of the combined delivery areas become strictly parcel delivery areas. As the double removing method determines whether a delivery area is a combined or parcel area, we conclude there is no usable information in the split between parcel and combined delivery areas.

In our further analysis, we use the double removing algorithm with the centroid as reference. By using the centroid as reference, we place more emphasis on maintaining Property 1, the centrality property. However, as the double removing method greatly impacts the division between parcel and combined delivery areas, we only analyse the delivery areas by delivery: either mail or parcel, whereby delivery areas delivered by the parcel deliverer might also contain mail.

Table 5: This table shows the comparison of centroid-focused double removing and time-focused double removing on the data of Saturday deliveries in Kolham.

Double removing method	# of mail	# of parcel	# of combined	Total # of areas
Centroid	7	71	60	138
Time	7	99	32	138

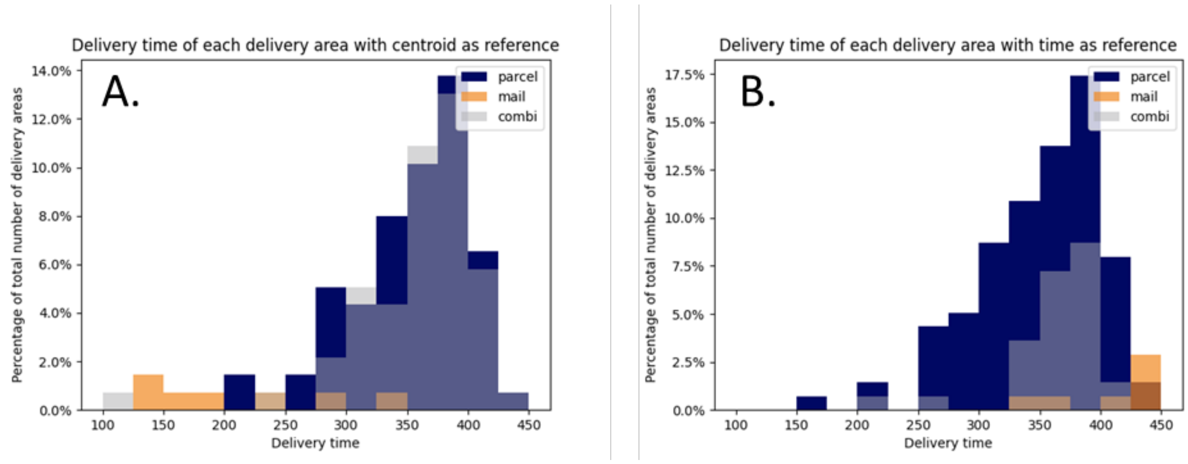

Figure 8: Histograms showing the total delivery time of each delivery area with different reference for removing double occurring locations. **A.** The centroid is the reference. **B.** The total delivery time is the reference.

5.6 Case study results

With the best algorithm and preprocessing parameters defined, we apply it to a case study of the data from the service region of the parcel depot in Kolham, a city in the eastnorth of the Netherlands. We perform our experiments within this region for two reasons. First, the northeast of the Netherlands contains a large area in which PostNL delivers mail by car, making it a suitable candidate for an extensive exploration of which types of neighbourhoods are ideal for integrating the delivery. Furthermore, to investigate the full impact on the parcel supply chain, we want to analyse an entire service area of a parcel depot. The delivery region of the parcel depot in Kolham satisfies both of these conditions.

We explore the effects of the integrated delivery on the quietest day (Saturday, Section 5.6.1), for which we think our supply chain model has the most potential, and after that, the busiest day (Tuesday, Section 5.6.2). Table 6 shows an overview of the sizes of the data sets. The data for each location is an average of eight analysed days. For each situation, we investigate three delivery options: to deliver mail and parcels separately as is currently the case, to deliver mail and parcels always combined, and the last option to choose between these, which we call hybrid delivery.

Table 6: This table shows an overview of the data used in case studies of delivery on Saturdays and Tuesdays in Kolham. It shows the number of locations specific for mail (m), parcel (p) and total (t), both before and after preprocessing and the final number of subregions.

	Saturday	Tuesday
# locations original (m/p/t)	9323/126,601/135,601	9207/145,236/154,443
# locations after preprocessing (m/p/t)	1255/3236/4491	1394/3987/5381
# subregions after preprocessing	3741	4721

5.6.1 Case study Saturday

We first analyse the delivery in the service area of Kolham on Saturdays. Figure 9 gives a general overview of the division of the different delivery areas. Table 7 shows the detailed results for the three different delivery options on Saturdays in Kolham. The option to deliver mail and parcels separately, which is the current practice, is the most expensive for our data. A fully integrated model with combined delivery of parcels and mail leads to a 3.5% decrease in required human resources, a loan saving of €400. Besides this saving in labour costs, the environmental benefits consist of an estimated reduction in kilometres of 720 km, a decrease of 9.5 %. With a fuel cost of €0.19 per kilometre, the total financial benefits account for €537 a day.

The hybrid model shows even more potential. Although the decrease in human resources and the environmental benefits are smaller, the hybrid model makes use of the lower wages for mail deliverers. The lower wages lead to an extra increase in financial benefits of €70 compared to the fully integrated approach, even though an additional employee is needed. For the hybrid model, the estimated total savings are €560 per Saturday in the region of Kolham.

Table 7: This table shows the results of the analysis of delivery on Saturdays in Kolham in terms of the objective, number of delivery areas specific for mail (m), parcel (p) and total (t) and estimated delivery distance.

Delivery method	Separate delivery	Combined delivery	Hybrid delivery
Objective (€)	22101.12	21700.30	21630.72
# of delivery areas (m/p/t)	12/130/142	0/137/137	7/131/138
Delivery distance (km)	7614	6894	7138

Now that we have established a possible improvement in business operations when we allow for combined delivery, we explore the properties of these integrated delivery areas. Our main finding is a rule-of-thumb regarding the size of mail in combined delivery areas. Figure 10A shows that at most 0.35 of the total delivery time is spent delivering mail in a combined delivery area. If the fraction is larger, it is better to deliver the mail in a separate mail delivery area.

Next, we explore the efficiency of the delivery areas. A delivery area is efficient if it contains a full-days work for a deliverer, as payment goes per day and not per hour. Figure 10B shows the delivery times per delivery area with locations still occurring in multiple delivery areas. Our model cannot use the full potential of working hours for a deliverer as Figure 10B shows. The model can do so for mail, with all delivery areas taking at least 440 minutes out of a possible 450 minutes to deliver. However, for the parcel and combined delivery areas, we see most delivery areas take between 390 and 420 minutes to deliver. That the model cannot create delivery areas fully utilising the delivery time is probably due to both the preprocessing step and the combination of assumptions. During the preprocessing phase, the algorithm can merge locations until they reach 7% of the capacity, which is 31.5 minutes. With our premises of a central seed location, it is impossible to use the potential delivery hours fully. We only consider the delivery times, as the volume of the delivery areas is never a limitation for our data set.

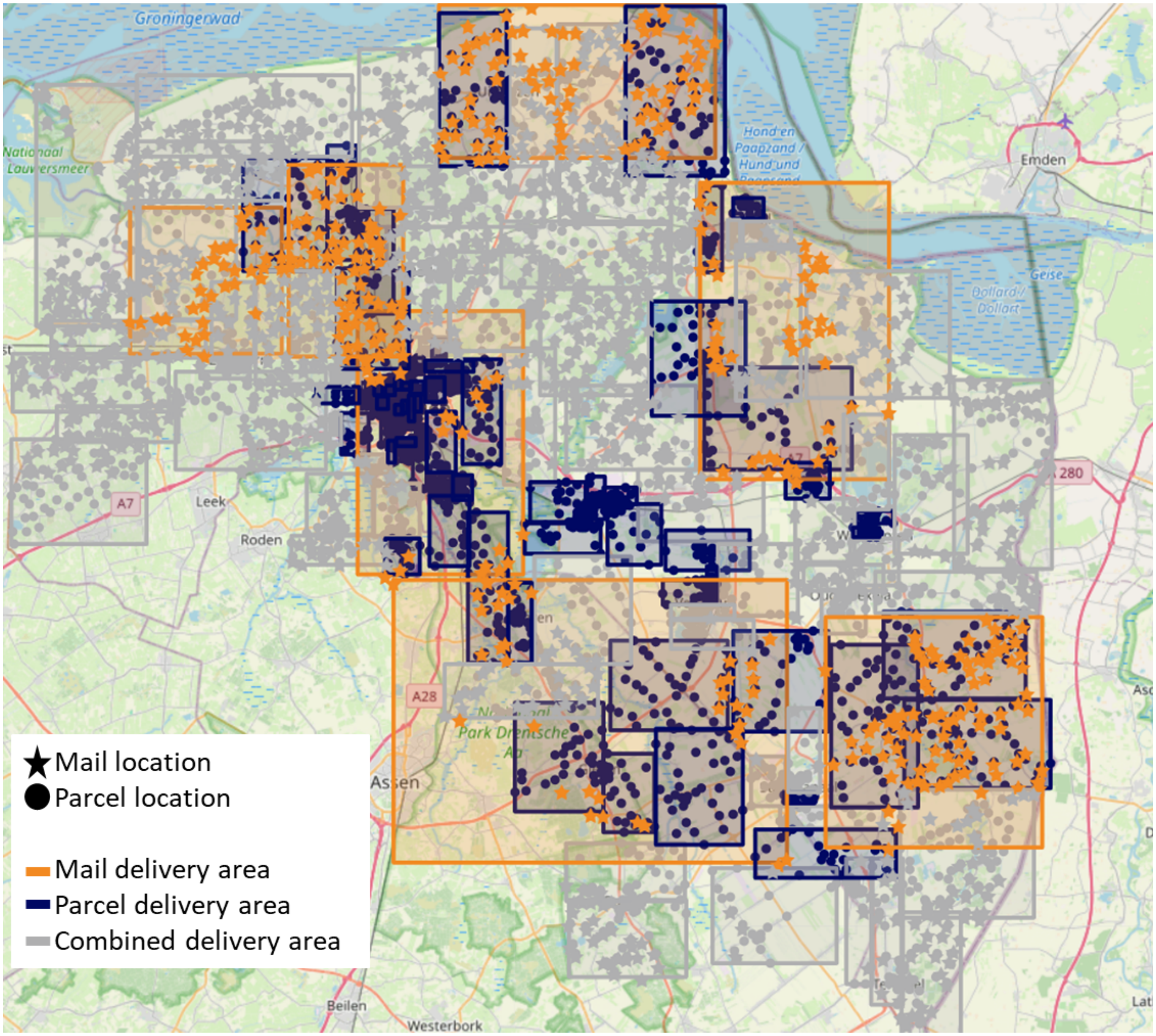

Figure 9: An overview of the delivery areas for the service area of Kolham on Saturdays. We show parcel and combined delivery areas separately for illustrative purposes. The colour of the mail or parcel location indicates to which type of delivery area it belongs.

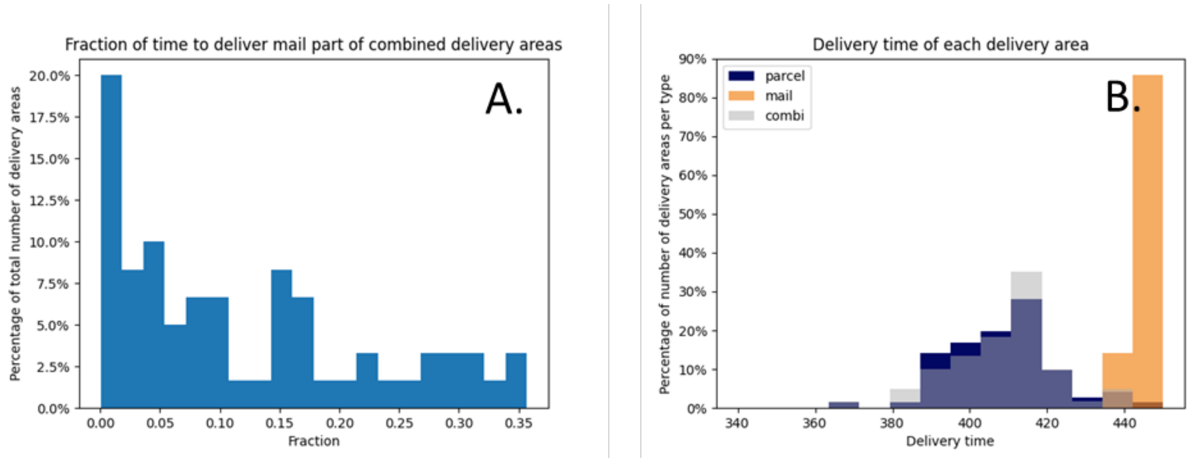

Figure 10: A. A histogram showing the fraction of time to deliver mail in combined delivery areas. **B.** A histogram showing the different delivery times for each delivery area.

5.6.2 Case study Tuesday

Now we explore the delivery on the busiest days, Tuesdays. Table 8 shows that separate delivery results in the best objective. A peculiar observation is that allowing for hybrid delivery does not result in the best objective value, which also happens in Section 5.4. Not resulting in the best objective can have several causes. First, it can again be due to the large optimality gap of the algorithm, which we show in Section 5.2 is already 17.6% for the small data set. Secondly, it can be due to non-optimal preprocessing parameters. In the experiments in Section 5.3, we see that the preprocessing parameters highly influence the algorithm's ability to find the best option for hybrid delivery. Finally, this can be due to the increase in subregions. With more subregions, the chances of randomly picking the subregion needed for improvement reduce. The implemented stopping criteria, five consecutive iterations without improvements, likely leads to an even larger optimality gap for larger data sets.

Separate and combined delivery use a similar number of subregions, so we expect comparable optimality gaps for these methods meaning that we can compare their results. Combined delivery requires one extra deliverer compared to the separate delivery, which is an increase of 0.6% in human resources. However, in total, the loan difference is €518 for one day of delivery, as the loans are lower for mail deliverers. Combined delivery has the environmental benefits of an approximate 600 km (6.5%) reduction. The fuel savings cannot counterbalance the extra loan costs as the financial benefits of combined delivery compared to the current separate delivery are €-406. Based on these results, we conclude that for Tuesday, the busiest day, it is best to maintain a separate delivery.

Similar to delivery on Saturdays, a fraction of at most 0.35 of the available delivery time is available to deliver mail, which confirms our rule of thumb previously established.

Table 8: This table shows the results of the analysis of delivery on Tuesdays in Kolham in terms of the objective, number of delivery areas specific for mail (m), parcel (p) and total (t), and estimated delivery distance

Delivery method	Separate delivery	Combined delivery	Hybrid delivery
Objective (€)	27994.56	28512.00	28851.84
# of delivery areas (m/p/t)	11/168/179	0/180/180	9/175/184
Delivery distance (km)	9126	8534	8893

5.7 Impact of only parallel delivery areas

In Section 4.1.3, we state that all delivery areas must be parallel to either the X-axis or Y-axis. We investigate the effect of this assumption by creating rotated data sets. We use the same data set of Saturday deliveries in Kolham as in Section 5.6.1, but we rotate all locations and depots by either 22.5° or 45° around the centroid of all locations.

Table 9 shows that not being able to rotate the delivery areas is a potential limitation of the model. The hybrid delivery option for our original, not rotated, data set is almost the best. Rotating the data by 45° and having separate delivery leads to an improvement of €5. However, we see that the estimated benefits in Section 5.6.1 are mere luck than wisdom. If our data, or axis, were rotated by 22.5°, the benefits would have been much smaller. It can very well be that a different rotation increases the benefits.

Moreover, when we apply a rotation of 45°, the model with hybrid delivery can no longer find the best solution. These findings are likely again due to a large optimality gap for the hybrid delivery, as we see in Section 5.4 and Section 5.6.2 as well. However, combined delivery does outperform separate delivery under all scenarios.

Table 9: This table shows the objective of the 22.5° and 45° rotated data and compares them to original data of delivery on Saturdays in Kolham.

Objective (€)	Separate delivery	Combined delivery	Hybrid delivery
Original data	22101.12	21700.30	21630.72
22.5° rotated	22101.12	22017.60	21914.88
45° rotated	21625.92	21542.40	21914.88

6 Discussion

This section discusses the limitations of our model and how future research can solve them in Section 6.1. We position our work within the theoretical research in Section 6.2.

6.1 Limitations

Although we notice in Section 5 that our model indicates substantial savings when mail and parcel delivery are integrated and delivered together on Saturdays, we should consider some significant limitations.

The most disturbing limitation is that hybrid delivery does not always result in the best objective value. However, knowing that the model contains all options available for either separate or combined delivery, we know allowing for hybrid delivery in the model must result in an objective as least as good as for the individual ones. As this is not always the case, we reach our stopping criteria, no improvement for five times while improvement is still possible. For the case study data sets, the optimality gaps are likely to be larger than for the small data sets used in Section 5.2. Several options exist to resolve this in future research. One could alter the used stopping criteria. Increasing the required consecutive iterations without improvement might lead to a more optimal solution. Another possibility is to increase the time cap on the RMP method. Moreover, one could extensively explore the impact of the tabu list. We used the values indicated by Bard and Jarrah (2013), which might not be optimal for our situation. By similar reasoning, a more extensive hyperparameter tuning for all other used parameters could also improve the results.

Secondly, we limit the flexibility of the delivery areas we can create by our assumptions. The parcel and combined delivery areas contain mostly between 390 and 420 minutes of work, instead of the allowed 450 minutes (Section 5.6.1). We base all our properties on the work of Bard and Jarrah (2013). An exception is the assumption of parallel delivery areas, which is also implemented in their work but not explicitly stated. We could not find any tested limitations of their introduced properties. To know these limitations, we investigate the impact of only parallel delivery areas in Section 5.7, which shows it is a considerable limitation. The size of the benefits depends on the rotation of the data set. However, combined delivery outperforms separate delivery under all scenarios, indicating that the quality is high enough to draw general conclusions.

Thirdly, the number of needed delivery areas for separate delivery of parcels and mail is much lower than the number of delivery areas in real life. About two times as many delivery areas exist for parcels, and for mail, it increases up to seven times. Having fewer, and thus smaller, delivery areas in our model is an indication that our model underestimates delivery times. This underestimation is because we use distances as the crow flies instead of the car drives. Furthermore, we use average velocities to calculate the time it takes to drive the distances. Even though we use different speeds when distances are below 20 km, the average rates underestimate short distances and overestimate long distances. This overestimation mainly results in too high haul times. Moreover, the estimation of the total number of kilometres in a delivery area is straightforward, and it is unclear with what quality the formula approximates the actually driven kilometres. Further work can compare the before-mentioned assumptions to reality and try to improve them where needed.

In Section 5.5, we observe the significant impact of the double-removing algorithm. Although we fully stand behind the shift from set partitioning constraints to set covering constraints in an attempt to keep the computation times within bounds, we do think it is a valuable change to incorporate the double removing algorithm within the iterative search. After we make all locations uniquely occurring, we are no longer guaranteed to satisfy the properties we enforce on our delivery areas. However, as we relax the properties this way, we might also relax them during the iterative optimisation. Relaxation of the properties will enable extra space for improvement. A good compromise between computation time and preventing covering locations very often is to include the covering as a soft constraint in the objective in future work. This way, the iterative search is already focused on preventing double-occurring locations as much as possible.

To conclude, we want to mention extrapolation limitations due to the region we investigate. We first select the day and region based on their potential to improve when we allow integrated delivery. Afterwards, we find that using Tuesdays gives significant worse results regarding the potential of combined delivery. Other regions than the service area of the depot in Kolham likely result in more minor benefits as well. Further research should analyse other areas to allow for better extrapolation of the results.

6.2 Theoretical implications

Because of the abovementioned limitations, our research can be considered a critical review of Bard and Jarrah (2013). The first idea was to mimic their model and use it to analyse the data. However, during the implementation, we deviate in three main aspects. First, we create the subregions differently. In mimicking their work, the subregions we constructed according to their procedure did not turn out to be logical, so we decided to use a different method. Second, we use two different methods to obtain an integer solution (RMP). Moreover, the idea arose to develop a heuristic instead of the exact Pricing model based on Bard and Jarrah (2013). The Randomised Constructive heuristic turns out to be of similar quality and faster, as we discuss in Section 5.2. An important note is that Bard and Jarrah (2013) does mention several extra constraints to speed up the exact Pricing model, which we did not implement. Besides these differences, we expect that the disclosure that rotating the delivery areas impacts the quality also applies to their work.

Finally, we want to emphasise the potential of the fact that we can create all delivery areas for the small data set of 2260 locations to generate a lower bound. We show there is a lot of potential in generating all delivery areas for our current assumptions and properties. A column generation approach is no longer needed when this procedure is enhanced to reduce the computation time. Solving all possible delivery areas with a commercial solver will give better results, as we see in Section 5.2 and indicates an optimality measure, which can function as stopping criteria. This approach is more straightforward, gives better results and directly shows the results' quality. When future work implements this approach, the same experiments can be conducted with higher certainty that the results are correct.

7 Conclusions

Within this section, we conclude our research. We first answer the subquestions (Section 7.1), which mainly function as a recap and summary of the work until now. Next, we answer our main research question in Section 7.2. We end it with some practical advice to PostNL in Section 7.3.

7.1 Answers to the subquestions

We answer the subquestions, which help us to summarize the process and get an overview of the approach.

1. **Which methods exist in the literature to create new optimal delivery areas?** After extensive investigation of the literature regarding Set Covering models, Capacitated Clustering models and Continuous Approximation, we find that the works of Bard and Jarrah (2013) and Lan et al. (2007) are the most important. The work of Bard and Jarrah (2013) contains the methods and frame on which we base our approach, and the work of Lan et al. (2007) explains how to construct the Set Covering heuristic.

2. **How do we formally define the problem of creating new last-mile delivery areas?** We formalise our problem to minimise the number of delivery areas used while satisfying a set covering constraint together with two capacity constraints for volume and delivery time.

3. **How do we adjust the methods from the literature to our problem?** We first make additional assumptions to fit our situation to the procedures in the literature. We assume that the delivery areas must be symmetric rectangles centred at a seed and parallel to either the X- or Y-axis. This way, we can use the framework presented by Bard and Jarrah (2013) and their RMP and Pricing method. We add the approach of Lan et al. (2007) to the possible RMP methods and create a new Pricing method, the Randomised Constructive heuristic, ourselves.

4. **What model best suits the situation of PostNL?** We conclude that the exact RMP model is best for solving the RMP, and the Randomised Constructive heuristic is best for solving the Pricing Problem. With this combination, we develop the new delivery areas, which we analyse to answer our main research question. A central finding of the experiments is that at most 0.35 of the total delivery time is spent delivering mail in a combined delivery area. If the fraction increases, it becomes better to have the mail in a separate mail delivery area.

5. **What is the quality of the model?** We discuss that several options exist to improve the quality of the model. These options include ensuring the hybrid delivery gives the best result, limiting the number of assumptions and using better time estimations. We emphasise that instead of column generation, a better approach is to generate all possible delivery areas and use them as input to a commercial solver.

7.2 Answer to the main research question

With the answers to subquestions in mind, we now answer our main research question. Our main research question is:

Is our new supply chain model for mail from car-delivery areas beneficial for PostNL in the last-mile delivery?

We answer our main research question through the lens of the three criteria as previously defined: Human resources, environmental, and financial benefits.

1. **Human resources:** For the delivery on the quietest day, Saturday, we note our new supply chain model is beneficial for PostNL. Integration has the potential to reduce the required personnel by 3.5%. For Tuesdays, the busiest day, we note that it leads to an increase of 0.6%, which is one person.

2. **Environmental benefits:** Both the hybrid and integrated models decrease the kilometres that deliverers must drive. On average, the savings are about 6% or 500 km.

3. **Financial benefits:** On Saturday, our newly proposed supply chain model leads to a daily saving of € 512 for the service area of Kolham, which can increase to € 560 when hybrid delivery is possible. However, on Tuesdays, combined delivery leads to extra costs of € 406 a day.

To conclude, we answer our main research question: our new supply chain model for mail from car-delivery areas is beneficial for PostNL in the last-mile delivery on quiet days, like Saturdays. Our model shows that keeping the delivery separately for busy days is more beneficial.

We think the model's quality is high enough that our general conclusion, as stated above, holds. However, as we explain in Section 6, our approach has several shortcomings. These shortcomings give such a significant level of uncertainty to our model that we advise to not use the actual numbers generated by our experiments.

Instead of realistic indications of the benefits, we have two main findings we want to mention. First, we find that at most 35% of the delivery time should be spent delivering mail for every deliverer. The maximum of 35% holds for both quiet and busy days.

Our second finding is an improvement to the theoretical knowledge regarding the optimisation of delivery areas. We show that instead of creating a column generation-like approach, there is a lot of potential in generating all delivery areas for our current assumptions and properties. This approach is more straightforward, gives better results and directly shows the results' quality.

7.3 Advice to PostNL

We advise PostNL first to investigate the bottleneck of the current supply chain model in more detail. An example is conducting field research with the deliverers to test the best improvement from their perspective. Moreover, PostNL should spend more time getting an overview of the current practices. When they know the current working hours of deliverers and their corresponding delivery areas, PostNL can explore possibilities for a less drastic implementation of our new proposed supply chain model. Saturday is the quietest day for both parcel and mail deliverers; it might very well be that parcel deliverers have time to deliver mail. PostNL can redistribute the mail to the parcel deliverers if this is the case while keeping the current delivery areas. This approach might already be effective, especially when new regulations will decrease mail on Saturdays as of next year. Within this approach, PostNL can use our finding that at most 35% of the delivery time should be spent delivering mail for every deliverer.

If PostNL wants to continue this research, we suggest changing the algorithm to generate all available delivery areas instead of a column generation approach. As mentioned before, this improves the generated results significantly.

Finally, we advise PostNL to adapt its way of working with an Agile mindset. The current state of the art is that once a PostNL employee draws a delivery area by hand, an algorithm indicates the expected volume and delivery times for the delivery area. The employee adjusts the delivery area until he creates a feasible delivery area. To fully automate this procedure within a thesis is a giant leap. A better approach is to improve this procedure with small incremental steps. First, PostNL can develop an algorithm to do the employee's work. Later, PostNL can realise full integration, which enables optimising all delivery areas instead of only one.

Bibliography

Egon Balas and Maria C Carrera. A dynamic subgradient-based branch-and-bound procedure for set covering. *Operations Research*, 44(6):875–890, 1996.

Jonathan F Bard and Ahmad I Jarrah. Large-scale constrained clustering for rationalizing pickup and delivery operations. *Transportation Research Part B: Methodological*, 43(5):542–561, 2009.

Jonathan F Bard and Ahmad I Jarrah. Integrating commercial and residential pickup and delivery networks: A case study. *Omega*, 41(4):706–720, 2013.

Jillian Beardwood, John H Halton, and John Michael Hammersley. The shortest path through many points. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 55, pages 299–327. Cambridge University Press, 1959.

John E Beasley. An algorithm for set covering problem. *European Journal of Operational Research*, 31(1):85–93, 1987.

Jack Brimberg, Nenad Mladenović, Raca Todosijević, and Dragan Urošević. Solving the capacitated clustering problem with variable neighborhood search. *Annals of Operations Research*, 272(1): 289–321, 2019.

José Ramón Cano, Oscar Cordón, Francisco Herrera, and Luciano Sánchez. A greedy randomized adaptive search procedure applied to the clustering problem as an initialization process using k-means as a local search procedure. *Journal of Intelligent & Fuzzy Systems*, 12(3-4):235–242, 2002.

Alberto Caprara, Matteo Fischetti, and Paolo Toth. A heuristic method for the set covering problem. *Operations research*, 47(5):730–743, 1999.

Alberto Caprara, Paolo Toth, and Matteo Fischetti. Algorithms for the set covering problem. *Annals of Operations Research*, 98(1):353–371, 2000.

CBS. Cbs in uw buurt, 2021. URL https://cbsinuwbuurt.nl/#gemeenten2020_bevolkingsdichtheid_inwoners_per_km2.

Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

Broderick Crawford, Ricardo Soto, Gino Astorga, and José García. Constructive metaheuristics for the set covering problem. In *International Conference on Bioinspired Methods and Their Applications*, pages 88–99. Springer, 2018.

Carlos F Daganzo. The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application. *Transportation science*, 18(4):331–350, 1984a.

Carlos F Daganzo. The length of tours in zones of different shapes. *Transportation Research Part B: Methodological*, 18(2):135–145, 1984b.

Carlos F Daganzo. *Logistics Systems Analysis*. Springer, 1999.

RF Daganzo, CF Souleyrette, and R Reginald. Implementing vehicle routing models. *Transport. Res. Part B: Methodol., Elsevier*, 24(4):263–286, 1990.

Yumin Deng and Jonathan F Bard. A reactive grasp with path relinking for capacitated clustering. *Journal of Heuristics*, 17(2):119–152, 2011.

S Eilon, C. D. T. Watson-Gandy, and N. Christofides. *Distribution Management: Mathematical Modelling and Practical Analysis.* Hafner, 1971.

Samuel Eilon, Carl Donald Tyndale Watson-Gandy, Nicos Christofides, and Richard de Neufville. Distribution management-mathematical modelling and practical analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, (6):589–589, 1974.

Marshall L Fisher and Ramchandran Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.

Marshall L Fisher and Pradeep Kedia. Optimal solution of set covering/partitioning problems using dual heuristics. *Management science*, 36(6):674–688, 1990.

Nader Ghaffarinasab, Tom Van Woensel, and Stefan Minner. A continuous approximation approach to the planar hub location-routing problem: Modeling and solution algorithms. *Computers & Operations Research*, 100:140–154, 2018.

Ola Jabali, Michel Gendreau, and Gilbert Laporte. A continuous approximation model for the fleet composition problem. *Transportation Research Part B: Methodological*, 46(10):1591–1606, 2012.

Patrick Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations research*, 36(6):929–936, 1988.

Ahmad I Jarrah and Jonathan F Bard. Large-scale pickup and delivery work area design. *Computers & operations research*, 39(12):3102–3118, 2012.

Guanghui Lan, Gail W DePuy, and Gary E Whitehouse. An effective and simple heuristic for the set covering problem. *European journal of operational research*, 176(3):1387–1403, 2007.

André Langevin, Pontien Mbaraga, and James F Campbell. Continuous approximation models in freight distribution: An overview. *Transportation Research Part B: Methodological*, 30(3): 163–188, 1996.

Gilbert Laporte. Fifty years of vehicle routing. *Transportation science*, 43(4):408–416, 2009.

Richard C Larson and Amedeo R Odoni. *Urban operations research.* Number Monograph. 1981.

Lucas Lessing, Irina Dumitrescu, and Thomas Stützle. A comparison between aco algorithms for the set covering problem. In *International Workshop on Ant Colony Optimization and Swarm Intelligence*, pages 1–12. Springer, 2004.

Mark Lewis, Haibo Wang, and Gary Kochenberger. Exact solutions to the capacitated clustering problem: A comparison of two models. *Annals of Data Science*, 1(1):15–23, 2014.

Luiz AN Lorena and Edson LF Senne. A column generation approach to capacitated p-median problems. *Computers & Operations Research*, 31(6):863–876, 2004.

N Mahmoodi Darani, P Bassiri, and M Yousefikhoshbakht. An effective algorithm in order to solve the capacitated clustering problem. *Journal of New Researches in Mathematics*, 1(4):81–102, 2016.

Anuj Mehrotra and Michael A Trick. Cliques and clustering: A combinatorial approach. *Operations Research Letters*, 22(1):1–12, 1998.

John M Mulvey and Michael P Beck. Solving capacitated clustering problems. *European Journal of Operational Research*, 18(3):339–348, 1984.

Marcos Negreiros and Augusto Palhano. The capacitated centred clustering problem. *Computers & operations research*, 33(6):1639–1663, 2006.

Gordon Frank Newell. Scheduling, location, transportation, and continuum mechanics: some simple approximations to optimization problems. *SIAM Journal on Applied Mathematics*, 25(3):346–360, 1973.

Temel Öncan. A survey of the generalized assignment problem and its applications. *INFOR: Information Systems and Operational Research*, 45(3):123–141, 2007.

Ibrahim H Osman and Nicos Christofides. Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, 1(3):317–336, 1994.

PostNL. Mail in the netherlands. URL https://www.postnl.nl/en/about-postnl/about-us/mail-in-the-netherlands/.

PostNL. *PostNL annual report 2021*. 2022. URL https://annualreport.postnl.nl/2021.

Karen R Smilowitz and Carlos F Daganzo. Continuum approximation techniques for the design of integrated package distribution systems. *Networks: An International Journal*, 50(3):183–196, 2007.

David M Stein. An asymptotic, probabilistic analysis of a routing problem. *Mathematics of Operations Research*, 3(2):89–101, 1978.

Matthias Winkenbach, Alain Roset, and Stefan Spinler. Strategic redesign of urban mail and parcel networks at la poste. *Interfaces*, 46(5):445–458, 2016.

Qing Zhou, Una Benlic, Qinghua Wu, and Jin-Kao Hao. Heuristic search to the capacitated clustering problem. *European Journal of Operational Research*, 273(2):464–487, 2019.

A Detailed derivations and algorithms

A.1 Detailed calculation of delivery distances

To estimate the expected driving time, we first need to estimate which neighbour will be delivered next. Based on this information, we calculate the expected driving time, which is the distance divided by the average speed. However, as uncertainty plays a crucial role in this calculation, we consider a range of neighbours, each with a different weight based on their distance and probability that they are delivered. The extensive calculations are described next, based on the expressions used by Bard and Jarrah (2009).

First, we calculate the delivery probability per location for both mail and parcels (π_{mail} and π_{parcel}). For location i, we do so with the following formulas

$$\pi_{mail,i} = \frac{\text{Unique dates of mail delivery to } i}{\text{Total unique delivery dates}} \text{ and } \pi_{parcel,i} = \frac{\text{Unique dates of parcel delivery to } i}{\text{Total unique delivery dates}},$$

for mail and parcel respectively. The extra index to indicate whether the calculation is for parcel or mail values is purely for explanatory means, as location i can only receive either mail or parcels. When an address receives both, it becomes two locations. We drop the extra index for further explanations.

Now that we know the delivery probability of each location, we can calculate the likelihood $w_{i,[j]}$ that location $[j]$ follows location i on the route. We do so by combining the delivery probability (π), the ranking of the jth closest locations to i ($R_{i,[j]}$) and the number of neighbours considered ($n = 15$) in the following way

$$w_{i,[j]} = \frac{\pi_j(n - R_{i,[j]} + 1)}{\sum_{l=1}^{n} \pi_j(n - R_{i,[j]})},$$

thereby combining the chances that a neighbour is delivered and the relative closeness to this neighbour.

Next, we use this likelihood to calculate the delivery time

$$t_i = \sum_{j-1}^{n} w_{i,[j]} \frac{D_{(i,[j])}}{v_{(i,[j])}},$$

whereby $D_{(i,[j])}$ is the distance to location $[j]$ and $v_{(i,[j])}$ is the average velocity when traveling there. $v_{(i,[j])}$ is equal to 30 if the distance is smaller than 20 km, and 50 otherwise. In the end the average delivery time is calculated as

$$\tau_i = \pi_i(STOP_i + t_i),$$

with $STOP_i$ being the norm time for the actual delivery of the items. For parcel locations, $STOP_i$ is increased by 0.1 minutes for every package that PostNL delivers at that location. Finally, we multiply the expression by the delivery probability to account for irregular delivery at this location.

A.1.1 Combining the data sets

With the key properties calculated for the separate data sets, we use these properties when combining both data sets. The grouped data set maintains locations where mail and parcel are delivered as separate locations to allow the model to decide between combined or separate delivery. However, the expected delivery time will be different for the combined delivery as the network is denser, and PostNL can deliver mail and parcels in one go.

The expected driving time to the next location changes automatically when we apply the calculations as explained in Section A.1 to the combined data set, as more, and thus closer, locations can be the next location. However, as PostNL can deliver mail and parcels at one location, the delivery probability of each location changes. We assume that the delivery probabilities of delivering mail and parcel at one location are independent. We use a Bernoulli trial to calculate the probability of a combined delivery $\pi_{\text{combined},i} = 1 - (1 - \pi_{\text{parcel},i})(1 - \pi_{\text{mail},i})$. We also assume that for combined delivery, the actual delivery time is equal to the average norm time of parcel delivery, indicating that delivery can include mail without inducing extra delivery time. A mail deliverer can often deliver the mail while seated in the car if the post box is near the road. However, this is not the case for a parcel deliverer, as the seating in a parcel van differs. We, therefore, assume that delivery of mail takes ten more seconds when delivered from a parcel van. The delivery time for the mail location becomes

$$\tau_{\text{combined,mail},i} = (STOP_{\text{combined,mail},i}(\pi_{\text{mail},i} - \pi_{\text{combined},i}) + \pi_{\text{mail},i} t_{\text{combined},i}.$$

For the parcel delivery the delivery time is calculated with the same formula as previously,

$$\tau_{\text{combined,parcel},i} = \pi_{\text{parcel},i}(STOP_{\text{parcel},i} + t_{\text{combined},i}).$$

The model uses these values of τ_{combined} for combined delivery, and the previously calculated values of τ_{mail} and τ_{parcel} for separate delivery. Again note that the index indicating whether the values regard mail or parcel are purely illustrative.

A.2 Creation of a grid

In Section 4.1.2, we explain the basics behind grid construction. Algorithm 2 provides the applied procedure in more detail.

Algorithm 2 Overview of grid procedure (*createGrid*)

Input: `preprocessed_locations`

`list_X` $\leftarrow$ $sortOnXCoordinate$(`preprocessed_locations`)

`list_Y` $\leftarrow$ $sortOnYCoordinate$(`preprocessed_locations`)

$\texttt{N}_{unassigned}$ $\leftarrow$ `preprocessed_locations` ▷ to keep track of the unassigned locations

while $\texttt{N}_{unassigned} \neq \emptyset$ **do**

  `range_X` $\leftarrow X^{max}(\texttt{N}_{unassigned}) - X^{min}(\texttt{N}_{unassigned})$ ▷ easily obtained from list_X

  `range_Y` $\leftarrow Y^{max}(\texttt{N}_{unassigned}) - Y^{min}(\texttt{N}_{unassigned})$ ▷ easily obtained from list_Y

  `ratio` $\leftarrow \frac{range\_X}{range\_Y}$

  `lowerbound` $\leftarrow \max(\lceil \frac{demand(N_{unassigned})}{xQ} \rceil, \lceil \frac{time(N_{unassigned})}{xT^{max}} \rceil$

  `vertical_lines` $\leftarrow \lceil \sqrt{ratio \cdot lowerbound} \rceil$

  if $\frac{demand(N_{unassigned})}{xQ} > \frac{time(N_{unassigned})}{xT^{max}}$ **then**

    `exp_locations` $\leftarrow \lceil \frac{demand(N_{unassigned})}{vertical\_lines} \rceil$

  else

    `exp_locations` $\leftarrow \lceil \frac{time(N_{unassigned})}{vertical\_lines} \rceil$

end if

`horizontal_locations` $\leftarrow$ $selectTillExpLocationsNew$(`list_X`) ▷ behind the last added location the vertical line is drawn ▷ For a line at the right, loop list_X backwards

while horizontal_locations $\neq \emptyset$ **do**

  `grid_locations` $\leftarrow$ `grid_locations` + $selectTillxCapacity$(`list_Y`, `horizontal_locations`) ▷ each time x times the limiting capacity has been reached a horizontal line is drawn

end while

`list_X` $\leftarrow$ `list_X` $\setminus$ `horizontal_locations`

`list_Y` $\leftarrow$ `list_Y` $\setminus$ `horizontal_locations`

$N_{unlocated} \leftarrow N_{unlocated} \setminus$ `horizontal_locations`

end while

return `grid_locations`