Construction and Application of a Memetic Algorithm Assigning Catchment Areas to Retailers for Consumer Parcel Flow

Master's Thesis

Author: Philip Moll (431983)

ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics
Econometrics and Management Science
Operations Research and Quantitative Logistics

Erasmus supervisor: Prof. dr. ir. Rommert Dekker
Erasmus co-reader: Dr. Remy Spliet
PostNL supervisor: Mathijs Udo

April 7, 2017

Abstract

The ongoing growth of the parcel shipping business at PostNL causes an increasing pressure on retailers that offer storage services, as storage capacity is limited. A significant share of this pressure follows from the catchment area of a retailer, the set of households for which a retailer stores certain types of parcel flow. By properly adjusting catchment areas according to the expected parcel volumes and retailer capacities, storage pressure can effectively be alleviated before turning to costly capacity expansion or acquisition of new retail locations. This thesis models the problem of determining suitable catchment areas as a Generalized Assignment Problem (GAP), minimising customer traveling distance while respecting capacity limitations at retailers. The constructed stand-alone solution method involves problem size reduction, LP-Relaxations, a Memetic Algorithm and iterated clustering, which all have been benchmarked against licenced optimisation software (Gurobi). Performance has been tested on several regions in the Netherlands, for which the results show that the method is capable of feasibly mapping catchment areas while minimising travelling distance in acceptable computation time. The method has been developed into a tool for PostNL and contributes in the decision making process of retailer catchment areas.

Keywords: Generalized Assignment Problem (GAP), Memetic Algorithm, Parcel Flow

Contents

1 Introduction

Driven by digitalisation and e-commerce, the mail and parcel industry has been drastically changing in recent years. The rise of online communication platforms and ease of connecting through mobile systems have caused a strong decline in the rate of letters sent. Simultaneously, parcel shipping shows an opposite trend as an effect of increased world trade and online shopping. As discussed in Bogers et al. (2014), globalisation and privatisation among postal companies increase competition, stimulate exploration of international markets and add pressure on daily operations. Such developments call out for efficiency transformations at PostNL, the main mail, parcel and e-commerce company in the Netherlands. Annual quantities of their two largest business units, mail and parcel shipping, are shown in Figure 1.

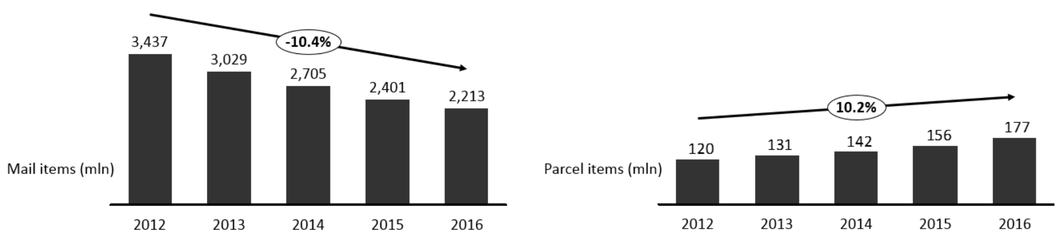

Figure 1: Compound annual growth rate of mail and parcel items shipped, PostNL (2012-2016)

Due to the changing demand for its services, PostNL needs to reconsider their currently applied methods to collect, sort, store and distribute mail and parcels efficiently. For instance, sorting centers to collect and sort mail have been closed down, merged or centralised, while new parcel sorting centers have opened to feasibly process the increasing amount of flow. As both markets are rapidly changing, the ability to adapt to new capacity requirements is crucial for PostNL. This thesis focuses on implications for the increasing parcel flow. More specifically, the focus will be on the increasing B2C and C2C flow of parcels that pass through retailers. A simplistic overview of parcel flow is shown in Figure 2.

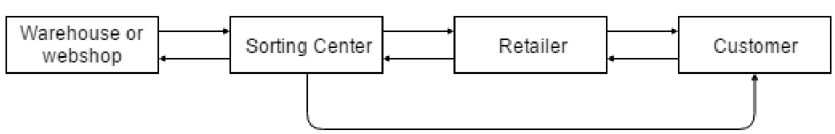

Figure 2: Flow of parcels

One important difference in the collection and delivery process of mail and parcels is the role of the retailer, one of many storage locations close to the customer. While mail can be delivered at customers' letterboxes any time of the day, parcels generally do not fit and therefore require to be accepted at the moment of delivery. Currently, a total of 2,850+ retailers in The Netherlands such as Bruna, Primera or local post offices offer capacity to temporarily store parcels for customers to pick up (or send) at a moment of their preference. One of the implications of the extra flow in parcels is the increased pressure on retailers to process and store parcels, especially in dense areas with relatively few retailers per household and where capacity expansion is often costly or infeasible. To alleviate pressure on retailers, computer-aided distribution tools may help to better match parcel flow with storage capacity. This thesis aims to construct a method to feasibly and efficiently distribute parcels over retailers by means of reconsidering the customer service area of a retailer, also referred to as its catchment area. This area denotes the set of households for which a retailer accepts certain types of parcel flow.

1.1 Problem Description

Retailers are crucial for PostNL as they provide a first contact point for customers to its services. Therefore, having a retailer with high service level close to many customers is a requisite for PostNL, not a bonus. For retailers, the reason for offering the service to process and store parcels shipped by PostNL is often twofold. First, the service results in a financial compensation by PostNL. However, the yield of the time consuming parcel processing in combination with volume reservation, is relatively low and has been decreasing the past years. More importantly, the service attracts extra customers into the retail shop, potentially spending money on other products or services once in the shop.

Due to the increasing flow of parcels at retailers, a growing number of retailers faces problems to feasibly store the received parcels, forming a bottleneck in the delivery process. If the intended storage space in the back office is full, a retailer is forced to keep parcels piled up in the front office. In such cases of disorganised storage, available shopping space decreases and the average time required to serve a customer rises, something I personally experienced while assisting a day during peak season at a Primera in Houten. Consequently, storage insufficiency limits the service quality.

As parcel storage capacity expansion for a retailer is often undesired or infeasible, retailers ask PostNL to unburden them by better distributing parcels over retailers according to their storage capacity. Simultaneously, customers wish to collect their parcel at a retailer conveniently close to their home. This combination results in two often conflicting objectives. Delivering a parcel at a retailer with available capacity far away from its destination, will likely lead to a dissatisfied customer. On the contrary, by solely considering the distance from parcel destination to a retailer, retailers in dense regions with many parcel receiving customers will face capacity issues. In such situations, retailers see parcel volume outgrow capacity, which in turn is inconvenient for customers and retailers. Dissatisfied customers turn to parcel shipping alternatives, while dissatisfied retailers might decide to completely stop offering parcel storage capacity, shifting the problem from one retailer to the next.

Traditionally, each retailer has been assigned a set of unique households for which it stores certain flows of parcels. A parcel which needs storage will be directed to the retailer which contains the household of the consignee in its catchment area. Due to the development of new residential districts, stochastic and dynamic parcel consumption and cases of retailers that permanently close, switch location or open up, the households in a catchment area of a retailer must change over time. Currently, catchment areas are a result of manual adjustment on adjustment as altering is often induced by a complaint by a retailer who has recently faced multiple incidents of capacity deficiency. PostNL alleviates parcel pressure on this retailer by assigning a proportion of households of its catchment area to a catchment area of a neighbouring retailer where required capacity may be available. Note that these adjustments are manually processed the moment a retailer faces or has faced capacity problems, while ideally such problems are prevented by planning ahead. In addition, from a customer perspective these adjustments on adjustments have resulted in assignments of households to retailers that, given capacity restrictions are respected, not necessarily minimise traveling distance. Therefore, the process to determine the catchment areas of retailers can be improved, both from retailer as from customer perspective. Ultimately, improvement in the decision making will benefit all three parties involved: customers, retailers and PostNL.

1.2 Research Objectives

In Section 1.1 the problem of currently inadequate matching process of households to retailers has been described. Therefore, the main research objective of this thesis is defined as:

"Can an efficient method be constructed and implemented at PostNL to improve decision making in the assignment of sets of households to retailers such that (1) the available capacity at retailers is respected and (2) the travelling distance for customers to retailers is minimised?"

The aim is to create a stand-alone solution method, such that no commercial or licenced software is required. Relevant questions in the process of modeling the situation at retailers and applying an appropriate method include: which types of parcel flow at retailers can be distinguished and which can be influenced by PostNL? Which requirements do retailers and customers have and how can these be processed into a mathematical optimisation model? Which method is most appropriate to solve the model? Can the method help to anticipate on future capacity problems?

In order to answer these questions, current parcel flow needs to be examined and requirements from both retailers and customers are to be considered. Together with several assumptions, the gained information will be processed into a mathematical optimisation model. Guided by recent research on similar problem formulations, a solution method will be constructed, tested and improved. The performance can be benchmarked against solutions found by theoretical bounds and licenced optimisation software (Gurobi), where the latter is only temporarily available through the Erasmus network.

The relevance of the research is threefold. Firstly, it serves as a tactical tool for PostNL to help unburden retailers without costly capacity expansion, while meeting customer and retailer expectations. Secondly, in case redistribution of households is insufficient to feasibly allocate expected parcel flow over retailers in a certain region, the proposed method helps to detect future capacity insufficiency. The method could then support focused contract negotiations with potential new retailers to increase capacity in an understaffed retail area. From an academic perspective, this research brings together and extends related work on a mathematical optimisation problem in order to construct an efficient method.

This thesis is structured as follows. Section 2 elaborates on most important definitions and assumptions used throughout this thesis. In Section 3, a mathematical formulation of the problem is described, on which Section 4 provides a literature review. Section 5 describes the proposed solution method, after which Section 6 synthesises the outline and presents experiments to test performance. Results are discussed in Section 7, followed by a conclusion in Section 8 and discussion on practical considerations in Section 9.

2 Terminology and General Assumptions

This section provides a qualitative elaboration on the most important definitions and assumptions used throughout this thesis and briefly explains their relation to the research problem.

2.1 Flow of Parcels

Reconsidering the flow of parcels as in Figure 2, PostNL distinguishes 4 different flows directly relevant for retailers and provides these with labels from a retailer perspective. An additional fifth flow affects retailers indirectly and is therefore also mentioned in this section for completeness. These flows will often be referred to throughout this thesis and are visualised in Figure 3.

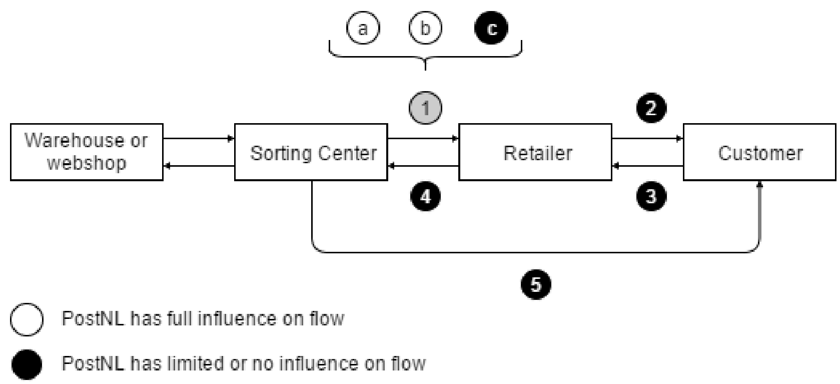

Figure 3: Types of parcel flow

1. *'Provide'*
   The retailer is provided with parcels from a sorting center, which customers can pick up at a moment to their preference. Within this *provision* of parcels, three different types of flow can be segmented:

   (a) *'Two times no answer (2TNA)'*
   In case a customer (or its neighbour) gives no answer at the moment of delivery by PostNL, the parcel returns to the sorting center in order to have a delivery attempt the next day. In case delivery the next day yields a second no answer, the parcel is shipped to a retailer. The customer is alerted at which retailer the parcel can be collected. PostNL decides at which retailer the parcel is stored, as denoted by the light circle of flow (a) in Figure 3.

   (b) *'Asian shipping (ASIA)'*
   Since mid 2015, Asian countries have been granted certain privileges to use services of PostNL at reduced cost in order to stimulate world trade. As an effect, parcel shipping from Asia to the Netherlands has been steadily increasing and has since then been separately labeled in the delivery process as one of the three flows within *provide*. These parcels from Asia are directly shipped from sorting center to a retailer and no costly and time consuming delivery attempts at customers are conducted. The customer is alerted at which retailer the parcel can be collected. PostNL decides at which retailer the parcel is stored.

(c) *'Customer request (CR)'*

An increasing amount of webshops offers customers a list of retailers at which the parcel can be stored in order to reduce time consuming delivery attempts. Through the webshop, the customer decides at which retailer the parcel is delivered. Therefore, PostNL has no influence on the *provision* flow of *CR*.

2. *'Serve out'*

A customer is served with its parcel originating from flow *provide* by visiting the retailer at a moment of its preference within two weeks after the parcel has been provided to the retailer. Throughout this thesis it is assumed PostNL has no influence on this flow. One might argue that by encouraging customers to quickly collect their parcel, the shelf time of parcels at retailers reduces and the average available capacity increase. However, the set of such steerable incentives by PostNL such as reminders or financial fines is limited and is not further elaborated on in this thesis. In addition, one might argue that a parcel stored at a retailer close to the customer results on average in less shelf time at the retailer in comparison to a situation in which a customer lives further away from the retailer. As storage capacity problems are mainly experienced in urban areas, the effect of distance between retailer and customer on the shelf time is not considered. As multiple retailers are often close to the consumer, relatively little difference in distance is likely to occur. Therefore, we assume no influence by PostNL on this flow.

3. *'Accept'*

In addition to the parcel storage service, most retailers offer customers the service to send out a parcel, either those parcels that need to be returned to a webshop, or those that must be send to another household. Retailers accept all parcels that customers bring in and customers choose which retailer they visit. Throughout this thesis it is assumed PostNL has no influence on this flow.

4. *'Skim'*

PostNL skims parcels that have either been on shelf for two weeks by flow *provide* or have been supplied by customers at a retailer *accept*. The parcels that have been on shelf for two weeks will be shipped through the sorting center and return to the sender, while those brought in by customers will be shipped through the sorting center to its destination. Skimming is performed multiple times on a daily basis by the same PostNL delivery employees who bring in the flow *provide*. Extra skimming would be costly and simultaneously have limited effect on the average volume of parcels present at retailers. Therefore, we assume no influence by PostNL on this flow.

5. *'Direct delivery'*

This flow consists of parcels that are accepted by the customer (or one of their neighbours) at the first or second delivery attempt such that the retailer is circumvented in the delivery process. The parcels that remain undelivered after two attempts, will be labeled as described in flow *2TNA*. By means of discussing time windows with customers, the hit rate, the ratio of direct deliveries over delivery attempts, might be increased. Such initiatives are currently tested in order to alleviate the costly parcel pressure at retailers, but not elaborated on in this thesis. We assume no influence by PostNL on the direct delivery flow and its effect on the *2TNA* flow.

As has been visualized in Figure 3, PostNL has influence on flows *2TNA* and *ASIA* by deciding to which retailer a parcel of this type is shipped. These two flows will be considered in the rest of this thesis. To illustrate their relative sizes, 70% of parcels shipped in the flow *provide* have been labeled as either *2TNA* or *ASIA* in the Netherlands from Q4 2015 up to and including Q3 2016. Key observation from this number is that the magnitude of the two parcel flows which are steerable by PostNL (*2TNA* and *ASIA*) is significant.

Currently, a retailer catchment area is forced to consist of a set of customer households for which both the flows *2TNA* and *ASIA* are accepted. From a customer point of view, this indicates a single retailer needs to be visited to collect any of the two types of parcel, bringing extra convenience. By allowing decoupling these flows, households might need to visit differing retailers in order to obtain their parcels, depending on the parcel type. While such a split likely reduces convenience, decoupling of the two flows also brings extra flexibility to determine feasible catchment areas. Therefore, looking into the potential benefits of decoupling the flows is an interesting direction.

2.2 Capacity

Each retailer has limited back office capacity to store parcels, which can be divided into two separate types of capacity: shelf space (m^3) and container space (m^3).

1. *'Shelf space'*
   This storage space is intended for incoming flow of *provide* as in Figure 3 until the parcel is *served out* to the customer. Parcels stored on shelf space must be easy to reach, such that a visiting consumer can be quickly served with its parcel. In case the parcel is not *served out* within two weeks time after *provision*, the parcel is removed from shelf and put in the second type of capacity, which consists of container space.

2. *'Container space'*
   This form of storage capacity stores the flows *accept* brought in by consumers and those parcels which have been on shelf for two weeks. Parcels stored in containers are directed to the sorting center and as direct access to these parcels is not necessary, piling them up in a container is allowed. With a rate of often multiple times a day, containers are *skimmed* by the vans which supply retailers with parcels.

These two types of capacity, shelf space and container space, are completely separate in order to prevent mix up of processes. Using both types of capacity for similar flow is undesired and might lead to unintended extra shipping, sorting actions and ultimately decrease service for customers. Based on retailer experience, shelf space is under pressure, which is the storage space for the flows *2TNA* and *ASIA*. The moment a retailer is structurally short on shelf space while it has ample container capacity, reclassification of storage room would be reasonable. Due to the high level of independence between the two types of capacity, this thesis solely focuses on shelf capacity, which can be interpreted as fixed volume (m^3) per retailer.

2.3 Capacity Problems and Potential Solution Directions

Now all the types of flow relevant to retailers have been discussed, consider the shelf space storage burden this brings at retailers. This burden on any moment in time can be decomposed into its two main factors: (1) the amount of parcels to be stored and (2) their corresponding volume (length times width times height). Given a fixed maximum shelf storage capacity at a retailer, reducing amounts or volumes, would reduce capacity issues at a particular moment in time. However, PostNL and retailers benefit from increasing amounts and volumes, gaining extra financial compensation for its services. Simultaneously, PostNL needs to respect the capacity limitation of each individual retailer, imposing a restriction on the objective to infinitely increase amounts and volume of which a fraction of parcels pass through retailers. As an alternative, reducing capacity problems can also be achieved by circumventing the retailer in the delivery process. However, recall that influence on the flow *direct delivery* is assumed to be out of scope, implying the total amount of parcels with their corresponding volume that flows through the total set of retailers is assumed to be non-steerable by PostNL. One way to reduce parcel pressure at retailers over time is to reduce the time a parcel needs to be stored at a retailer, referred to as the shelf time. For this factor, recall that the influence on the flows *serving out* and *skimming* is assumed to be limited and considered out of scope throughout this thesis for reasons stated in the previous section. Thus, factors contributing to the burden at retailers are assumed to be non-steerable by PostNL and considered out of scope in this study.

Given the amount, volume and shelf time of parcels to be stored at the total set of retailers, an obvious alternative to reduce the capacity burden is to increase shelf capacity. This either be achieved by reserving or acquiring additional back office space for existing retailers, or by cooperating with new location which currently do not store parcels for PostNL. However, expansion of shelf space for an existing retailer is often undesired as it is costly or would directly reduce space destined to store other products. Therefore, partnering up with new retailers might be a solution to alleviate storage burden in a region, incurring a corresponding set up cost for PostNL.

Before turning to any of the mentioned potentially fruitful but costly resorts of capacity expansion, this thesis reconsiders the currently applied rules for deciding which retailer is assigned to process and store certain flow. As discussed, the significant flows *2TNA* and *ASIA* are flexible in the sense that PostNL can decide for any parcel receiving household in the Netherlands at which retailer the parcel must be stored. By reconsidering the catchment area of each individual retailer based on their capacity, storage space might be better utilised, limiting expansion costs. In case a region still contains retailers with long term insufficient capacity after efficiency gains in catchment areas have been achieved, expansion is required.

2.4 Locations, Distance Measures and Catchment Areas

In order to decide to which retailer a parcel of flow *2TNA* or *ASIA* must be shipped, consumer households and retailers need to be evaluated based on their location. By means of the RD coordinate system (*Rijksdriehoekscoördinatensysteem*), a commonly applied Cartesian system with meter as unit measure, every location in the Netherlands (including retailers and customer households) has a unique centre of gravity, a combination of strictly positive X and Y values. X is increasing in the west to east direction and

Y in the south to north direction. Therefore, each location can be identified and compared based on their unique pair of RD coordinates. Practical reasons for applying this distance measure is its availability and computational simplicity. A limitation of this system arises in case no direct connection exists between locations. Most significant practical problems occur when obstacles such as rivers or highways prevent a customer from conveniently reaching the retailer at which its parcel is stored. Therefore, by using the RD coordinate system, one needs to remain critical when assessing the practical value of matches between retailers and customers. Other distance measures such as distance by road or distance by foot would be preferred but are not applied due to lack of availability. Thus, the RD coordinate system will be applied throughout this thesis, but note its limitations when implementing solutions in practice.

In addition to a pair of RD coordinates, every address in the Netherlands has been given a postal code by PostNL. Such postal codes are a combination of four numbers and two characters, for instance '3011AB'. One method applied by PostNL to define different sets containing consumer households is based on varying level of detail in postal code: PC4 ('3011'), one of its subsets PC5 ('3011A') and corresponding subset PC6 ('3011AB'), sizable on the number of elements. As neighbouring households tend to have similar postal codes, households can easily be grouped together based on their postal code. Note that only the combination of PC6 with street number necessarily defines a unique household, where other sets likely define an area containing multiple households. Consider the following illustrations for clarification.

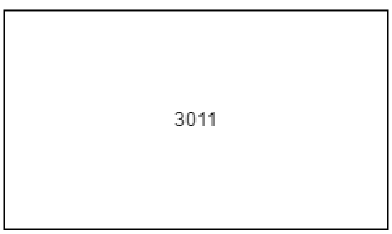

Figure 4: PC4 level

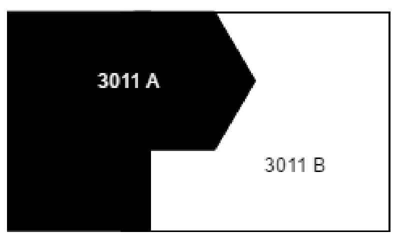

Figure 5: PC5 level

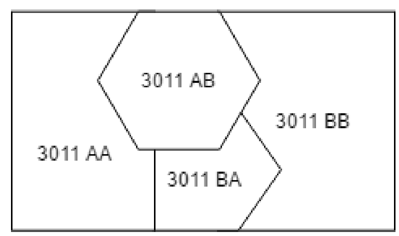

Figure 6: PC6 level

In Figures 4, 5 and 6 a fictive region is separated based on PC4, PC5 and PC6, respectively. Now let H be the set containing all households in the area with $|H| = m$ elements. For the case under consideration this implies the PC4 '3011' has m elements. Denote for each element $i \in H$ the corresponding RD coordinate as (X_i, Y_i). Then, as the RD coordinates of customer households are strictly positive, define the RD coordinate of PC4 '3011' as the weighted center:

$$(\bar{X}, \bar{Y}) = \left(\frac{\sum_{i=1}^{m} X_i}{m}, \frac{\sum_{i=1}^{m} Y_i}{m} \right) \tag{1}$$

Similar methodology can be applied to calculate RD coordinates for each PC5 and PC6 area in Figure 5 and 6. Possessing the RD coordinates at PC4, PC5, PC6 as well as the individual household level, allows for adjustments in the level of detail when assigning households to retailers, which will be useful for computational reasons in later stage.

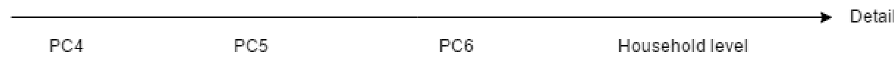

Figure 7: Level of detail on locations for consumer households

As a rule imposed by PostNL, households with similar PC6 are required to be served by the same retailer, limiting the the assignment of catchment areas to PC6 level. This rule will last during the entire thesis.

2.5 Problem Instances

Most parcel capacity problems occur in dense areas where storage capacity is scarce and costly. In particular, many retailers in Amsterdam have recently struggled to store the parcels they have been assigned to process. As largest improvement potential can be found urban areas, the proposed method will be applied on Amsterdam and the medium sized city Breda. In order to test performance of the algorithm in a region with rather different characteristics, the province of Flevoland is included as problem instance. This region consists of the dense cities Almere and Lelystad, but also includes rural areas. See Table 1 for a general overview the three problem instances which will be considered throughout this thesis.

Region or city	# Retailers	# Households	# PC6	# PC5	# PC4
The Netherlands	**2,866**	**8,630,593**	**455,140**	**32,954**	**4,049**
1. Amsterdam	93	442,748	16,711	1,018	74
2. Flevoland	70	186,032	11,196	833	94
3. Breda	23	81,962	3,195	228	21

Table 1: Overview of three problem instances to be considered

3 Problem Formulation

Incorporating the previously described terminology and assumptions, the problem boils down to feasibly allocating the parcel burden caused by steerable flows *2TNA* and *ASIA* per PC6 element to a retailer with available shelf space, minimising distance to be covered by customers to collect their parcel at the retailer. In order to mathematically formalise the problem, the following notation will be applied to any region containing retailers and sets of households. Determination of appropriate values for this notation will be discussed in Section 5.2.

3.1 Retailers

Let R be the set of retailers in a region under consideration, with $|R| \geq 2$. Each retailer $i \in R$ has a pair of strictly positive RD coordinates (X_i, Y_i) and a strictly positive maximum shelf space volume (m^3) capacity, denoted as $c_i > 0$. As some of the shelf space will be lost due to inefficient packing caused by heterogeneous volumes of parcels, define $\alpha \in [0, 1]$ as the proportion of practically effective capacity at retailers. In addition, by including $\beta \in [0, 1]$, a buffer can be provided, resulting in more robust solutions in case parcel flow strongly exceeds expectation. Define $\bar{q}_i \geq 0$ as a parameter for volume (m^3) caused by the non-steerable *Customer Request* flow for each retailer $i \in R$.

3.2 Customer Households

Any region under consideration contains a fixed amount of households. Define P_h as the set containing elements based on household level, such that $m_h \geq 1$ denotes the number of households that is present in a region. Next, define the sets P_4, P_5 and P_6, containing the elements corresponding to groups of households which possess similar PC4, PC5 and PC6, respectively. Let m_4, m_5 and m_6 denote the number of elements in the sets P_4, P_5 and P_6, respectively. Note that by the nature of subsets for any region holds that $m_4 \leq m_5 \leq m_6 \leq m_h$. As discussed, each element $j \in P_4$ (P_5, P_6, P_h) has a centre of location and thus a pair of RD coordinates (x_j, y_j). For both steerable flows *2TNA* and *ASIA*, let the weights $\bar{w}_j$ and $\bar{v}_j$ denote values for the relative size of expected number of parcels to be stored at retailers among elements $j \in P_4$ (P_5, P_6, P_h), respectively. These weights are important from a customer point of view and serve to correct for the distance to be covered to retailers. For example, a weight with high value indicates this element (household or postal code area) is expected to collect many parcels at a yet to be determined retailer. As a consequence, this information can be used to prioritise matching the element to an adjacent retailer compared to an element with low expected number of parcels. Additionally, denote $\bar{f}_j$ and $\bar{g}_j$ as a parameter for the expected volume (m^3) of respective flows *2TNA* and *ASIA* from element $j \in P_4$ (P_5, P_6, P_h). Any element j matched to a retailer reduces the expected available capacity with the value of the weight. For example, matching an element $j \in P_4$ (P_5, P_6, P_h) that is expected to receive relatively little volume, will cause less effect on the available capacity compared to those elements with higher expected volumes.

3.3 Distance

Define for each $i \in R$ and each $j \in P_4$ (P_5, P_6, P_h) the distance measure $d_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$, with meter as unit measure.

3.4 Decision Variables

Define the following binary decision variables:

$$x_{ij} = \begin{cases} 1 & \text{if retailer } i \in R \text{ is matched to flow } \textit{2TNA} \text{ of element } j \in P_4(P_5, P_6, P_h) \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if retailer } i \in R \text{ is matched to flow } \textit{ASIA} \text{ of element } j \in P_4(P_5, P_6, P_h) \\ 0 & \text{otherwise} \end{cases}$$

3.5 IP Formulation

Combining the previously defined sets, parameters and decision variables, we can construct for each level of detail P_4 (P_5, P_6, P_h) the following IP model.

IP Formulation for P_6:

$$\min \quad \sum_{i \in R} \sum_{j \in P_6} d_{ij}(\bar{w}_j x_{ij} + \bar{v}_j y_{ij}) \tag{2}$$

$$\sum_{i \in R} x_{ij} = 1 \qquad \forall j \in P_6 \tag{3}$$

$$\sum_{i \in R} y_{ij} = 1 \qquad \forall j \in P_6 \tag{4}$$

$$\sum_{j \in P_4} (\bar{f}_j x_{ij} + \bar{g}_j y_{ij}) \leq \alpha\beta c_i - \bar{q}_i \qquad \forall i \in R \tag{5}$$

$$x_{ij}, y_{ij} \in \mathbb{B} \qquad \forall i \in R, \forall j \in P_6 \tag{6}$$

$$(x_{ij} = y_{ij}) \qquad \forall i \in R, \forall j \in P_6 \tag{7}$$

The objective function states that the sum of traveling distances over all retailers to their matched elements are minimised, where distance is weighted per element by the expected number of parcels to be stored. Equations (3) and (4) indicate that each PC6 element in the region under consideration is matched to a single retailer for the flows *2TNA* and *ASIA*, respectively. Equation (5) concerns capacity and restricts the assignment of PC6 areas of both flows to retailers. It states that the total sum of volumes of both flows from those PC6 assigned to a retailer cannot exceed the corresponding workable capacity minus the volume of inflexible parcel storage destined for the *'CR'* flow. Equation (6) assures the matches between retailers and PC6 areas are binary, while the optional constraint (7) limits the assignment of types of flow per PC6 area to a single retailer. Recall that PostNL currently applies a rule to assign a similar retailer for both flows on PC6, such that the equality $x_{ij} = y_{ij}$ holds for all $i \in R$ and $j \in P_6$. Assume constraint (7) holds unless stated differently, as the effect of decoupling will be tested in later phase. The following section discusses literature on a comparable problem formulation and examines appropriate solution methods to solve the IP for instances of considerable size.

4 Related Work

The previously stated IP formulation is similar to the Generalized Assignment Problem (GAP), which can be formally defined as follows. Let $I = \{1, 2, ..., m\}$ be the set of knapsacks and $J = \{1, 2, ..., n\}$ be the set of items such that for each knapsack $i \in I$ we have a maximum capacity $b_i > 0$. For each $i \in I$ and each $j \in J$ there are costs $c_{ij} > 0$ and resource requirement $r_{ij} > 0$ for assigning item j to knapsack i. The aim is to find the assignment of items to knapsacks with least amount of cost such that all items are assigned to a knapsack and capacity limitations at knapsacks are not exceeded.

Generalized Assignment Problem (GAP):

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{8}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j \in J \tag{9}$$

$$\sum_{j=1}^{n} r_{ij}x_{ij} \leq b_i \qquad \forall i \in I \tag{10}$$

$$x_{ij} \in \mathbb{B} \qquad \forall i \in I, \forall j \in J \tag{11}$$

The term Generalized Assignment Problem (GAP) is formally introduced by Ross and Soland (1975) and the interpretation of the IP formulation mentioned in the previous section can be supported by the analogy of items (households), knapsacks (retailers), costs (distance) and resource requirements (volume). GAP is a variation of the well-known knapsack problem and its extension, the multiple knapsack problem (MKP). Numerous real life applications have been modeled as a GAP since it has been introduced in 1975. Examples can be found in the assignment of tasks to computers (Balachandran (1976)), land use allocation (Cromley and Hanink (1999)), retrieval of data blocks in disks (Aerts et al. (2003)), scheduling of advertising campaigns (Adany et al. (2016)) and many more. I propose an application in the determination of catchment areas of retailers facing capacity restrictions.

4.1 Exact Solution Methods

The GAP is known to be NP-hard as described by Garey and Johnson (1979). Note that the solution space of GAP is equal to m^n, with m the number of knapsacks and n the number of items to be assigned. Allowing the binary decision variables to take fractional values by means of the LP-relaxation provides a theoretical lower bound but will result in unpractical solutions as splitting PC6 elements not allowed. Commercial optimisation software like CPLEX, GLPK or Gurobi are capable of solving 'large-scale' linear, mixed and integer programs, such as the GAP. Due to licencing, no description of the applied algorithms is openly available. Several open source exact solution algorithms have been reported on, able to yield optimal solutions only for small to medium sized instances within reasonable computation time. Savelsbergh (1997) reformulated the GAP to a set-covering problem for which an exact Branch and Price algorithm has been employed. Later, Pigatti et al. (2005) applied a stabilisation method on this Branch and Price algorithm to speed up convergence of column generation. Posta et al. (2012) reformulated the GAP into a sequence of decision problems for which solutions are found by means of a lagrangian Branch and Bound method that is improved by variable fixing rules. These exact methods provide optimal solutions in small to medium sized problem instances taken from the OR library Beasley (1990). However, as the problem size of real life areas containing retailers and PC6 elements outgrows these instances, the need to turn to efficient heuristics grows.

Considering the retailer parcel storage problem of PostNL, recall from Table 1 that 455.140 PC6 need to be reassigned to 2,850+ retailers in The Netherlands. One can easily reduce the size of the problem by focusing on one specific city or region at a time, as a retailer in Groningen will likely not be serving a PC6 element south of Assen, drastically reducing the size of the solution space. Another way to reduce the problem size is to consider PC5 elements to find a feasible assignment. Still, by focusing on a single region or city on PC5, significant problem sizes may occur, calling out for suitable heuristics able to provide quality solutions within reasonable computation time. In order to evaluate solutions on their quality, two metrics will be applied. First, similar as described by Chu and Beasley (1997), let the fitness function $F(\cdot)$ represent the cost of the solution, equal to the objective function in the GAP formulation. Recall that for the problem under consideration this is equal to the sum over all distances from centres of postal codes to their matched retailer, weighted on the number of parcels flowing to each postal code. Fitness and average distance will be used interchangeably throughout this thesis. The second metric to evaluate the quality of a solution is the unfitness function $U(\cdot)$, equal to the sum of capacity exceeding over all retailers. An unfitness function with positive value indicates the solution is infeasible. For some cases, relative preference between solutions is trivial, but for some this is not the case. Consider two solutions, s_1 and s_2, such that their fitness and unfitness are $0 < F(s_1) < F(s_2)$ and $U(s_1) > U(s_2) > 0$. Note that both solutions are infeasible, where the first exceeds capacity most but results in a better objective value. As the capacity constraint is a hard constraint imposed by PostNL, the second solution will be preferred by my method. Stated differently, solutions are ranked on sorted unfitness from low to high. In case a tie in unfitness occurs, a low value for fitness is preferred. Ties in both metrics are broken arbitrarily.

Numerous (meta)heuristics to find such solutions to the GAP have been reported on, including Genetic Algorithms (GA), Tabu Search (TS), Simulated Annealing (SA), hybrid approaches and Memetic Algorithms (MA). Their reasoning, benefits and disadvantages when applied to instances of the GAP are discussed in the following subsections.

4.2 Genetic Algorithms (GA)

The most widely applied metaheuristic to find solutions for large instances of GAP is the Genetic Algorithm (GA), firstly described by Holland (1975). This evolutionary algorithm is based on a population of solutions, for which favourable components are encouraged in selection for reproduction, while mutation causes diversification needed to escape non-global optima. This way, solutions potentially evolve towards a desired global optimum. An often discussed downside of this method is the many parameter settings in combination with the role of probability, hindering the direct effect of different parameter settings on the quality of the method. Wilson (1997) solves the dual form of the GAP formulation using GA as it restores feasibility to near-optimal solutions. Once a feasible solution has been found, the algorithm switches to local search to further improve the objective function while remaining feasible. The algorithm was tested on a randomly generated set of 50 knapsacks and 500 items. More recently, Liu and Wang (2015) presented a scalable GA-algorithm to solve GAP based on parallel computation architecture. Due to a significant level of independence, populations generated by GA are well-suited to be distributed over multiple cores, facilitating exploration of more and often better solutions within reasonable time.

4.3 Tabu Search (TS)

In addition to GA-algorithms, Tabu Search (TS) has also been widely applied in solving instances of the GAP. By keeping track of recently found solutions, TS allows exploration of solution areas with decreasing objective value in its aim to escape non-global optima. Compared to other discussed algorithms, TS is the only deterministic method. Its corresponding downsides include parameter setting (e.g. tenure), memory requirement and problem specific tabu list construction. Dıaz and Fernández (2001) discuss a TS algorithm applied on instances of the GAP based on first improvement by interchanging two jobs. A relaxed formulation allows considering solutions with alternating feasibility and in order to increase flexibility, varying tenure settings have been tested. The results of the method have been benchmarked against algorithms such as GA and Variable Depth Search with Branching (BVDS) on three publicly available problem sets. While the algorithm has a high level of simplicity, comparable solutions are found within excellent computational time. Recently, Sadykov et al. (2015) applied a column generation method on the GAP after which diversification is explored, skipping recently selected columns using a Tabu list. The method improved primal bounds on several instances in literature, varying from size 20 knapsacks and 200 items to 80 knapsacks and 1600 items.

4.4 Simulated Annealing (SA)

Less often than GA or TS, Simulated Annealing (SA) algorithms have been discussed in literature to find solutions to the GAP. SA is a search method that diversifies at the beginning of the search by allowing large jumps in solutions, considering distinct parts of the solution space. As the algorithms proceeds, diversification switches to intensification. Qian and Ding (2007) presents a SA algorithm to solve the closely related multidimensional knapsack problem, iteratively accepting solutions that improve the best found solution, while accepting non-improving solutions with a non-increasing probability. Downsides of such methods include parameter settings, the potentially required repetition of running the algorithm and that the algorithms does not provide information onto a next run. However, the paper mentioned above compares the performance of its SA to the GA described in Chu and Beasley (1997) and results show improvement in speed, while producing competitive objective values. Their method is claimed to be favourable for the considered instances when available computational time is strongly limited.

4.5 Memetic Algorithms (MA)

Hybrid methods integrate elements of multiple search algorithms into a single algorithm. Note that this is different from simply running the individual algorithms in sequence and that hybrid methods are intended to combine best of both (or more) methods. Several hybrid approaches have been presented in literature to obtain solutions for GAP. Yagiura et al. (2006) based a hybrid approach on Path Relinking (PREC) with tabu search, resulting in a method to solve large instances of GAP. Memetic Algorithms (MA) are often referred to as 'Cultural Evolutionary' or 'Hybrid Evolutionary' Algorithms and form a subset of hybrid algorithms. The evolutionary aspect of creating populations with solutions is accompanied with a local improvement heuristic, resulting in more directed search or learning component in each iteration of the search. Motivated to overcome limitations of two techniques, Norman and Moscato (1991) were one of the first to present an MA by integrating aspects of SA into evolutionary GA on a well-studied combinatorial optimisation problem, a Traveling Salesman Problem. Since then, MAs have turned out to be useful method for a numerous NP-hard optimisation problems, including the GAP. Chu and Beasley (1997) discuss an MA on publicly available problem instances of the GAP which separates violation measurement of capacity constraints (unfitness) from the objective function (fitness). Lau and Tsang (1998) successfully applied an MA method based on GA and Guided Local Search (GLS), labeled as Guided Genetic Algorithm (GGA). Lorena et al. (2002) discuss an application of the MA, the Constructive Genetic Algorithm (CGA), to GAP. In addition to the algorithm mentioned by Chu and Beasley (1997), dynamic population sizes and an assignment heuristics are applied. Performance was tested on publicly available problem instances up to 20 knapsacks and 200 items and showed promising results. Later, Feltl and Raidl (2004) built an MA based on the approach described by Chu and Beasley (1997) by including improved initialisation heuristics, selection scheme and mutation operator. The approach is tested on multiple instances from literature and indicate an average better performance based on solution quality and computational time.

4.6 Concluding Remarks

In the goal to develop a stand alone solution method to solve large instances with varying characteristics of the GAP, different approaches from literature have been discussed and evaluated. Based on the reviews, practical success in a variety of similar studies applying MAs gives confidence to construct a method containing its aspects. Crucial in its success will be to include an appropriate hybrid evolutionary or learning component into the outline of a GA. Also, parameter settings and strategies per function require sufficient attention in order to balance the depth search with diversification of the solution space, as instances with different characteristics (e.g. urban Amsterdam vs rural Flevoland) are expected be solved by this single method. From a customer point of view, having assigned a single retailer for a long period of time contributes to convenience. Therefore, catchment areas in a region are updated only periodically (e.g. once every 2 months) such that low computation time is favourable, but no strict time limitation is forced on the method. Similar as the described alternatives, MA cannot be expected with certainty to find optimum solutions for large instances within reasonable time. However, it is capable to obtain near optimal solutions, adequate for the parcel storage case under consideration. Application of MA may thus be a suitable method for the problem at hand and may automate and improve the decision making on catchment areas. The next section discusses the complete methodology, including data pre-processing and its aspects of MA.

5 Methodology

For PostNL, a method is succesful when it is capable of assigning PC6 elements to retailers for any city or reasonable region such that the corresponding total customer traveling distance is minimised and the capacity limitation at each retailer is respected. Reconsidering the instances depicted in Table 1 of Section 2.5, it can be seen the problem sizes considered in this thesis range from 23 retailers and 3,195 PC6 in Breda to 93 retailers and 16,711 PC6 in Amsterdam. Based on discussed literature on methods to solve instances of the NP-hard GAP formulation, such problem sizes are extremely large and would be time consuming (>24 hours) to directly solve to (near-) optimality. In order to overcome the complexity of the problem at PC6, I propose a method involving MA which breaks down the problem in multiple steps, visually explained in the following example. The open source program RStudio is used to construct the method. After the example of Section 5.1, in depth considerations on the construction of all steps in the method are discussed.

5.1 Example of Proposed Method

This section illustrates the method by means of a small and fictive example. In order to start assigning PC6 to retailers, an appropriate and thereby manageable level of detail must be set, depending on the number of retailers and postal code elements in the city or region under consideration. As discussed, households located close to each other tend to have similar postal codes such that the problem size strongly reduces by grouping households together from PC6 to PC5 or even PC4. Consider Figure 8 in which 4 retailers and 35 centres of unique PC6 locations are depicted, each with their corresponding amount of available storage capacity and expected parcel numbers and volumes (m^3), respectively. While real life instances are much larger, assume that the size of this small example already causes complexity to feasibly assign PC6 elements to retailers such that distances are minimised. As a consequence, the first step in the proposed method is to group PC6 elements together based on similar PC5, as shown in Figure 9.

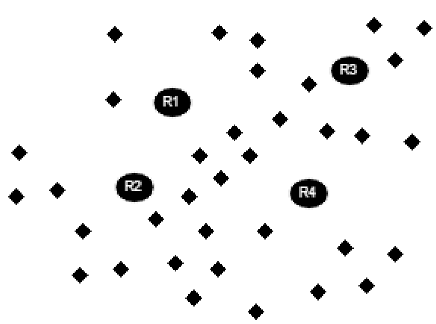

Figure 8: Difficult PC6 instance unmanageable by MA method

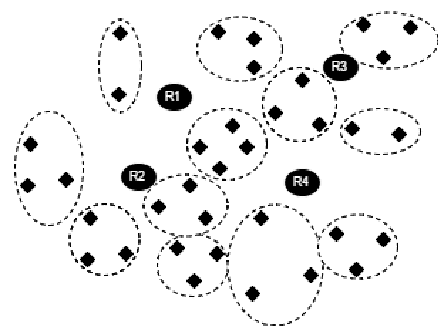

Figure 9: Grouping PC6 based on PC5, reducing problem size

After grouping the PC6 elements into their centre of gravity indicated by the PC5 location, the problem size reduces to 4 retailers and 12 PC5 as shown in Figure 10. Suppose this new size is manageable such that an MA can successfully be applied. Consider the best feasible solution found by the MA method in Figure 11.

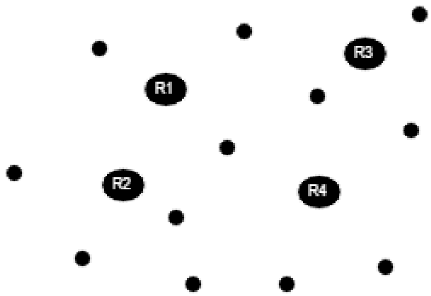

Figure 10: PC5 instance manageable by MA method

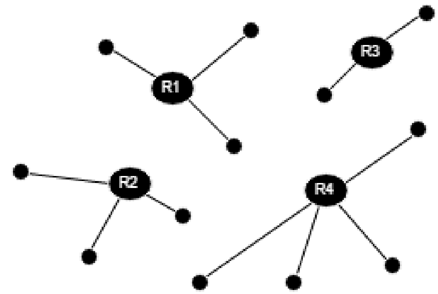

Figure 11: Feasible solution by MA method on PC5

Once a feasible solution has been found, note that by the subset structure of postal codes this match on PC5 is feasible for each retailer on PC6 as well. This is an important observation and gives us a first feasible solution on PC6. The next steps aim to improve the found solution by dividing the set of retailers into non-overlapping clusters, based on proximity of retailers and a maximum number of retailers per cluster. Two clusters $C_1 = \{R_1, R_2\}$ and $C_2 = \{R_3, R_4\}$ are shown in Figure 12. By dividing the region into clusters containing several retailers together with their previously found matches on PC5, manageable sub problems occur on PC6. Hence, the benefit of this way of clustering is that each created subset has the property that the capacity of retailers is sufficient to allocate the burden caused by the postal codes. Without the solution on PC5, a cluster of retailers does not necessarily contain information on a feasible corresponding set of postal codes. The next step is to consider each cluster individually such that the corresponding PC5 elements are replaced by their PC6 elements, as shown for C_1 in Figure 13.

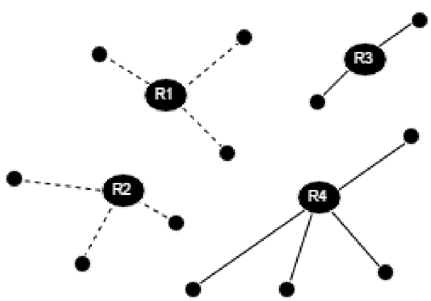

Figure 12: Solution clustering

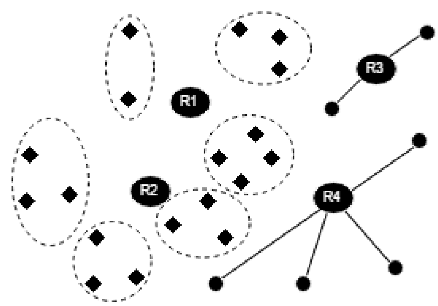

Figure 13: Cluster 1 on PC5

As the match found on PC5 level contains relatively large chunks of grouped PC6, the sub problem on PC6 increases the flexibility of matches to be made and potential improvements may be found. The sub problem C_1 consists of 2 retailers and 18 PC6 elements. Given the small number of retailers and increased number of PC6, the next step focuses to turn the LP-Relaxation of each cluster problem feasible. The solution caused by the LP-Relaxation often consists of few fractional assignments which are mostly the elements that have similar distance to multiple retailers. As a result of this step, a potentially improved solution can be found, as shown in Figure 14. Note that the PC6 indicated in white has changed retailer compared to the solution found by the previous solution which allocated postal codes on PC5. Next, the PC5 elements of C_2 are replaced in a similary way by their PC6 elements, shown in Figure 15.

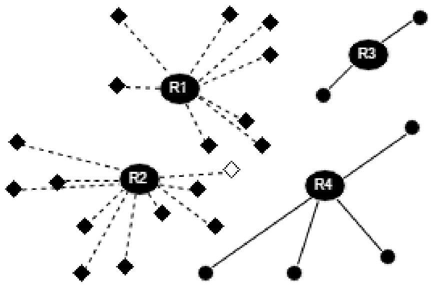

Figure 14: Cluster 1 solution of LP-Heuristic

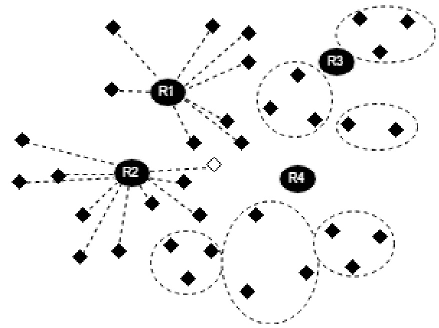

Figure 15: Cluster 2 on PC5

Solving C_2 provides the solution as depicted in Figure 16. Again, the elements shown in white have been reassigned compared to the previously found solution due to increased flexibility. Once each cluster has been solved by the LP-Heuristic, the best found feasible solution is shown in Figure 17 and the method terminates.

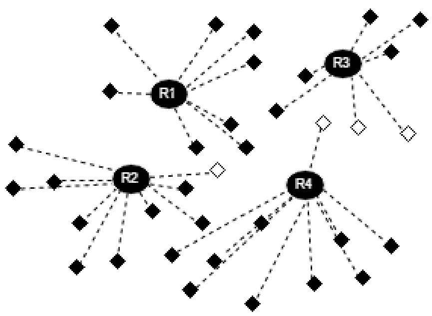

Figure 16: Cluster 1 and 2 solution of LP-Heuristic

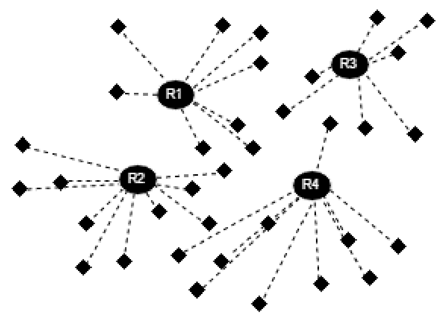

Figure 17: Final solution

While the example illustrated by Figures 8 - 17 provides an overview of all steps performed by the method, three non-trivial steps receive attention in the section on methodology:

1. Many approaches are possible to come from Figure 10 to Figure 11. Two methods, referred to as the Starting Heuristics (SH) and MA, will be discussed in Section 5.3 and 5.4, respectively.

2. The clustering method to come from Figure 11 to Figure 12 will be discussed in Section 5.5.

3. The LP-Heuristic applied on each non-overlapping cluster to come from Figure 13 to Figure 14 (and Figure 15 to Figure 16) will be discussed in Section 5.6.

5.2 Data

First, assumptions and basic pre-processing steps of data to serve as appropriate input for the IP formulation will be discussed. While realistic input will be crucial in the attempt to practically implement improved retailer catchment areas, the regions Breda, Flevoland and Amsterdam will be tested using fictitious parcel flow parameters. Reasons for doing so, is that the focus will be on testing the performance of the proposed method, which should be applicable to any input. Secondly, retailer capacities are being surveyed and information on retailer capacities c_i is currently not available at PostNL and will be collected in the coming months. Once all required information on capacities is available, effort on realistic parameter setting for parcel amounts and volumes will be meaningful. In this thesis, only the real life locations of both retailers and postal codes will be applied. This section only touches upon some of the considerations required to generate reasonable input.

5.2.1 Assumptions and Basic Pre-Processing

Detailed historic information is available on parcel flow through the use of scanning equipment at retailers. For any parcel that passed through a retailer, data is available on its flow type (*2TNA*, *ASIA*, *CR*), volume, arrival and departure time, destination on household level and thus also on any postal code level. This information is processed into a forecast by PostNL which states the number of parcels to expect for each type of flow on a national level on a weekly basis over a fixed time horizon. Based on the assumption that recent proportions per city or region remain fairly equal over time, this forecast can be scaled down to any region under consideration by their historic relative size. Recall the parameters on parcel flow discussed in the IP formulation in section 3.5, which follow from this scaled down forecast. The weights $\bar{w_j}$ and $\bar{v_j}$ refer to the expected number of parcels on PC6 level, which can be aggregated on PC5 and PC4 levels if necessary. The values for $\bar{f_j}$ and $\bar{g_j}$ denote the expected volume (m^3) at a retailer caused by a PC6, PC5 or PC4. Note that the catchment areas are intended to last for a pre-defined horizon, for example 2 months. Therefore, in order to account for shelf time and fluctuations during the week, the parameters for expected volume can be scaled to incorporate this information. Thus, the values will be dimensioned on the day of the week which historically has proven to cause most burden at retailers. Next, as the non-steerable flow *CR* forces partly reservation of retailer shelf capacity, similar scaling can be applied, but now on retailer level instead of PC level. Again, as amounts and volumes have a certain amount of variance, capacities are dimensioned towards the day in the week which historically has proven to be the bottleneck. Setting the parameters based on this day for each region under consideration in combination with the piling parameter α and buffer parameter β, reasonable input estimates can be generated for the IP. In addition, the values for shelf capacity c_i (m^3) are straightforward and are equal to the sum over all shelves with their corresponding length times width and height per retailer. For all PC and retailers, their locations (X_i, Y_i) are available such that the corresponding crow-fly distances d_{ij} can be easily computed.

5.3 Starting Heuristics (SH)

The Starting Heuristics (SH) refer to a set of easy to implement rules to find initial matches between postal codes and retailers, both deterministic as probabilistic. While the solutions found by the SH differ in quality depending on the problem instance, they provide a starting point to improve on in later phase of the method. A total of 4 rules have been included in the SH:

1. *'Largest Volume First'*. This rule iteratively evaluates non matched postal codes in descending order of volume. Per postal code, the method checks retailers in ascending order of distance. In case the volume corresponding to the postal code fits at the retailer, the two are matched. Due to this assignment, the available capacity at the retailer reduces, after which the method moves on to the next non matched postal code in line with largest volume, until all postal codes have been

assigned. In case the volume corresponding to a postal code does not fit at any of the retailers, the method assigns this postal code to the closest retailer and accepts this particular solution in the SH is infeasible.

2. *'Shortest Distance First'*. This rule evaluates potential assignments between non matched postal codes and retailers in ascending order of distance. In case the non matched postal code fits the currently available capacity at the retailer, the method assigns this postal code and moves on to the next non matched postal code in line. If a postal code does not fit at the retailer, the method continues with the next element in line of ascending distances. In case one or more postal codes have not been feasibly assigned after all distances of non matched postal codes have been checked, assign these postal codes to their closest retailer and accept this particular solution in the SH is infeasible.

3. *'Rounding the LP-Relaxation'*. This rule considers the LP-Relaxation and rounds the fractional assignments to their closest integer such that each postal code is matched to a single retailer. In cases of an even split in assignment found by the LP-Relaxation, the tie is broken by considering which assignment adds least to unfitness, followed by least fitness in case of another tie.

4. *'LP-Heuristic'*. In addition to the 3 deterministic rules described above, a probabilistic variation of the solution found by the LP-Relaxation has been included in the SH. One practical way to look for improvements on the rounded LP-Relaxation is to fix the binary assignments and to consider the fractional assignments as non matched. When assigning these non matched postal codes to retailers, note that the order in which these fractional assignments are evaluated, determines the outcome of the final solution as the first match influences the available capacity at retailers for the fractional postal codes which follow. Hence, by iteratively considering the set of fractional assignments in different order, multiple solutions can be created in case multiple fractional assignments are present in the LP-Relaxation. For realistic problem instances such as Breda, Flevoland and Amsterdam, the number of fractional elements after the LP-Relaxation is significant, leading to varying solutions found by the LP-Heuristic. In case assigning a non matched fractional element must cause infeasibility for any retailer, the method exchanges this postal code with one of the postal codes that already has been assigned to the retailer which has most available capacity. For this retailer, a currently matched postal code is removed such that the released capacity burden allows the non matched postal code to fit. This type of exchange continues a predefined number of times or terminates once a feasible solution has been found, after which it is included in the set SH. Pseudo-code on this last method, referred throughout this thesis as the LP-Heuristic, is shown in Appendix A and will be discussed in more depth in Section 5.6.

Note that the rules which form the SH do not guarantee feasibility but will for many reasonable instances find acceptable starting solutions, containing useful elements to recombine during the MA phase. As the LP-Heuristic is probabilistic, the number of solutions contained in the SH can depend on the preference of the user. For example, in case the SH are requested to contain 50 solutions, the set contains the 3 deterministic rules described above, complemented by 47 solutions found by the LP-Heuristic.

5.4 MA Method

This section discusses the relevant steps and considerations while developing an appropriate MA method applicable to assign postal codes to retailers as illustrated in Figure 10 and Figure 11.

5.4.1 Solution Representation

An important requirement for MA to be applicable is a suitable representation of solutions and its atoms, also referred to as building bricks by Holland (1975), structures by Chu and Beasley (1997) or -based on the analogy of evolution- chromosomes as in Liu and Wang (2015). With respect to the GAP formulation, consider the binary solution matrix with decision variables x_{ij}, containing the information which postal codes are assigned to which retailer. Each match, indicated by a 1 in a column, contributes as a building brick to the solution. A complete assignment contains a single match for all columns. This matrix formulation can be represented as a one dimensional vector z, such that index z_j contains the retailer to which postal code j is matched. An example of both feasible solution representations of a match between four postal codes $(1, 2, 3, 4)$ and three retailers (A, B, C) is shown in Figure 18.

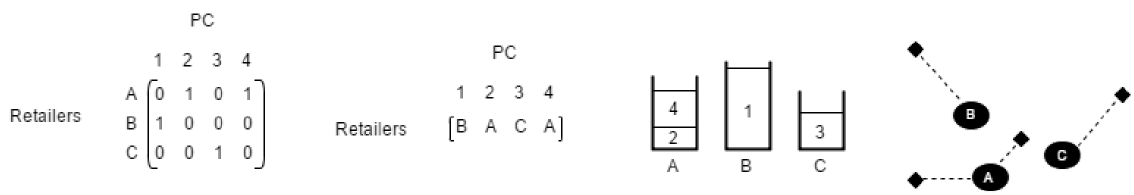

Figure 18: Matrix, vector, capacity burden and map of a single solution

Consider the vector representation of a solution as in Figure 18. Define every index containing a matched retailer as a building brick such that each building brick contributes to the burden at the retailer to which it is matched. Next, define components to be a collection of one or more building bricks. The vector solution in the example of Figure 18 can thus be decomposed into $2^j - 1 = 2^4 - 1 = 15$ components, by either including or excluding building brick j into the component such that at least one building brick is present in the component. Following this terminology, a solution is a component which includes building bricks on all its indices. Step-wise inclusion of building bricks into components gradually gets you closer to a solution, which is feasible if the capacity restriction at each retailer is not violated (e.g. right in Figure 18). Note that the complexity of this problem comes from the fact that each decision to include a building brick into a component limits the options to feasibly assign non-matched postal codes. A reasonable but greedy match between a postal code and a retailer impacts future matches, potentially forcing building bricks to add significant distance to remain feasible. In the formal GAP notation, matrix representation is applied for the decision variables, although vector representation and terminology on building bricks and components will be useful when constructing solutions using MA.

5.4.2 Outline Memetic Algorithm

Consider the pseudo-code in Algorithm 1, providing the outline of the MA. Functions in bold require special attention while constructing and testing the method, due to their numerous strategies with corresponding effect on performance. In case no feasible solution can be found, the method will return information on those retailers with capacity problems after assigning postal codes based on shortest distance. In such cases, the method provides preferred capacity expansion locations, as it will alleviate pressure on retailers and contribute least to the distance to be covered by customers.

Data: Any city or region containing info on postal codes and retailers
Result: Assignment of postal codes to retailers

```
BestSolution ←Generate(RandomSolution);
if RetailerCapacities ≥ VolumesToBeStored then
    Population ← InitializePopulation(StartingHeuristics);
    BestSolution ← StoreBestSolution(Population);
    while StoppingCriteriumNotReached do
        NextPopulation ← ∅ ;
        while | NextPopulation | < PopulationSize do
            TwoParents ← Select(SelectionRules, Population);
            TwoChildren ← Crossover(CrossoverRules,TwoSelectedParents);
            TwoChildren ← Mutate(MutationRules,TwoChildren);
            TwoChildren ← LocalSearch(UnfitnessReductionRules,TwoChildren);
            TwoChildren ← LocalSearch(FitnessReductionRules,TwoChildren);
            NextPopulation ← NextPopulation ∪ TwoChildren;
        end
        Population ← NextPopulation;
        if NewBestSolution(Population, BestSolution) then
            BestSolution ← StoreNewBestSolution(Population);
        end
    end
    if CheckIfFeasible(BestSolution)=TRUE then
        return BestSolution;
    end
end
if CheckIfFeasbile(BestSolution)=FALSE then
    'No feasible solution can be found';
    InfeasibleSolution ← Assign(ShortestDistanceRule,Households,Retailers);
    return RetailersWithCapacityExceeding(InfeasibleSolution);
end
```

Algorithm 1: Outline Memetic Algorithm

5.4.3 Population Initialisation

In the initialisation phase of the algorithm, a population must be created containing a set of solutions. This population must contain solutions with characteristics to allow for diversification as well as the potential of quick convergence to (near-)optimality. Setting the size of the population depends on the size of the problem, the computational time available and required quality of the solution. Multiple strategies can be applied to generate a population containing varying parent solutions. The Starting

Heuristics as described in Section 5.3 are considered appropriate to serve as the initialisation population for the MA. As time is not a strict restriction at PostNL, MA is applied to produce offspring of the initial population containing the Starting Heuristics in order to explore larger parts of the solution space, aiming to find improved local and potentially global optima.

5.4.4 Selection

While the fitness and unfitness function give direction to which solutions contain building bricks favourable for reproduction, different strategies to select solutions exist in literature. The elitist strategy selects a percentage of the best solutions in the population to breed for offspring, while a binary tournament strategy makes a selection based on evaluation of a defined number of randomly selected solutions in the population. A third commonly applied selection strategy appoints a selection probability to each solution based on fitness and unfitness values. This strategy is often referred to as the roulette wheel selection strategy and while it favours those solutions with low fitness and low unfitness value, it allows selection of unfavoured solutions in order to escape non-global optima. In addition, rank based (or proportional) roulette wheel serves as an alternative, assigning step-wise selection probabilities to solutions based on on their rank instead of absolute evaluation differences. Noraini and Geraghty (2011) studied performance of various parent selection strategies for the GA on instances of a different NP-hard problem, the Travelling Salesman problem. In this paper the authors conclude that for increasing problem sizes the rank based selection is a favourable strategy as it continues to explore the search space while other strategies face premature convergence. The paper states that if the solution quality is favoured in the trade off against computation time, rank based selection is the best choice. For the PostNL case under consideration, updates of catchment areas are only periodically desired, allowing for extra computational costs if necessary. Therefore, rank based selection seems as the most appropriate parent selection strategy applicable in the MA. However, through trial and exploration, the performance of this selection has failed to meet the expectations and often infeasible solutions are generated. One way to overcome this, is to favour better solution by the inclusion of an extra parameter ρ, assigning the proportion of best solutions in the generation to consider in rank based selection. The probabilities have been determined as follows. Let P denote the population size and ρ the percentage of best solutions to consider. In such case, the best $B = \rho \cdot P$ ranks can be assigned to solutions in descending order of preference. This implies rank B is assigned to the best solution in the population, $B - 1$ to the second best and so forth. Consider the example in Table 2 in which the proportion of selection is set to $\rho = 0.5$ for a population of size $P = 10$. The sum of all these ranks is $S = \frac{(B+1)B}{2}$, such that for each solution the probability of selection is $p(sol) = \frac{rank(sol)}{S}$.

Solution	Unfitness	Fitness	Rank	Prob
1	0	751	5	5/15
2	0	756	4	4/15
3	3	751	3	3/15
4	5	740	2	2/15
5	11	738	1	1/15
6	12	731	0	0
...	...	...	...	...
10	50	743	0	0
			$S = 15$	1

Table 2: Example of rank based selection with population size $P = 10$ and selection proportion $\rho = 0.5$

As an alternative to rank based selection, binary tournament selection with a user defined number of elements is included as option in the MA. In Noraini and Geraghty (2011) it is claimed that this selection strategy is faster and results in better results for smaller problem instances. The paper states that as the problem size increases, the binary tournament selection is susceptible to premature convergence. As problem sizes of most interesting regions (cities) typically contain many PC elements, rank based selection is expected to outperform the binary selection strategy on solution quality. From a cost perspective, binary tournament selection is expected to find solution more quickly compared to rank based selection. Both the rank based and tournament selection strategies will be applied and tested on performance for the MA on Breda, Flevoland en Amsterdam.

5.4.5 Crossover

When two parents are selected to breed, their components can be crossed over using multiple strategies, which will be explained using a small example. Reconsider the situation of Figure 18 in which each postal code $\{1, 2, 3, 4\}$ is to be matched to one of the three available retailers $\{A, B, C\}$. Suppose the vector representations of two selected solutions in the parent population are given by $s_1 = (B, A, C, A)$ and $s_2 = (C, B, B, A)$. The four elements of the child solution (c_1) will be filled by either of the two corresponding elements of its parent such that $c_1 \in (\{B, C\}, \{A, B\}, \{B, C\}, A)$. For each building brick, selection of a parent can be equal (coin flip), implying the building brick on the first index of the child solution is either B or C with equal probability. Instead of simulating a coin flip, a more directed greedy strategy can be applied, such that the probability of building brick crossover depends on their effect on the fitness and unfitness function. Choose the parent building brick which adds least to unfitness, after which ties are broken by the effect on fitness. Following this directed crossover, the order in which the building bricks are evaluated, determines the corresponding child solutions. As performance of the order is problem specific, postal codes are evaluated in random order during crossover.

5.4.6 Mutation

The mutation rate ensures perturbation by altering a percentage of the building bricks in child solutions to a value which might not be present at both of its parent solutions. The initial mutation rate must balance diversification and intensification, while the mutation rate in later stage may decrease to obtain local, potentially global, optima. In order to set the mutation rate for the problem instance at hand, trial and error have led to a suitable initial value of 4 building bricks per solution. In addition, the proposed method decreases the mutation rate in case two successive generations have not yield a new best solution in order to intensify the search based on the solutions found so far, with a minimum of 2 bricks per solution.

5.4.7 Local Search on Unfitness Reduction

As only those solutions with an unfitness value of 0 are considered feasible, two local improvement heuristics are applied in lexicographic manner. First, an unfitness reduction search is applied after mutation to switch from non-feasible to feasible solutions if necessary. Pseudo-code is shown in Algorithm 5 in Appendix B. The greedy search iteratively considers retailers with capacity problems. For those retailers, their currently matched postal codes are evaluated on alternative retailers in ascending order from close to far, such that a switch is made in case the total unfitness after the switch reduces compared to status quo. The iteration stops whenever all retailers have a set of postal codes which can be feasibly allocated, or a maximum number of checks per retailer has been performed. Note that this local search has been applied to the infeasible solutions of the SH as well.

5.4.8 Local Search on Fitness Reduction

After local search on unfitness reduction has been applied, the next step is to improve the objective function. Whereas the unfitness search is only activated for infeasible solutions, every solution is checked on feasibility improvement. Therefore, it is vital that the local search is efficient and effective, as it strongly adds to the computational expense. Also, the fitness reduction search ensures the unfitness value is non-increasing. Parts of the applied local search can be found in Algorithm 6 in Appendix C. Its reasoning is straightforward: the retailer with most available capacity is considered, for which iteratively switches of potentially fitting currently non-matched postal codes are evaluated. If a switch reduces distance, the switch is applied. The pseudo-code is only shown for the retailer which has maximum capacity, while the code can easily be extended such that it considers multiple retailers and their corresponding potential switches. Similar as for solutions generated by the MA, this local search is applied to all solutions in the SH.

5.4.9 Termination

Several criteria may be applied when deciding termination of the MA. A first strategy would be to stop when a feasible solution satisfies certain criteria, for instance when the fitness of the solution found is within a range from a theoretical bound such as the LP-Relaxation or the shortest distance assignment where the capacity constraint is disregarded. Also, the number of generations to consider before terminating the algorithm is a commonly applied rule. Other termination rules include a fixed computation time, a fixed number of generations without improvement or a combination of such rules. For the proposed method, termination is applied after 4 consecutive generations without improvement.

5.5 Clustering

After the MA has finished and found a feasible solution on the level of PC4 or PC5, the next step is to appropriately divide the total set of retailers and their corresponding matches into clusters based on the location of retailers. The motivation for clustering is to shift the balance of the problem size such that the reduction of retailers allows the PC4 or PC5 element to be reconsidered at PC6 level. Ultimately, exchange of PC6 elements between retailers within a cluster can reduce the total average distance to be covered. Note that a cluster containing a single retailer and its catchment area is unable to improve, as no exchange is possible. One requirement for a cluster is that the number of retailers must be 'manageable' in the sense that the problem size can be handled by the last phase of the method which applies an LP-Relaxation heuristic. Without clustering, the problem size will be similar the initial problem on PC6 and thus for most reasonable regions be unsuitable to directly solve. Also, clustering is required in order for the LP-Heuristic to run, for two main reasons. First, the LP-Relaxation package of Rstudio can only handle a limited problem size. Secondly, if the size is appropriate for the program to run, the many retailers and PC6 per cluster add to complexity to turn the numerous fractional assignments into quality binary elements. Given this restriction on cluster size, note that large clusters are favoured due to its increased improvement potential. Thus, the objective is to find clusters containing many retailers, bounded by the restriction that each cluster must be manageable in size. In order to determine the maximum number of retailers in a cluster, consider the problem size of Amsterdam which shows the largest ratio PC6 over retailers ($16,711/93 \approx 180$). By means of multiple experiments with varying problem size, the input limits of the LP-Heuristic have been tested. From 13 retailers and more, the on average corresponding 13 times 180 PC6 elements start to outgrow compatibility. As the distribution of PC6 over retailers depends on the capacities and volumes and might thus differ, a maximum has been imposed of 10 retailers per cluster to be on the safe side. This maximum ensures the heuristic can start for any problem under consideration and that improvement due to exchange of PC6 within clusters is often possible. The appropriate number of clusters follows from the first k for which the K-means cluster solution respects this condition. Pseudo-code is shown in Algorithm 2.

Data: Retailerlocations, Max number of retailers per cluster (C)
Result: K clusters containing retailers

```
Initial values: k = 0 ; C = 10 ; I_Continue = TRUE ;
while I_Continue do
    k ← k + 1 ;
    ClusteredRetailers ← ApplyKMeansClustering(RetailerLocations , k) ;
    NumberOfRetailersInCluster ← Sequence(1 to k) ;
    for j in 1 : k do
        NumberOfRetailersInCluster [ j ] ← Count(ClusteredRetailers = j) ;
    end
    if max(NumberOfRetailersInCluster) ≤ C then
        I_Continue ← FALSE ;
    end
end
return ClusteredRetailers ; k ;
```

Algorithm 2: Determination of k clusters

When the appropriate k clusters have been determined by minimising the within sum of squared distances, k subproblems arise which are solved by the heuristic described in the next section. Note that the initialisation centers of the k-means clustering function are chosen at random, allowing the resulting clusters to slightly vary per iteration of the same region. This randomness, similar for other aspects in the method involving chance, can be controlled by setting seeds for random number generation. However, slightly varying clusters might have effect on the resulting assignment on PC6. This effect will not be looked into during this thesis and the initial clustering found will be considered for the next phase of the method.

5.6 LP-Heuristic

The final step in the method is to resolve each cluster individually on PC6. Due to the clustering described in the previous subsection, it is ensured that each cluster contains 1 to 10 retailers, together with their feasible corresponding assignment found by the SH and the MA. Note that the number of PC6 elements varies per cluster, but is for all regions very likely to be less than 2.000 for any cluster. Also, note that for each cluster the total spare capacity, the difference between the sum of total retailer capacity and the sum of total volume from PC6, is non-decreasing for the number of retailer in a cluster. Stated differently, a cluster with many retailers is likely to see a large absolute spare capacity, adding to flexibility during exchange. For each cluster, the LP-Relaxation is able to quickly generate a solution with potential fractional assignments. By means of inspection of multiple clusters of varying problem instances, it has been noted that the proportion of binary assignments is large, often over 95 %. The fractional assignments generally occur at PC6 elements for which multiple retailers have equal distance. A practical step would be to simply round the fractional assignments, resulting in potential capacity exceeding. Based on the hard constraint that the retailer capacity must be respected, this final step, the LP-Heuristic, aims to feasibly assign the fractional elements found by the LP-Relaxation. Pseudo-code on the outline of this heuristic can be found in Appendix A and is similar to the one applied during the SH phase. Note the only difference in that that we have a feasible solution found by MA as input. The heuristic iteratively considers the fractional and thus unmatched PC6 by fitting them into the available capacity at retailers resulting from the binary assignments. In case the volume corresponding to an unmatched PC6 does not fit at any of the retailers in the cluster due to the dispersion of available capacity of multiple retailers, the retailer with maximum available capacity is considered. For this retailer, a currently applied match is undone such that the resulting available retailer capacity is greater or equal than the volume of the PC which could not be stored. As this type of exchange might in rare cases result in a loop due to repeated exchange, the volumes of exchanging PC are restricted to be different, limiting but not excluding such loops. Additionally, due to the large number of PC6 assigned to a retailer and their corresponding differences in parameters on volume, loops are unlikely and the method will likely turn the LP-Relaxation feasible on PC6. As an alternative, the LP-Relaxation can be turned feasible by means of rounding the fractional assignments and applying the local searches on feasibility improvement as described in the MA method. While these first two solution methods involving the LP-Relaxation are likely but not bounded to turn feasible, the initial feasible MA solution can be reconsidered, which may be improved by means of the local fitness reduction search. This third and final method in the LP-Heuristic differs from the one described in the SH. Given these three solutions of which the last is certain to be feasible, the best will be returned by the method.

6 Outline Complete Method and Experimental Setup

Figure 19 combines the steps of the proposed method into one decision tree. The experiments to test performance of is elements are discussed in this section.

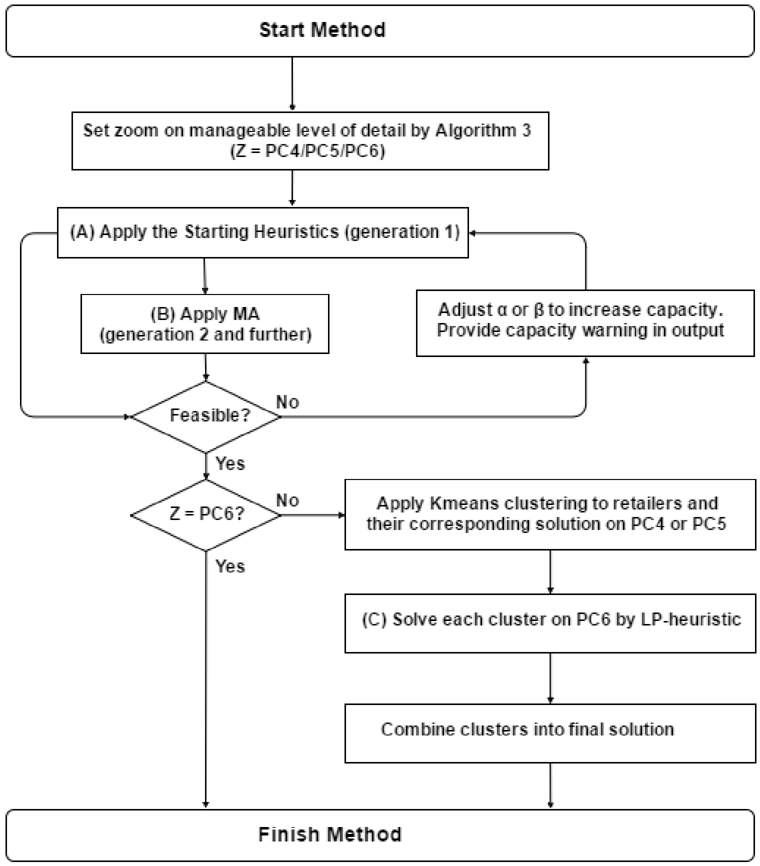

Figure 19: Outline complete method

First, for any region to be solved, an appropriate level of detail must be set. As discussed, this level of detail depends on the size of the region under consideration, specifically the number of retailers and PC4, PC5 and PC6 elements. As we aim to find assignments on PC6, preferably, PC6 is chosen from the start. However, this will lead to unmanageable problem sizes for most regions of considerable size. The rules which have been applied to determine the level of zoom are presented in Algorithm 3.

Data: Any region containing retailers and postal codes
Result: Manageable zoom (= level of detail)

Zoom ← 0 ;
if *(NrRetailers ≤ NrPC6) & (NrRetailers ≤ 10) & (NrPC6 ≤ 2000)* **then**
  Zoom ← PC6 ;
end
else
  if *(NrRetailers ≤ NrPC5) & (NrRetailers ≤ 100) & (NrPC5 ≤ 1200)* **then**
    Zoom ← PC5 ;
  end
  else
    if *(NrRetailers ≤ NrPC4) & (NrPC4 ≤ 100)* **then**
      Zoom ← PC4 ;
    end
  end
end
if *Zoom ≠ 0* **then**
  return Zoom
end
else
  return 'Unmanageable region'
end

Algorithm 3: Rules to set initial zoom

Note that for any zoom the number of postal code elements must be greater than or equal to the number of retailers. Else, we are certain some retailers will have no catchment area. In addition, given a limitation of Rstudio on the LP-Relaxation size and discussed literature on appropriate MA sizes, the boundaries follow from trial and error and serve to properly allow the method to start. For reasonable regions to be solved such a cities and its neighbourhood, the initial zoom will likely be set on PC5. In case Algorithm 3 returns that the region is unmanageable, adjustments are required on the problem instance setting.

Next in Figure 19, the Starting Heuristics as described in Section 5.3 generate a set of initial solutions on the previously determined zoom. While the best solution in this set of solutions may be adequate, the MA can be applied to explore larger parts of the solution space and improve the best solution of the Starting Heuristics. Note that the inclusion of MA after SH is optional, as can be seen in Figure 19. Varying population sizes (50, 100, 150) and selection strategies (binary tournament and rank based) are applied and tested for MA. Note that the population size affect the SH as more sequences of the LP-Heuristic can be tested, potentially improving as the population size grows. By means of a student version of Gurobi Optimisation software, solutions found by both the SH and the MA can be appropriately evaluated on performance. In order to appreciate the quality of the two methods, the first experiment must answer:

Experiment 1: 'Given the currently applied level of zoom, what is the performance of the SH (A) compared to licenced optimisation software and what is the contribution of the MA (B) in terms of solution quality and computational time?'

Next is Figure 19, a feasibility check is shown. In case the SH or the MA terminates without containing a feasible solution, the parameters for parcel piling efficiency α or the buffer β might be reconsidered, affecting capacity as discussed in Section 3.5. For any region and in any case, independent of capacities and expected parcel flow, each retailer must be assigned a catchment area at any moment such that the method generates a workable solution for PostNL. In case adjustments on α or β are required to assign postal codes, the method provides a capacity warning. In such cases, retailer capacity might not be adequate to handle the parcel flow. While the method reruns with increased values for α or β to find a feasible solution, simultaneously, expansion of capacity is desired. Therefore, the method provides insight where to increase capacity by showing the map of catchment areas based on the shortest distance rule. Locations of retailers which face most capacity insufficiency are shown in red and direct to most appropriate locations to increase capacity, as this would alleviate pressure of the region and add little distance for customers to travel. Iteratively increasing α or β will eventually ensure the SH or MA find feasible solutions, such that the method proceeds.

Next is Figure 19, the level of zoom is reconsidered. An initial zoom of PC6 indicates the method is finished as a solution on PC6 is what we aim for. In cases where the initial zoom is set on PC4 or PC5, the clustering steps and LP-Heuristic are applied as described in Sections 5.5 and 5.6, respectively. The second experiment will answer the following question:

Experiment 2: 'Given the constructed K-means clusters, what is the performance of the LP-Heuristic (C) compared to licenced optimisation software in terms of solution quality and computational time?'

Note that in cases where the initial zoom is PC4 or PC5, the interpretation of the first experiment described in this section only concerns a sub problem such that a third experiment must answer:

Experiment 3: 'What is the contribution of MA (B) to the final solution on PC6 in case the initial zoom is PC4 or PC5, compared to the SH in relation to additional computation time?'

This experiment tests the lasting effect of MA on the final PC6 solution compared to the SH in case Z $\neq$ PC6. Eventually, when the method has finished, the assignment on PC6 will be returned together with relevant information on the expected capacity burden at retailers following from the assignment. Also, mappings of the solution may be presented together with visuals on the progress while executing the method.

Another relevant question concerns the two steerable flows *ASIA* and *2TNA*. As discussed in the IP formulation in Section 3.5, splitting the assignment of these flows per PC6 by neglecting constraint (7) doubles the problem size and gives extra flexibility, allowing for solutions with shorter traveling distance. On the other side, customers might need to visit differing retailers depending on the type of parcel they need to collect, unfavourable for most customers. The fourth question to consider is:

Experiment 4: 'What is the effect of allowing a split for both steerable flows *ASIA* and *2TNA* per PC6 (neglecting constraint (7)) on travel distance and what proportion of PC6 is expected to receive two differing retailers?'

The answer on the latter question supports the decision making whether the steerable flows must similar or may vary. Summarising, the 4 questions stated in this section will be answered in Section 7 by applying the proposed method on the retailers and postal codes of Breda, Flevoland and Amsterdam with information presented in Table 3. In order to test the method by lack of data on capacity, fictitious demand and capacities have been distributed over the regions.

	Breda	**Flevoland**	**Amsterdam**
Initial level of zoom	PC5	PC5	PC5
Retailer elements (#)	23	70	93
PC5 elements (#)	228	833	1,018
Total Demand (m^3)	59,206	247,015	284,574
Total Capacity (m^3)	67,070	310,923	317,300
Overcapacity (%)	13.28	25.87	11.5

Table 3: Settings for considered problem instances

Given these settings, Table 4 presents computable bounds per level of detail in the form of the optimal solution value, LP-Relaxation and solution where the capacity constraint is relaxed and postal codes are matched to the retailer which are located most closely. As the initial level of zoom is PC5, the information on PC4 can be disregarded. Note that the computation time required to reach optimality using Gurobi optimisation software grows in problem size and is 2,654 sec ($\approx$ 44 min) for Amsterdam on PC5. For PC6, Gurobi and the LP-Relaxation are unavailable due to the large problem size. Therefore, only the shortest distance bound is computable, giving only little insight on a global bound.

		Breda	**Flevoland**	**Amsterdam**
PC5	Opt. Av. Fitness (m)	806.19	973.43	521.18
	Opt. Comp. Time (sec)	56	842	2,654
	LP-Relaxation Av. Fitness (m)	796.68	965.48	515.18
	Unlimited Capacity Av. Fitness (m)	621.28	805.79	398.84
PC6	Opt. Av. Fitness (m)	-	-	-
	Opt. Comp. Time (sec)	-	-	-
	LP-Relaxation Av. Fitness (m)	-	-	-
	Unlimited Capacity Av. Fitness (m)	637.98	857.37	417.63

Table 4: Information on bounds

7 Results

This section discusses the results of the experiments to test the performance of the proposed method. While the experiments are performed on Breda, Flevoland and Amsterdam, I will take Breda as main example to discuss and explain the logic of the method in detail using multiple plots, figures and tables. Details on results for Flevoland and Amsterdam will be discussed using multiple tables.

Before discussing the experiments, an example is now shown to illustrate the necessity of the method. Consider Figure 20, showing a situation for Breda where each PC5 is matched to its closest retailer, a situation which would be most desirable for customers. As discussed in the section on data, fictitious capacities, parcel numbers and parcel volumes have been distributed over the 23 retailers and 228 PC5 elements in Breda. For each retailer, the catchment area is denoted by a unique color in Figure 20, while Figure 21 shows the exact same infeasible assignment, presenting retailers with capacity problems in red. For both figures, the average travelling distance of 621 meters is shown in the title. As can be seen from Figure 21, this shortest distance assignment might lead to a great imbalance of parcel burden at retailers. For the retailers in red, the expected burden caused by their matched PC5 elements outgrows capacity such that switches are required to alleviate pressure, while other retailer would gladly accept more parcels to store. In our objective to feasibly distribute the parcel burden over retailers and reach a global minimum concerning the average travelling distance as described in the IP formulation, the solution methods SH, MA, clustering and LP-Heuristic will now be tested on performance.

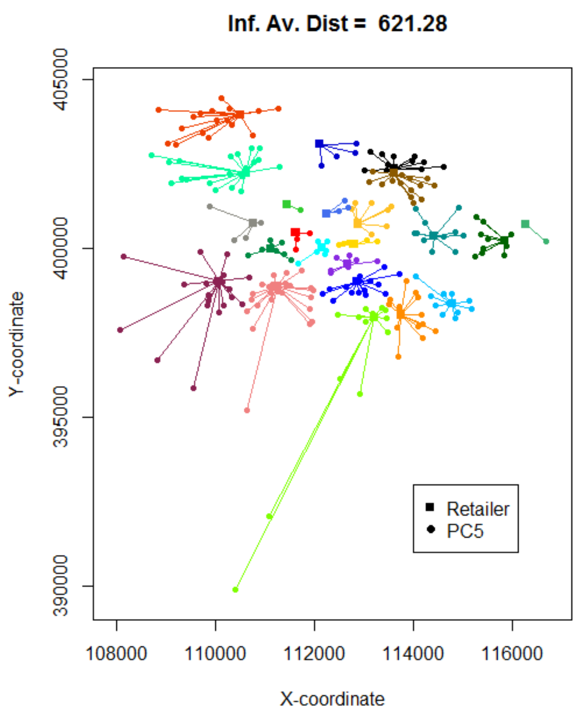

Figure 20: Catchment areas in Breda (by Shortest Distance Rule)

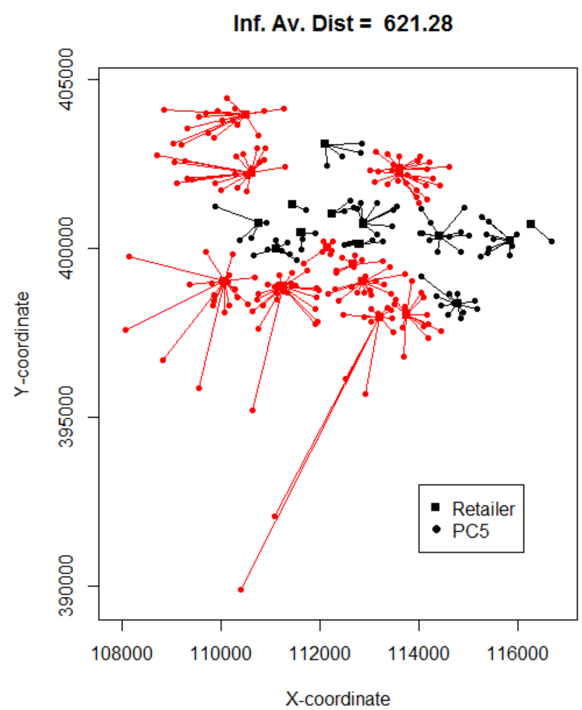

Figure 21: Expected capacity problems in red (by Shortest Distance Rule)

7.1 Experiment 1: Performance of SH and MA

In this experiment we aim to measure performance of SH and MA in their assignment of PC5 to retailers, a first step in our goal to assign postal codes on PC6. As the level of zoom is PC5 for all three regions under consideration, the contribution of MA on the PC6 solution compared to SH will be tested in Section 7.3. In order to test performance of the SH and the contribution of the MA in terms of solution quality and computational time on PC5, a single run for Breda will be discussed first, as shown in Figure 22.

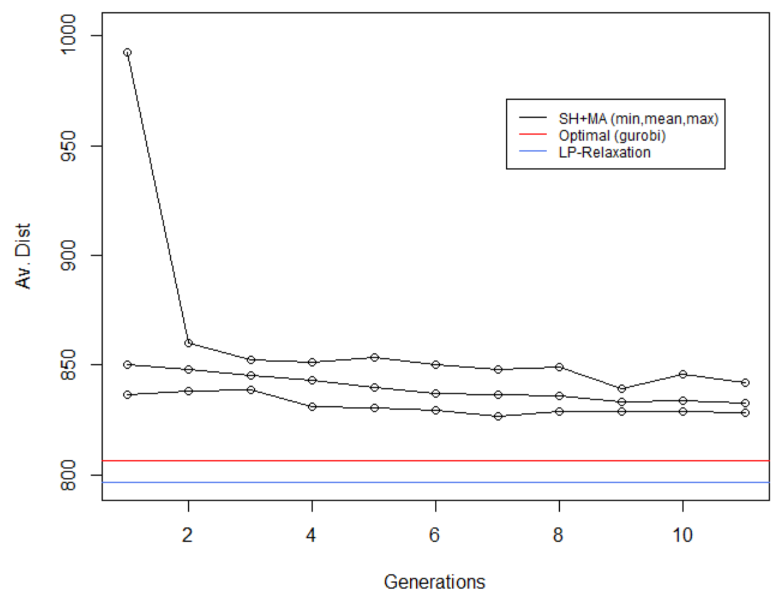

Figure 22: Development of SH + MA generations (Breda, Zoom = PC5, Popsize = 100, $\rho = 20\%$)

The Figure shows the LP-Relaxation, optimal solution and the range (min, mean, max) of feasible solutions found by SH + MA per generation of size 100 for the city of Breda on PC5. Note that the first generation is formed by the SH, where generation 2 up to and including 11 are generated by MA applying rank based selection with $\rho = 20\%$. As a result of the rule to terminate the MA after 4 consecutive generations without improvement, generation 7 shows the best found (minimum) average travelling distance (826 m) while respecting the capacities of all retailers. Note that the optimal solution value is 806 m (red line) and that the shortest distance solution of 621 m presented in figures 20 and 21 could have been included in Figure 22 as an additional lowerbound, next to the LP-Relaxation. Next, consider Figure 23 and Figure 24 which show the best feasible solution found in generation 7 and the optimal solution for Breda, respectively.

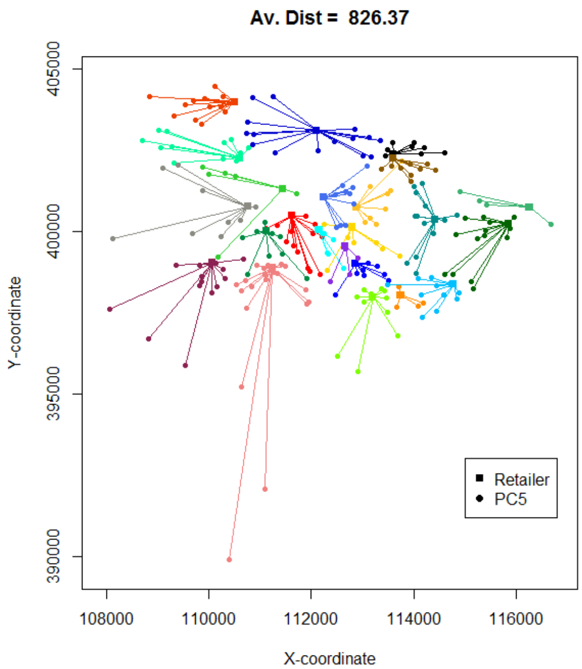

Figure 23: A feasible solution
(by SH + MA, Popsize = 100, $\rho = 20\%$)

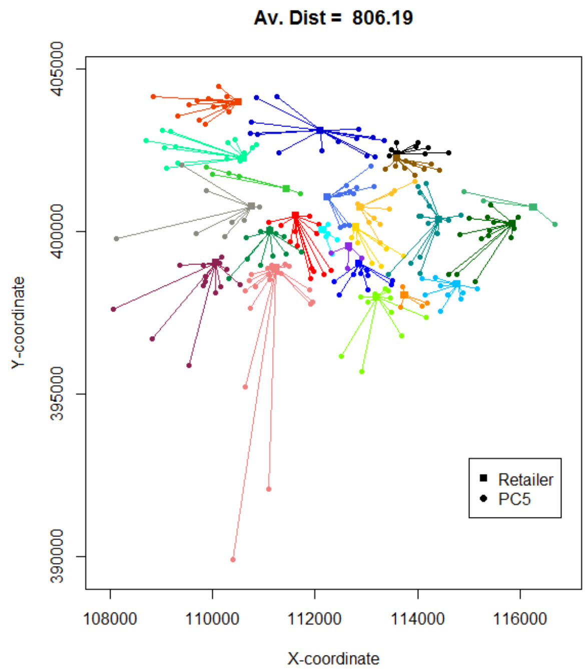

Figure 24: Optimal solution
(by Gurobi optimisation software)

When looking closely at these Figures, small differences can be seen in the assignments of PC5 to retailers. Again, the average distance per PC5 element is presented in the titles of both figures. The optimality gap of Figure 23 is thus $((826.37 - 806.19)/806.19) = 2.50\%$, in this example indicating an average distance reduction potential of $826.37 - 806.19 \approx 20$ meters. As both the SH as well the MA contain elements exposed to chance, multiple runs for varying settings are conducted, for which the results are presented in Table 5.

Breda (PC5)	Popsize	Runs (#)	Av. Gap (%)	Best Gap (%)	Av. Time (sec)	Runs improving SH (#)
SH	50	30	4.08 %	3.70 %	16.2	-
	100	30	3.99 %	3.63 %	34.7	-
	150	30	3.93 %	3.63 %	51.8	-
SH + MA	50	30	3.60 %	2.82 %	47.6	26
Bin. Tour. 5	100	30	3.38 %	2.46 %	94.5	30
	150	30	3.31 %	2.40 %	146.9	29
SH + MA	50	30	3.05 %	2.15 %	155.9	30
Rank $\rho = 20\%$	100	30	2.71 %	1.40 %	341.5	30
	150	30	2.81 %	1.45 %	473.0	30

Table 5: Results on SH and MA for Breda on PC5

For population sizes of 50, 100 and 150 a total of 30 runs have been performed for the SH. For similar population sizes, two types of MA have been tested, varying on discussed selection criteria. Note that the 30 SH runs make up for the first generation of both types MA, depending on population size. As can be seen from Table 5, the SH with population size 50 tends to find solutions with an optimality gap of 4.08% in an average of 16 seconds per run. Both MA settings show slightly improved average optimality gaps and enhance the SH in most (if not all) of 30 cases on solution value. Rank based selection shows most average improvement, at the cost of most average computation time. In order to reduce the average gap using MA, average computational expense must drastically increase, visualised in Figure 25.

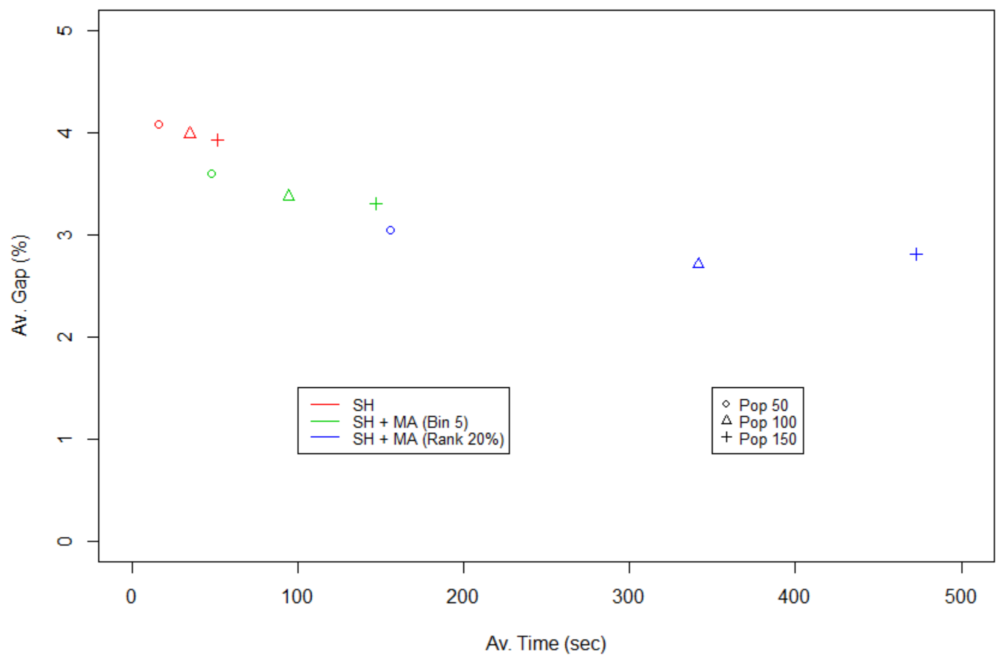

Figure 25: Av. Time required to reduce Av. Gap for Breda

The results for Flevoland and Amsterdam can be found in Appendix D in Table 10 and Table 11, respectively. Similarly as for Breda, the MA is able to improve the SH only marginally. For Flevoland, an average optimality gap of 9.52% found within 3 minutes by SH for population size 50 has mostly been improved by rank based MA of population size 150 to an average gap of 7.15%, using an extra 2,700 seconds (= 45 minutes) of computation time. Note that the average optimality gaps for Flevoland are strictly larger compared to Breda. One explanation for this is that the distribution of retail locations over the region is very different. Due to the potential lack of close alternatives, non-optimal assignments may add to substantial travelling distance for Flevoland, while non-optimal alternatives for Breda have relatively less effect on traveling distance. For Amsterdam, an average optimality gap of 7.01% found by the SH in little over 2 minutes has been improved by rank based MA upto 5.80% at the cost of an additional 10 minutes computation time. Based on the results on the subproblems on PC5 for Breda, Flevoland and Amsterdam, it can be stated that performance of SH is slightly outperformed by the MA, at the cost of significant computational costs.

7.2 Experiment 2: Performance of LP-Heuristic

After the SH or MA have finished improving Breda, Flevoland or Amsterdam on PC5, the next step is to break the solution into manageable clusters which can be considered on PC6. As discussed, these clusters are formed by applying k-means clustering on the retail locations as described in Algorithm 2, with a maximum of 10 retailers per cluster. Consider Figure 26, which shows the previously discussed solution of Breda on PC5. Figure 27 shows for the same solution a result of k-means clustering such that a total of 5 clusters have been formed containing 3, 3, 5, 5 and 7 retailers, summing to the total of 23 retailers in Breda.

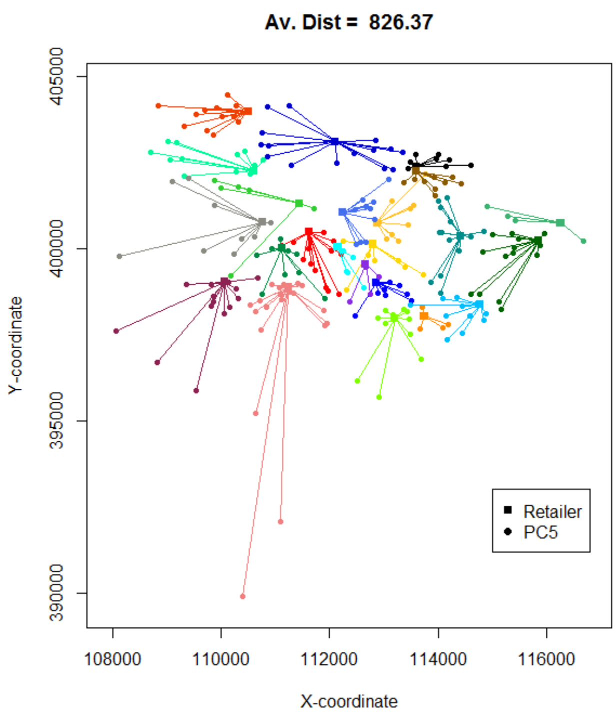

Figure 26: A feasible PC5 solution

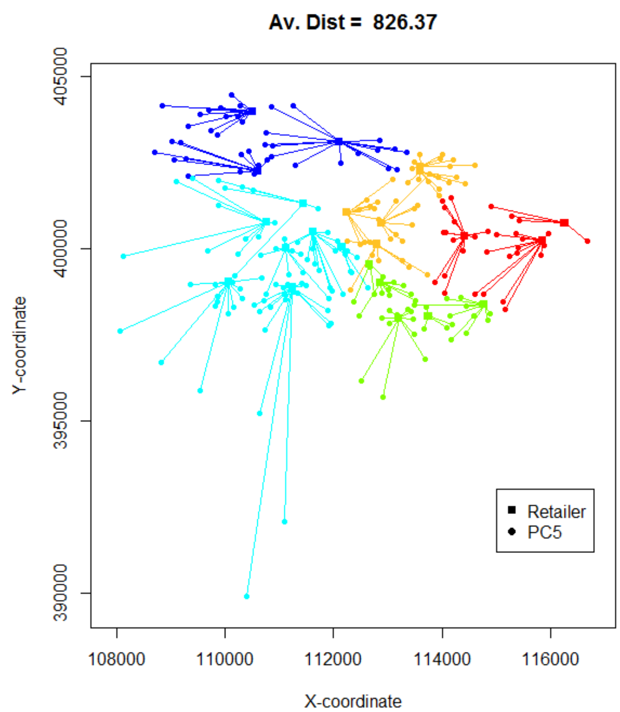

Figure 27: A feasible and clustered PC5 solution

Recall that as each retailer is feasibly assigned to a set of PC5 elements, each cluster in Figure 27 containing retailers and their corresponding set of PC5 elements is feasible as well. Next, the PC5 elements can be replaced by their PC6 elements and the LP-Heuristic can be individually applied on each cluster. For the cluster containing 5 retailers depicted in green in Figure 27, results on the LP-Heuristic and Gurobi solver are shown in Figures 28 and 29.

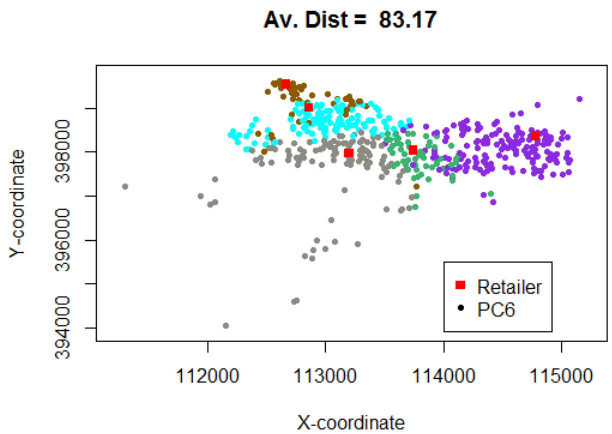

Figure 28: A feasible cluster solution on PC6 (by LP-Heuristic)

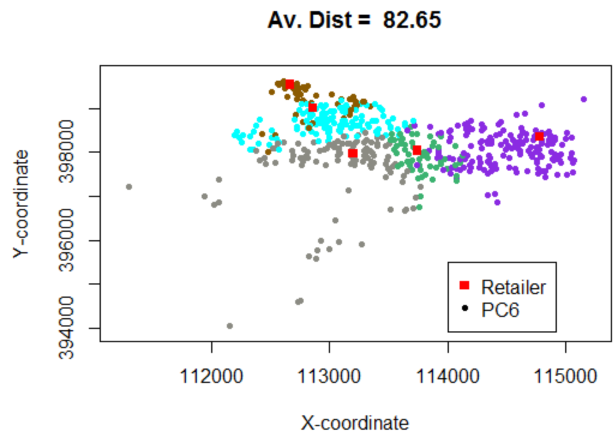

Figure 29: Optimal cluster solution on PC6 (by Gurobi optimisation software)

The differences between these feasible solutions are hard to see with the naked eye as 464 out of 476 PC6 elements are matched to the same retailer ($\approx$ 97.5%). The titles show summed weighted distance between PC6 and retailers per cluster, divided by the total number of parcels in all clusters. Iterating the LP-Heuristic and Gurobi software over all clusters in the region and adding the corresponding distances (titles), respectively, allows to compare performance and is denoted as a single run. In Table 6 the results are shown on cluster solutions for multiple complete runs on Breda, Flevoland and Amsterdam.

PC6	Runs (#)	Av. Opt. Gap Sum of Clusters (%)	Av. Time Sum of Clusters (sec)
Breda	270	0.27 %	3
Flevoland	90	0.20 %	12
Amsterdam	90	0.11 %	12

Table 6: Results on LP-Heuristic

The 270 runs for Breda follow from the 9 settings in Table 5 for which 30 runs are completed each, while for the other regions the 9 settings are runs 10 times each. The difference between the LP-Heuristic and the solution found by Gurobi optimisation software is often negligible and solutions for all clusters can relatively quickly be generated. However, note that in Table 6 the performance is measured against its potential, given the cluster. Take into account that in a situation where all 23 retailers of Breda are divided into 23 individual clusters, no exchange would be possible and the average optimality gap of the sum of clusters would be 0.00%. Ideally, the performance of the PC6 solution would be measured against a global PC6 bound, which could be either the Gurobi solution, the LP-Relaxation or the solution in which capacity is unlimited and thus the shortest distance is applied. The first two bounds cannot be computed by Gurobi and Rstudio due to limitations in size, whereas the latter bound provides very little (if any) information on solution quality due to binding capacity limitations, making it more difficult to fully measure the global performance of the LP-Heuristic. From a practical point of view, the LP-Heuristic is appropriate to be applied after the SH or SH + MA. Given the reasonably formed clusters, it is able to quickly find workable solutions with little room for improvement.

Considering the discussed cluster run for Breda, iterating the LP-Heuristic over the other clusters and combining their solutions into a mapping results in Figures 30 and 31. Both figures show the same solution indicating retailer catchment areas. Note that the cluster solution presented in Figure 28 can be found in the south-eastern part of Figures 30 and 31 such that catchment areas are indicated by the same color.

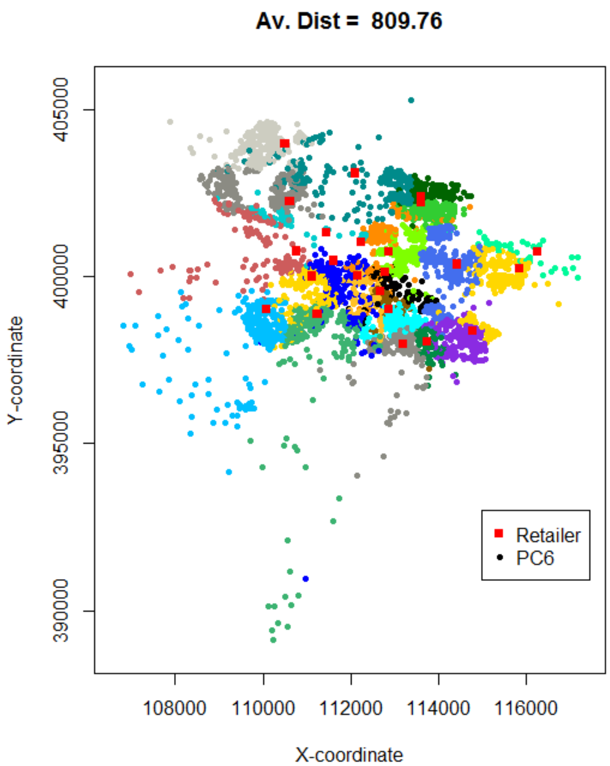

Figure 30: A feasible solution on PC6 for Breda (by complete method)

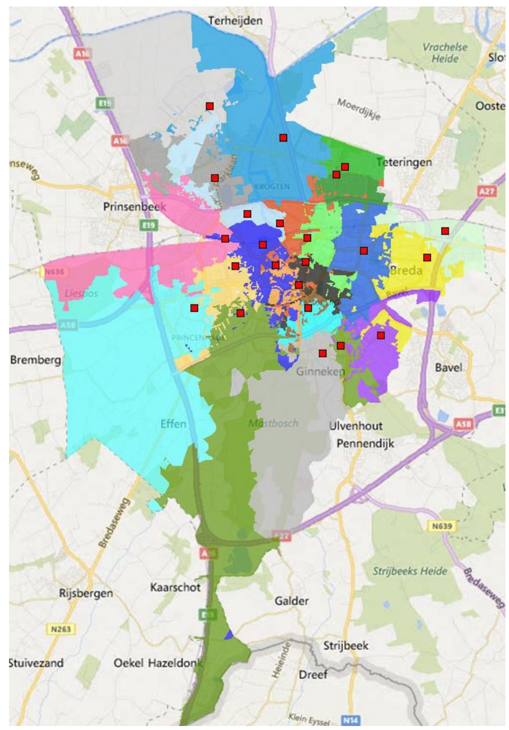

Figure 31: Feasible catchment areas of Breda (by complete method)

7.3 Experiment 3: Contribution MA on PC6 Solution if Initial Zoom ≠ PC6

While Section 7.1 considered the contribution of MA to SH on PC5 level, recall that the goal is to find an assignment of PC6 to retailers. In order to test the lasting effect of MA on the resulting PC6 solution, consider Table 7. This table shows results for Breda of runs which have been processed by the complete method from start to finish, either by the SH, or a combination of SH and MA, before being clustered and solved by the LP-Heuristic.

Breda (PC6)	Popsize	Runs (#)	Av. Dist (m)	Runs improv. SH (#)	Av. Time (sec)
Unl. Cap. Sol.	-	-	638.0	-	-
SH	50	30	819.8	-	19.1
	100	30	819.8	-	37.8
	150	30	819.6	-	54.9
SH + MA	50	30	815.6	26	49.3
Bin. Tour. 5	100	30	817.4	30	96.2
	150	30	814.5	29	148.6
SH + MA	50	30	815.2	30	160.0
Rank $\rho = 20\%$	100	30	814.5	30	345.9
	150	30	815.7	30	477.4

Table 7: Results complete method for Breda on PC6

The lasting improvement of MA in comparison to SH on the PC6 solution in combination with clustering and the LP-Heuristic can be found in Table 7. Under the currently tested settings, the PC6 solution found after applying the SH is on average 819.8, while the PC6 solutions for combinations of SH and MA tend to be slightly better, upto and average distance of 814.5, indicating 5 meter average reduction. Results on Flevoland and Amsterdam can be found in Appendix E and show similar marginal effect, upto an average reduction of 16 meters for Flevoland and 7 meters for Amsterdam. However, keep in mind that multiple elements of chance are involved when running the method from start to finish, such as the clustering and the order in which unmatched postal codes are evaluated while running the LP-Heuristic. These random elements slightly conceal the direct effect of MA to SH on the PC6 solution. On the other hand, the large number of complete iterations per setting (30 for Breda, 10 for Flevoland and Amsterdam) give reason to believe the MA does have a slight lasting effect on the PC6 solution. For all three tables, the only bound available is shown for reference, which is the situation where unlimited capacity would be available and shortest distance can be applied. This global bound however, provides little insight as the capacity restriction is in all three regions a binding constraint.

7.4 Experiment 4: Splitting Flows per PC6

In order to support the decision whether splits for the catchment area of the two flows *2TNA* and *ASIA* might be beneficial in the future, regard Table 8 which provides information on bounds for Breda.

Breda		**No Split**	**Split**
PC5	Number of postal code elements (#)	223	446
	Opt. Av. Fitness (m)	806.19	795.59
	Opt. Comp. Time (sec)	56	181
	LP-Relaxation Av. Fitness (m)	796.68	792.92
	Unlimited Capacity Av. Fitness (m)	621.28	621.28
PC6	Number of postal code elements (#)	3195	6390
	Opt. Av. Fitness (m)	-	-
	Opt. Comp. Time (sec)	-	-
	LP-Relaxation Av. Fitness (m)	-	-
	Unlimited Capacity Av. Fitness (m)	637.98	637.98

Table 8: Information on bounds

When splitting the flows *ASIA* and *2TNA*, the number of postal codes and corresponding problem double in size, adding flexibility to feasibly match postal codes at reduced distance. Again, information on bounds on PC6 is limited, inconvenient to fully appreciate the solutions found by the method. Table 9 presents results of 10 complete runs of the method for both situations 'No Split' and 'Split'. For practical reasons, the method is applied without MA, using an initial population for SH of size 50, followed by clustering and the LP-Heuristic per cluster.

Breda (PC6)	**No Split**	**Split**
Av. Dist (m)	819.8	797.0
Prop. PC6 with equal retailer (%)	100%	73.4%
Av. Time (sec)	19.1	34.4

Table 9: SH + Clustering + LP-Heuristic (10 runs, Popsize = 50)

Allowing a split for Breda under the current settings results in a substantial average reduction from 819.8 meter to 797.0, a reduction of over 20 meters per PC6 element. The corresponding proportion of PC6 elements with differing retailers for the two flows is on average 26.6%. Also, the extra computation time is relatively small compared to the previously discussed additional effect of the MA. For a customer, the potentially inconvenient varying parcel pick locations are compensated by reduced average travelling distance. The decision whether to split will in most cases hardly affect the retailer, as the capacity constraint is binding for both situations (split vs no split). For extreme situations in which capacity is highly under pressure, a split might be needed to gain feasibility. A practical downside of allowing flows to be split, is the extra uncertainty when assessing the values for parameters on parcel flow. While the focus of this thesis is on the performance of the method with fixed sets of input, the parameter estimation will be crucial in the objective to practically implement solutions. Splitting sparse historical information might cause parameter estimation for PC6 elements to become difficult and susceptible to outliers due to the limited number of recent observations.

8 Conclusion

This thesis focused to construct a stand-alone method to alleviate consumer parcel storage pressure at retailers without costly expansion or acquisition. The automation and optimisation of assigning a catchment area to a retailer has been looked into, referring to the set of households for which a retailer stores parcels in case direct delivery at the consumer is unsuccessful or unintended. The problem has been formulated as an NP-hard Generalised Assignment Problem (GAP), minimising weighted traveling distance between customers and retailers in a pre-defined region such that retailer capacity (m^3) is not exceeded. Input for the model is data on retailer capacities and scaled forecasts on amounts and volumes of parcels destined on household level for the considered region.

In order to solve the model, a set of rules including variations of the LP-Relaxation have been combined, referred to as the Starting Heuristics (SH). In addition, solution methods from literature to solve large instances of the GAP have been evaluated. Based on performance requirements from PostNL, application of a Memetic Algorithm (MA) has been further examined. Fictitious parameter settings have been used on the real life problem instances Breda, Flevoland and Amsterdam, appropriate to test performance of the SH and MA in Rstudio against Gurobi Optimisation Software, a licenced optimiser. As a result of the size and complexity of these regions, the subproblems on PC5 have been considered first. Based on the results from the experiments, it can be concluded that the MA slightly outperforms the SH under current settings at the cost of significant computation time. As time is no severe restriction for PostNL, the MA might be more suitable, depending on the preferences of the user. In a next experiment, clusters are formed using k-means clustering with constraints on cluster size, after which the performance of the proposed LP-Heuristic assigning PC6 elements to retailers is tested. Due to the absence of reasonable global bounds on detailed PC6 level such as an LP-Relaxation or optimal solution found by licensed software, performance relative to its cluster potential is tested. While the found solutions seem reasonable, this lack of information limits accurate appreciation of the quality of the solution on a global level. After clustering and applying the LP-Heuristic multiple times, the lasting effect of MA on the PC6 solution has been tested. It can be concluded that the MA contributes to the PC6 solution for all regions. However, the additional required computation time might become problematic, averaging at 44 minutes for one instance setting of Flevoland. Depending on the preference of the user, the SH or combination of SH + MA may be applied. Another experiment confirms the effect of splitting two types of flow per PC6, allowing for extra flexibility to respect capacity and minimise distance. The downside of splitting is the decrease of customer service for those who need to visit varying retailers. Based on the experiment, the effect of splitting the flows on distance is substantial for Breda, giving reason to believe that splits might be an appropriate way to reduce distance or an alternative in a situation where capacity in a region is highly limited and expansion is undesired or too costly.

It can be concluded that the research objective to construct a stand-alone method assigning catchment areas appears to be achieved. The method has proven to adequately construct areas while taking capacity limitations and travelling distance into account for all regions and instances considered. With the expectation of further parcel flow increase over the coming years, the proposed method has not only been tested on performance, but has also been further developed into a suitable tool to support decision making on catchment areas in the operations of PostNL, beneficial for customers, retailers and PostNL.

9 Practical Considerations, Limitations and Further Research

While the proposed method has tested to be suitable to support decision making on catchment areas for the discussed areas, limitations should be taken into account when practically implementing the provided solutions.

9.1 Appropriate Problem Instances Setting

Instance setting is an important element which affects the catchment areas found by the method. While Breda may seem as a separate, logically chosen area to solve, inclusion of suburban areas or surroundings might result in a partly different solution. For instance, by reconsidering Figure 31, one might argue why the adjacent village of Bavel (west of Breda) is not included in the problem instance of Breda. Inclusion might change catchment areas, depending on the demand from Bavel and the amount of present storage capacity. In the goal to minimise average travelling distance for all PC6 to their matched retailer the Netherlands, it could be argued that Bavel must be included. However, similar arguments lead to inclusion of the adjacent Teteringen, followed by Oosterhout (and so on). Following this logic will result in inclusion of all regions in The Netherlands, giving the problem an inappropriate level of complexity.

In contrast to the previous reasoning, it might be argued that the considered area of Flevoland may be split into non-overlapping areas, concerning for instance Almere, Lelystad, Dronten, Zeewolde and Emmeloord such that the union of these sets contain all of Flevoland. Households in Emmeloord will most likely not be asked to pick up their parcel in Almere due to their distance. In addition, when solving the areas separately the solutions can be found in less computation time. However, by splitting Flevoland into smaller regions, potential capacity insufficiency in an area can no longer be solved by exchange due to the created boundaries.

Synthesising, the problem instance setting phase is an important, subjective step which influences the solutions found by the method. The instances must be a balanced mix of size and logical, natural boundaries in the region to be solved, such that the union of instances covers the entire country. The fixed areas Breda, Flevoland and Amsterdam are taken as problem instances to test the method, but the user should be aware of the effect of alternative region setting has effect on the to be found catchment areas.

9.2 Data Pre-processing

Before practically implementing the tool at PostNL, elaboration on pre-processing steps of data is crucial. The discussed steps only touch upon some of the considerations when determining appropriate values for the IP formulation. Concepts as setting the time horizon for which catchment areas are expected to last, seasonality on parcel flow and buffer or piling assumptions are highly prioritised for future work. While this thesis merely focuses on the construction and application of the stand-alone method, this pre-processing will require most attention once all retailer capacities are known. Also, in order to streamline the process, periodically updates must be automated, securing the retail locations, capacities and demand forecasts are accurate. For the locations, another discussed practical limitation is the use of the distance measure 'as the crow flies'. This might result in unreasonable solutions when obstacles as rivers are present in the area under consideration. An improvement to the currently applied measure would incorporate such information, by for instance taking the distance by road.

9.3 Methodology

A downside of the MA is its numerous options for parameter settings such as mutation rate, selection and termination. While under the currently tested settings the contribution to SH was measurable but little, parameter optimisation per problem instance may improve the solution quality. However, from a practical point of view, little knowledge on parameter setting should be demanded from operators of the tool, limiting potential contribution of MA to unknown problem instances. Also, as the MA is subject to elements involving chance, the seed functions are applied to orchestrate the probabilistic runs. This is useful for future operators of the tool, as runs under similar settings will generate identical answers when ran more than once. As an alternative, as solutions vary for differing seed settings, one might decide to run the method for multiple seeds and choose the best one, potentially adding to solution quality while required computational time will grow. An additional improvement might be found in the efficiency of coding or improvements on the local searches on fitness and unfitness reduction. These local searches only consider a small part of the potential improvements. In general, the absence of useful bounds on PC6 in Rstudio slightly limits the appreciation of the solutions found by the method. Other statistical or (licenced) optimisation programs might be able to handle the problem sizes of PC6 and compute the LP-Relaxation or even optimal solution to use as bounds. Future work might also be directed to the clustering phase. As discusses, the k-means cluster solution is based on chance such that the found clusters for the same instance might differ. Differing clusters influence the domain of the LP-Heuristic and corresponding final solution. The effect of a cluster on the final PC6 solution can only be evaluated in retrospect, once the PC6 solution is found. Again, from a practical point of view, multiple cluster runs might be applied such that the best final PC6 solution can be chosen.

Appendices

A Outline LP-Heuristic

Data: A potential fractional solution
Result: A binary solution

```
Initial values: LP-Relaxation matrix solution, attempt = 0;
Optional initial value: Solution found by MA;
Solution ← SetFractionalValuesToZero(LP-RelaxationMatrixSolution) ;
SetOfUnmatchedPC ← FindColumnsOnlyZero(Solution) ;
while (SetOfUnmatchedPC ≠ ∅) & (attempt < 100) do
    attempt ← attempt + 1 ;
    NextPC ← SelectRandomPC(SetOfUnmatchedPC) ;
    if Volume(NextPC) ≤ max(AvRetCap(Solution)) then
        FittingRetailers ← FindRetailers(AvRetCap ≥ Volume(NextPC)) ;
        NextRetailer ← FindMinDist(FittingRetailers,NextPC) ;
        Solution[ NextRetailer , NextPC ] ← 1 ;
        Update(SetOfUnmatchedPC, AvRetCap) ;
    end
    else
        NextRet ← RetailerWithMaxAvRetCap ;
        DeltaVolume ← Volume(NextPC) - AvRetCap(NextRet) ;
        MatchedPCSet ← FindPCMatchedToRetailer(Solution,NextRet) ;
        NewUnmatchedPC ← FindPCWithMinVolume[ (Volume(MatchedPCSet) ≥ DeltaVolume) &
          (Volume(MatchedPCSet) ≠ Volume(NextPC)) ] ;
        Solution [ NextRet, NewUnmatchedPC ] ← 0 ;
        Solution [ NextRet, NextPC ] ← 1 ;
        Update(SetOfUnmatchedPC,AvRetCap) ;
    end
end
if SetOfUnmatchedPC ≠ ∅ then
    Solution ← Combine(Solution,AssignShortestDistance(SetOfUnMatchedPC)) ;
end
Solution ← LocalSearch(UnfitnessReductionRules,Solution) ;
Solution ← LocalSearch(FitnessReductionRules,Solution) ;
AlternativeSolution ← RoundFractionalColumns(LP-RelaxationMatrixSolution) ;
AlternativeSolution ← LocalSearch(UnfitnessReductionRules,AlternativeSolution) ;
AlternativeSolution ← LocalSearch(FitnessReductionRules,AlternativeSolution) ;
BestSolution ← StoreBestSolution(Solution,AlternativeSolution) ;
if Available(SolutionFoundByMA) then
    ImprovedMASolution ← LocalSearch(FitnessReductionRules,SolutionFoundByMA) ;
    BestSolution ← StoreBestSolution(Solution,AlternativeSolution,ImprovedMASolution) ;
end
return BestSolution;
```

Algorithm 4: Pseudo-code on LP-Heuristic

B Outline Local Search on Unfitness Reduction

Data: Any solution

Result: A solution with non increasing unfitness

```
if Unfitness(Solution) > 0 then
    ToBeConsideredRet ← RandomOrder(FindRetWithCapProblems(Solution);
    for i in 1 : count(ToBeConsideredRet) do
        NextRet ← ToBeConsideredRet [ i ];
        I_UpdateDist ← TRUE;
        I_StopSearch ← FALSE;
        while (I_StopSearch = FALSE) & (AvRetCap(NextRet) < 0) & (Counter < MaxCheck) do
            PCInNextRet ← FindAllColumns(Solution = NextRet);
            if I_UpdateDist = TRUE then
                SelectedDistMatrix ← SelectColumns( PCInNextRet , FullDistMatrix );
                SetAllElementsInRow(SelectedDistMatrix [ NextRet ] ) ← ∞;
            end
            AltRet ← FindRowWithMinValue(SelectedDistMatrix);
            AltPC ← PCInCurrentRet[ FindColWithMinValue(SelectedDistMatrix) ];
            Switch ← min(AvRetCap(AltRet) - Vol(AltPC) , 0) + min(AvRetCap [ NextRet ] +
             Vol(AltPC) , 0);
            NoSwitch ← min(AvRetCap(AltRet) , 0) + min(AvRetCap(NextRet) , 0);
            if Switch > NoSwitch then
                if min(AvRetCap(AltRet) - Vol(AltPC) , 0) < 0 then
                    ToBeConsideredRet ← AddElement(AltRet , ToBeConsideredRet);
                end
                Solution [ AltPC ] ← AltRet;
                I_UpdateDist ← TRUE;
            end
            else
                SelectedDistMatrix [ AltRet , FindColWithMinValue(SelectedDistMatrix) ] ← ∞;
                I_UpdateDist ← FALSE;
            end
            if FindMinValue(SelectedDistMatrix) = ∞ then
                I_StopSearch ← TRUE;
            end
            Counter ← Counter + 1 ;
        end
    end
end
return  Solution;
```

Algorithm 5: Pseudo-code on unfitness reduction

C Part of Outline Local Search on Fitness Reduction

Data: Any solution
Result: A solution with non increasing fitness and unfitness

```
Initial values: j = 1 ; RandomOrderPC = 1 ;
while j ≤ RandomOrderPC do
    CurrentRetailer ← FindRetailerWithMostAvailableCap(Solution) ;
    CurrentMostAvailableCap ← Cap(CurrentRetailer) ;
    FittingPC ← FindPC(Volume ≤ CurrentMostAvailableCap) ;
    RandomOrderPC ← sample(FittingPC,length(FittingPC)) ;
    I_Change ← FALSE ;
    while (I_Change = FALSE) & (j ≤ length(RandomOrderPC) do
        Switch ← Distance [ CurrentRetailer,RandomOrder [ j ] ] ;
        NoSwitch ← Distance [ Solution [ RandomOrderPC [ j ] ] , RandomOrderPC [ j ] ] ;
        if Switch < NoSwitch then
            I_Change ← TRUE ;
            Solution[ RandomOrder [ j ] ] ← CurrentRetailer ;
            if FindRetailerWithMostAvailableCap(Solution) != CurrentRetailer then
                j ← 0 ;
            end
        end
        j ← j+1 ;
    end
end
return Solution;
```

Algorithm 6: Pseudo-Code on fitness reduction

D Additional Tables for Section 7.1

Flevoland (PC5)	Popsize	Av. Gap (%)	Best Gap (%)	Av. Time (sec)	Runs improving SH (#)
SH	50	9.52 %	8.98 %	153	-
10 runs	100	9.42 %	8.98 %	307	-
	150	9.37 %	8.97 %	641	-
SH + MA	50	8.38 %	7.19 %	256	8
Bin. Tour. 5	100	8.70 %	7.76 %	483	6
10 runs	150	8.36 %	7.49 %	1,578	10
SH + MA	50	8.12 %	7.26 %	902	9
Rank $\rho = 20\%$	100	7.52 %	6.95 %	1,682	10
10 runs	150	7.15 %	6.28 %	2,842	10

Table 10: Results on SH and MA for Flevoland on PC5

Amsterdam (PC5)	Popsize	Av. Gap (%)	Best Gap (%)	Av. Time (sec)	Runs improving SH (#)
SH	50	7.01 %	6.81 %	133	-
10 runs	100	6.83 %	6.54 %	259	-
	150	6.72 %	6.54 %	390	-
SH + MA	50	6.49 %	4.96 %	351	5
Bin. Tour. 5	100	6.60 %	5.46 %	653	3
10 runs	150	6.39 %	5.67 %	1,108	7
SH + MA	50	6.24 %	5.01 %	339	7
Rank $\rho = 20\%$	100	5.80 %	4.36 %	741	8
10 runs	150	6.22 %	5.12 %	865	7

Table 11: Results on SH and MA for Amsterdam on PC5

E Additional Tables for Section 7.3

Flevoland (PC6)	Popsize	Runs (#)	Av. Dist (m)	Runs improv. SH (#)	Av. Time (sec)
Unl. Cap. Sol.	-	-	857	-	-
SH	50	10	1,090	-	165
	100	10	1,089	-	319
	150	10	1,088	-	657
SH + MA	50	10	1,083	8	262
Bin. Tour. 5	100	10	1,084	6	489
	150	10	1,083	10	1,592
SH + MA	50	10	1,085	9	920
Rank $\rho = 20\%$	100	10	1,077	10	1,699
	150	10	1,074	10	2,859

Table 12: Results complete method for Flevoland on PC6

Amsterdam (PC6)	Popsize	Runs (#)	Av. Dist (m)	Runs improv. SH (#)	Av. Time (sec)
Unl. Cap. Sol.	-	-	417.6	-	-
SH	50	10	567.5	-	145
	100	10	566.2	-	270
	150	10	565.3	-	402
SH + MA	50	10	563.9	5	363
Bin. Tour. 5	100	10	565.5	3	665
	150	10	563.7	7	1,112
SH + MA	50	10	564.7	7	351
Rank $\rho = 20\%$	100	10	561.6	8	752
	150	10	564.3	7	877

Table 13: Results complete method for Amsterdam on PC6

References

Ron Adany, Moran Feldman, Elad Haramaty, Rohit Khandekar, Baruch Schieber, Roy Schwartz, Hadas Shachnai, and Tami Tamir. All-or-nothing generalized assignment with application to scheduling advertising campaigns. *ACM Transactions on Algorithms (TALG)*, 12(3):38, 2016.

Joep Aerts, Jan Korst, and Frederik Spieksma. An approximation algorithm for a generalized assignment problem with small resource requirements. 2003.

V Balachandran. An integer generalized transportation model for optimal job assignment in computer networks. *Operations Research*, 24(4):742–759, 1976.

John E Beasley. Or-library: distributing test problems by electronic mail. *Journal of the operational research society*, 41(11):1069–1072, 1990.

Marcel Bogers, Kristian J Sund, and Juan Andrei Villarroel Fernandez. The organizational dimension of business model exploration: Evidence from the european postal industry. *Business Model Innovation: The Organizational Dimension. Oxford: Oxford University Press, Forthcoming*, 2014.

Paul C Chu and John E Beasley. A genetic algorithm for the generalised assignment problem. *Computers & Operations Research*, 24(1):17–23, 1997.

Robert G Cromley and Dean M Hanink. Coupling land use allocation models with raster gis. *Journal of Geographical Systems*, 1(2):137–153, 1999.

Juan A Dıaz and Elena Fernández. A tabu search heuristic for the generalized assignment problem. *European Journal of Operational Research*, 132(1):22–38, 2001.

Harald Feltl and Günther R Raidl. An improved hybrid genetic algorithm for the generalized assignment problem. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 990–995. ACM, 2004.

Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. ISBN 0716710447.

John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

Tung Leng Lau and Edward PK Tsang. The guided genetic algorithm and its application to the generalized assignment problem. In *Tools with Artificial Intelligence, 1998. Proceedings. Tenth IEEE International Conference on*, pages 336–343. IEEE, 1998.

Yan Y Liu and Shaowen Wang. A scalable parallel genetic algorithm for the generalized assignment problem. *Parallel Computing*, 46:98–119, 2015.

Luiz AN Lorena, Marcelo G Narciso, and JE Beasley. A constructive genetic algorithm for the generalized assignment problem. *Evolutionary Optimization*, 5:1–19, 2002.

Mohd Razali Noraini and John Geraghty. Genetic algorithm performance with different selection strategies in solving tsp. 2011.

Michael G Norman and Pablo Moscato. A competitive and cooperative approach to complex combinatorial search. In *Proceedings of the 20th Informatics and Operations Research Meeting*, pages 3–15. Citeseer, 1991.

Alexandre Pigatti, Marcus Poggi De Aragao, and Eduardo Uchoa. Stabilized branch-and-cut-and-price for the generalized assignment problem. *Electronic Notes in Discrete Mathematics*, 19:389–395, 2005.

Marius Posta, Jacques A Ferland, and Philippe Michelon. An exact method with variable fixing for solving the generalized assignment problem. *Computational Optimization and Applications*, 52(3): 629–644, 2012.

PostNL. Postnl jaarverslagen, 2012-2016.

Fubin Qian and Rui Ding. Simulated annealing for the 0/1 multidimensional knapsack problem. *NUMERICAL MATHEMATICS-ENGLISH SERIES-*, 16(4):320, 2007.

G Terry Ross and Richard M Soland. A branch and bound algorithm for the generalized assignment problem. *Mathematical programming*, 8(1):91–103, 1975.

Ruslan Sadykov, François Vanderbeck, Artur Pessoa, and Eduardo Uchoa. Column generation based heuristic for the generalized assignment problem. *XLVII Simpósio Brasileiro de Pesquisa Operacional, Porto de Galinhas, Brazil*, 2015.

Martin Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations research*, 45(6):831–841, 1997.

JM Wilson. A genetic algorithm for the generalised assignment problem. *Journal of the Operational Research Society*, 48(8):804–809, 1997.

Mutsunori Yagiura, Toshihide Ibaraki, and Fred Glover. A path relinking approach with ejection chains for the generalized assignment problem. *European journal of operational research*, 169(2):548–569, 2006.