Recent Mathematical Approaches to Service Territory Design

Zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

von der Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte
Dissertation

von
Dipl.-Wi.-Ing. Matthias Bender

Tag der mündlichen Prüfung:	10. Oktober 2017
Referent:	Prof. Dr. Stefan Nickel
Korreferent:	Prof. Dr. Jörg Kalcsics

Karlsruhe, 2017

Acknowledgments

This thesis would not have been possible without the support of several people. In the following, I would like to express my gratitude to them.

First of all, I thank my supervisor Prof. Dr. Stefan Nickel for giving me the opportunity to write this thesis. He provided me with continuous support and guidance over all the years and, at the same time, gave me the freedom to pursue my own ideas.

I want to thank my co-supervisor Prof. Dr. Jörg Kalcsics for numerous fruitful discussions and his valuable feedback to my work. I am particularly grateful for the intense exchange of ideas during our joint research stay in Montréal.

I would like to thank all colleagues at the Department of Logistics and Supply Chain Optimization of the Research Center for Information Technology (FZI). Writing this thesis was greatly facilitated by the pleasant working atmosphere. A special thanks is due to Alexander Butsch, Anne Meyer, Boris Amberg, David Sayah and Martin Pouls for proofreading this thesis. Moreover, I thank Alexander and Anne for many inspiring discussions on territory design.

Furthermore, I want to thank my project partner PTV Group, particularly Werner Heid, Frank Radaschewski, Nitin Ahuja, Jürgen Pöhler and Dietmar Flohs, for the excellent cooperation for more than five years. This thesis has benefited considerably from the meetings and discussions in our joint project. I also thank PTV Group for providing real-world problem data as well as their PTV xServers.

I would like to thank my parents for their encouragement and support throughout my entire life. Last and most importantly, I want to thank my wife Jennifer for her love, patience and understanding, especially during the last months of writing this thesis.

Abstract

Many companies and institutions operate a field service workforce to provide services at their customers' sites. Examples include the sales force of consumer goods manufacturers, the field service technicians of engineering companies, and the nurses of home-health care providers. To obtain clearly defined areas of responsibility, the geographical region under study is in many cases subdivided into service territories, each of which is served by a single field worker or a team of field workers. The design of service territories is subject to various planning criteria. The most common ones are geographical compactness, contiguity, and balance in terms of workload or income potential, but there can be several additional criteria and requirements depending on the specific application.

In this thesis, we deal with the development of mathematical models and methods for service territory design problems. Our focus is on planning requirements that are relevant for practice, but have received little attention in the existing literature on territory design so far. We address the question how these requirements can be incorporated into mathematical models and mathematical programming based solution methods. We first present requirements that restrict the feasible assignments of customers to field workers and provide components for their integration into mathematical models. We further consider the requirement that customers must be served multiple times during a given planning horizon. We introduce the resulting problem, which we call the multi-period service territory design problem (MPSTDP). It has not yet been studied in the literature. The emphasis is put on the scheduling task of the MPSTDP, which deals with the assignment of service visits to the days of the planning horizon. We formally define this task and devise a heuristic solution method. Our heuristic produces high-quality solutions and clearly outperforms the existing software product of our industry partner. Moreover, we present the first specially-tailored exact solution method for this task: a branch-and-price algorithm that incorporates specialized acceleration techniques, such as a fast pricing heuristic and symmetry reduction techniques. Ultimately, we study the design of territories for parcel delivery companies. We address the tactical design of the territories and their daily adjustment in order to cope with demand fluctuations. The problem involves determining the number of territories and assigning heterogeneous resources to the territories, a combination not yet addressed in literature. We propose different models as well as a heuristic solution approach, and we perform an extensive case study on real-world problem data.

Contents

List of Figures

List of Tables

List of Abbreviations

CVRP	Capacitated vehicle routing problem
IP	Integer program(ming)
IRP	Inventory routing problem
LP	Linear program(ming)
LS branching	Largest split branching
MIP	Mixed integer program(ming)
MPSTDP	Multi-period service territory design problem
MPSTDP-P	Partitioning subproblem of the multi-period service territory design problem
MPSTDP-S	Scheduling subproblem of the multi-period service territory design problem
MPSTDP-S*	Special planning scenario of the scheduling subproblem of the multi-period service territory design problem
OR	Operations research
PSD branching	Pseudocost branching
PTV	PTV Group
PVRP	Period vehicle routing problem
RMP	Restricted master problem

SR inequality	Subset-row inequality
TSP	Traveling salesman problem
VRP	Vehicle routing problem

1 Introduction

Territories play an important role in the provision of various services. Consider, for example, a company that operates a sales force to perform selling and promotional activities. Among the typical tasks that the sales manager of the company faces is the assignment of customer accounts to salespersons with the aim of establishing clearly defined sales territories. An exemplary design of 130 sales territories is illustrated in Figure 1.1. Each sales territory constitutes the area of responsibility for a specific salesperson or team, and, hence, the design of sales territories has an enormous impact on the daily work of the sales force. According to Zoltners and Sinha (2005), well-designed territories provide equitable workload and income opportunities for all salespeople as well as short travel times, which results in an improved morale of the salespeople, an increase in selling time, and reduced travel costs. They estimate that a good territory design yields a 2% to 7% increase in sales compared to an average design. Moreover, they report that roughly 11% of all full-time employees in the USA are retail or field salespeople, and that the associated costs amount to more than a trillion dollars per year. This points out the high economic importance of sales territory design. Besides the management of a sales force, territories are also used to organize the provision of services in several other applications, which emphasizes the great relevance of service territory design for practice. Common examples include mail and parcel delivery (Bodin and Levy, 1991; Wong, 2008), home-health care (Benzarti et al., 2013), the collection of solid waste (Hanafi et al., 1999), snow disposal and winter gritting (Muyldermans et al., 2002; Perrier et al., 2008), road maintenance (Chen et al., 2017), and emergency services (Baker et al., 1989).

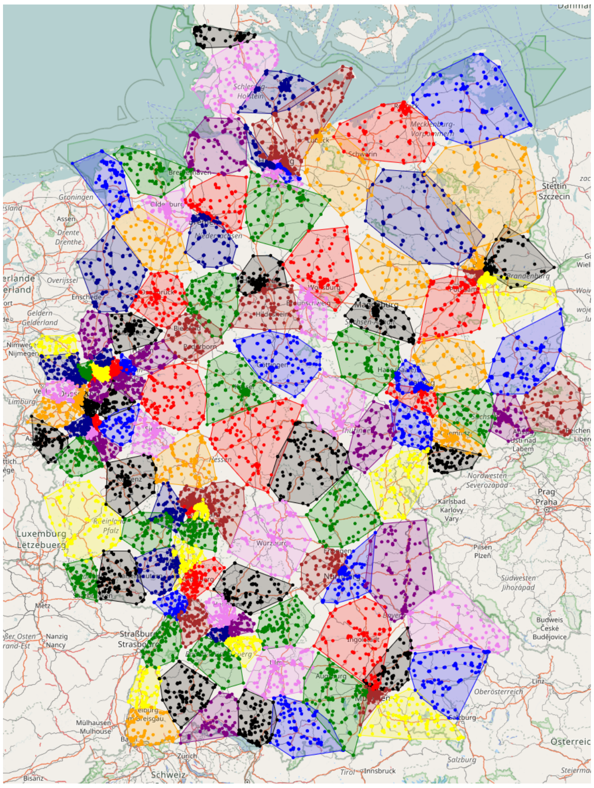


Figure 1.1: Exemplary territory design solution for a problem consisting of over 17,000 customers grouped into 130 sales territories (map data © OpenStreetMap contributors)

In general, territory design or districting is the task of grouping small geographic units, which are called basic areas, into larger clusters, which are called territories or districts, such that some relevant planning criteria are satisfied (Kalcsics, 2015). The most common planning criteria are geographical compactness, contiguity, and balance (see Section 2.1.3 for a more detailed description of these criteria). A district is considered compact if it is fairly round-shaped and undistorted. Contiguity means that it is possible to travel from a basic area to any other basic area in the same district without leaving the district. Balance describes the desire for districts that have roughly the same size with respect to one or multiple quantifiable attributes. Besides these criteria, there can be various additional planning criteria and requirements depending on the specific application.

Since real-world problems may comprise thousands of basic areas which must be grouped into several dozens of districts while multiple planning criteria have to be taken into account (Fleischmann and Paraschis, 1988; López-Pérez and Ríos-Mercado, 2013), territory design is a very challenging task. Hence, methods of operations research (OR) are commonly used to support decision-makers. Solutions that are computed using OR models and methods will, however, only be accepted by decision-makers if all important planning criteria and requirements are appropriately taken into account. Otherwise, the generated solutions must be revised manually, which is very time-consuming and, therefore, should be reduced to a minimum.

This thesis deals with the development of mathematical models and solution methods capable of solving real-world territory design problems. The peculiarities of the various applications and the desire to keep manual post-processing at a minimum call for application-specific approaches. We focus on applications arising in the provision of on-site services, i.e., services that are provided at the customers' sites. We further concentrate on planning requirements that have widely been neglected in the districting community, although we consider them as highly relevant in practice. The practice-oriented approach followed in this thesis is further underlined by the fact that the considered planning requirements have been identified in a joint project with our industry partner PTV Group (PTV)[1], which has many years of experience in providing commercial software and consulting services for territory design.

In the remainder of this introductory chapter, we present the main contributions of this thesis and explain its organization.

1.1 Contribution

The main contributions of this thesis can be summarized as follows.

Besides the typical planning criteria encountered in almost every territory design problem, we present some **additional planning requirements that arise in the provision of on-site services**. The requirements have been identified in cooperation with PTV and are motivated by real-world problems. Despite their high practical relevance, they have, to date, received little, some of them even no attention in the districting literature.

We address the question how these requirements can be reflected in **mathematical models and mathematical programming based solution approaches**. Our focus is on approaches that are capable of solving problems of realistic size. First, we deal with restrictions regarding the assignment of basic areas to districts which arise, for example, from customers requiring special skills from the employee responsible for serving them. We adopt a well-known integer programming model from the districting literature and provide model components that can be used to incorporate such **assignment restrictions**.

[1] `http://www.ptvgroup.com`

Furthermore, we consider the requirement that customers must be visited several times in a given planning horizon. The resulting problem is an extension of the classical territory design problem to a multi-period setting and has not yet been studied in the scientific literature. We refer to it as the **multi-period service territory design problem** (MPSTDP). In addition to grouping basic areas into districts, it includes the **scheduling of recurring service visits** over the planning horizon subject to customer-specific restrictions. This gives rise to the question what role the classical criteria compactness and balance play in a multi-period setting. In this thesis, we focus on the scheduling task of the MPSTDP, i.e., we concentrate on the assignment of service visits to the days of the planning horizon. We elaborate on relevant criteria, define the task formally and propose a location-allocation heuristic. Our heuristic is, as far as we are aware, the first extension of the well-known location-allocation approach of Hess et al. (1965) to a multi-period setting. Moreover, we present the first exact branch-and-price algorithm for the scheduling task of the MPSTDP. It incorporates specialized acceleration techniques, in particular a fast pricing heuristic and techniques to reduce the symmetry inherent to the proposed model. Both the location-allocation heuristic and the branch-and-price algorithm are extensively evaluated on real-world problem instances.

We further study a territory design problem faced by parcel delivery companies, such as DHL and FedEx. The problem extends the classical territory design problem since it involves the **assignment of resources to districts**. Two categories of resources are considered, namely a heterogeneous fleet of vehicles, which differ in their loading capacities, and a heterogeneous crew of drivers, who differ in their contractual working times. In addition, it is part of the problem to **determine the number of districts**. To the best of our knowledge, the assignment of resources to districts in combination with the determination of the number of districts has not been studied before. Furthermore, we address the question what **day-to-day adjustments** in the district design should be made in order to cope with demand fluctuations. We present mathematical models for the tactical design of the districts and their adjustment in day-to-day business. The focus is on finding a reasonable trade-off between resource efficiency, compliance with the drivers' working times, workload balance between drivers, and service consistency. We propose a simple, yet effective heuristic, which is evaluated in a case study based on real-world problem data.

The work presented in this thesis has yielded two scientific articles:

1. Bender, M., Meyer, A., Kalcsics, J., Nickel, S. (2016). The multi-period service territory design problem – An introduction, a model and a heuristic approach. *Transportation Research Part E: Logistics and Transportation Review*, 96:135–157.

2. Bender, M., Kalcsics, J., Nickel, S., Pouls, M. A branch-and-price algorithm for the scheduling of customer visits in the context of multi-period service territory design. Submitted to *European Journal of Operational Research* in February 2017; under revision as of June 2017.

1.2 Organization

This thesis is structured as illustrated in Figure 1.2.

Contribution and Organization	
Chapter 1:	Introduction
Foundations	
Chapter 2:	Fundamentals of Service Territory Design
Chapter 3:	Mathematical Modeling and Solution Principles
Applications	
Chapter 4:	The Multi-Period Service Territory Design Problem – An Introduction, a Model and a Heuristic Approach
Chapter 5:	A Branch-and-Price Algorithm for the Scheduling of Customer Visits in the Context of Multi-Period Service Territory Design
Chapter 6:	Districting for Parcel Delivery Services – A Two-Stage Solution Approach and a Real-World Case Study
Summary	
Chapter 7:	Conclusions and Outlook

Figure 1.2: Organization of this thesis

In Chapter 2, we provide a short introduction to territory design. In particular, we present typical applications and planning criteria commonly encountered. Moreover, we describe additional planning requirements that play an important role in the provision of on-site services, and we outline how we address them in this thesis.

In Chapter 3, we present a well-known mathematical model and demonstrate how this model can be extended such that geographical contiguity is ensured and assignment restrictions are satisfied. In addition, we give a short overview of solution principles that are relevant for this thesis.

Chapters 4–6 address specific applications and constitute the main part of this work. Each of these chapters is self-contained, hence allowing readers to pick and choose the topics they are most interested in. In Chapter 4, we introduce the MPSTDP, which involves the scheduling of service visits over a given planning horizon. The emphasis is put on the scheduling task of the MPSTDP. We propose a mixed integer programming model for this task and develop a heuristic solution method. In Chapter 5, we study a highly relevant planning scenario of this task and propose an exact branch-and-price algorithm including specialized acceleration techniques. Chapter 6 deals with the design of districts for parcel delivery companies. We formulate different models, present a heuristic solution approach, and perform an extensive case study.

Finally, we sum up the major contributions of this thesis and suggest directions for future research in Chapter 7.

2

Fundamentals of Service Territory Design

This chapter begins with a short primer on territory design in general. We present common applications, introduce the main components of territory design problems and describe the associated planning task. Then, we go into the peculiarities of service territory design, which is the focus of this thesis. We present various additional planning requirements that arise specifically in territory design applications for on-site services and sketch how we address them in the remainder of this thesis.

2.1 Territory Design in General

In this section, we give a brief overview of the most important territory design applications to illustrate the typical planning tasks and criteria. Based on this, we explain the main components of territory design problems, describe the general problem statement and provide a summary of the most common planning criteria. For a more detailed introduction to territory design, we refer the reader to Kalcsics et al. (2005) and Kalcsics (2015).

2.1.1 Applications

Districting applications can broadly be categorized into political districting and service territory design. The former category deals with the design of electoral districts, whereas the latter category addresses the design of districts for the provision of public or private services.

Political Districting

In political districting, a governmental area is subdivided into districts in which elections are held taking into account several relevant planning criteria. This application has attracted the attention of many researchers since the 1960s (e.g., Bozkaya et al., 2003, 2011; Forman and Yue, 2003; Garfinkel and Nemhauser, 1970; George et al., 1997; Goderbauer, 2016; Hess et al., 1965; Hojati, 1996; Mehrotra et al., 1998; Ricca and Simeone, 2008; Ricca et al., 2008; Vickrey, 1961; Yamada, 2009).

Consider, for example, the districts for the election of the German Bundestag, the national parliament of Germany. Currently, there exist 299 such districts, and in each district, the candidate who receives the majority of the votes is elected to parliament. Since roughly half of the members of the Bundestag are elected in this way, a fair design of the districts is of utmost importance. Therefore, German electoral law defines strict guidelines for them (Bundeswahlgesetz, 2017): It is mandatory to adhere to the boundaries of the federal states, and, if possible, municipality and county boundaries should also be respected. Districts are supposed to be contiguous. The number of districts in each federal state must correspond to its share of the population. The population in each district should not deviate more than 15% from the average population of a district, and districts must be redesigned if this deviation exceeds 25%. As a consequence, changes in the population distribution necessitate the realignment of districts. For the election of the Bundestag in September 2017, 34 of the 299 districts have been redefined, mainly due to changes in the population (Bundeswahlleiter, 2017).

This example illustrates most of the criteria that are typically encountered in political districting problems. One criterion is population equality, i.e., the requirement that the population should approximately be the same in each district to ensure that each vote has the same power. Moreover, contiguity is desired to prevent the deliberate manipulation of district boundaries in favor of one political party, the so-called *gerrymandering*. Especially majority voting systems are vulnerable to this practice. The example also includes the requirement to adhere to existing administrative boundaries such that administrative units are not split into various electoral districts. Not contained in the example is the compactness criterion, which is frequently considered and which aims also at the prevention of gerrymandering. Other criteria that are less frequently encountered are socio-economic homogeneity, representation of ethnic minorities, integrity of communities, the consideration of natural barriers such as mountains and rivers, and similarity with the previous solution (see Bozkaya et al., 2003, 2011; George et al., 1997; Williams, 1995).

Extensive reviews on planning criteria, models and solution methods for political districting can be found in Williams (1995) and Ricca et al. (2013).

Design of Service Territories

A large body of literature is dedicated to the design of service territories. We categorize it further into territories for on-site services in which services are provided at the locations where demand is present, territories for services that are provided at fixed locations, and territories for services which do not fall within the former two subcategories.

On-Site Services Many companies operate a field sales force to provide services at their customers' sites, e.g., to perform selling activities, set up promotional displays, fill shelves, or provide information about the product range. In the USA alone, there are approximately six million full-time salespeople with field responsibility (Zoltners and Sinha, 2005). The design of sales territories has been studied by a large number of researchers (e.g., Drexl and Haase, 1999; Fleischmann and Paraschis, 1988; Hess and Samuels, 1971; López-Pérez and Ríos-Mercado, 2013; Ríos-Mercado and López-Pérez, 2013; Salazar-Aguilar et al., 2011; Shanker et al., 1975; Zoltners and Sinha, 1983, 2005). The associated planning task is to uniquely allocate customers to salespeople. The resulting sales territories serve as areas of responsibility for the salespeople, i.e., a salesperson or sales team is responsible for all customers within a territory. Typically, sales territories are supposed to be geographically compact and contiguous since this helps reduce unproductive travel time. Moreover, they should be balanced with respect to one or multiple attributes, such as expected workload or earning opportunities. Most authors assume that the number of sales territories is given in advance. Regarding the salespersons' locations, some authors assume them to be fixed (e.g., Ríos-Mercado and López-Pérez, 2013), while others consider them as part of the planning process (e.g., Fleischmann and Paraschis, 1988). As opposed to the above mentioned balancing approach, some authors follow a profit maximization approach, in which the profit contribution of a customer is modeled as a function of the time spent on the customer (e.g., Drexl and Haase, 1999).

Another field of application is concerned with the design of work areas for pickup and delivery operations (e.g., Bard and Jarrah, 2009; Carlsson, 2012; Carlsson and Delage, 2013; Galvão et al., 2006; González-Ramírez et al., 2011; Haugland et al., 2007; Jarrah and Bard, 2012; Lei et al., 2012, 2016; Novaes and Graciolli, 1999; Novaes et al., 2000; Ouyang, 2007; Wong and Beasley, 1984). In this application, a given geographical region is partitioned into pickup or delivery districts. Typically, one vehicle is responsible for serving all customers within a district, and all vehicle tours start and end at one or several given depots. Districts are commonly supposed to be compact and contiguous. Similarly to sales districting, it is frequently desired that the workload or the number of customers is roughly equal in all districts. In some cases, however, only an upper bound on the workload in each district is considered (e.g., Haugland et al., 2007). Since workload includes travel time and integrating exact travel time calculations into the solution approaches is computationally prohibitive, many authors rely on travel time estimations (Bard and Jarrah, 2009; Galvão et al., 2006; Haugland et al., 2007; Jarrah and Bard, 2012; Lei et al., 2012, 2016). There is no clear tendency in literature whether the number of districts is given in advance (e.g., Carlsson, 2012; Carlsson and Delage, 2013; Galvão et al., 2006) or is supposed to be minimized (e.g., Bard and Jarrah, 2009; Jarrah and Bard, 2012; Lei et al., 2012, 2016). Since customer demand is usually uncertain at the time when the districts are designed, some authors do not partition the whole region under study, but only a certain proportion of it, into districts in order to leave routing flexibility for day-to-day operations (Schneider et al., 2015; Zhong et al., 2007). A more detailed review of districting approaches for the design of pickup or delivery districts is provided in Chapter 6 of this thesis.

Furthermore, the provision of services to streets is a widely studied field of application. Such services range from the delivery of leaflets or mail (Bodin and Levy, 1991; Butsch et al., 2014), and the collection of solid waste (Hanafi et al., 1999; Mourão et al., 2009), to winter services and road maintenance (Chen et al., 2017; Muyldermans et al., 2002, 2003; Perrier et al., 2008) as well as meter reading (Silva de Assis et al., 2014). In these applications, the streets within the region under study must be partitioned into service districts, each being served by a single person, a vehicle or a fleet of vehicles. Districts are commonly desired to be balanced with respect to service time or total workload including travel time. In most applications, compactness and connectedness of districts is considered. Another criterion that is encountered in some applications is the minimization of deadheading time, i.e., the time for traversing a street without providing service (e.g., Butsch et al., 2014). Also the assignment of districts to depots or disposal sites, whose locations can either be given in advance or must be determined as part of the planning problem, is considered in some cases (Chen et al., 2017; Perrier et al., 2008).

Another application is the design of districts for home-health care services (Benzarti et al., 2013; Blais et al., 2003). Here, a dedicated team of nurses and other medical personnel provides home care to the patients in each district. Relevant criteria comprise compactness, contiguity, workload balance, and conformity with existing administrative boundaries. Patients must be uniquely assigned to districts since this leads to clear responsibilities and the establishment of long-term relations between caregivers and patients. Moreover, mobility plays an important role as caregivers need to travel between the patients in their district, e.g., by means of public transportation.

Finally, there exist applications addressing the design of police command districts or patrol sectors as well as the design of primary response areas for ambulances (Baker et al., 1989; Camacho-Collados et al., 2015; D'Amico et al., 2002). Prevalent criteria in these applications include again workload balance, compactness, and contiguity. As the district design affects emergency services, an important additional criterion is the consideration of response time to calls for service.

Services Provided at Fixed Locations An application where service is provided at fixed locations is school districting (Caro et al., 2004; Ferland and Guénette, 1990; Schoepfle and Church, 1989, 1991). In this application, students must be assigned to existing schools taking into account the capacities of the schools. Desirable criteria comprise compactness in the sense of minimum total distance that students must travel to schools, compliance with a prespecified maximum travel distance, the consideration of geographical obstacles, contiguity, the assignment of all students in a city block to the same school, the number of students to be bused, and racial balance. In a redistricting setting, an upper bound on the proportion of city blocks that may be reallocated can be desirable to ensure a certain degree of similarity with the existing solution.

The design of districts for social facilities, such as hospitals, is another application where service is provided at fixed locations (Minciardi et al., 1981). Each district must be contiguous, contain at least one facility providing service, and its dimension, e.g., in terms of population, must be between given lower and upper bounds. Moreover, the total capacity in each district must be sufficient to meet demand. Beyond that, several additional criteria are deemed relevant, such as the average distance to be traveled by the inhabitants of a district.

Other Services Two applications deal with service territory design in a wider sense. The first one is financial product districting, in which customers are partitioned into districts such that the expected customer-dependent cost price of a financial product is roughly the same for all customers within each district (de Fréminville et al., 2015). Moreover, each district must be contiguous and contain a given minimum number of customers. Geographical compactness is of no importance for this application.

The second application arises in the context of electrical power distribution (Bergey et al., 2003). For the transition from a state-owned, monopolistic electricity service provider to multiple competitive business units, the physical assets of a power grid must be divided into economically viable districts. The districts must be non-overlapping and contiguous.

Furthermore, they should be geographically compact, as this increases profitability, and have roughly the same earning potential to promote competition.

2.1.2 Main Components

According to Kalcsics (2015), territory design problems generally contain the following main components.

Basic Areas

The smallest geographical units considered in territory design problems are called *basic areas* or *basic units*. Depending on the application, they can be represented by points (e.g., geocoded customer locations), lines (e.g., streets) or polygons (e.g., zip code areas). Furthermore, each basic area has one or several quantifiable attributes, which are called *activity measures*. Typical examples for activity measures include service time, sales potential, number of customers, and population size. For the sake of simplicity, we assume in the following explanations only one activity measure.

Districts

A *district* or *territory* is a subset of basic areas. The number of districts can either be predetermined, or it can be part of the problem to determine their number. The activity measure of a district, also called its *size*, is usually calculated as the sum of the activity measures of the assigned basic areas. However, the district size can also contain solution-dependent measures, e.g., travel times, whose exact value cannot be computed by simply summing up the individual activity measures of all basic areas assigned to a district. Moreover, in some applications, a so-called *center*, which typically coincides with one of the basic areas, is associated with each district. Again, centers can be given in advance, or they can be the result of the planning process. Note that we will use the terms “territory” and “district” interchangeably throughout this thesis.

Territory Design Solution

A *territory design solution* consists of a set of districts. Synonyms that are also used in this thesis are *districting solution*, *solution*, and *district design*.

2.1.3 Problem Statement and Common Planning Criteria

Territory design or *districting* can be described as the problem of partitioning a set of basic areas into districts that are compact, contiguous and balanced. Additionally, and only if required by the application, a center must be determined for each district (Kalcsics, 2015).

Based on Kalcsics (2015), the planning criteria that are encountered most commonly in districting problems can briefly be summarized as follows.

Complete and Exclusive Assignment

Complete and exclusive assignment means that each basic area must be assigned to exactly one district, which corresponds to the requirement that a partition of the set of basic areas is sought. In political districting, it is obvious that each eligible voter is allocated to one and only one electoral district. In service districting, this requirement results in clear responsibilities for the persons providing the service and helps establish and foster personal relations with customers.

Compactness

A district is considered *compact* if it is fairly round-shaped and undistorted. In political districting, compactness is required to prevent gerrymandering. In service districting, either customers must travel to the sites providing service or field workers have to travel to the customers; in both cases, the motivation for compactness is to reduce travel time. Although compactness seems to be an intuitive concept, there exists no rigorous definition. Numerous different measures have been proposed to quantify compactness. None of them is comprehensive, and all have some weaknesses (see Chapter 3 in Butsch, 2016, for an extensive review). Compactness measures can broadly be categorized into geometric and distance-based measures. The measures of the former category rely on geometric properties of the districts, such as the area or perimeter, and are mainly used for basic areas represented by polygons. The measures of the latter category are based on the distances between the basic areas of a district and the associated center or on pairwise distances between basic areas. The distances can also be squared or weighted by the activity measures of the basic areas. Distance-based measures are predominantly used when basic areas are represented by points or lines.

Contiguity

As with compactness, there is no rigid mathematical definition of contiguity. If neighborhood information is naturally available, as is the case if basic areas are represented by lines or polygons, or can reasonably be derived for point representations of basic areas, a neighborhood graph can be constructed. A district is considered *contiguous* if it induces a connected subgraph in the neighborhood graph. Otherwise, a different approach for defining compactness is based on the overlap of districts: A district is considered contiguous if the convex hull of its basic areas does not intersect the convex hull of the basic areas of another district (Jarrah and Bard, 2012; Kalcsics et al., 2005). The rationale behind contiguity is very similar to the one for compactness. Depending on the underlying application, contiguity either aims at preventing gerrymandering or at reducing travel time.

Balance

The balance criterion requires districts to have approximately the same size with respect to the activity measure, hence expressing the desire for fairness. In political districting, this criterion is used to ensure that each vote has the same power. In service districting, it is typically incorporated to allocate workload or sales potential evenly to districts. Again, there are different approaches to measure balance. The most common one is to compute the relative deviation of a district's size from the mean district size. A solution is considered perfectly *balanced* if this deviation equals zero for all districts.

District Center

Even though in most cases the determination of district centers is not a planning criterion on its own, *centers* are often used for compactness measurement. Hence, determining centers is in many approaches an integral part of the solution process. If network distances (e.g., distances based on the road network) are used in the compactness measure, the potential centers have to be restricted to the locations included in the precalculated distance matrix, which is usually based only on the locations of the basic areas. Otherwise, the center of gravity is also a valid choice.

Number of Districts

The *number of districts* is frequently predetermined. For example, it can be determined in such a way that the expected workload in a district is below the maximum feasible working time of a field worker and above some minimum value to achieve a reasonable utilization of the field workers. Predetermining this number is particularly appropriate if the activity measure is assumed to be independent of the solution. However, if the activity measure consists largely of solution-dependent components, as is the case, for example, if workload is mainly driven by travel time, it can be more appropriate to consider the number of districts as an outcome of optimization.

2.2 Territory Design for On-Site Services

In this thesis, we focus on territory design applications for on-site services. These applications share the following characteristics: Basic areas correspond to *customers* (e.g., supermarkets) who demand a certain service, or they represent an aggregation of individual customers (e.g., based on their zip codes). The requested service must be provided at the customers' premises by *service providers* (e.g., salespersons), each being responsible for all customers within a single service district. Since the locations of the customers are distributed over a geographical region, the service providers need to travel to their customers in order to perform the service. This means that their workload not only consists of the actual provision of the service, but also of the trips to the customers.

2.2.1 Deriving Additional Planning Requirements

Now, we present various planning requirements arising in the context of on-site services that go beyond the most common criteria introduced in the previous section. These requirements have been identified in a project with our industry partner PTV and are also backed by corresponding findings in scientific literature. Their integration into districting models and solution methods will be addressed in the subsequent chapters.

Assignment Restrictions

There can be requirements which restrict the possibilities of assigning customers to districts. We call these requirements *assignment restrictions*. Some of them are due to interdependencies between customers, whereas others result from customers having special requirements with respect to the service provider who is assigned to them.

Interdependencies Between Customers It can be required to assign a specified subset of customers to the same district (see Caballero-Hernández et al., 2007, who report the requirement that certain pairs of customers must be assigned to the same district). We call this a *joint assignment* requirement. Consider, for example, a pharmacy with multiple branch offices, each requiring service by a salesperson. In this case, it might be desirable to assign all branch offices to the same district such that they are served by the same salesperson.

Also the opposite can be true, i.e., there can be cases where some specified customers have to be assigned to different districts, e.g., due to political or strategic decisions (see López-Pérez and Ríos-Mercado, 2013, who report such a requirement with regard to pairs of customers). We call this a *disjoint assignment* requirement.

Requirements Concerning the Service Providers Blakeley et al. (2003) report planning requirements arising in the context of periodic elevator and escalator maintenance operations. In this problem, service technicians have to carry out regular maintenance work, and each technician is responsible for a dedicated set of buildings. When buildings are assigned to service technicians, one important restriction is that a service provider must have the necessary skill set (e.g., can service elevators, can service escalators, is a hydraulic-elevator specialist, etc.) to perform the corresponding tasks.

More generally speaking, service providers can have different *skills*, and customers can require one or several skills. Thus, a customer can only be assigned to a district if the associated service provider has all required skills. A special case of this requirement arises if the assignment of a customer to a service provider is prescribed, e.g., to maintain the personal relationship between a service provider and a key customer, or prohibited, e.g., if a service provider has been banned by a customer. We call these assignment restrictions *fixed assignment* and *forbidden assignment* requirements, respectively.

Scheduling of Recurring Services

In some applications, customers have to be visited by the service providers on a regular basis. For example, Fleischmann and Paraschis (1988) report a problem encountered at a German manufacturer of consumer goods, whose salespersons regularly visit retailers for purposes of sales promotion and advertising.

If customers have to be visited several times during a planning horizon, the classical districting problem is extended by a temporal dimension. In addition to partitioning the set of customers into districts, *visiting schedules* have to be determined for each salesperson, i.e., visits have to be assigned to the days of the planning horizon subject to customer-specific requirements. For a problem that was brought to our attention by PTV, we have identified the following planning criteria, which have, in the meantime, also been scientifically recognized (see Bender et al., 2016): Visits must be distributed evenly over the weeks of the planning horizon according to customer-specific visiting rhythms. The schedules must adhere to customer-specific weekday patterns, which limit the combinations of weekdays on which a customer can be served. Additionally, customers might be required to be served always on the same weekdays. Moreover, the notion of compactness and balance must be extended to a multi-period setting. Motivated by the desire for short travel times, compactness in a multi-period setting means that all customers to be served by the same service provider on the same day or in the same week should be geographically close to each other. Balance in the context of a multi-period planning horizon means that workload should be approximately the same on each day and in each week in order to avoid time periods with excessive workload and time periods in which a service provider is underutilized.

Assignment of Resources with Different Capacities

Capacity constraints, e.g., with respect to working time or vehicle load, can be encountered in several problems in which on-site services are provided (e.g., Bard and Jarrah, 2009; Jarrah and Bard, 2012; Novaes et al., 2000). An additional planning requirement arises if district capacities are variable and must be determined as part of the problem. Consider, for example, the case where some service providers are full-time employees and others are part-time employees, and, thus, service providers distinguish themselves by their contractual working times. If there is no predefined allocation of employees to districts, designing service districts involves the assignment of an employee type (full-time or part-time) to each district. Obviously, a district associated with a part-time employee must contain less workload than a district intended to be served by a full-time employee. The same reasoning applies if a vehicle is required for the provision of the service, and various vehicle types with different capacities are available to choose from.

More generally, the planning requirement can be described as follows. If several types of *resources* are available which differ in their capacities, a resource type must be assigned to each district and the corresponding capacity limits must be reflected in the size of the districts, e.g., with respect to their expected workload, the total amount of goods to be transported, and so on.

Determination of the Number of Districts

As already mentioned in the previous section, the number of districts is predetermined in many cases. A reasonable number of districts can be determined in advance especially if the activity measure is not solution-dependent. However, in the context of on-site services, travel time can make up a substantial part of the overall workload. It is therefore hardly possible to precisely determine the number of required districts (and associated resources) in advance. Hence, it can be necessary to consider the determination of the number of districts as part of the planning process, as it is done, for example, in Bard and Jarrah (2009), Haugland et al. (2007), and Lei et al. (2016). This, in turn, entails another important requirement, namely the integration of travel time estimates of sufficiently high quality into the solution approach.

Day-to-Day Adjustments

According to Wong (2008), service providers in the parcel shipping industry have pre-assigned service territories, and each service provider is responsible for the pickup and delivery operations in his or her territory. Although it is deemed desirable that service territories remain stable from day to day in order to maintain service continuity, Wong reports that territories may have to be adjusted on a daily basis due to workload fluctuations. The goal of such adjustments is to balance workload and to avoid overloading of individual service providers.

This shows that, in the presence of heavy workload fluctuations, there is a conflict of objectives between service consistency resulting from stable service districts, on the one hand, and working time related objectives, such as workload balance or compliance with the service providers' contractual working times, on the other hand. Hence, it is necessary to take into account these conflicting objectives and to develop an approach for the day-to-day adjustment of service districts that is capable of finding a reasonable trade-off.

2.2.2 Addressing Additional Planning Criteria in This Thesis

We address the planning requirements of Section 2.2.1 as follows. In Chapter 3, we take a well-known mathematical programming model from literature and show how this model can be enhanced by the presented assignment restrictions. The scheduling of recurring services is addressed in Chapters 4 and 5. In these chapters, we formally define the problem of scheduling customer visits and propose a heuristic solution approach as well as an exact branch-and-price algorithm. The assignment of resources with different capacities and the determination of the number of districts are simultaneously tackled in Chapter 6. In the context of parcel shipping, we propose three models for designing tactical delivery districts. Moreover, we present a model that is capable of adjusting an existing tactical district design to the concrete workload on a certain day.

3

Mathematical Modeling and Solution Principles

In the previous chapter, we have informally introduced the main ingredients of districting problems. Now we show how districting problems can formally be modeled by means of a mathematical program. For this purpose, we first present the model of Hess et al. (1965), which was the first mathematical program proposed for districting, and we describe how the model can be extended by exact contiguity constraints. Then, we show how each of the assignment restrictions introduced in Chapter 2 can be integrated into the model. Individual model components can be rediscovered in the application-specific models that will be introduced in Chapters 4–6. Furthermore, we explain solution principles that will play an important role in the development of application-specific solution methods in the subsequent chapters.

3.1 A Basic Model

In the following, we present a basic model to illustrate how districting problems can be formulated mathematically. This model is based on the well-known model of Hess et al. (1965), which takes into account the compactness and balance criteria, but not the contiguity criterion. Therefore, we explain two existing approaches to incorporate contiguity into the model.

3.1.1 The Model of Hess et al.

Denote by $B = \{1, ..., |B|\}$ the set of basic areas, which we call customers in the following. Further, denote by $w_b \in \mathbb{R}^+$ the activity measure associated with customer $b \in B$. With $n \in \mathbb{N}^+$ being the number of districts to be planned, the average activity measure per district can be computed as $\mu = \frac{1}{n} \sum_{b \in B} w_b$. Parameters $s_{min} \leqslant 100$ and $s_{max} \geqslant 100$ state the minimum and the maximum size of a district defined as percentage of μ. $c_{bi} \in \mathbb{R}^+$ denotes the distance between customer $b \in B$ and customer $i \in B$. Moreover, define the following binary decision variables:

$$x_{bi} = \begin{cases} 1 & \text{if customer } b \in B \text{ is assigned to the district represented by customer } i \in B \\ 0 & \text{otherwise} \end{cases}$$

Customer $i \in B$ is selected as district center if and only if $x_{ii} = 1$.

Then, the model of Hess et al. (1965) can be stated as the following integer programming (IP) model, which we denote by $\text{LOCALLOC}_{\text{IP}}$:

$$(\text{LOCALLOC}_{\text{IP}}) \qquad \sum_{b \in B} \sum_{i \in B} w_b c_{bi}^2 x_{bi} \to \min \tag{3.1}$$

$$\text{s.t.} \qquad \sum_{i \in B} x_{bi} = 1 \qquad b \in B \tag{3.2}$$

$$\sum_{i \in B} x_{ii} = n \tag{3.3}$$

$$\sum_{b \in B} w_b x_{bi} \geqslant \frac{s_{min}}{100} \mu x_{ii} \qquad i \in B \tag{3.4}$$

$$\sum_{b \in B} w_b x_{bi} \leqslant \frac{s_{max}}{100} \mu x_{ii} \qquad i \in B \tag{3.5}$$

$$x_{bi} \in \{0, 1\} \qquad b, i \in B \tag{3.6}$$

The Objective Function (3.1) aims at optimizing compactness: It minimizes the sum of the squared distances between customers and district centers, weighted by the customers' activity measure. Constraints (3.2) in combination with the integrality of the decision variables defined in Constraints (3.6) make sure that each customer is assigned to exactly one district. Constraints (3.3) guarantee that exactly n district centers are selected. The balance criterion is reflected by Constraints (3.4) and (3.5), which limit the size of each district to the feasible range defined by the allowable deviation from the average district size μ. These constraints also guarantee that customers can only be assigned to districts represented by a customer that is selected as district center. Observe that contiguity is not enforced in this model.

3.1.2 Modeling Contiguity

In the following, we present the approaches of Drexl and Haase (1999) and Shirabe (2009) for the exact formulation of the contiguity criterion as part of a mathematical programming model. Both approaches rely on the definition of an adjacency relation. Let this relation be given by $A_b \subset B$, which represents all customers that are adjacent to customer $b \in B$.

The Approach of Drexl and Haase

Drexl and Haase (1999) enforce contiguity with the following constraints, which are similar to the subtour elimination constraints used in vehicle routing problem formulations:

$$\sum_{b \in \bigcup\limits_{b' \in S} (A_{b'} \setminus S)} x_{bi} - \sum_{b \in S} x_{bi} \geqslant 1 - |S| \qquad i \in B, S \subseteq B \setminus (\{i\} \cup A_i), S \neq \emptyset \tag{3.7}$$

Consider a district center represented by customer $i \in B$ and a non-empty subset of customers $S \subseteq B$ which contains neither the district center i nor any customer adjacent to i. Clearly, if only the customers in S were assigned to district center i, they would be geographically separated from i and, thus, the district would not be contiguous. Hence, if all customers in S are assigned to i, Constraints (3.7) enforce that at least one customer which is not contained in S and which is adjacent to a customer in S is also assigned to i. Considering all the subsets thus ensures that the districts are contiguous. Since the number of subsets and, consequently, also the number of constraints grow exponentially in the number of customers, a cut generation approach is typically applied in order to add only the violated constraints to the model (e.g., Ríos-Mercado and López-Pérez, 2013; Salazar-Aguilar et al., 2011).

The Approach of Shirabe

The approach of Shirabe (2009) is based on network flows. Suppose that $i \in B$ is a district center and consider a network with nodes representing customers and arcs between nodes that correspond to adjacent customers. The idea is that the district center i is the sink in the network and each customer that is assigned to i has one unit of supply which is sent to this sink. This is achieved by adding the following constraints and variables:

$$\sum_{b' \in A_b} f^i_{bb'} - \sum_{b' \in A_b} f^i_{b'b} = x_{bi} \qquad b \in B \setminus \{i\} \tag{3.8}$$

$$\sum_{b' \in A_b} f^i_{b'b} \leqslant (|B| - 2) x_{bi} \qquad b \in B \setminus \{i\} \tag{3.9}$$

$$\sum_{b' \in A_i} f^i_{b'i} \leqslant (|B|-1)x_{ii} \tag{3.10}$$

$$f^i_{bb'} \geqslant 0 \qquad b \in B, b' \in A_b \tag{3.11}$$

The continuous variables $f^i_{bb'}$ defined in Constraints (3.11) state the quantity that flows from customer $b \in B$ to customer $b' \in A_b$ within the network associated with district center i. Constraints (3.8) make sure that each customer that is assigned to i, except for i itself, sends one unit. Moreover, Constraints (3.9) and (3.10) ensure that the flow to customers not assigned to i equals zero. If, in contrast to the assumption above, i is not selected as a district center, all flows in the associated network are forced to zero. To enforce contiguity for all districts, (3.8)–(3.11) must be added for each $i \in B$.

3.2 Integrating Assignment Restrictions

Now we formulate model components for the integration of assignment restrictions into model LOCALLOC$_{IP}$. While joint and disjoint requirements have already been considered in the districting literature (Caballero-Hernández et al., 2007; López-Pérez and Ríos-Mercado, 2013; Ríos-Mercado and López-Pérez, 2013), we are not aware of any districting paper that addresses different skills of service providers or fixed/forbidden assignments of customers to service providers in a setting where the allocation of service providers to districts is not given in advance. In that case, the restrictions cannot be incorporated by simple variable fixations.

The model components that we introduce in the following are grouped analogously to the requirements in Section 2.2.1. We first present components resulting from interdependencies between customers. Then, we propose components to integrate customer requirements with respect to the service providers.

3.2.1 Interdependencies Between Customers

Joint and disjoint assignment requirements can easily be modeled. Let $\mathcal{J} = \{J_1, ..., J_{|\mathcal{J}|}\}$ with $J_k \subset B$, $1 \leqslant k \leqslant |\mathcal{J}|$, denote the set that contains all subsets of customers that must be assigned to the same district. Further, let $\theta(S) \in B$ represent the customer of set $S \subseteq B$ with the smallest index. Then, the joint assignment requirement can be integrated by the following additional constraints:

$$x_{bi} = x_{\theta(J_k),i} \qquad J_k \in \mathcal{J}, b \in J_k, b \neq \theta(J_k), i \in B \tag{3.12}$$

For each subset of customers $J_k \in \mathcal{J}$ that must be jointly assigned, these constraints guarantee that all contained customers are assigned to the same district as the customer of J_k with the smallest index. Obviously, instead of using these constraints, joint assignment can also be realized by aggregating all customers that must be assigned to the same district in a preprocessing step.

Let $\mathcal{D} = \{D_1, ..., D_{|\mathcal{D}|}\}$ with $D_l \subset B$, $1 \leqslant l \leqslant |\mathcal{D}|$, denote the set of all subsets of customers that must be assigned to different districts. Analogously to the constraints used in Ríos-Mercado and López-Pérez (2013) for the special case $|D_l| = 2$, disjoint assignment is then enforced by the following additional constraints:

$$\sum_{b \in D_l} x_{bi} \leqslant 1 \qquad D_l \in \mathcal{D}, i \in B \tag{3.13}$$

The constraints make sure that at most one customer of each subset of customers $D_l \in \mathcal{D}$ is assigned to each district.

3.2.2 Requirements Concerning the Service Providers

For the integration of requirements with respect to the service providers, we introduce the following notation. Let the set of service providers be denoted by $\mathcal{P}$. Further, let the set of service providers that are eligible for customer $b \in B$ with respect to the required skills be defined as $\mathcal{P}(b) = \{p \in \mathcal{P} \mid$ service provider p has all skills required to serve customer $b\}$. Fixed assignments of customers to service providers are represented by the set $\mathcal{FIX} = \{(b, p) \in B \times \mathcal{P} \mid$ customer b must be assigned to service provider $p\}$. Analogously, forbidden assignments are defined by the set $\mathcal{FORB} = \{(b, p) \in B \times \mathcal{P} \mid$ customer b must not be assigned to service provider $p\}$.

Note that fixed and forbidden assignments could also be modeled via skills. A fixed assignment of a customer to a certain service provider can be achieved by introducing an exclusive skill, i.e., a skill that only the required service provider has, and by letting the customer demand this skill. To model a forbidden assignment using skills, all service providers except the forbidden one must receive an additional skill which is demanded by the customer. However, we refrain in the following from modeling these requirements via skills since skills and fixed/forbidden assignments are motivated by different practical requirements.

In the following, we distinguish two cases: (a) the determination of district centers and the assignment of a service provider to each district is part of the planning process, (b) district centers and associated service providers are predetermined. Let us first consider case (a). Since, in this case, service providers must be assigned to districts, we introduce the following additional decision variables:

$$y_{pi} = \begin{cases} 1 & \text{if service provider } p \in \mathcal{P} \text{ is assigned to the district represented by customer } \\ & i \in B \\ 0 & \text{otherwise} \end{cases}$$

The desired assignment restrictions can be introduced to the model through the following additional constraints:

$$\sum_{p \in \mathcal{P}} y_{pi} = x_{ii} \qquad i \in B \tag{3.14}$$

$$\sum_{i \in B} y_{pi} = 1 \qquad p \in \mathcal{P} \tag{3.15}$$

$$x_{bi} \leqslant \sum_{p \in \mathcal{P}(b)} y_{pi} \qquad b, i \in B \tag{3.16}$$

$$x_{bi} = y_{pi} \qquad (b,p) \in \mathcal{FIX}, i \in B \tag{3.17}$$

$$x_{bi} \leqslant 1 - y_{pi} \qquad (b,p) \in \mathcal{FORB}, i \in B \tag{3.18}$$

$$y_{pi} \in \{0,1\} \qquad p \in \mathcal{P}, i \in B \tag{3.19}$$

Constraints (3.14) and (3.15) in conjunction with the binary nature of the decision variables defined by Constraints (3.19) ensure that exactly one service provider is assigned to each district and each service provider is allocated exactly once. Due to Constraints (3.16), customers can only be assigned to a district if the associated service provider has the required skills. Fixed assignments are enforced through Constraints (3.17). Constraints (3.18) make sure that customers can only be assigned to a district if the district is not associated with a forbidden service provider.

For case (b), let the customers that represent district centers be denoted by $I \subset B$. Moreover, let $\delta(p)$ denote the district center $i \in I$ that service provider $p \in \mathcal{P}$ is associated with. Since model LOCALLOC_{IP} is simplified if the locations of the district centers are known in advance, we state in the following the entire model, not just the model components resulting from the assignment restrictions:

$$\sum_{b \in B} \sum_{i \in I} w_b c_{bi}^2 x_{bi} \to \min \tag{3.20}$$

$$\text{s.t.} \qquad \sum_{i \in I} x_{bi} = 1 \qquad b \in B \tag{3.21}$$

$$\sum_{b \in B} w_b x_{bi} \geqslant \frac{s_{min}}{100} \mu \qquad i \in I \tag{3.22}$$

$$\sum_{b \in B} w_b x_{bi} \leqslant \frac{s_{max}}{100} \mu \qquad i \in I \qquad (3.23)$$

$$x_{b,\delta(p)} = 0 \qquad b \in B, p \in \mathcal{P} \setminus \mathcal{P}(b) \qquad (3.24)$$

$$x_{b,\delta(p)} = 1 \qquad (b,p) \in \mathcal{FIX} \qquad (3.25)$$

$$x_{b,\delta(p)} = 0 \qquad (b,p) \in \mathcal{FORB} \qquad (3.26)$$

$$x_{bi} \in \{0,1\} \qquad b \in B, i \in I \qquad (3.27)$$

The model is quite similar to the original model (3.1)–(3.6), but it differs in the following aspects. Since the district centers are predetermined, Constraints (3.3) defining that exactly n district centers are selected are not required any more. Moreover, index i of variables x_{bi} is restricted to the set I. The assignment restrictions translate into the variable fixations defined by Constraints (3.24)–(3.26). Constraints (3.24) prohibit the assignment of a customer to a service provider without sufficient skills. Fixed and forbidden assignments are incorporated through Constraints (3.25) and (3.26) by forcing the corresponding assignment variables to one and zero, respectively.

3.3 Solution Principles

Even if modern computer hardware and state-of-the-art mixed integer programming (MIP) solvers are used, model LOCALLOC_{IP} is computationally intractable even for fairly small problem instances. For example, we tried to solve four problem instances with 500 customers and five districts using the MIP solver Gurobi 7.0.2[1] on a Windows 8 machine with 12 GB of RAM and an Intel Core i7-5600U CPU running at a clock rate of 2.6 GHz. With a time limit of one hour, none of the four instances could be solved to proven optimality, and the average optimality gap was 51.1%. Since districting problems encountered in practice often consist of several thousand customers or necessitate the consideration of various additional planning requirements, simply plugging the model into a MIP solver is obviously not a feasible solution approach. Instead, either heuristics must be used or specially-tailored exact methods have to be developed. In the following, we introduce some principles which form the basis of our solution approaches for the applications presented in Chapters 4–6. For a general overview of solution approaches for districting problems, we refer the reader to the reviews provided by Kalcsics et al. (2005) and Ricca et al. (2013).

[1] http://www.gurobi.com

3.3.1 Location-Allocation Heuristic

Model $\text{LOCALLOC}_{\text{IP}}$ includes two types of decisions, namely the location of district centers and the allocation of customers to centers. Since it is computationally intractable to tackle both types of decisions simultaneously, Hess et al. (1965) decompose the problem into the corresponding subproblems, which are termed location and allocation subproblem, and solve the subproblems alternately.

The idea of Hess et al. (1965) is summarized in Algorithm 3.1. In the first iteration, the location subproblem is solved by guessing a set of initial district centers $I \subset B$ with $|I| = n$. In all subsequent iterations, it is solved based on the solution of the previous iteration, e.g., by selecting for each district the customer that optimizes the compactness measure as the new center. Given the set of fixed district centers, the allocation subproblem is solved using the linear programming (LP) model which results from model $\text{LOCALLOC}_{\text{IP}}$ by setting the minimum and maximum district size $s_{min} = s_{max} = 100$ and relaxing the integrality of the decision variables. We denote this model by ALLOC_{LP}.

$$(\text{ALLOC}_{\text{LP}}) \qquad \sum_{b \in B} \sum_{i \in I} w_b c_{bi}^2 x_{bi} \to \min \tag{3.28}$$

$$\text{s.t.} \qquad \sum_{i \in I} x_{bi} = 1 \qquad b \in B \tag{3.29}$$

$$\sum_{b \in B} w_b x_{bi} = \mu \qquad i \in I \tag{3.30}$$

$$x_{bi} \geqslant 0 \qquad b \in B, i \in I \tag{3.31}$$

Due to Constraints (3.30), the resulting districts have exactly the same size, but since the decision variables are continuous, solutions may contain split assignments, i.e., customers that are allocated to more than one district center. Hence, in the next step these split assignments are resolved by allocating each customer uniquely to a district center. If the solution process has converged, the algorithm terminates; otherwise, another iteration of the algorithm is performed. If more than one set of initial district centers is available, the approach is easily extended to a multi-start heuristic. Note that we assume in Algorithm 3.1 that only customers are eligible as district centers to allow the use of precalculated network distances.

Hess et al. (1965) resorted to the LP model ALLOC_{LP} for allocating customers to given district centers because it was impossible at that time to solve the allocation subproblem as an IP model. Nowadays, advances in the performance of MIP solvers and the increase in computational power have made it possible to solve the IP model for the allocation of customers to predetermined district centers in short computing times even for problem instances with

Algorithm 3.1 Location-allocation heuristic of Hess et al. (1965)

Input: Basic areas B, number of districts n
Output: A feasible districting solution
1: Initialize iteration counter $t = 1$
2: Guess initial district centers $I^t \subset B$, $|I^t| = n$
3: Solve model ALLOC_{LP} with current district centers I^t
4: Resolve split assignments to obtain feasible solution $\mathcal{S}^t$
5: **if** $(t > 1)$ and $(\mathcal{S}^t = \mathcal{S}^{t-1})$ **then**
6:     **return** solution $\mathcal{S}^t$
7: **else**
8:     Set $t = t + 1$
9:     Select new district centers $I^t \subset B$ based on solution $\mathcal{S}^{t-1}$
10:     Go to line 3
11: **end if**

several thousand customers. For example, setting the minimum district size $s_{min} = 90$ and the maximum district size $s_{max} = 110$ we were able to solve four problem instances, each consisting of 5,000 customers and 50 predetermined district centers, to proven optimality in less than six seconds on average (again using Gurobi 7.0.2 on a Windows 8 machine with 12 GB of RAM and an Intel Core i7-5600U CPU at 2.6 GHz). Note that if an IP is used to solve the allocation subproblem, it is, in general, not possible to find a feasible solution for $s_{min} = s_{max} = 100$ due to the discrete structure of the problem, and, hence, appropriate values for s_{min} and s_{max} must be selected.

3.3.2 Column Generation

Column generation is an iterative method for solving large-scale linear programs. An extensive introduction to this topic is provided, e.g., by Desrosiers and Lübbecke (2005). Column generation is particularly suited for linear programs consisting of a huge number of variables or columns as it works only with a subset of these columns and adds more columns only if needed. It exploits the fact that only a subset of columns needs to be considered explicitly to find a provably optimal solution.

Consider the following linear program, which we refer to as the *master problem* (MP):

$$(\text{MP}) \qquad \sum_{j \in N} c_j x_j \to \min \tag{3.32}$$

$$\text{s.t.} \qquad \sum_{j \in N} a_{ij} x_j \geqslant b_i \qquad i \in M \tag{3.33}$$

$$x_j \geqslant 0 \qquad j \in N \tag{3.34}$$

Suppose that the number of columns $|N|$ in the master problem is prohibitively large, meaning that it is not possible to explicitly consider all columns. The basic idea of column generation is to work with a *restricted master problem* (RMP), which differs from the master problem solely in that it contains only a small subset $N' \subseteq N$ of columns, and to add new columns to the RMP only if they might improve the objective value. The initial set of columns N' can be obtained, for example, from a heuristic solution. If no feasible solution is available, another way of initializing the column generation process is to introduce artificial columns, i.e., columns with very high objective coefficients which ensure that the RMP is feasible.

Solving the RMP with the current set of columns N' yields optimal primal and dual solutions. The task is now to find out if there exists a column currently not contained in the RMP which could improve the objective value. In other words, we search for a column with negative reduced cost. Let the optimal dual multipliers associated with Constraints (3.33) be denoted by π_i, $i \in M$. Then, the reduced cost $\bar{c}_j$ of column $j \in N$ can be computed as

$$\bar{c}_j = c_j - \sum_{i \in M} a_{ij}\pi_i. \tag{3.35}$$

The problem of checking whether a column with negative reduced cost exists is termed the *pricing problem* (PP) and can be stated as follows:

$$(\mathrm{PP}) \qquad \bar{c}^* := \min \left\{ c_j - \sum_{i \in M} a_{ij}\pi_i \,\middle|\, j \in N \right\} \tag{3.36}$$

If $\bar{c}^* < 0$, it means that a column with negative reduced cost exists. This column is then added to the RMP, and another iteration of column generation is performed, i.e., the RMP is re-optimized with the updated set of columns and the resulting pricing problem is solved. If $\bar{c}^* \geqslant 0$, there are no more columns that might improve the objective value, and, thus, the current primal solution to the RMP is an optimal solution to the master problem. The solution process is summarized in Algorithm 3.2.

Note that it is not necessary to solve the pricing problem to optimality in each iteration of the column generation process. It is also a valid choice to identify promising columns by means of a heuristic and to switch to an exact method only if the heuristic does not find a solution with negative reduced cost. This way, a significant speedup might be achieved, while overall optimality is still guaranteed. Moreover, generating many columns with negative reduced cost in a single iteration may help reduce the number of iterations and, hence, accelerate the solution process.

Algorithm 3.2 Column generation

Input: Master problem MP
Output: An optimal solution x to MP or the finding that MP is infeasible if x contains artificial columns

1: Determine initial set of columns N'
2: Solve RMP with columns N' to obtain optimal primal and dual solutions x and π
3: Solve $PP(\pi)$ to obtain minimum reduced cost $\bar{c}^*$ and associated column j
4: **if** $(\bar{c}^* < 0)$ **then**
5:     Set $N' = N' \cup \{j\}$
6:     Go to line 2
7: **else**
8:     **return** x
9: **end if**

3.3.3 Branch-and-Bound

Branch-and-bound is a technique, in which a problem is divided into smaller subproblems that are easier to solve, and the information obtained from solving them is used to determine an optimal solution to the original problem. A detailed introduction can be found, e.g., in Wolsey (1998, pp. 91–111) and in Nemhauser and Wolsey (1999, pp. 354–367).

Branch-and-bound is a common way to solve mixed integer programs. In the following, we consider the case of a mixed integer program with minimization objective. Moreover, we focus on LP-based branch-and-bound, i.e., branch-and-bound which uses LP relaxations. The *LP relaxation* of a MIP is a simplification of the problem in which integrality constraints on the variables are removed.

Typically, a search tree is used to represent a branch-and-bound approach. In the root node of the tree, the LP relaxation of the original problem is solved. If its solution violates an integrality constraint of the original problem, the problem is divided into two or more smaller subproblems, each corresponding to a node in the search tree. This division is termed *branching*. Branching can be done, for example, by choosing an integer variable which assumes a fractional value in the solution of the LP relaxation and by creating two subproblems as follows: Suppose the selected variable is x_j and its value in the LP relaxation is $\bar{x}_j$. Then, the feasible region can be split into two by adding the constraint $x_j \leqslant \lfloor \bar{x}_j \rfloor$ to one subproblem, and the constraint $x_j \geqslant \lfloor \bar{x}_j \rfloor + 1$ to the other subproblem. Repeatedly dividing (sub)problems into smaller subproblems yields the complete search tree.

Since in many problems the huge number of feasible solutions prohibits their explicit enumeration, *bounds* are used to prune parts of the search tree, i.e., to identify nodes of the tree which require no further investigation. An upper bound on the optimal objective value of the original problem is provided by any feasible solution to the original problem. Lower bounds on the objective values of nodes can be obtained by LP relaxations. Based on Wolsey (1998, pp. 94–98), a node of the search tree can be pruned in any of the following cases:

- Pruning by optimality: The subproblem associated with the node has been solved to optimality, which is the case if the solution of the LP relaxation is a feasible solution to the original problem (i.e., all integrality constraints of the original problem are satisfied).

- Pruning by bound: The lower bound for the subproblem associated with the node is not better than that of a known feasible solution to the original problem, and, hence, the subtree does not contain an improved solution.

- Pruning by infeasibility: The LP relaxation of the node is infeasible, and, hence, the subtree does not contain a feasible solution to the original problem.

In Algorithm 3.3 (adapted from Nemhauser and Wolsey, 1999, p. 355) we illustrate the steps of a general LP-based branch-and-bound algorithm. Lines 8, 10, and 14 correspond to pruning by infeasibility, bound, and optimality, respectively. Observe that in line 16 the lower bound of each node is initialized with the optimal objective value obtained for the LP relaxation in its parent node. Based on these initial lower bounds, additional pruning by bound occurs in line 13 after a new best solution to the original problem is found.

Two important questions arising in the design of a branch-and-bound algorithm are the following: Which node that needs further investigation should be explored next? Which variable should be branched on? Regarding the first question, possible strategies include *depth-first search* and *best-first search* (Wolsey, 1998, pp. 99–101). Motivated by the fact that a feasible solution to the original problem is needed to prune the search tree significantly, depth-first search aims at quickly descending in the tree in the hope that a good upper bound is obtained. Best-first search, on the other hand, selects among all unexplored nodes the one with the best lower bound, and is geared towards minimizing the number of explored nodes. Regarding the second question, a variety of rules exist, for example, *most infeasible branching*, which chooses the variable with fractional part closest to 0.5, *largest split branching*, which selects the variable whose value is closest to an integer number, and *pseudocost branching*, which picks the variable that is expected to lead to the largest change in the objective value. For a study of rules for node and variable selection we refer the reader to Achterberg et al. (2005) and Linderoth and Savelsbergh (1999).

Adding cutting planes to the LP relaxations of the nodes yields a common extension of branch-and-bound, which is termed *branch-and-cut*. The use of cutting planes aims at tight-

Algorithm 3.3 LP-based branch-and-bound

Input: A MIP with feasible region X_{MIP}
Output: An optimal solution x^* to the MIP and its objective value z^* or the finding that the MIP is infeasible (indicated by $z^* = +\infty$)
1: Initialize the problem set $P = \{P^0\}$ with the MIP, set the lower bound for P^0 to $\underline{z}^0 = -\infty$, and initialize $z^* = +\infty$
2: **if** $(P = \emptyset)$ **then**
3:     **return** z^* and an optimal solution x^* if one exists
4: **end if**
5: Select and remove a problem P^k from P
6: Solve the LP relaxation of P^k; denote its feasible region by X^k_{LP}, its optimal solution, if one exists, by x^k_{LP} and its objective value by z^k_{LP}
7: **if** $(X^k_{LP} = \emptyset)$ **then**
8:     Go to line 2
9: **else if** $(z^k_{LP} \geqslant z^*)$ **then**
10:     Go to line 2
11: **else if** $(x^k_{LP} \in X_{MIP})$ **then**
12:     Set $x^* = x^k_{LP}$ and $z^* = z^k_{LP}$
13:     Remove from P all problems with $\underline{z}^k \geqslant z^*$
14:     Go to line 2
15: **else**
16:     Divide P^k into subproblems $\{P^{kl}\}^m_{l=1}$ with $\underline{z}^{kl} = z^k_{LP}$, $l = 1, ..., m$, and add them to P
17:     Go to line 5
18: **end if**

ening the LP relaxations such that better lower bounds are obtained and fewer nodes must be explored.

3.3.4 Branch-and-Price

A (mixed) integer program may suffer from a weak LP relaxation. However, the LP relaxation can, in many cases, be tightened by a reformulation consisting of a huge number of variables, as for example Barnhart et al. (1998) note. According to the authors, column generation can be combined with branch-and-bound to solve such a reformulation, and they refer to this method as *branch-and-price*. More precisely, a branch-and-price algorithm uses column generation to solve the LP relaxations of the (reformulated) problems associated with the nodes of a branch-and-bound search tree. Branching decisions in each node are taken into account in the RMP, e.g., by removing all columns that violate a branching decision, and also in the corresponding pricing problem to make sure that no forbidden columns are generated. Combining branch-and-price with the use of cutting planes yields a so-called *branch-price-and-cut* algorithm.

4

The Multi-Period Service Territory Design Problem – An Introduction, a Model and a Heuristic Approach

In this chapter, we consider service territory design applications in which a field service workforce is responsible for providing recurring services at their customers' sites. We introduce the associated planning problem, which consists of two subproblems: In the partitioning subproblem, customers must be grouped into service territories. In the scheduling subproblem, customer visits must be scheduled throughout the multi-period planning horizon. The emphasis of this chapter is put on the scheduling subproblem. We propose a mixed integer programming model for this subproblem, present a location-allocation heuristic, and perform extensive experiments on real-world instances.

This chapter is based on the following article:

Bender, M., Meyer, A., Kalcsics, J., Nickel, S. (2016). The multi-period service territory design problem – An introduction, a model and a heuristic approach. *Transportation Research Part E: Logistics and Transportation Review*, 96:135–157.

4.1 Introduction

Many companies employ a field service workforce for providing recurring services at their customers' sites. For example, manufacturers and wholesalers of consumer goods typically operate a sales force that regularly visits their customers to promote sales or to supply product range information (see, e.g., Fleischmann and Paraschis, 1988; Polacek et al., 2007). Also, some engineering companies employ field service technicians to carry out regular technical maintenance at their customers' sites (see, e.g., Blakeley et al., 2003). The frequency and duration of the visits depend on customer-specific factors, e.g., the customer's sales volume or the tasks to be performed at the customer. To increase customer satisfaction, two aspects of service consistency play an important role in these applications: personal and temporal consistency. The former means that always the same field worker is responsible for a particular customer, which is desirable as it helps establish and foster long-term personal relationships with customers (see, e.g., Kalcsics et al., 2005; López-Pérez and Ríos-Mercado, 2013; Zoltners and Sinha, 2005). The latter expresses the expectation of customers to be visited on a regular basis (see, e.g., Groër et al., 2009, for a similar consistency requirement arising in the small package shipping industry). Regularity means, on the one hand, that the visits should be equally distributed over the weeks of the planning horizon according to customer-specific visiting rhythms. On the other hand, regularity refers to the weekdays on which visits take place as customers might prefer to be served always on the same weekdays.

Typically, the following three planning tasks arise in these applications. (1) The customer base must be partitioned into service territories with one field worker being responsible for each territory. This partition is usually maintained over a long period of time to promote the development of personal relationships between field workers and customers. (2) On a tactical level, the visit schedules have to be created, which means that the visiting days for each customer must be determined. The planning horizon for this task is typically between 3 and 12 months. (3) On an operational level, the detailed planning must be performed, which includes the planning of the daily routes and, when necessary, the rescheduling of visits. It is important to note that short-term customer requests and unexpected events must be taken into account in this step. According to estimates of our project partner, about 20% of the customer visits need to be rescheduled to another day in the short term. Therefore, both the route planning and the rescheduling are done by the field worker in the daily business. Ideally, planning tasks (1) to (3) would be tackled by a single, integrated approach, but the size of realistic problem instances (sometimes with ten thousand or more customers) prohibits an integrated approach. Moreover, integrating the calculation of the daily routes and the visit schedules is only of little use due to the potential necessity to reschedule customer visits in the daily business.

The above problem was brought to our attention by our project partner PTV Group (PTV), a commercial provider of districting and clustering software headquartered in Karlsruhe, Germany. In our joint project, we tackled the partitioning task (1) and the scheduling task (2); we omitted the routing and rescheduling task (3) as this task can only be solved reasonably in the short term when all operational details are known.

One of PTV's products is the xCluster Server (PTV, 2014), which solves the optimization problem resulting from the scheduling task (2). When the planning algorithm for the xCluster Server was initially designed several years ago, the technological possibilities were limited, in particular with regard to the availability of high-performance mixed integer programming (MIP) solvers and computational power in general, which lead PTV to develop a simple local search procedure. The goal of the cooperation with PTV is the development of a new solution approach that takes advantage of recently available technologies. Since PTV has many different customers, it is important that the new solution approach covers a wide range of real-world requirements. Additionally, it must be easily adaptable to further planning requirements.

The main contributions of this chapter are the following:

- We introduce a new problem, which we call the multi-period service territory design problem (MPSTDP). Despite its high practical relevance, it has not been studied in the literature before. To the best of our knowledge, we are the first to elaborate the problem from a scientific point of view.
- We formally define the scheduling subproblem, i.e., the subproblem corresponding to planning task (2), as a mixed integer linear programming model.
- We propose a heuristic solution approach for the scheduling subproblem. The approach is capable of considering the relevant planning requirements of PTV's customers. It involves the repeated solution of an integer programming model, which can easily be extended by additional planning requirements.
- We perform extensive computational experiments on real-world instances and on instances that were derived from real-world data by varying the values of some parameters. The results show that the new approach produces high-quality solutions and outperforms the existing solution method of PTV.

The remainder of this chapter is organized as follows. In Section 4.2, we give a detailed description of the problem under study. In Section 4.3, we review related problems and point out the differences to our problem. In the subsequent section, we introduce a mathematical model for the subproblem that corresponds to the scheduling task (2). In Section

4.5, we propose a heuristic approach based on a location-allocation scheme. To evaluate our approach, we introduce appropriate evaluation measures in Section 4.6. In Section 4.7, we report the results of extensive computational experiments on real-world data and benchmark our approach against PTV's xCluster Server (PTV, 2014). Finally, we provide some concluding remarks in Section 4.8.

4.2 Problem Description

In this section, we describe the MPSTDP and introduce the notation for the scheduling subproblem, which is the major focus of this chapter.

There is a given set of *customers* (e.g., supermarkets), represented by index set $B = \{1, ..., |B|\}$, which demand recurring on-site services. The services must be carried out by a given set of field workers, which we call *service providers*. Corresponding to planning tasks (1) and (2), the MPSTDP consists of the following two subproblems.

Partitioning subproblem (MPSTDP-P): This subproblem corresponds to the well-known territory design or districting problem (see Kalcsics, 2015, for an overview of typical planning criteria). The set of customers must be partitioned into service territories with exactly one service provider being responsible for each service territory. As the service providers have to travel within their territories, geographically compact and connected territories are desired because they lead to short travel times for the service providers. Furthermore, for reasons of fairness, all service territories should have approximately the same workload.

Scheduling subproblem (MPSTDP-S): In this subproblem, a valid visit schedule must be determined for each service territory, i.e., customer visits must be assigned to the weeks and days of the planning horizon subject to customer-specific visiting requirements. The planning horizon comprises $|W|$ weeks and m days per week, resulting in $m|W|$ days in total. Weeks and days are indexed by $w \in W = \{1, ..., |W|\}$ and $d \in D = \{1, ..., |D|\}$, respectively. The customer-specific visiting requirements restrict the temporal distribution of customer visits at two levels.

At the level of weeks, the visits of each customer must be periodically recurring according to a customer-specific *week rhythm* $r_b \in \mathbb{N}^+$, $b \in B$, meaning that each customer $b \in B$ must be visited every r_b weeks. We call a week in which a customer is visited by a service provider a *visiting week* of the customer. As the first visit of each customer $b \in B$ must be in the first r_b weeks of the planning horizon, a customer's week rhythm can be translated into r_b valid combinations of visiting weeks P_b, which we call *week patterns*. If, for example, a customer's week rhythm is $r_b = 2$ and the planning horizon consists of $|W| = 6$ weeks, P_b contains the week patterns {1, 3, 5} and {2, 4, 6}, i.e., the customer must be visited either in weeks one, three and five or in weeks two, four and six.

At the level of days, there are restrictions on the number of visits per visiting week and on the weekdays on which customers may be visited. More precisely, each customer $b \in B$ must be visited n_b times in each visiting week. A day on which a customer is visited is said to be a *visiting day* of the customer. The visiting days within each visiting week must correspond to one of the customer's valid *weekday patterns* Q_b. A weekday pattern is a combination of weekdays on which the customer may be visited. For example, for a customer with $n_b = 2$, the set Q_b could consist of the weekday patterns {Monday, Thursday} and {Tuesday, Friday}, meaning that the customer must be visited either on Monday and Thursday or on Tuesday and Friday. Additionally, if regularity is required with respect to the weekdays on which a customer is visited, we call this a *weekday regularity* of the customer.

The number of weeks in the planning horizon, $|W|$, is typically chosen as the least common multiple of the week rhythms r_b, $b \in B$ since, after this time, the schedule could be repeated identically. Therefore, a customer $b \in B$ must be visited $\frac{|W|}{r_b} n_b$ times during the entire planning horizon. Each visit of a customer requires an individual service time. By t_{bj}, $j \in \{1, ..., \frac{|W|}{r_b} n_b\}$ the service time associated with the j-th visit of customer $b \in B$ is given.

When customer visits are scheduled, compactness – in the sense of geographically concentrated customer visits – plays a crucial role. As in the partitioning subproblem, this is again due to the fact that the service providers have to travel to their customers. On each day in the planning horizon, a service provider has to visit those customers within his or her service territory that are scheduled for that day. Hence, in order to reduce travel time, all customers that need to be visited on the same day should form a geographically compact area. Note that compactness does, of course, not necessarily lead to the shortest possible routes. In fact, there might be less compact solutions that lead to shorter travel times than a highly compact solution. But compact solutions have a significant advantage when it comes to short-term customer requests and unexpected events in the daily business as they provide a high degree of flexibility with respect to the sequence in which customers can be visited. This is illustrated by the example in Figure 4.1. The figure depicts the visits that are scheduled for a specific day. The right-hand side shows a fairly compact solution, whereas the solution on the left-hand side is less compact. In the example on the left-hand side, the service provider starts his route from the depot and intends to visit customer A as the first customer of the route, followed by customers B, C and D. But suppose that in the morning of that day, customer A calls the service provider and tells him that the only

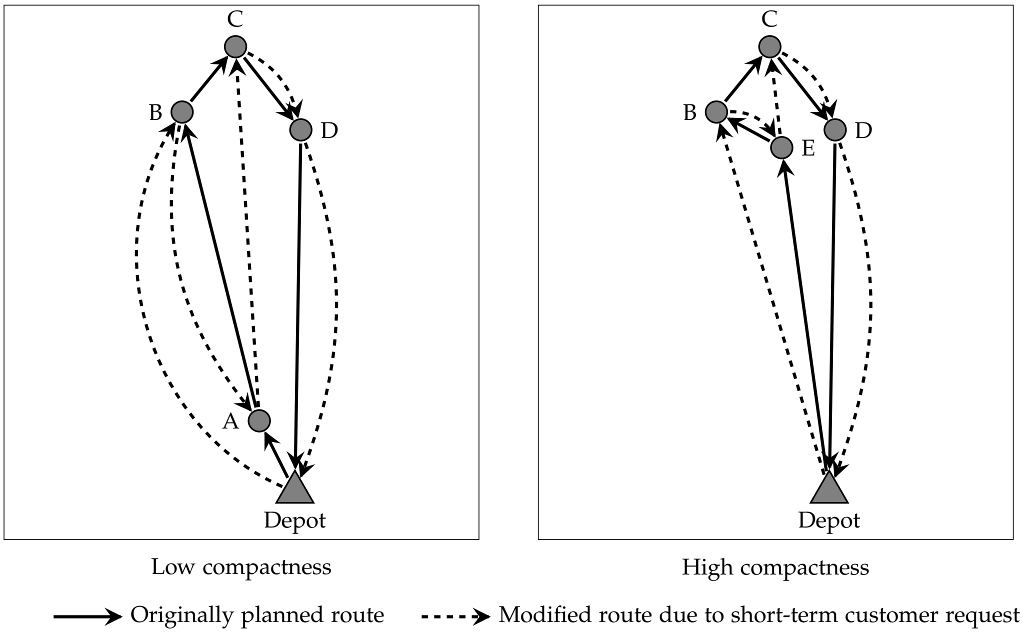


Figure 4.1: Flexibility provided by compact solutions with respect to the sequence in which customers can be visited

possible visiting time is 12 p.m., which is in the middle of the service provider's working day. In this case, the service provider would have to visit customer B first, then travel all the way back to customer A, then visit customers C and D, and finally return to the depot. This would lead to a significant increase in travel time compared to the originally planned route and possibly even to the violation of maximum working hours. In contrast, a more compact solution, such as the example on the right-hand side of the figure, allows the service provider to fulfill short-term customer requests without a substantial increase in travel time. Suppose, for instance, that the service provider originally planned to visit the customers in the sequence E, B, C and D, and that, again, a customer visit has to be rescheduled in the short term. Let us assume in this example that customer E requests to be visited at noon, i.e., customer E cannot be visited as the first customer of the route as it was originally planned. In this case, only a small detour compared to the original plan would be necessary.

Besides the planning criterion that each service provider's daily customer visits should be geographically close to each other, there is an additional compactness requirement related to the customer visits of each week. More precisely, all customers that must be visited by the same service provider in the same week should be geographically concentrated. This requirement is motivated by the fact that, in practice, a visit which is scheduled for a certain day may not be carried out on that day, e.g., because the service provider does not arrive at

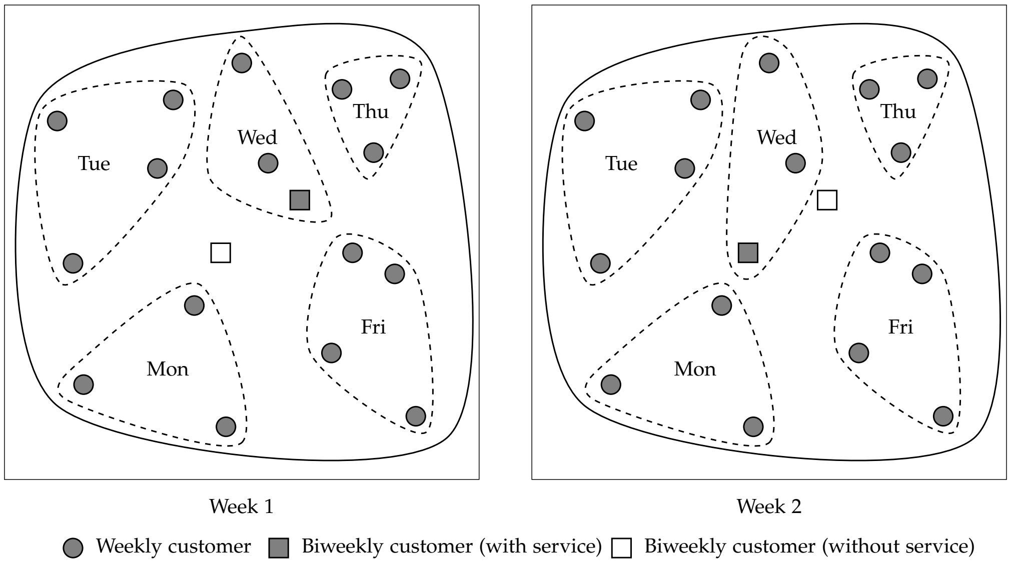


Figure 4.2: Solution to a problem with many weekly customers

the customer on time due to a traffic jam. If the customers that are scheduled for this week are geographically close to each other, the service provider can catch up on the missed visit on another day of the week without having to travel overly long distances.

The achievable compactness of the week clusters depends not only on the geographical distribution of the customers but, to a large extent, also on their week rhythms. This is illustrated by the examples in Figures 4.2 and 4.3. Let us assume for these examples that the planning horizon consists of $|W| = 2$ weeks and $m = 5$ days per week and that all customers must be visited once per visiting week, i.e., $n_b = 1$ for all $b \in B$. Figure 4.2 depicts the solution to a problem with almost only weekly customers that are spread evenly over the entire service territory. In this case, there exists no feasible schedule that would prevent the service provider from traveling almost through the whole service region every week. However, when the customers' week rhythms are more favorable, it is possible to schedule the visits in such a way that the service provider needs to travel only through a relatively small area of the service territory every week. This situation is depicted in Figure 4.3.

In order to avoid time periods with workload peaks and time periods with very little work, another important planning criterion is workload balance over time. Each service provider's workload should be evenly distributed over the planning horizon, i.e., the workload should be roughly the same on all days and in all weeks of the planning horizon.

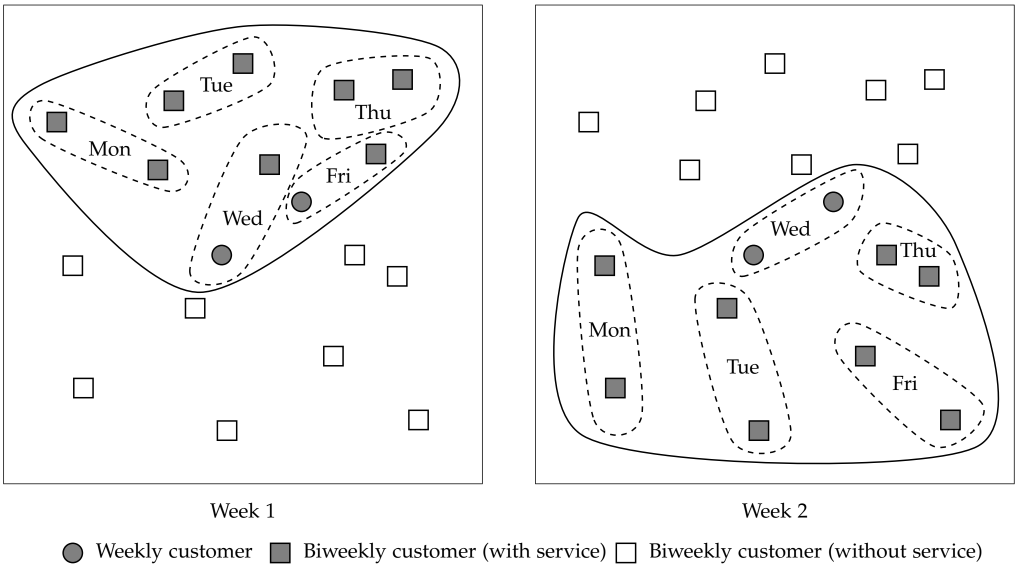


Figure 4.3: Solution to a problem with only few weekly customers

In summary, the MPSTDP-S consists of finding a visit schedule for each service territory that satisfies the following criteria:

- The schedule is feasible with respect to the customers' visiting requirements.
- The customers to be visited on each day form a geographically compact area, which we call *day cluster*.
- The customers to be visited in each week form a geographically compact area, which we call *week cluster*.
- The service time is distributed evenly across the days of the planning horizon.
- The service time is distributed evenly across the weeks of the planning horizon.

With the aim of establishing and maintaining long-lasting customer relationships, the design of the service territories remains fairly stable over a long period of time, typically several years. As opposed to this, visit schedules are valid only for at most 12 months and, hence, have to be redetermined more frequently. Therefore, a solution approach specifically for the scheduling subproblem MPSTDP-S is required. When the schedule expires, this approach can be used to determine a new schedule without modifying the service territories. If, from time to time, the service territories need to be redesigned, we solve the subproblems MPSTDP-P and MPSTDP-S sequentially. This means that we solve a classical districting problem in the first stage. For this purpose, any existing solution method for districting problems can be used. In the second stage, we solve the scheduling subproblem by designing the week and day clusters for each service territory independently.

The partitioning subproblem MPSTDP-P has been studied extensively in the districting literature (see, e.g., Kalcsics, 2015, for a survey of applications and solution methods). However, we believe that this is the first academic work to deal with the scheduling subproblem MPSTDP-S. Therefore, we concentrate on the MPSTDP-S in the remainder of this chapter.

4.3 Related Work

To the best of our knowledge, there are only two papers dealing with multi-period territory design problems. Lei et al. (2015) consider a problem in which the occurrence of customers changes from period to period. They assume that the customers of each period are known in advance and that a period comprises several weeks. In each period, all customers must be visited exactly once on a route which starts and ends at one of the available depots. The following decisions must be made: For each period, the customers must be partitioned into districts, and a depot must be assigned to each district. Furthermore, the customers of each district must be partitioned into subdistricts with each subdistrict representing the customers that must be visited on a particular working day. As the objective function the authors use a weighted sum of four measures, namely the number of districts, the compactness of subdistricts, district similarity in subsequent periods and balance with respect to salesmen's profit. They propose an adaptive large neighborhood search and solve modified Solomon and Gehring & Homberger test instances with up to 400 customers and a maximum of three periods. Lei et al. (2016) describe a similar problem, in which customers are either deterministic or stochastic. Districts must be determined for each period of the planning horizon before the stochastic customers are revealed. All customers (deterministic and stochastic) of the same district have to be served on a single vehicle route from a central depot. The objectives are the same as in Lei et al. (2015), but instead of using a weighted sum as the objective function, the authors treat the problem as a true multi-objective optimization problem and solve it with a multi-objective evolutionary algorithm. Although the problems in Lei et al. (2015) and Lei et al. (2016) consider a multi-period planning horizon, they do not contain a scheduling component comparable to the MPSTDP-S. In Lei et al. (2015), the service days within each period must be decided, but, in contrast to the MPSTDP-S, each customer must be served exactly once per period and, hence, there are no restrictions on the temporal distribution of visits. In particular, Lei et al. (2015) do not consider week rhythms and weekday patterns, which are essential components of the MPSTDP-S. In Lei et al. (2016), there is no scheduling aspect at all since the customers that have to be served in a particular period are given by the concrete demand realization. Hence, a transformation of the MPSTDP-S to the problems studied in Lei et al. (2015) or Lei et al. (2016) is not possible.

The task of scheduling regular customer visits throughout a planning horizon arises also in some extensions of the vehicle routing problem and in multi-period scheduling problems. Since there exist different variants of regularity considered in these problems, we introduce a short classification. Figure 4.4 contains examples for the most important types of regularity. In the figure, we consider one exemplary customer and a planning horizon of four weeks and five days per week. The filled circles indicate the visiting days of the customer. Regularity type (1) means that the visiting weeks are periodically recurring, i.e., the number of weeks between consecutive visiting weeks is constant. In the example, the customer is visited every second week, beginning from the first week of the planning horizon. Regularity type (2) is similar to type (1), but refers to days instead of weeks. A customer is said to have regularity type (2) if the number of days between consecutive visits is constant. Regularity type (3) is a special case of type (1). Here, besides the periodicity with respect to visiting weeks, the weekdays on which the visits take place are the same in each visiting week. The customer in the example is visited biweekly on the second and fifth weekday. Finally, regularity type (4) is given if the number of days between consecutive visits is constant and the weekdays of the visits are identical throughout the planning horizon. Note that in the MPSTDP-S, regularity type (1) or (3) is considered, depending on the presence of weekday regularity requirements.

Scheduling and regularity aspects are considered in the period vehicle routing problem (PVRP) and the inventory routing problem (IRP). In the classical vehicle routing problem (VRP), customers must be assigned to vehicles and vehicle routes must be determined. The PVRP extends the classical VRP by a multi-period planning horizon in which customers must be visited several times. As an additional decision, the PVRP contains the selection of a feasible visit schedule for each customer. Regularity types (1)–(4) can be considered through an appropriate choice of valid visit schedules. For reviews on the PVRP, we refer the reader to Francis et al. (2008) and Irnich et al. (2014). Recent papers on specific variants can be found in Archetti et al. (2015), Miranda et al. (2015) and Rahimi-Vahed et al. (2015). We would like to stress one particular paper from the PVRP literature, namely the paper by Mourgaya and Vanderbeck (2007). The problem studied by Mourgaya and Vanderbeck is quite similar to our problem as it is a tactical variant of the PVRP, in which customer visits are scheduled and assigned to vehicles in such a way that workload is balanced and compact clusters are achieved, whereas routing cost are not explicitly taken into account. But in contrast to our problem, their tactical model does not contain weeks as a separate time scale, i.e., they do not take into account the compactness of week clusters. Moreover,

Week	1					2					3					4				
Type of regularity / Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(1) Periodic w.r.t. weeks	●			●								●	●							
(2) Periodic w.r.t. days		●		●		●		●		●		●		●		●		●		●
(3) Periodic w.r.t. weeks + weekday regularity		●			●							●			●					
(4) Periodic w.r.t. days + weekday regularity		●					●					●					●			

Figure 4.4: Examples for different types of regularity (a filled circle indicates that the customer is visited on that day)

the planning horizon considered in their experiments consists only of up to six days, and it appears questionable if their column generation-based heuristic can be applied to planning horizons of several months.

In the IRP, a supplier is responsible for replenishing the inventory of its costumers. To this end, products must be delivered to the customers on vehicle routes starting and ending at the supplier. Besides the routing decision, the decisions in the IRP include the timing and the quantities of the deliveries. Regularity type (2) can be observed, e.g., in the cyclic IRP studied by Raa and Aghezzaf (2009). A less restrictive approach is described by Coelho et al. (2012), who consider (among other consistency features) the possibility of specifying a minimum and maximum time interval between consecutive visits of the same customer, which results in regularity type (2) if the minimum and maximum time interval are set to the same value. Extensive reviews on the IRP can be found in Bertazzi et al. (2008) and Coelho et al. (2014). Recent papers on specific variants are provided by Chitsaz et al. (2016), Dong and Turnquist (2015), Ekici et al. (2015) and Li et al. (2016).

The main difference to our problem is that both the PVRP and the IRP explicitly aim at minimizing routing costs. In our problem, however, we aim at geographical compactness.

Another class of problems related to the MPSTDP-S are multi-period scheduling problems in which tasks have to be scheduled according to strict, predefined rhythms. In these problems, the time period between consecutive executions of a task is constant, corresponding to regularity type (2) with the only difference that time is not necessarily discretized into days. Applications of this kind of multi-period scheduling problems can be found in maintenance scheduling (e.g., Wei and Liu, 1983), processor scheduling (e.g., Korst et al., 1991), and logistics (e.g., Campbell and Hardin, 2005; Delgado et al., 2005; Kazan et al., 2012). However, these problems have in common that geographical aspects are not taken into account, i.e., compactness is not considered a relevant planning criterion. For this reason, solution approaches for this class of problems cannot directly be applied to the MPSTDP-S.

In summary, the main differences between the MPSTDP-S and the related problems are the following: The presented multi-period territory design problems do not contain a scheduling aspect comparable to the MPSTPD-S. The objective in the PVRP and IRP is to optimize routing cost, whereas in our problem compact week and day clusters are desired. Multi-period scheduling problems lack the consideration of any geographical aspects.

4.4 Mathematical Formulation of the MPSTDP-S

In this section, we state the subproblem MPSTDP-S as a mixed integer linear program. To this end, we introduce the following additional notation.

Let $P = \bigcup_{b \in B} P_b$ denote the set of all week patterns and $Q = \bigcup_{b \in B} Q_b$ the set of all weekday patterns. Then, for each week pattern $p \in P$ the parameter ψ_p^w is 1 if the week pattern contains week $w \in W$, and 0 otherwise. Analogously, ω_q^d states whether weekday pattern $q \in Q$ contains day $d \in D$. Due to the rigid week rhythms, it is easy to transform the service times t_{bj}, $j \in \{1, ..., \frac{|W|}{r_b} n_b\}$ into parameters t_b^w, which state the time for serving customer $b \in B$ in week $w \in W$, and parameters t_{bq}^d, which denote the time required for the service of customer $b \in B$ on day $d \in D$ if weekday pattern $q \in Q_b$ is selected. The average weekly and daily service times are denoted by $\mu^{week} = \frac{T}{|W|}$ and $\mu^{day} = \frac{T}{|D|}$, respectively, with $T = \sum_{b \in B} \sum_{j \in \{1,...,|W|/r_b n_b\}} t_{bj}$ being the total service time over all customers. Parameters τ^{week} and τ^{day} define the maximal allowable percentage that the actual service times may deviate from the average weekly and daily service times, respectively. The week of day $d \in D$ is represented by $\phi(d) \in W$. The distance from customer i to customer b is given by c_{ib}, $i, b \in B$.

We introduce the following decision variables:

$$g_{bp} = \begin{cases} 1 & \text{if week pattern } p \in P_b \text{ is assigned to customer } b \in B \\ 0 & \text{otherwise} \end{cases}$$

$$h_{bq}^w = \begin{cases} 1 & \text{if weekday pattern } q \in Q_b \text{ is assigned to customer } b \in B \text{ in week } w \in W \\ 0 & \text{otherwise} \end{cases}$$

These variables are sufficient to describe the temporal distribution of the visits, but they do not suffice to take into account the compactness criteria. As the compactness measure in our approach, we use the sum of the distances between the customers that are served on a particular day (week) and a customer that is selected as the cluster center for this day (week). Such a center-based compactness measure is quite common in literature (see, e.g., Fleischmann and Paraschis, 1988; Hess et al., 1965; Hojati, 1996; Salazar-Aguilar et al., 2011). There are also other ways to measure compactness, e.g., based on pairwise distances between customers. However, these measures are computationally intractable when incorporated into a MIP model and can, therefore, only be used for an a posteriori evaluation of solutions.

To integrate the compactness measure into the model, we introduce the following auxiliary variables:

$$u_{ib}^{w} = \begin{cases} 1 & \text{if customer } b \in B \text{ is assigned to week center } i \in B \text{ in week } w \in W \\ 0 & \text{otherwise} \end{cases}$$

$$v_{ib}^{d} = \begin{cases} 1 & \text{if customer } b \in B \text{ is assigned to day center } i \in B \text{ on day } d \in D \\ 0 & \text{otherwise} \end{cases}$$

$$x_{b}^{w} = \begin{cases} 1 & \text{if customer } b \in \text{ is the center in week } w \in W \\ 0 & \text{otherwise} \end{cases}$$

$$y_{b}^{d} = \begin{cases} 1 & \text{if customer } b \in B \text{ is the center on day } d \in D \\ 0 & \text{otherwise} \end{cases}$$

For a better overview, the notation used in the basic model of the MPSTDP-S is summarized in Table 4.1.

4.4.1 Basic Model

Using the introduced notation, the MPSTDP-S can be formulated as the following MIP, which we denote by $\mathsf{SCHEDULE}_{\mathsf{MIP}}$:

$$\lambda \sum_{b\in B}\sum_{i\in B}\sum_{w\in W} n_b c_{ib} u_{ib}^{w} + (1-\lambda)\sum_{b\in B}\sum_{i\in B}\sum_{d\in D} c_{ib} v_{ib}^{d} \quad \to \min \tag{4.1}$$

$$\text{s.t.} \quad \sum_{p\in P_b} g_{bp} = 1 \qquad b \in B \tag{4.2}$$

$$\sum_{i\in B} u_{ib}^{w} = \sum_{p\in P_b} \psi_p^{w} g_{bp} \qquad b \in B, w \in W \tag{4.3}$$

$$u_{ib}^{w} \leqslant x_i^{w} \qquad b, i \in B, w \in W \tag{4.4}$$

$$\sum_{b\in B} x_b^{w} = 1 \qquad w \in W \tag{4.5}$$

$$\sum_{b\in B}\sum_{p\in P_b} t_b^{w} \psi_p^{w} g_{bp} \geqslant (1-\tau^{week})\mu^{week} \qquad w \in W \tag{4.6}$$

$$\sum_{b\in B}\sum_{p\in P_b} t_b^w \psi_p^w g_{bp} \leqslant (1+\tau^{week})\mu^{week} \qquad w\in W \qquad (4.7)$$

$$\sum_{q\in Q_b} h_{bq}^w = \sum_{p\in P_b} \psi_p^w g_{bp} \qquad b\in B, w\in W \qquad (4.8)$$

$$\sum_{i\in B} v_{ib}^d = \sum_{q\in Q_b} \omega_q^d h_{bq}^{\phi(d)} \qquad b\in B, d\in D \qquad (4.9)$$

$$v_{ib}^d \leqslant y_i^d \qquad b,i\in B, d\in D \qquad (4.10)$$

$$\sum_{b\in B} y_b^d = 1 \qquad d\in D \qquad (4.11)$$

$$\sum_{b\in B}\sum_{q\in Q_b} t_{bq}^d \omega_q^d h_{bq}^{\phi(d)} \geqslant (1-\tau^{day})\mu^{day} \qquad d\in D \qquad (4.12)$$

$$\sum_{b\in B}\sum_{q\in Q_b} t_{bq}^d \omega_q^d h_{bq}^{\phi(d)} \leqslant (1+\tau^{day})\mu^{day} \qquad d\in D \qquad (4.13)$$

$$g_{bp} \in \{0,1\} \qquad b\in B, p\in P_b \qquad (4.14)$$

$$h_{bq}^w \in \{0,1\} \qquad b\in B, q\in Q_b, w\in W \qquad (4.15)$$

$$u_{ib}^w \geqslant 0 \qquad b,i\in B, w\in W \qquad (4.16)$$

$$v_{ib}^d \geqslant 0 \qquad b,i\in B, d\in D \qquad (4.17)$$

$$x_b^w \in \{0,1\} \qquad b\in B, w\in W \qquad (4.18)$$

$$y_b^d \in \{0,1\} \qquad b\in B, d\in D \qquad (4.19)$$

The Objective Function (4.1) aims at optimizing compactness. The first term represents the compactness of the week clusters, whereas the second term expresses the compactness of the day clusters. Parameter $\lambda \in [0,1]$ is used to weight between weekly and daily compactness. Constraints (4.2) guarantee that a valid week pattern is assigned to each customer. Constraints (4.3) and (4.4) ensure that a customer which is served in a particular week is assigned to a week center of the same week. Constraints (4.5) guarantee that exactly one week center per week is chosen. Balanced service times across the weeks are enforced by Constraints (4.6) and (4.7) by limiting the feasible deviation from the average weekly service time. Constraints (4.8) link the week pattern choice and weekday pattern choice for each customer. If the selected week pattern for a customer implies service in a particular week, a valid weekday pattern must be selected for this week. Otherwise, no weekday pattern may be selected. Constraints (4.9)–(4.13) are analogous to Constraints (4.3)–(4.7), but refer to decisions at day level instead of week level. Constraints (4.14)–(4.19) are the domain constraints. Note that Constraints (4.16) and (4.17) define continuous variables, but due to Constraints (4.3), (4.4), (4.9) and (4.10) these variables are implicitly binary.

Note that since the week patterns imply periodicity with respect to the visiting weeks of each customer, the basic model considers regularity type (1) for all customers.

Table 4.1: Summary of the notation for the basic model of the MPSTDP-S

Index sets	
B	Customers
W	Weeks in the planning horizon
D	Days in the planning horizon
P	All week patterns
P_b	Valid week patterns for customer $b \in B$
Q	All weekday patterns
Q_b	Valid weekday patterns for customer $b \in B$
Parameters	
$c_{ib} \in \mathbb{R}^+$	Distance from customer $i \in B$ to customer $b \in B$
$n_b \in \mathbb{N}^+$	Number of visits of customer $b \in B$ per visiting week
$t_b^w \in \mathbb{R}^+$	Time for serving customer $b \in B$ in week $w \in W$
$t_{bq}^d \in \mathbb{R}^+$	Time for serving customer $b \in B$ on day $d \in D$ if weekday pattern $q \in Q_b$ is selected
$\phi(d) \in W$	Week of day $d \in D$
$\psi_p^w \in \{0,1\}$	Indicates whether week pattern $p \in P$ contains week $w \in W$ (1) or not (0)
$\omega_q^d \in \{0,1\}$	Indicates whether weekday pattern $q \in Q$ contains day $d \in D$ (1) or not (0)
$\mu^{week} \in \mathbb{R}^+$	Average weekly service time
$\mu^{day} \in \mathbb{R}^+$	Average daily service time
$\tau^{week} \in \mathbb{R}^+$	Maximum allowable deviation of the actual from the average weekly service time
$\tau^{day} \in \mathbb{R}^+$	Maximum allowable deviation of the actual from the average daily service time
$\lambda \in [0,1]$	Weight for weekly compactness
Variables	
$g_{bp} \in \{0,1\}$	Takes a value of 1 if and only if week pattern $p \in P_b$ is selected for customer $b \in B$
$h_{bq}^w \in \{0,1\}$	Takes a value of 1 if and only if weekday pattern $q \in Q_b$ is selected for customer $b \in B$ in week $w \in W$
$u_{ib}^w \in \{0,1\}$	Takes a value of 1 if and only if customer $b \in B$ is assigned to week center $i \in B$ in week $w \in W$
$v_{ib}^d \in \{0,1\}$	Takes a value of 1 if and only if customer $b \in B$ is assigned to day center $i \in B$ on day $d \in D$
$x_b^w \in \{0,1\}$	Takes a value of 1 if and only if customer $b \in B$ is selected as the center for week $w \in W$
$y_b^d \in \{0,1\}$	Takes a value of 1 if and only if customer $b \in B$ is selected as the center for day $d \in D$

4.4.2 Weekday Regularity

Recall that we defined weekday regularity as regularity with respect to the weekdays on which a particular customer is visited. We distinguish two variants, namely *strict weekday regularity* and *partial weekday regularity*. In the following, we describe the two variants and explain how model $\mathsf{SCHEDULE}_{MIP}$ must be adapted in each case.

Strict Weekday Regularity

If strict weekday regularity is required for a particular customer, the customer must be visited according to the same weekday pattern in every visiting week. In other words, the weekdays on which the customer is visited must always be the same throughout the entire planning horizon. Hence, a customer with strict weekday regularity has regularity type (3).

Let $B_{strict} \subseteq B$ denote the set of customers that demand strict weekday regularity. Then, the following modifications of the model must be made. The first r_b weeks of the planning horizon contain exactly one week in which customer $b \in B_{strict}$ is visited. Since, in the presence of strict weekday regularity, the same weekday pattern must be selected in every visiting week, the weekday pattern which is selected for the first r_b weeks determines the weekday patterns for all remaining weeks of the planning horizon. Hence, for all customers that require strict weekday regularity, variables h_{bq}^{w} need to be introduced for the first r_b weeks only. For all $b \in B_{strict}$, Constraints (4.15) are therefore modified as follows:

$$h_{bq}^{w} \in \{0,1\} \qquad b \in B_{strict}, q \in Q_b, w \in W, w \leqslant r_b \tag{4.15a}$$

Moreover, for all $b \in B_{strict}$, Constraints (4.8), which link the week pattern and weekday pattern decisions, also need to be introduced for the first r_b weeks only:

$$\sum_{q \in Q_b} h_{bq}^{w} = \sum_{p \in P_b} \psi_p^{w} g_{bp} \qquad b \in B_{strict}, w \in W, w \leqslant r_b \tag{4.8a}$$

In Constraints (4.9), (4.12) and (4.13) all variables h_{bq}^{w} with $b \in B_{strict}$, $w > r_b$ must be replaced by the corresponding variables of the first r_b weeks. For this purpose, we define function $\bar{\phi}(b,d)$ for all $b \in B$, $d \in D$:

$$\bar{\phi}(b,d) = \begin{cases} \phi(d) & \text{if } b \notin B_{strict} \\ ((\phi(d)-1) \bmod r_b)+1 & \text{if } b \in B_{strict} \end{cases}$$

For all customers without strict weekday regularity, i.e., $b \notin B_{strict}$, $\bar{\phi}(b,d)$ returns the week that contains the given day $d \in D$. For all customers which require strict weekday regularity, i.e., $b \in B_{strict}$, $\bar{\phi}(b,d)$ returns the week within the first r_b weeks of the planning horizon that determines the weekday pattern for customer b in the week which contains day $d \in D$.

All occurrences of $\phi(d)$ in the original model are replaced by $\bar{\phi}(b,d)$, which yields the modified Constraints (4.9a), (4.12a) and (4.13a):

$$\sum_{i \in B} v_{ib}^{d} = \sum_{q \in Q_b} \omega_q^{d} h_{bq}^{\bar{\phi}(b,d)} \qquad b \in B, d \in D \tag{4.9a}$$

$$\sum_{b \in B} \sum_{q \in Q_b} t_{bq}^{d} \omega_q^{d} h_{bq}^{\bar{\phi}(b,d)} \geqslant (1-\tau^{day})\mu^{day} \qquad d \in D \tag{4.12a}$$

$$\sum_{b \in B} \sum_{q \in Q_b} t_{bq}^{d} \omega_q^{d} h_{bq}^{\bar{\phi}(b,d)} \leqslant (1+\tau^{day})\mu^{day} \qquad d \in D \tag{4.13a}$$

Partial Weekday Regularity

Similarly to strict weekday regularity, partial weekday regularity also describes the requirement that a customer must be visited according to a regular weekday pattern. However, partial weekday regularity allows a predefined number of deviations from the regular weekday pattern and is, therefore, less restrictive than strict weekday regularity.

Let $B_{partial} \subseteq B$ denote the set of customers which require partial weekday regularity and $f_b \in \mathbb{N}^+$, $b \in B_{partial}$, the number of allowed deviations from the regular pattern for customer b. Then, for each customer $b \in B_{partial}$, additional variables and constraints need to be added to model $\mathsf{SCHEDULE}_{MIP}$:

$$\sum_{q \in Q_b} h'_{bq} = 1 \qquad b \in B_{partial} \tag{4.20}$$

$$\sum_{w \in W} h^w_{bq} \geqslant h'_{bq} \left(\frac{|W|}{r_b} - f_b \right) \qquad b \in B_{partial}, q \in Q_b \tag{4.21}$$

$$h'_{bq} \in \{0, 1\} \qquad b \in B_{partial}, q \in Q_b \tag{4.22}$$

Variables h'_{bq} defined in Constraints (4.22) describe whether weekday pattern $q \in Q_b$ is selected as the regular weekday pattern for customer $b \in B_{partial}$:

$$h'_{bq} = \begin{cases} 1 & \text{if weekday pattern } q \in Q_b \text{ is selected as the regular weekday pattern} \\ & \text{for customer } b \in B_{partial} \\ 0 & \text{otherwise} \end{cases}$$

Constraints (4.20) guarantee that for each customer $b \in B_{partial}$ exactly one regular weekday pattern is selected. $\frac{|W|}{r_b}$ is the number of weeks in which customer $b \in B_{partial}$ is visited throughout the planning horizon. Hence, Constraints (4.21) make sure that the selected weekday patterns deviate in at most f_b weeks from the selected regular weekday pattern.

4.4.3 Remarks on the Model

Using model $\mathsf{SCHEDULE}_{MIP}$, we tried to compute optimal solutions for small test instances with 30 and 50 customers, four weeks and five days per week. Only three out of ten 30-customer instances could be solved to optimality within a time limit of one hour. The average optimality gap of the remaining seven 30-customer instances was 3.6%. Out of the ten 50-customer instances, none could be solved to optimality, even with a time limit of ten hours (the average optimality gap was 4.5%). Hence, it seems impossible to solve this model to optimality for realistic instance sizes, which typically comprise more than 100 customers and several months. This is mainly due to two reasons, namely the high symmetry of the model and the great number of variables. In the following, we describe our attempts to address these two issues.

Model $\mathsf{SCHEDULE}_{\mathsf{MIP}}$ contains variables to describe the selection of week patterns, g_{bp}, and variables to describe the selection of weekday patterns within weeks, h^{w}_{bq}. The weekday pattern variables contain more information than the week pattern variables. In fact, the values of the week pattern variables can be derived from the values of the weekday pattern variables. It is easily possible to formulate the MPSTDP-S without week pattern variables g_{bp} and, hence, reduce the number of variables in the model. But experiments showed that the performance of the model is better if it contains both weekday and week pattern variables. Therefore, we decided to use both groups of variables.

There is a lot of symmetry in model $\mathsf{SCHEDULE}_{\mathsf{MIP}}$, i.e., there exist many different feasible solutions that have the same objective function value. For example, consider the case where the week rhythm, r_b, is from the set {1, 2, 4} and the number of visits per visiting week, n_b, is equal to one for all customers $b \in B$. Suppose that there are no weekday regularity requirements and no restrictions in terms of valid weekdays, i.e., the set of valid weekday patterns, Q_b, $b \in B$, contains a valid pattern for each weekday. Further, let the planning horizon consist of four weeks and five days per week. Let a given feasible solution consist of the four week clusters C^1, C^2, C^3 and C^4, which represent the customers that are scheduled for week one, two, three and four, respectively. Symmetric solutions can be determined by assigning the week clusters to different weeks. However, this rearrangement is subject to restrictions due to the customers' week rhythms. Customers with a week rhythm of one or four do not impose any restrictions on the rearrangement. But due to the biweekly customers, week clusters C^1 and C^3 as well as week clusters C^2 and C^4 must not be assigned to subsequent weeks. Thus, eight symmetric solutions can be obtained by rearrangements of week clusters (assuming feasibility with respect to the balance constraints), see Table 4.2. Additionally, the model contains a lot of symmetry at the level of day clusters. Since there are no restrictions with respect to the weekdays on which customers are served, there are 5! different ways of assigning day clusters to weekdays within each week. In a four-week planning horizon, this results in $(5!)^4$ symmetric solutions due to rearrangements of day clusters. When the symmetry of week and day clusters is combined, $8 \cdot (5!)^4 - 1 = 1,658,879,999$ symmetric solutions can be determined to each feasible solution.

In order to deal with the high symmetry of the model, we tested instance-specific symmetry breaking constraints. The idea was to order the service times of the weeks and of the days within each week in such a way that many symmetric solutions become infeasible.

Table 4.2: Symmetric solutions obtained by rearrangements of week clusters

	Visit in week			
Symmetric solution no.	1	2	3	4
1	C^1	C^2	C^3	C^4
2	C^3	C^2	C^1	C^4
3	C^1	C^4	C^3	C^2
4	C^3	C^4	C^1	C^2
5	C^2	C^1	C^4	C^3
6	C^4	C^1	C^2	C^3
7	C^2	C^3	C^4	C^1
8	C^4	C^3	C^2	C^1

However, we experienced a deterioration in the running times, presumably because the symmetry breaking constraints make it more difficult for the heuristics of the MIP solver to find new feasible solutions. A different idea for symmetry breaking, which proved to be effective for a special variant of model $\mathsf{SCHEDULE}_{\mathsf{MIP}}$, is presented in Section 5.5.5.

4.5 Location-Allocation Heuristic

Due to the high complexity of the problem, we propose a heuristic solution approach. Our approach – as many approaches in territory design – is based on the old idea of Hess et al. (1965) to decompose the problem into a location subproblem and an allocation subproblem (see Kalcsics et al., 2005, for an overview of papers using this idea). Therefore, we briefly describe the approach of Hess et al. in the following.

Hess et al. (1965) deal with a (single-period) political districting problem. In this problem, a set of basic areas must be partitioned into electoral districts in such a way that the districts are compact, balanced with respect to population, and contiguous. In the location subproblem, they determine a subset of the basic areas which serve as district centers. For the first iteration of the algorithm, they use initial trial centers; for all subsequent iterations, they calculate the centers of gravity for each temporary district and use them as the new district centers. Then, in the allocation subproblem, they assign each basic area to exactly one district center. To this end, they solve a transportation problem and uniquely resolve all split assignments (a customer has a split assignment if he is assigned to more than one center). Location and allocation are repeated in an iterative manner until the solution process converges.

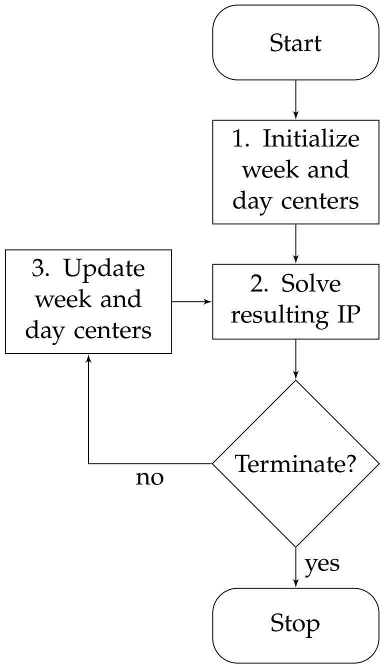


Figure 4.5: Location-allocation heuristic of Hess et al. (1965) adapted to the MPSTDP-S

We adopt this decomposition approach for the MPSTDP-S. The general procedure of our adapted location-allocation heuristic is outlined in Figure 4.5. The algorithm starts with selecting an initial set of week and day centers (Step 1). By fixing the center decisions, we obtain an integer program (IP) which is solved by a general-purpose MIP solver (Step 2). Then, the week and day centers are updated: For each week cluster and for each day cluster, the customer $b \in B$ which, when picked as the cluster center, leads to the smallest contribution to the Objective Function (4.1) is used as the new center (Step 3). Steps 2 and 3 are performed iteratively. The algorithm terminates if the current iteration has not produced an improved solution or if a user-defined maximum number of iterations, $iter_{max}$, has been performed.

To the best of our knowledge, our approach is the first that extends the work of Hess et al. (1965) to a multi-period setting. The major novelties of our location-allocation heuristic are the initialization procedure and the resulting IP. In the following, we go into the details of these two components.

4.5.1 Selection of Initial Centers

The selection of good initial centers for the MPSTDP-S differs greatly from the single-period districting problem. In the single-period case, one wants to achieve compact, non-overlapping districts. Therefore, a reasonable strategy is to distribute the initial centers relatively evenly across the region under study, probably with a higher concentration in areas with high demand, i.e., in areas with a large number of customers or with a high level of activity. However, the strategy for the single-period districting problem is not applicable to the MPSTDP-S where customers are visited several times throughout the planning horizon. In the multi-period case, non-overlapping week or day clusters can, in general, not be achieved.

In the following, we develop a suitable initialization procedure for the MPSTDP-S based on the following observations:

1. At the level of individual customers, there is a weekly regularity due to week rhythms r_b, $b \in B$. These regularities can result in similarities at the level of week clusters, i.e., week clusters in different weeks may have a large number of customers in common. Such similarities can establish, at the earliest, after r_{min} weeks, with $r_{min} = \min_{b \in B} r_b$ denoting the smallest week rhythm of all customers. To account for this, only r_{min} different initial week centers should be selected. If the number of weeks within the planning horizon, $|W|$, is greater than r_{min}, these week centers as well as their corresponding day centers should recur every r_{min} weeks.

2. The r_{min} different week centers should be evenly distributed over the entire region under study to facilitate the formation of compact week clusters, i.e., week clusters which span a relatively small geographical area.

3. The day centers of each week should obviously be close to their corresponding week center.

4. The day centers should, however, not (or at least not all) coincide with the corresponding week center, but rather be evenly distributed in the vicinity of the week center to promote the formation of compact day clusters.

5. The smaller the week rhythm r_b of a customer $b \in B$, the more likely it should be that the customer is selected as a week center or a day center. This favors the selection of customers $b \in B$ with $r_b = r_{min}$ and, therefore, increases the probability that the visits of these customers can be scheduled in accordance with their occurrence as centers.

We adapt the well-known initialization procedure of k-means++ (Arthur and Vassilvitskii, 2007), a popular seeding technique for cluster analysis, to take these observations into account. Let $c(b, J)$, $b \in B$, $J \subseteq B$ denote the minimum distance between customer b and any customer in set J. Then, given a set of candidate centers $I \subseteq B$ and the set of already selected centers $J \subseteq B$, the probabilistic function in Algorithm 4.1 is used to determine the next initial week or day center. Algorithm 4.1 is equivalent to the procedure used in k-means++ with the only difference that, in our adapted version, also the week rhythms are taken into account. Hence, in accordance with observations 2, 4 and 5, the probability that a candidate center is selected depends on its distance to the closest center already chosen and on its week rhythm. This means, the farther away from an already selected center and the smaller the week rhythm, the more likely it is that a customer is selected as the next initial week or day center.

Algorithm 4.1 Function to pick the next initial week or day center based on k-means++ (Arthur and Vassilvitskii, 2007)

Input: Set of candidate centers $I \subseteq B$, set of already chosen centers $J \subseteq B$
Output: The next center $b \in I$

function NEXT_CENTER(I, J)
  if $J = \emptyset$ **then**
    return $b \in I$ with probability $\frac{1}{r_b} / \sum_{b' \in I} \frac{1}{r_{b'}}$
  else
    return $b \in I$ with probability $\frac{c^2(b,J)}{r_b} / \sum_{b' \in I} \frac{c^2(b',J)}{r_{b'}}$
  end if
end function

The function in Algorithm 4.1 is used in Algorithms 4.2 and 4.3 to select the initial week and day centers, respectively. As in k-means++ (Arthur and Vassilvitskii, 2007), this is done in an iterative fashion, but we adapt the procedure of k-means++ in such a way that observations 1 and 3 are considered.

In the first while-loop of Algorithm 4.2, r_{min} different customers are selected as the week centers, $\gamma^w \in B$, $w \in W$ for the first r_{min} weeks of the planning horizon. The set of candidate centers consists of all customers, i.e., $I = B$. According to observation 1, the second while-loop makes sure that these centers repeat periodically every r_{min} weeks.

To select the initial day centers, $\gamma^d \in B$, $d \in D$, we proceed as illustrated in Algorithm 4.3. We subdivide the entire region into temporary week clusters by assigning each customer – independently of his week rhythm – to the closest week center, i.e., the temporary week cluster $\widetilde{C}^w$ is defined as $\widetilde{C}^w = \{b \in B : c_{\gamma^w b} < c_{\gamma^{w'} b}, w \neq w'\}$ for each week $w \in W$ with $w \leqslant r_{min}$. We use again the function in Algorithm 4.1 to determine suitable day centers, but we restrict the day center candidates to the customers within each temporary week cluster, i.e., $I = \widetilde{C}^w$. Through this, we make sure that the day centers of each week are close to the corresponding week center, as is required by observation 3. Analogously to the initialization of the week centers and according to observation 1, the day centers recur every r_{min} weeks.

Algorithm 4.2 Initialization of week centers

Input: Set of customers B
Output: Initial week centers γ^w, $w \in W$

```
procedure INIT_WEEK_CENTERS
    w ← 1
    J ← ∅
    while w ⩽ r_min do
        γ^w ← NEXT_CENTER(B, J)
        J ← J ∪ {γ^w}
        w ← w + 1
    end while
    while w ⩽ |W| do
        γ^w ← γ^((w−1) mod r_min)+1
        w ← w + 1
    end while
end procedure
```

Algorithm 4.3 Initialization of day centers

Input: Temporary week clusters $\widetilde{C}^w$, $w \in W$
Output: Initial day centers γ^d, $d \in D$

```
procedure INIT_DAY_CENTERS
    w ← 1
    while w ⩽ r_min do
        J ← ∅
        for all days d in week w do
            γ^d ← NEXT_CENTER(C̃^w, J)
            J ← J ∪ {γ^d}
        end for
        w ← w + 1
    end while
    while w ⩽ |W| do
        for all days d in week w do
            γ^d ← γ^((d−1) mod (m r_min))+1
        end for
        w ← w + 1
    end while
end procedure
```

An example of initial week and day centers is visualized in Figure 4.6. In this example, we assume that the planning horizon consists of $|W| = 8$ weeks and that the minimum week rhythm $r_{min} = 4$. Hence, the initial centers of week one correspond to the initial centers of week five, the initial centers of week two correspond to the initial centers of week six, and so on. The dashed lines indicate the borders of the temporary week clusters. The dark triangles represent the locations of the week centers and the light triangles the locations of the day centers within the respective weeks.

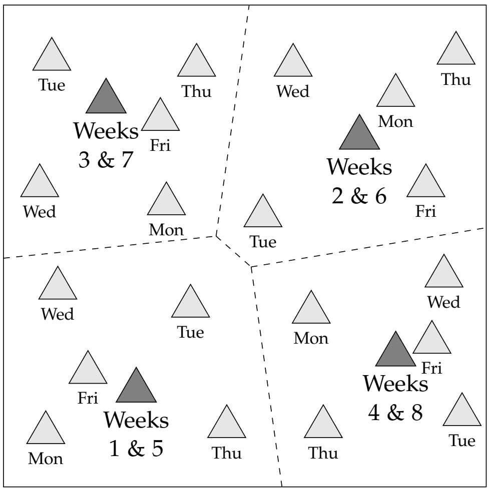


Figure 4.6: Example of initial week and day centers with $r_{min} = 4$ and $|W| = 8$

4.5.2 Integer Linear Program with Fixed Centers

When week and day center decisions are fixed, variables u^w_{ib}, v^d_{ib}, x^w_b and y^d_b (defined in Constraints (4.16)–(4.19)) can be removed from model $\mathsf{SCHEDULE}_{MIP}$. The only remaining variables are the pattern variables g_{bp} and h^w_{bq} (Constraints (4.14) and (4.15)). Note that the compactness criterion in the objective can now be expressed as a function of the pattern variables since the distances between customers and centers can be attached directly to the pattern variables. Denote again by $\gamma^w \in B$ the customer that represents the center of week $w \in W$, and by $\gamma^d \in B$ the customer that represents the center of day $d \in D$. Further, define $\bar{c}_{bp} = \sum_{w \in W} \psi^w_p n_b c_{\gamma^w b}$ and $\bar{c}^w_{bq} = \sum_{d \in D(w)} \omega^d_q c_{\gamma^d b}$ with $D(w)$ representing the days in week $w \in W$. Then, model $\mathsf{SCHEDULE}_{MIP}$ reduces to the following integer linear program, which we denote by ALLOC_{MIP}:

$$\lambda \sum_{b \in B} \sum_{p \in P_b} \bar{c}_{bp} g_{bp} + (1 - \lambda) \sum_{b \in B} \sum_{q \in Q_b} \sum_{w \in W} \bar{c}^w_{bq} h^w_{bq} \to \min \tag{4.23}$$

s.t. (4.2), (4.6), (4.7), (4.8), (4.12), (4.13), (4.14) and (4.15).

If weekday regularity is required, this model is modified as described in Section 4.4.2.

4.6 Evaluation Measures

Recall that in the model $\mathsf{SCHEDULE}_{\mathsf{MIP}}$, we use a center-based compactness measure in the objective function because other compactness measures, e.g., measures based on pairwise distances, are computationally intractable. For the a posteriori evaluation of solutions, we are, however, not restricted to measures that are suitable for a MIP model. Hence, we use this section to do some groundwork for our extensive experiments in the next section by proposing appropriate measures to evaluate and compare solutions to the MPSTDP-S.

We introduce the following notation to represent solutions to the MPSTDP-S. Let C^{day} denote the set of day clusters and $C^d \in C^{day}$ denote the day cluster of day $d \in D$, i.e., $C^d = \{b \in B : b \text{ is served on day } d\}$. Analogously, denote by C^{week} the set of week clusters and by $C^w \in C^{week}$ the week cluster of week $w \in W$, i.e., $C^w = \{b \in B : b \text{ is served in week } w\}$. A solution to the MPSTDP-S is represented by the set of day and week clusters $C = \{C^{day}, C^{week}\}$. Note that the day clusters would be sufficient to fully describe a solution since the week clusters can be derived from the day clusters. Nevertheless, we use this redundant representation because this allows us to keep the formulation of the evaluation measures simple.

4.6.1 Compactness Measures

In the context of the MPSTDP-S, compactness refers to the geographical distribution of customers within the week and day clusters. Clusters with geographically concentrated customers are considered more compact than clusters that span a large geographical area. There are many ways to quantify the concept of compactness. We decided to use measures based on pairwise distances since this seems to be the most intuitive approach for our problem. More precisely, we measure the average distance between any two customers that belong to the same week or day cluster. The lower this distance, the higher is the geographical concentration of the customers in the cluster.

To evaluate the geographical compactness of the week clusters of solution C, we define the measure $WComp(C)$:

$$WComp(C) = \frac{\sum\limits_{C^w \in C^{week}} \sum\limits_{b \in C^w} \sum\limits_{b' \in C^w, b \neq b'} c_{bb'}}{\sum\limits_{C^w \in C^{week}} |C^w|(|C^w| - 1)} \tag{4.24}$$

Analogously, we define $\mathrm{DComp}(C)$ to measure the geographical compactness of the day clusters of solution C:

$$\mathrm{DComp}(C) = \frac{\sum\limits_{C^d \in C^{day}} \sum\limits_{b \in C^d} \sum\limits_{b' \in C^d, b \neq b'} c_{bb'}}{\sum\limits_{C^d \in C^{day}} |C^d|(|C^d| - 1)} \tag{4.25}$$

4.6.2 Travel Time Measures

The main motivation behind the compactness objective is the fact that the service providers have to travel to their customers and that geographically concentrated clusters are assumed to reduce the overall travel time. To account for this aspect, we propose additional measures based on route lengths. Please note that we assume that all daily routes start and end at the service provider's depot (e.g., the office or home), although, in practice, there can be overnight stays, meaning that the service provider does not return to the depot after all customers of the day have been served.

To evaluate a solution in terms of travel time, we solve a symmetric traveling salesman problem (TSP) for each day of the planning horizon and add up the daily travel times. The TSP for each day is defined on a complete graph. The nodes for day $d \in D$ correspond to the customers that are scheduled for that day, $C^d \in C^{day}$, plus the service provider's depot, E. Each pair of nodes is connected via edges and the edge cost corresponds to the travel time between the nodes. Let $\theta(N)$ be the travel time of an optimal solution (i.e., shortest travel time, optimality gap of max. 1%) to the TSP with nodes N. Then, the total travel time, $TT(C, E)$, of a solution C with depot E is calculated as the sum of travel times of the daily routes:

$$TT(C, E) = \sum_{C^d \in C^{day}} \theta(C^d \cup \{E\}) \tag{4.26}$$

The time needed to travel from the depot to the first customer of the daily route and from the last customer of the route back to the depot can only be reduced significantly if customers nearby the depot are assigned to the day cluster, even when other customers of the day cluster are far from the depot. In this case, the travel time from/to the depot is artificially decreased at the cost of a reduced cluster compactness. Apart from this undesirable case, daily compactness mainly effects the travel time *within* the day cluster, i.e., the travel time between customers. The travel time from/to the depot is more or less constant. Thus, it is interesting to have a measure which only considers the proportion of the total travel time that is related to trips between customers. For this purpose, we introduce the measure $TT_{IC}(C, E)$, which describes the total intra-cluster (IC) travel time of a solution C with depot E. Let $\eta(N, E)$ denote the travel time of an optimal solution to the TSP with nodes N

minus the travel time associated with those edges of the solution that link the customers to the depot E. Then, measure $TT_{IC}(C, E)$ is defined as follows:

$$TT_{IC}(C, E) = \sum_{C^d \in C^{day}} \eta(C^d \cup \{E\}, E) \tag{4.27}$$

4.6.3 Balance Measures

Balance describes the requirement that the time needed to serve the customers should be evenly distributed throughout the planning horizon. This means that each day and each week should have roughly the same amount of service time. Perfect balance is achieved if the service time in each week is equal to the average weekly service time μ^{week}, and the service time on each day is equal to the average daily service time μ^{day}. As it is common in districting problems, we measure the maximum relative deviation from the average. We calculate the weekly balance, $WBal(C)$, and the daily balance, $DBal(C)$, of a solution C as follows:

$$WBal(C) = \max_{C^w \in C^{week}} \frac{|\chi(C^w) - \mu^{week}|}{\mu^{week}}, \tag{4.28}$$

$$DBal(C) = \max_{C^d \in C^{day}} \frac{|\xi(C^d) - \mu^{day}|}{\mu^{day}}, \tag{4.29}$$

where $\chi(C^w)$ is the service time that arises in week cluster $C^w \in C^{week}$, and $\xi(C^d)$ is the service time that arises in day cluster $C^d \in C^{day}$. The smaller the values of these measures, the more balanced we consider the solution.

4.7 Computational Experiments

We now present the results of extensive computational experiments. First, we report the results obtained from solving model $SCHEDULE_{MIP}$ on small test instances using the standard MIP solver Gurobi and derive some insights on the solution quality of our location-allocation heuristic. The main focus of this section is, however, on the evaluation of our location-allocation heuristic on test instances of realistic size. For this purpose, we develop an experimental design which covers a wide range of parameter values and problem characteristics. Since, for these realistic instance sizes, model $SCHEDULE_{MIP}$ cannot be solved by a standard MIP solver in a reasonable time, we benchmark our approach against the PTV xCluster Server (PTV, 2014), a commercial software product for scheduling customer visits. Additionally, we perform experiments to examine the impact of different types of weekday regularity on the travel time of the location-allocation solutions as well as on the running time behavior of the location-allocation heuristic, and we present a small extract of the solutions on a map.

4.7.1 Optimality Gap on Small Instances

As already mentioned in Section 4.4.3, we tried to compute optimal solutions to ten 50-customer test instances. The planning horizon for each instance consisted of four weeks and five days per week. We used the MIP solver Gurobi, warm started with the location-allocation solution, to solve model $\mathsf{SCHEDULE}_{\mathsf{MIP}}$. Gurobi could not find a (proven) optimal solution to any of the ten instances within a time limit of ten hours.[1] Hence, we do not know exactly how far the solutions of the location-allocation heuristic are from the optimal solutions. We can, however, compare the solutions of the location-allocation heuristic with the best incumbent and the best lower bound found by Gurobi for each test instance to obtain a range for the gap between the location-allocation solutions and the optimal solutions. We found out that the location-allocation solutions are, on average, 3.0% worse than the best incumbent found for each instance by Gurobi. On the other hand, the objective values of the location-allocation solutions are, on average, 8.0% higher than the best lower bound found by Gurobi. This means that the location-allocation approach produces high-quality solutions with an average optimality gap between 3.0% and 8.0%. The average runtime of the location-allocation approach was 4.6 seconds.

To provide a comparison with known optimal solutions, we briefly report in the following the results we obtain on the three 30-customer instances that could be solved optimally within one hour.[2] The optimality gaps for the location-allocation heuristic on these instances are 4.2%, 6.0%, and 7.3%. This means that high-quality solutions with an average optimality gap of 5.9% are found. The average running time per instance was 0.3 seconds.

4.7.2 Experimental Design

For the evaluation of the location-allocation heuristic we use 20 real-world instances provided by PTV. The data describe the planning task arising at a manufacturer of fast moving consumer goods whose sales force has to visit retailers, such as supermarkets and gas stations, on a regular basis. Each instance contains the service provider's depot and, on average, $|B| = 115$ customers. The customers' week rhythms, r_b, $b \in B$, are from the set $\{1, 2, 4, 8, 16\}$, which implies a planning horizon of $|W| = 16$ weeks. Each week consists

[1] Gurobi version 6.0.2 was used for these tests. The tests were performed on a machine with an Intel Xeon E5-2650 v2 CPU with eight cores, running at 2.6 GHz, and 128 GB of RAM.

[2] Gurobi version 6.0.5, Intel Core i5-760, four cores at 2.8 GHz, 8 GB of RAM.

of $m = 5$ days. All customers must be visited exactly once per visiting week, i.e., $n_b = 1$ for all customers $b \in B$. The weekdays on which visits may take place are not restricted, i.e., each weekday represents a valid weekday pattern $p \in P_b$ for all customers $b \in B$. The customers do not have weekday regularity requirements. Their service times (in minutes), t_{bj}, $b \in B$, $j \in \{1, ..., \frac{|W|}{r_b} n_b\}$, are from the set {22, 28, 34, 39, 42}, and each visit of a customer takes the same amount of time, i.e., $t_{bj} = t_{bk}$, for all $b \in B$, $j, k \in \{1, ..., \frac{|W|}{r_b} n_b\}$.

In order to test our location-allocation heuristic under many diverse conditions, we generated additional test instances by modifying some parameters of the original real-world instances. The parameters we modified are the weekday regularity, the week rhythms, the number of visits per visiting week, and the service times (see Table 4.3 for a summary of the different parameter values that are covered by our test instances):

- *Weekday regularity:* First of all, we generated instances with strict weekday regularity for all customers as well as instances with partial weekday regularity for all customers. In the case of partial weekday regularity, we allowed one deviation from the regular weekday pattern, but we required that more than half of the visiting weeks of each customer must follow the regular weekday pattern. This means that all customers with at least three visiting weeks are allowed to deviate once from the regular weekday pattern, whereas all other customers are not allowed to deviate.

- *Week rhythms / Number of weeks:* With respect to the week rhythms, we generated instances in which all weekly customers of the original instances were changed to customers with a week rhythm of eight, and all biweekly customers of the original instances were changed to customers with a week rhythm of 16. This yields {4, 8, 16} as the set of week rhythms and a planning horizon of 16 weeks. Furthermore, we generated instances in which the week rhythms were randomly drawn from the set {3, 4, 6, 12, 16} with probabilities 15%, 20%, 30%, 20%, and 15%, respectively, resulting in a planning horizon of 48 weeks.

- *Number of visits per visiting week:* Concerning the number of visits per visiting week, we generated additional instances in which the number of visits per week were picked uniformly at random from the set {1, 2, 3}. Multiple visits per visiting week were, however, only eligible for weekly customers, since, from a practical point of view, it does not appear to make sense to serve non-weekly customers multiple times per visiting week. As in the original data, we assumed that there are no restrictions with respect to the combinations of weekdays on which visits may take place, i.e., the set of weekday patterns, P_b, $b \in B$, comprises all combinations of weekdays for which the number of contained weekdays equals the number of visits per visiting week.

Table 4.3: Parameter values covered by the test instances

Parameter	Values
Weekday regularity	no regularity, partial regularity (1 deviation allowed), strict regularity
Week rhythms / Number of weeks in planning horizon	{1, 2, 4, 8, 16} / 16, {4, 8, 16} / 16, {3, 4, 6, 12, 16} / 48
Number of visits per visiting week	{1}, {1, 2, 3}[1]
Service times (minutes)	{22, 28, 34, 39, 42}, {15, 20, ..., 55, 60}

[1] Only in combination with week rhythms {1, 2, 4, 8, 16}

- *Service times:* Finally, we generated additional instances by modifying the service times. For each visit of a customer, we picked a service time uniformly at random from the set {15, 20, ..., 55, 60}.

We choose a full factorial design, i.e., we consider all combinations of the above mentioned parameter values for all of the original 20 test instances. This yields, in total, 480 test instances. Using these instances, we perform computational experiments to compare the performance of the location-allocation heuristic with that of the PTV xCluster Server (PTV, 2014). Furthermore, we perform additional experiments to gain insights into the effect of weekday regularity on the travel time of the solutions as well as on the running time behavior of our algorithm.

4.7.3 Implementation Details and Parametrization

In the presence of partial weekday regularity requirements, the integer linear program $\mathsf{ALLOC}_{\mathsf{MIP}}$ must be modified as explained in Section 4.4.2. Additional variables and constraints must be added to $\mathsf{ALLOC}_{\mathsf{MIP}}$, which makes it harder to find a feasible solution. To speed up the solution process, we first use the location-allocation algorithm to solve an auxiliary problem. In this auxiliary problem, all partial weekday regularity requirements are replaced by strict weekday regularity requirements, which, instead of introducing additional variables and constraints, leads to a reduction of the number of variables in model $\mathsf{ALLOC}_{\mathsf{MIP}}$. Note that the solution to the auxiliary problem is feasible for the original problem. Therefore, we use this solution to warm start the location-allocation algorithm on the original problem.

Both the location-allocation heuristic and the PTV xCluster Server (PTV, 2014) were run on a Windows 7 machine with 8 GB of RAM and an Intel Core i5-760 at a clock rate of 2.8 GHz. The location-allocation heuristic was coded in Java, and Gurobi 6.0.5 was used to solve model $\mathsf{ALLOC}_{\mathsf{MIP}}$. For all tests, the Gurobi MIP gap parameter was set to 1%, which we consider sufficiently small for all practical applications. Moreover, the maximum time spent by Gurobi on solving the integer program in Step 2 of the algorithm was limited to 15 seconds. The maximum number of location-allocation iterations was set to $iter_{max} = 20$, which did not impose a restriction for the vast majority of the test instances in our experiments. In combination with the time limit of 15 seconds for the solution of the integer program, the maximum runtime of our heuristic is limited to five minutes per instance, which is according to our experiences with our industry partner PTV an acceptable computation time for human planners. If the objective function value did not improve by more than 0.1% compared to the previous iteration, the algorithm terminated early. The user parameter λ in Objective Function (4.23) was set to 0.33.

Depending on the focus of the experiments, we set the values of the balance tolerance parameters τ^{week} and τ^{day} differently. In Section 4.7.4, we compare the performance of the location-allocation heuristic and the PTV xCluster Server (PTV, 2014). For a fair comparison, we make sure that for all test instances the balance achieved with the location-allocation heuristic is at least as good as the balance of the PTV xCluster solution. To this end, we first solve each test instance with the PTV xCluster Server, and then use the values obtained for the weekly and daily service time balance as the values for the balance tolerance parameters of the location-allocation heuristic. As a consequence, all test instances in Section 4.7.4 are solved with different values for the balance tolerance. In Sections 4.7.5 to 4.7.7, we focus on the impact of different types of weekday regularity on the travel time of the location-allocation solutions and on the running time behavior of the location-allocation heuristic. To guarantee the comparability of the results from this analysis, the same balance tolerance must be used for all instances. Therefore, we choose $\tau^{week} = 15\%$ as the weekly balance tolerance and $\tau^{day} = 30\%$ as the daily balance tolerance for all experiments in Sections 4.7.5 to 4.7.7.

4.7.4 Comparison with PTV xCluster Server

Since the MPSTDP-S cannot be solved by a standard MIP solver for realistic instance sizes, we use the PTV xCluster Server version 1.18.1.3 (PTV, 2014) as the benchmark for the location-allocation heuristic. PTV xCluster Server uses a local search to determine a visit schedule that is valid with respect to the customers' visiting requirements. The optimization criteria of the local search are compactness and balance. At the beginning of the local search, the focus of the optimization is on improving compactness. During the course of the optimization, the focus shifts to the improvement of balance. Two types of moves are considered, namely the relocation of a customer to a different week or day cluster and the exchange of the week or day clusters of two customers. The algorithm terminates after a user-specified number of iterations or if no more improvements are found.

Remember that, for a better comparability of the location-allocation approach and the PTV xCluster Server (PTV, 2014), we set the balance tolerances, τ^{week} and τ^{day}, of the location-allocation heuristic to the actual service time balance of the xCluster solutions. Table 4.4 shows the average results of the two approaches with respect to compactness and travel time, grouped according to different types of weekday regularity. The first eight columns contain the average absolute values. $DComp$ and $WComp$ are measured in kilometers, TT and TT_{IC} are measured in hours. The last four columns show the relative deviation between the location-allocation solutions and the xCluster solutions with respect to the four measures. The relative deviation between the location-allocation solution $C_{LocAlloc}$ and the corresponding xCluster solution $C_{xCluster}$ on measure M is computed as

$$Dev(C_{LocAlloc}, C_{xCluster}, M) = \frac{M(C_{LocAlloc}) - M(C_{xCluster})}{M(C_{xCluster})}.$$

Hence, a negative deviation means that the location-allocation solution is better than the xCluster solution with respect to measure M. In the table, these deviations are averaged over all test instances of a row.

The results show that the location-allocation approach clearly outperforms the PTV xCluster Server (PTV, 2014) in all four compactness and travel time measures. With respect to measure $DComp$, the location-allocation solutions are, on average, 26.26% better than the xCluster solutions. Measure $WComp$ is improved by 13.47% compared to the xCluster solutions. The total travel time TT is reduced, on average, by 15.36 hours, the intra-cluster travel time TT_{IC} by 20.46 hours, which translates into relative improvements of 6.55% and 18.74%, respectively. It is noticeable that the reduction in the total travel time TT is smaller than the reduction of the intra-cluster travel time TT_{IC}. This means that the travel time between the depot and the day clusters increases compared to the xCluster solutions, but this increase is overcompensated by improvements of the intra-cluster travel time TT_{IC}. A possible explanation for this effect are outliers in the xCluster solutions, i.e., single customers that are relatively far from the other customers of a day cluster. Such outliers are, in some cases, produced by xCluster in an attempt to improve the balance of a solution. They can lead to a reduced travel time between the depot and the day cluster at the cost of intra-cluster compactness.

It can further be seen from Table 4.4 that, the higher the degree of freedom in terms of weekday regularity, the higher is the improvement of the location-allocation solutions over the xCluster solutions. For example, the average relative improvement on measure $DComp$ is 22.34% in the case of strict weekday regularity. When weekday regularity is relaxed to partial and none, the improvement increases to 25.55% and 30.88%, respectively. Similar effects can be observed for measures TT and TT_{IC}. Only on measure $WComp$ are the values almost the same for all three types of weekday regularity.

Table 4.4: Comparison between location-allocation approach and xCluster (PTV, 2014): Average compactness and travel time grouped by the three types of weekday regularity

Weekday regularity	Location-allocation				PTV xCluster Server				Relative deviation between location-allocation and xCluster			
	$DComp$	$WComp$	TT	TT_{IC}	$DComp$	$WComp$	TT	TT_{IC}	$DComp$	$WComp$	TT	TT_{IC}
None	7.94	22.79	223.83	87.85	11.28	26.05	239.58	111.25	-30.88%	-13.42%	-7.18%	-22.42%
Partial	8.66	23.55	227.12	96.53	11.44	27.17	243.61	117.06	-25.55%	-13.74%	-6.72%	-18.14%
Strict	9.01	23.68	229.76	99.59	11.44	27.17	243.61	117.06	-22.34%	-13.25%	-5.74%	-15.66%
Average	**8.53**	**23.24**	**226.90**	**94.66**	**11.39**	**26.80**	**242.26**	**115.12**	**-26.26%**	**-13.47%**	**-6.55%**	**-18.74%**

Table 4.5: Comparison between location-allocation approach and xCluster (PTV, 2014): Relative compactness and travel time deviation grouped by the different sets of week rhythms and planning horizons

Week rhythms / Number of weeks in planning horizon	Relative deviation between location-allocation and xCluster			
	$DComp$	$WComp$	TT	TT_{IC}
{1, 2, 4, 8, 16} / 16	-20.19%	+0.49%	-5.70%	-13.47%
{4, 8, 16} / 16	-37.55%	-39.69%	-8.91%	-26.57%
{3, 4, 6, 12, 16} / 48	-27.10%	-15.17%	-5.89%	-21.46%
Average	**-26.26%**	**-13.47%**	**-6.55%**	**-18.74%**

Table 4.5 provides a different view of the same results by grouping the relative deviation between the two approaches according to the three different sets of week rhythms and associated planning horizons. The location-allocation heuristic clearly beats xCluster (PTV, 2014) in all dimensions except one. When weekly customers are present, the $WComp$ values of the location-allocation approach and xCluster are nearly identical. This can be explained by the fact that the weekly customers force the service provider to travel almost across the whole service territory in every week, which leads to very similar solutions in terms of weekly compactness. In the cases without weekly customers, the location-allocation approach is able to produce solutions that have a significantly higher weekly compactness than the xCluster solutions.

The average weekly and daily balance values, $WBal$ and $DBal$, are reported in Table 4.6. Remember that the balance tolerances τ^{week} and τ^{day} of the location-allocation approach were set to the actual balance values of the xCluster solutions. Consequently, the balance values of the two approaches are almost the same, with the location-allocation solutions having a slightly better balance.

Table 4.7 contains the average and maximum running times per instance in seconds. The location-allocation approach has significantly longer running times than the PTV xCluster

Table 4.6: Comparison between location-allocation approach and xCluster (PTV, 2014): Average service time balance (in percent)

Weekday regularity	Location-allocation		PTV xCluster Server	
	$DBal$	$WBal$	$DBal$	$WBal$
None	45.69	11.31	46.03	12.02
Partial	21.40	7.10	21.46	7.48
Strict	21.36	7.01	21.46	7.48
Average	**29.48**	**8.47**	**29.65**	**8.99**

Table 4.7: Comparison between location-allocation approach and xCluster (PTV, 2014): Average and maximum running time (in seconds)

Weekday regularity	Location-allocation		PTV xCluster Server	
	Average	Max	Average	Max
None	14.54	103.07	14.80	73.40
Partial	41.57	156.26	7.23	33.60
Strict	25.94	109.27	6.96	32.41
Avg/Max	**27.35**	**156.26**	**9.66**	**73.40**

Server (PTV, 2014). With an average of approximately 27 seconds, the location-allocation running times are almost three times as high as those of xCluster. However, one has to keep in mind that the MPSTDP-S is a tactical planning problem, which has to be solved only every few months. In such a tactical context, the location-allocation running times are completely acceptable. In fact, rather than having very short running times, solution quality is of utmost importance in practice since high-quality solutions can prevent the necessity of manual post-processing by a human planner.

4.7.5 The Cost of Weekday Regularity

In practice, many customers appreciate weekday regularity because it leads to a reduction in the time needed for coordination and to an increase in efficiency. However, enforcing partial or strict weekday regularity means that the solution space is restricted compared to the situation without weekday regularity. One would expect that such a restriction leads to a deterioration in the compactness and the travel time of the solutions produced by the location-allocation approach. In this section, we investigate this "cost of weekday regularity". Concretely, we analyze the increase in travel time (measure TT) when weekday regularity is imposed relative to the situation without weekday regularity. Remember that we choose $\tau^{week} = 15\%$ and $\tau^{day} = 30\%$ for all experiments in this section.

Table 4.8: Cost of weekday regularity for the two types of service times measured as the increase in travel time relative to the case without weekday regularity

Weekday regularity	Service times		
	Original	Randomly picked	**Average**
Partial	+1.27%	+6.08%	**+3.68%**
Strict	+1.82%	+8.88%	**+5.35%**

Table 4.8 contains the cost of weekday regularity for the two different types of service times considered in the test instances. On average over all 480 test instances, we observe a 3.68% increase in travel time when partial weekday regularity (max. one deviation from the regular weekday pattern) is enforced. In the case of strict weekday regularity, the total travel time is increased by 5.35%. A more detailed analysis shows that the cost of weekday regularity differs greatly depending on the values of the service times and week rhythms. The cost of weekday regularity is modest for instances with original service times: 1.27% in the case of partial weekday regularity and 1.82% in the case of strict weekday regularity. For the randomly generated service times, the cost of weekday regularity is much higher: It amounts to 6.08% and 8.88%, respectively. This result can be explained as follows. Remember that in the original real-world data all service times are from the set {22, 28, 34, 39, 42} and the same service time is incurred for each visit of the same customer. In our randomly generated test instances, the service time for each customer visit is randomly drawn from the set {15, 20, ..., 55, 60}, i.e., the service times may vary between different visits of the same customer. For example, a customer may require a 15-minute service on the first visit, a 60-minute service on the second visit and a 35-minute service on the third visit. Moreover, the range of the randomly drawn service times is more than twice as high as the range of the original service times. This means that there is more variability in the randomly drawn service times than in the original service times. When weekday regularity is imposed, the higher variability of the randomly generated instances leads to a greater increase in travel time.

Table 4.9 shows the cost of weekday regularity for the three types of week rhythms. Again, huge differences in the impact of weekday regularity can be observed. When week rhythms are from the sets {1, 2, 4, 8, 16} and {4, 8, 16}, the cost of weekday regularity is marginal (and even negative in one case). On the other hand, when the week rhythms are from the set {3, 4, 6, 12, 16}, weekday regularity leads to a significant increase in travel time of up to 18.51%. In the first two cases, all week rhythms are a power of two and, consequently, higher week rhythms are an integer multiple of smaller week rhythms. This facilitates the balancing of service times. The week rhythms in the third case do not have this beneficial

Table 4.9: Cost of weekday regularity for the three types of week rhythms measured as the increase in travel time relative to the case without weekday regularity

Weekday regularity	Week rhythms		
	{1, 2, 4, 8, 16}	{4, 8, 16}	{3, 4, 6, 12, 16}
Partial	+0.59%	-0.17%	+13.69%
Strict	+1.06%	+0.78%	+18.51%

property. Thus, the restrictions that go along with the introduction of weekday regularity cannot be compensated as easily as in the case of more favorable week rhythms.

In summary, we observed that enforcing weekday regularity leads to an increase in travel time. However, the extent of the increase is different under different circumstances. In our experiments, we identified the service times and the week rhythms as the major influencing factors on the cost of weekday regularity.

4.7.6 Running Time Analysis

Based on the experiments of Section 4.7.5, we now investigate the running time behavior of the location-allocation approach. The average and maximum running times are listed in Table 4.10, grouped according to different types of weekday regularity and week rhythms. The average running time over all test instances is roughly 28 seconds, the maximum running time is 280 seconds.

None and strict weekday regularity yield very similar running times of approximately 22 seconds on average and 140 seconds at the maximum. In contrast, partial weekday regularity results in significantly longer running times of 40 seconds on average and 280 seconds at the maximum. The reason for this is that we need to adopt a more involved procedure when partial weekday regularity requirements are present than in the other two cases. The additional variables and constraints that must be introduced to the model (see Section 4.4.2) make it hard for the MIP solver to find an initial feasible solution. Therefore, we perform two runs of the location-allocation heuristic consecutively (see Section 4.7.3). We first solve an auxiliary problem with strict weekday regularity and then take this solution to warm start the location-allocation heuristic for the problem with partial weekday regularity. This two-stage procedure is obviously more time-consuming than performing just a single run of the location-allocation heuristic as in the other two cases.

Regarding the week rhythms and the resulting planning horizons, one can see that the 48-week planning horizon results in considerably longer running times than the 16-week

Table 4.10: Running times of the location-allocation approach (in seconds)

Weekday regularity	Week rhythms / Number of weeks in planning horizon							
	{1, 2, 4, 8, 16} / 16		{4, 8, 16} / 16		{3, 4, 6, 12, 16} / 48			
	Average	Max	Average	Max	Average	Max	**Avg/Max**	
None	11.17	50.95	10.36	57.80	54.66	141.10	**21.84**	**141.10**
Partial	13.18	174.09	15.90	47.67	118.00	280.04	**40.06**	**280.04**
Strict	8.94	97.56	13.14	35.50	57.99	139.96	**22.25**	**139.96**
Avg/Max	**11.09**	**174.09**	**13.13**	**57.80**	**76.88**	**280.04**	**28.05**	**280.04**

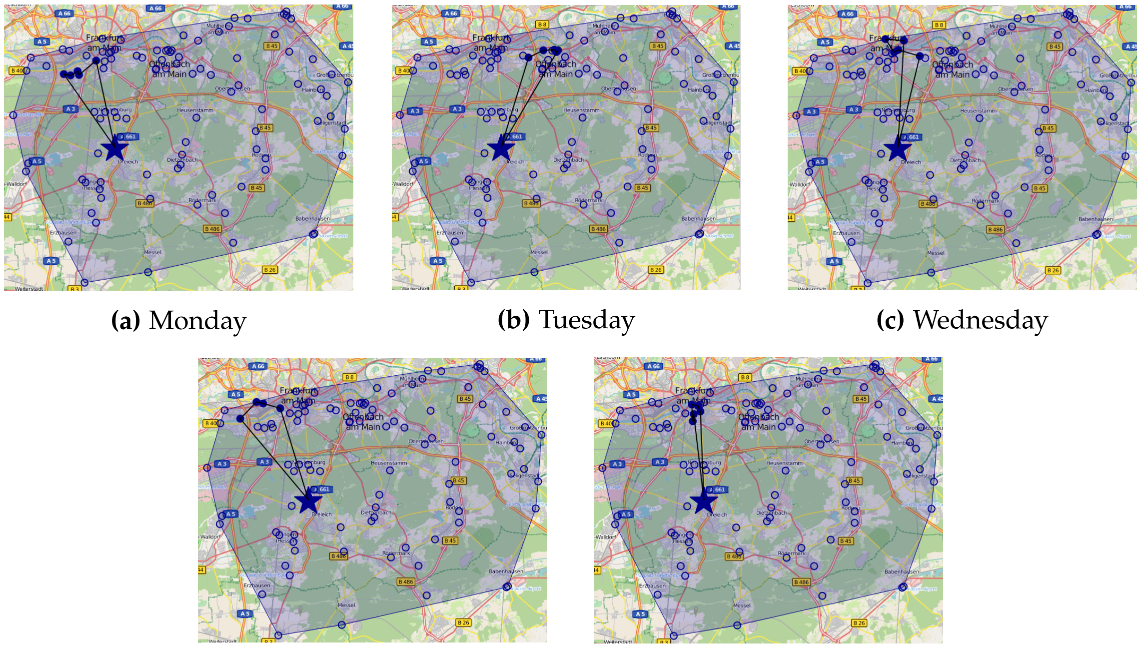
planning horizons (on average 77 seconds vs. 11 and 13 seconds, respectively). The running times for week rhythms {1, 2, 4, 8, 16} and {4, 8, 16}, both with a planning horizon of 16 weeks, are very similar.

4.7.7 Visualization of Results

To give a visual impression of the solutions obtained with the location-allocation approach, we visualize the day clusters for the five working days of an exemplary week in Figure 4.7. The big star represents the service provider's depot, the circles represent the customers. A filled circle means that the customer must be served on that particular day, whereas an empty circle stands for a customer without a service request. The solid lines indicate the service provider's routes, which have been calculated a posteriori by solving a TSP for each day cluster. The darker area represented by the convex hull of the customers is the entire service territory, i.e., the region for which the service provider is responsible. The figure shows that the location-allocation approach produces geographically compact day clusters. Furthermore, all day clusters of the week are within a relatively small sub-area of the service territory, meaning that also a good weekly compactness could be achieved.

4.8 Conclusions

In this chapter, we introduced the multi-period service territory design problem. To the best of our knowledge, this problem has not been treated before in the literature, although its practical relevance is high. The MPSTDP combines two subproblems, namely a partitioning subproblem and a scheduling subproblem. Since the partitioning subproblem corresponds to the well-known (classical) territory design problem, we laid the emphasis of this chapter on the scheduling subproblem. We formulated the scheduling subproblem



(a) Monday **(b)** Tuesday **(c)** Wednesday

(d) Thursday **(e)** Friday

Figure 4.7: Day clusters and corresponding TSP routes for the five working days of an exemplary week (map data © OpenStreetMap contributors)

as a mixed integer linear program. Due to the great number of variables and the high symmetry, it is – even on small instances – not possible to solve this formulation to optimality using a standard MIP solver. Therefore, we proposed a location-allocation heuristic. Extensive experiments on real-world instances and on instances derived from real-world data have shown that this heuristic produces high-quality solutions in reasonable running times. Our heuristic clearly outperforms the PTV xCluster Server version 1.18.1.3 (PTV, 2014) in terms of solution quality. This lead PTV to replace their existing algorithm with an algorithm based on the presented location-allocation heuristic in version 2.1.0 of their xCluster Server, which was released in December 2016. Furthermore, we examined the cost of weekday regularity, i.e., the increase in travel time when partial or strict weekday regularity is introduced. We found out that the cost of weekday regularity depends to a great extent on the characteristics of the test instances. The variability of the service times and the compatibility of the week rhythms have turned out to be the main influencing factors.

5

A Branch-and-Price Algorithm for the Scheduling of Customer Visits in the Context of Multi-Period Service Territory Design

As we have seen in the previous chapter, a problem that arises in the context of multi-period service territory design is the scheduling of customer visits. In this problem, customer visits must be assigned to the days of the planning horizon subject to customer-specific requirements. Now we consider a highly relevant planning scenario of this problem and present an exact branch-and-price algorithm. We propose specialized acceleration techniques, particularly a fast pricing heuristic and techniques to reduce the symmetry inherent to the problem, and we evaluate the algorithm on real-world data sets.

This chapter is based on the following article:

> Bender, M., Kalcsics, J., Nickel, S., Pouls, M. A branch-and-price algorithm for the scheduling of customer visits in the context of multi-period service territory design. Submitted to *European Journal of Operational Research* in February 2017; under revision as of June 2017.

5.1 Introduction

Classical service territory design problems consist of grouping customers into larger clusters, which are called territories or districts, such that some relevant planning criteria are met (Kalcsics, 2015). In each district, a service provider, e.g., a salesperson or service technician, is responsible for providing services at the customers' sites. In many cases, these services must be provided several times during a given planning horizon, which extends the classical problem to a multi-period setting. The resulting problem, the multi-period service territory design problem (MPSTDP), has been introduced in Chapter 4. One of the subproblems that arises in the MPSTDP is the scheduling subproblem MPSTDP-S. In this subproblem, the districts are already given and customer visits need to be scheduled for each district individually. In this chapter, we consider a highly relevant planning scenario of the MPSTDP-S, which we denote by MPSTDP-S*. It can formally be described as follows.

Given a planning horizon consisting of weeks $W = \{1, ..., |W|\}$ and days $D = \{1, ..., |D|\}$, and given the set of customers $B = \{1, ..., |B|\}$ of a district, the task is to assign customer visits to the weeks and days of the planning horizon. Each customer $b \in B$ must receive on-site service by the service provider who is responsible for the district, and the service must be provided according to a customer-specific *week rhythm* $r_b \in \mathbb{N}^+$, which means that each customer must be visited regularly every r_b weeks, with the first service taking place in the first r_b weeks of the planning horizon. The number of weeks $|W|$ in the planning horizon is defined as the least common multiple of week rhythms $\{r_b\}_{b \in B}$. Each service of a customer requires a *service time* $t_b \in \mathbb{R}^+$. In order to balance the service provider's workload over the time periods of the planning horizon, the total service time on each day must be within the interval $\left[LB^{day}, UB^{day}\right]$, and the total service time in each week is limited to the interval $\left[LB^{week}, UB^{week}\right]$, where LB^{day}, UB^{day}, LB^{week} and UB^{week} denote appropriate minimum and maximum cumulative service times. The distance from customer $b \in B$ to customer $b' \in B$ is given by $c_{bb'} \in \mathbb{R}^+$. In order to reduce the travel time required for serving the customers, the objective is to schedule the customer visits in such a way that customers who are served on the same day or in the same week are geographically close to each other. More precisely, the objective is to minimize the sum of the distances between all customers that are served in the same time period (day or week) and a customer that is selected as the center for that time period. We adopt the terminology of Chapter 4 and call the latter customers *day centers* and *week centers*. Note that a week center does not have to be served in the week it acts as the week center. This applies analogously to day centers. Furthermore, we denote the subsets of customers that are served on the same day or in the same week as *day clusters* and *week clusters*, respectively.

One might argue that, rather than striving for geographically compact day and week clusters, the daily route lengths should be optimized. However, since service visits might have to be rescheduled in day-to-day business (e.g., due to short-term customer requests), explicitly considering routing decisions is only of little use. Moreover, geographically compact clusters provide a high degree of flexibility to cope with short-term customer requests and other unexpected events in day-to-day operations. A detailed discussion on these aspects is provided in Section 4.2.

Compared to the MPSTDP-S studied in Chapter 4, the MPSTDP-S* contains the following assumptions. As opposed to the MPSTDP-S, the MPSTDP-S* does not consider the possibility that a customer demands more than one service per week. We assume in the MPSTDP-S* that there are no restrictions with respect to the days on which a customer can be served, whereas the MPSTDP-S provides the opportunity to take into account customer-specific weekday patterns, which can be used to restrict service to particular combinations of weekdays. Moreover, we assume in the MPSTDP-S* that always the same service time t_b is incurred for customer $b \in B$, while the MPSTDP-S allows the specification of different service times for each visit of a customer. These assumptions hold for the majority of the real-word projects of our industry partner PTV Group (PTV), a commercial provider of districting and clustering software. Hence, we study a highly relevant planning scenario of the MPSTDP-S.

For a review of related problems, we refer the reader to Section 4.3. Since the problem under study has been introduced only recently, no specialized exact solution methods have been proposed yet. However, we are aware of three papers that use column generation for similar problems. Mehrotra et al. (1998) study a single-period political districting problem and propose a branch-and-price based heuristic. The master problem corresponds to a set-partitioning problem with an additional constraint enforcing the required number of territories. The objective is to optimize compactness. Each column in the master problem represents a feasible territory, i.e., a territory which is contiguous and balanced in terms of population. Accordingly, the pricing problems correspond to two-sided knapsack problems with contiguity constraints. The authors incorporate some heuristic elements to increase computational efficiency, e.g., simplified contiguity constraints and distance-based variable fixing. de Fréminville et al. (2015) deal with a special single-period districting problem which they call the financial product districting problem. In this problem, customers must be partitioned into territories such that the expected customer-dependent cost price of a financial product is relatively the same for all customers that belong to the same territory. The authors formulate the master problem as a set-partitioning problem with additional side constraints. They aim at minimizing a weighted sum of the cost price variances within the territories. Each column corresponds to a feasible territory, which means that it must be contiguous and contain a given minimum number of customers. As the reduced cost of a column includes the cost price variance, the objective function of the pricing problem is nonlinear. The authors propose a greedy multi-start heuristic to solve the pricing problem and two heuristic procedures to determine an integer solution to the master problem. Mourgaya and Vanderbeck (2007) study a tactical variant of the period vehicle routing problem. The objective is to obtain geographically compact clusters for each time period and vehicle, and to balance workload between vehicles. In the master problem of their column generation reformulation, clusters, i.e., subsets of customers whose workload does not exceed a given upper bound, are selected for the time periods of the planning horizon. The authors propose a greedy insertion heuristic to solve the pricing problems, which correspond to quadratic knapsack problems. They alternately solve the linear programming (LP) relaxation of the restricted master problem and fix some of the variables to construct an integer solution. The problems studied by Mehrotra et al., de Fréminville et al., and Mourgaya and Vanderbeck differ from our problem in the following aspects: The problems tackled by Mehrotra et al. and de Fréminville et al. consider a single-period setting where each customer must be assigned to exactly one territory. Furthermore, contiguity is explicitly required in both problems. In contrast to this, we deal with a multi-period problem in which customers have to be assigned to multiple clusters, and we do not consider contiguity as a relevant planning criterion. Moreover, geographical compactness, which is the objective in our problem, is not taken into account by de Fréminville et al. In the problem studied by Mourgaya and Vanderbeck, geographical compactness is relevant only with respect to one time scale (days), whereas we consider geographical compactness with respect to two time scales (days and weeks). Finally, in terms of solution methodology, the authors of the three papers propose heuristics, whereas we strive for the development of an exact method.

The main contributions of this chapter are as follows:

- We are the first to present an exact branch-and-price algorithm for the scheduling task of the MPSTDP.
- We propose specially-tailored techniques to speed up the algorithm, such as a fast greedy heuristic to solve the pricing problems and techniques to reduce the symmetry inherent to the MPSTDP-S*.
- We show the effectiveness of our algorithm through extensive computational experiments on real-world instances and investigate the impact of individual algorithmic features. Instances with up to 55 customers can be solved to optimality in reasonable running times.

- Compared to solving the compact formulation of the MPSTDP-S* with a general purpose mixed integer programming (MIP) solver, we achieve an average reduction in running time of more than 98.1%.

The remainder of this chapter is organized as follows. In Section 5.2, we present a compact linear integer programming (IP) model for the MPSTDP-S*. This model is reformulated in Section 5.3 into a master problem and several pricing problems, which serve as the basis for our branch-and-price algorithm. Moreover, we introduce some definitions and basic concepts about symmetry in this section. In Section 5.4, we present the details of our algorithm, including specialized techniques that aim at reducing running time. In Section 5.5, we report the results of extensive experiments on real-word test instances, which prove the effectiveness of the proposed algorithm. Finally, we provide a short conclusion in Section 5.6.

5.2 A Compact Formulation

In this section, we present a compact IP formulation for the MPSTDP-S*. It is based on the formulation of Section 4.4, but adapts this formulation to the planning scenario studied in this chapter. We introduce the following additional notation. Let $D(w) \subset D$ represent the days in week $w \in W$, and denote by $\lambda \in [0, 1]$ a user parameter to weight the importance of compact week clusters versus compact day clusters. 1 (0) means that the compactness of day clusters (week clusters) is irrelevant to the user, intermediate values represent trade-offs between the two extremes. Furthermore, define the following decision variables:

$$u_{ib}^{w} = \begin{cases} 1 & \text{if customer } b \in B \text{ is served in week } w \in W \text{ and assigned to week} \\ & \text{center } i \in B \\ 0 & \text{otherwise} \end{cases}$$

$$v_{ib}^{d} = \begin{cases} 1 & \text{if customer } b \in B \text{ is served on day } d \in D \text{ and assigned to day center} \\ & i \in B \\ 0 & \text{otherwise} \end{cases}$$

$$x_{b}^{w} = \begin{cases} 1 & \text{if customer } b \in B \text{ is the week center in week } w \in W \\ 0 & \text{otherwise} \end{cases}$$

$$y_{b}^{d} = \begin{cases} 1 & \text{if customer } b \in B \text{ is the day center on day } d \in D \\ 0 & \text{otherwise} \end{cases}$$

Using this notation, the MPSTDP-S* can be modeled as the following compact IP, which we denote by (COMP):

$$
\begin{alignat}{3}
&(\mathrm{COMP}) \quad && \lambda \sum_{b \in B} \sum_{i \in B} \sum_{w \in W} c_{ib} u_{ib}^{w} + (1-\lambda) \sum_{b \in B} \sum_{i \in B} \sum_{d \in D} c_{ib} v_{ib}^{d} \quad \rightarrow \min && && (5.1)\\
&\text{s.t.} && \sum_{i \in B} \sum_{w \in W, w \leqslant r_b} u_{ib}^{w} = 1 && b \in B && (5.2)\\
& && \sum_{i \in B} u_{ib}^{w} = \sum_{i \in B} u_{ib}^{((w-1) \bmod r_b)+1} && b \in B, w \in W, w > r_b && (5.3)\\
& && u_{ib}^{w} \leqslant x_{i}^{w} && b, i \in B, w \in W && (5.4)\\
& && \sum_{b \in B} x_{b}^{w} = 1 && w \in W && (5.5)\\
& && \sum_{b \in B} \sum_{i \in B} t_b u_{ib}^{w} \geqslant LB^{week} && w \in W && (5.6)\\
& && \sum_{b \in B} \sum_{i \in B} t_b u_{ib}^{w} \leqslant UB^{week} && w \in W && (5.7)\\
& && \sum_{i \in B} \sum_{d \in D(w)} v_{ib}^{d} = \sum_{i \in B} u_{ib}^{w} && b \in B, w \in W && (5.8)\\
& && v_{ib}^{d} \leqslant y_{i}^{d} && b, i \in B, d \in D && (5.9)\\
& && \sum_{b \in B} y_{b}^{d} = 1 && d \in D && (5.10)\\
& && \sum_{b \in B} \sum_{i \in B} t_b v_{ib}^{d} \geqslant LB^{day} && d \in D && (5.11)\\
& && \sum_{b \in B} \sum_{i \in B} t_b v_{ib}^{d} \leqslant UB^{day} && d \in D && (5.12)\\
& && u_{ib}^{w} \in \{0,1\} && b, i \in B, w \in W && (5.13)\\
& && v_{ib}^{d} \in \{0,1\} && b, i \in B, d \in D && (5.14)\\
& && x_{b}^{w} \in \{0,1\} && b \in B, w \in W && (5.15)\\
& && y_{b}^{d} \in \{0,1\} && b \in B, d \in D && (5.16)
\end{alignat}
$$

The Objective Function (5.1) optimizes the geographical compactness of the week and day clusters as a weighted sum. Constraints (5.2) ensure that the first service visit of each customer $b \in B$ is scheduled for the first r_b weeks, and Constraints (5.3) guarantee that the service recurs every r_b weeks. Constraints (5.4) make sure that assignments can only be made to customers that are selected as the week center of the respective week. Constraints (5.5) enforce that exactly one week center is selected for each week. The total service time of each week is guaranteed to be within the feasible time interval through Constraints (5.6) and (5.7). The weeks and days of the planning horizon are linked by Constraints (5.8).

Constraints (5.9)–(5.12) impose restrictions at the level of days that are analogous to those defined by Constraints (5.4)–(5.7) for the level of weeks. Lastly, Constraints (5.13)–(5.16) define the binary decision variables.

Experiments have shown that model $(COMP)$ can be solved only for very small problem instances to proven optimality in reasonable running time by a general purpose MIP solver (see the computational results in Section 5.5.5). This motivated the development of our branch-and-price algorithm.

5.3 A Column Generation Reformulation

In the following, we reformulate the compact model $(COMP)$ of the previous section as a master problem and several pricing problems. Furthermore, we define what we understand by the term *symmetry* and show that the master problem exhibits a high degree of symmetry.

5.3.1 Master Problem

For the formulation of the master problem, we need to introduce some additional notation. Let the set S^{week} contain all feasible week clusters, i.e., all subsets of customers B that yield in total a service time within the interval $[LB^{week}, UB^{week}]$. Analogously, denote by S^{day} the set containing all feasible day clusters, i.e., all subsets of customers B that yield in total a service time in the interval $[LB^{day}, UB^{day}]$. Furthermore, denote by $S^w \subseteq S^{week}$ the clusters that can be selected for week $w \in W$ and by $S^d \subseteq S^{day}$ the clusters that can be selected for day $d \in D$. This notation might appear redundant since the unrestricted master problem that we present in the following contains the entire set of feasible clusters, i.e., $S^w = S^{week}$ for each week $w \in W$ and $S^d = S^{day}$ for each day $d \in D$. However, the restricted master problem in Section 5.4 may contain proper subsets $S^w \subset S^{week}$ and $S^d \subset S^{day}$ of all feasible clusters, and these subsets may vary from time period to time period. Therefore, we need to differentiate between specific weeks and days. Moreover, let $c_s = \min_{i \in B} \sum_{b \in s} c_{ib}$ denote the compactness for each cluster $s \in S = S^{week} \cup S^{day}$. The lower the value of c_s, the more compact cluster $s \in S$ is. Let parameter a_{sb} be equal to 1 if cluster $s \in S$ contains customer $b \in B$, and 0 otherwise.

Further, introduce the following binary decision variables:

$$\delta_s^w = \begin{cases} 1 & \text{if cluster } s \in S^w \text{ is selected for week } w \in W \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_s^d = \begin{cases} 1 & \text{if cluster } s \in S^d \text{ is selected for day } d \in D \\ 0 & \text{otherwise} \end{cases}$$

Then, the master problem can be formulated as the following IP, which we denote by model (MP):

$$(MP) \quad \lambda \sum_{w \in W} \sum_{s \in S^w} c_s \delta_s^w + (1-\lambda) \sum_{d \in D} \sum_{s \in S^d} c_s \delta_s^d \to \min \tag{5.17}$$

$$\text{s.t.} \quad \sum_{s \in S^w} \delta_s^w = 1 \qquad w \in W \tag{5.18}$$

$$\sum_{w=1}^{r_b} \sum_{s \in S^w} a_{sb} \delta_s^w = 1 \qquad b \in B \tag{5.19}$$

$$\sum_{s \in S^w} a_{sb} \delta_s^w = \sum_{s \in S^w} a_{sb} \delta_s^{((w-1) \bmod r_b)+1} \qquad b \in B, w \in W, w > r_b \tag{5.20}$$

$$\sum_{s \in S^d} \delta_s^d = 1 \qquad d \in D \tag{5.21}$$

$$\sum_{d \in D(w)} \sum_{s \in S^d} a_{sb} \delta_s^d = \sum_{s \in S^w} a_{sb} \delta_s^w \qquad b \in B, w \in W \tag{5.22}$$

$$\delta_s^w \in \{0,1\} \qquad w \in W, s \in S^w \tag{5.23}$$

$$\delta_s^d \in \{0,1\} \qquad d \in D, s \in S^d \tag{5.24}$$

The Objective Function (5.17) optimizes the compactness. Constraints (5.18) make sure that exactly one cluster per week is selected. Constraints (5.19) guarantee that there is exactly one service visit of each customer $b \in B$ in the first r_b weeks, and Constraints (5.20) ensure that each customer $b \in B$ is served every r_b weeks. Constraints (5.21) make sure that exactly one cluster per day is selected. Weeks and days are linked by Constraints (5.22). Constraints (5.23) and (5.24) are the domain constraints.

5.3.2 Pricing Problems

Let π_0^w, π_1^b, $\pi_2^{b,w}$, π_3^d, and $\pi_4^{b,w}$ denote the dual variables for Constraints (5.18), (5.19), (5.20), (5.21), and (5.22), respectively. Then, the pricing problem can be formulated as follows:

$$
\begin{aligned}
(\mathrm{PP}) \quad & \lambda \sum_{b\in B} \sum_{i\in B} \sum_{w\in W} c_{ib} u_{ib}^w + (1-\lambda) \sum_{b\in B} \sum_{i\in B} \sum_{d\in D} c_{ib} v_{ib}^d \\
& - \sum_{w\in W} \pi_0^w - \sum_{b\in B} \pi_1^b \sum_{w=1}^{r_b} \sum_{i\in B} u_{ib}^w \\
& - \sum_{b\in B} \sum_{w\in W, w>r_b} \pi_2^{b,w} \sum_{i\in B} \left(u_{ib}^w - u_{ib}^{((w-1) \bmod r_b)+1} \right) \qquad (5.25) \\
& - \sum_{d\in D} \pi_3^d - \sum_{b\in B} \sum_{w\in W} \pi_4^{b,w} \sum_{i\in B} \left(\sum_{d\in D(w)} v_{ib}^d - u_{ib}^w \right) \to \min \\
\text{s.t.} \quad & (5.4)–(5.7), (5.9)–(5.16)
\end{aligned}
$$

Model (PP) decomposes into $|W|$ independent pricing problems for the weeks and into $|D|$ independent pricing problems for the days, which gives us the following result.

Pricing Problems for the Weeks Define parameters $\bar{c}_{ib}^w$, $b, i \in B$, $w \in W$, as follows:

$$
\bar{c}_{ib}^w = \begin{cases} \lambda c_{ib} - \pi_1^b + \sum_{\hat{w}=1}^{\frac{|W|}{r_b}-1} \pi_2^{b,(w+\hat{w}r_b)} + \pi_4^{b,w} & \text{if } w \leqslant r_b \\ \lambda c_{ib} - \pi_2^{b,w} + \pi_4^{b,w} & \text{otherwise} \end{cases} \qquad (5.26)
$$

The pricing problem for week w can then be stated as the following IP (for better readability, the superscript w of the variables is omitted):

$$
\begin{aligned}
(\mathrm{PP}^w) \quad & \sum_{b\in B} \sum_{i\in B} \bar{c}_{ib}^w u_{ib} - \pi_0^w \to \min && && (5.27) \\
\text{s.t.} \quad & \sum_{b\in B} x_b = 1 && && (5.28) \\
& u_{ib} \leqslant x_i && b, i \in B && (5.29) \\
& \sum_{b\in B} \sum_{i\in B} t_b u_{ib} \geqslant LB^{week} && && (5.30) \\
& \sum_{b\in B} \sum_{i\in B} t_b u_{ib} \leqslant UB^{week} && && (5.31) \\
& u_{ib} \in \{0,1\} && b, i \in B && (5.32) \\
& x_b \in \{0,1\} && b \in B && (5.33)
\end{aligned}
$$

Pricing Problems for the Days With $\phi(d) \in W$ representing the week that contains day $d \in D$ and parameter $\bar{c}^{d}_{ib} = (1-\lambda)c_{ib} - \pi_4^{b,\phi(d)}$, the pricing problem for day $d \in D$ can be formulated as follows (again, the superscript d of the variables is omitted):

$$(PP^d) \qquad \sum_{b \in B} \sum_{i \in B} \bar{c}^{d}_{ib} \nu_{ib} - \pi_3^d \to \min \tag{5.34}$$

$$\text{s.t.} \qquad \sum_{b \in B} y_b = 1 \tag{5.35}$$

$$\nu_{ib} \leqslant y_i \qquad b, i \in B \tag{5.36}$$

$$\sum_{b \in B} \sum_{i \in B} t_b \nu_{ib} \geqslant LB^{day} \tag{5.37}$$

$$\sum_{b \in B} \sum_{i \in B} t_b \nu_{ib} \leqslant UB^{day} \tag{5.38}$$

$$\nu_{ib} \in \{0,1\} \qquad b, i \in B \tag{5.39}$$

$$y_b \in \{0,1\} \qquad b \in B \tag{5.40}$$

5.3.3 Symmetry in Model (MP)

As we have already noted in Section 4.4.3, problem MPSTDP-S contains a lot of symmetry. This applies also to model (MP) of the column generation reformulation. Symmetry can be present on the level of weeks and days. In the following, we formally define week and day symmetry, and derive a minimum amount of symmetry that can be found in any solution. Note that we use vectors in the remainder of this chapter to specify week (day) clusters in chronological sequence. This means that the first component of such a vector represents the week (day) cluster of the first week (day) of the planning horizon, the second component represents the week (day) cluster of the second week (day), and so on.

Week symmetry

By the term *week symmetry* we mean the symmetry that is due to the temporal rearrangement of a solution's week clusters. It is defined as follows.

Definition 5.1. Given two feasible solutions with respective week clusters $C = (C^1, ..., C^{|W|})$ and $\tilde{C} = (\tilde{C}^1, ..., \tilde{C}^{|W|})$, the two solutions are said to be week-symmetric if there exists a permutation $\sigma : W \mapsto W$ with $C^{\sigma(w)} = \tilde{C}^w$ for each week $w \in W$.

Next, we define what we mean by a *feasible week cluster permutation* for a solution and by a *maximally week-symmetry constrained solution*.

Definition 5.2. Given the week clusters $C = (C^1, ..., C^{|W|})$ of a feasible solution, a permutation $\sigma : W \mapsto W$ is said to be a feasible week cluster permutation for that solution if in the week clusters $(C^{\sigma(1)}, ..., C^{\sigma(|W|)})$ each customer $b \in B$ is served every r_b weeks.

Definition 5.3. A solution consisting of week clusters $C = (C^1, ..., C^{|W|})$ is said to be maximally week-symmetry constrained with respect to the set of week rhythms $R \subseteq \{r_b\}_{b \in B}$ if each week cluster C^w, $w \in W$, contains for each week rhythm $r \in R$ a customer $b \in B$ with $r_b = r$.

In the following, we state a special property of week cluster permutations that are feasible for maximally week-symmetry constrained solutions. This property will play an important role in the development of symmetry reduction techniques in Section 5.4.3.

Lemma 5.1. If a week cluster permutation is feasible for a solution that is maximally week-symmetry constrained with respect to the set of week rhythms R, it is feasible for any other solution that consists only of customers $b \in B$ with $r_b \in R$.

Proof. Consider a solution that is maximally week-symmetry constrained with respect to the set of week rhythms R. Clearly, removing a customer from the solution does not reduce the number of feasible week cluster permutations for that solution. Likewise, (feasibly) inserting an additional customer with $r_b \in R$ does not reduce the number of feasible week cluster permutations for that solution since each week cluster already contains a customer $b \in B$ with $r_b = r$ for each $r \in R$ and, hence, the newly inserted customer does not impose any additional restrictions. Since, starting from a maximally week-symmetry constrained solution, any other solution can be generated by inserting additional customers and removing present customers, a week cluster permutation that is feasible for a maximally week-symmetry constrained solution with respect to R is also feasible for any other solution that consists only of customers $b \in B$ with $r_b \in R$. □

From Lemma 5.1 we can derive a minimum amount of week symmetry inherent in any solution. Consider, for example, a planning horizon of $|W| = 4$ weeks, and suppose that $r_b \in R = \{1, 2, 4\}$ for each customer $b \in B$. The week cluster permutations shown in Table 5.1 are feasible for a maximally week-symmetry constrained solution with respect to R and, hence, also for any other solution in which the customers' week rhythms are restricted to the set R. This means that there are (at least) eight week-symmetric solutions to any solution consisting only of customers $b \in B$ with $r_b \in R$. Note that, when a solution is not maximally week-symmetry constrained, there might be even more week symmetry than given by Lemma 5.1.

Table 5.1: Feasible week cluster permutations for a maximally week-symmetry constrained solution with respect to $R = \{1, 2, 4\}$ and a planning horizon of $|W| = 4$ weeks (example adopted from Section 4.4.3)

Permutation no.	$\sigma(1)$	$\sigma(2)$	$\sigma(3)$	$\sigma(4)$	Permutation no.	$\sigma(1)$	$\sigma(2)$	$\sigma(3)$	$\sigma(4)$
1	1	2	3	4	5	3	4	1	2
2	1	4	3	2	6	3	2	1	4
3	2	3	4	1	7	4	1	2	3
4	2	1	4	3	8	4	3	2	1

Day symmetry

With $m = \frac{|D|}{|W|}$ denoting the number of days per week, we define *day symmetry* as follows.

Definition 5.4. Given a week $w \in W$, two feasible solutions, and their respective day clusters $C = (C^{w,1}, ..., C^{w,m})$ and $\tilde{C} = (\tilde{C}^{w,1}, ..., \tilde{C}^{w,m})$ in week w, the two solutions are said to be day-symmetric with respect to week w if there exists a permutation $\sigma : \{1, ..., m\} \mapsto \{1, ..., m\}$ with $C^{w,\sigma(d)} = \tilde{C}^{w,d}$ for each weekday $d \in \{1, ..., m\}$ in week w.

Since there are no restrictions with respect to the distribution of a customer's service visits to the days *within* a week, any rearrangement of the day clusters within a week is feasible. Consequently, there are $m!$ day-symmetric solutions for each week $w \in W$, which results in $(m!)^{|W|}$ day-symmetric solutions for the entire planning horizon. Consider again the example with a planning horizon of $|W| = 4$ weeks from the previous subsection and suppose that each week consists of $m = 5$ days. Combining week and day symmetry, there are (at least) $8 \cdot (5!)^4$ symmetric solutions to any feasible solution. We will propose techniques to reduce this tremendous amount of symmetry in Section 5.4.3.

5.4 Branch-and-Price Algorithm

We propose a branch-and-price algorithm (see, e.g. Barnhart et al., 1998; Lübbecke and Desrosiers, 2005) to solve model (MP). A branch-and-price algorithm is a branch-and-bound algorithm for solving integer programs, in which the LP relaxation in each node of the branch-and-bound tree is solved using column generation. When the solution in a node is fractional and better than the current incumbent solution, branching is performed. In the following, we explain these steps in detail and present specialized techniques to reduce week and day symmetry. Furthermore, we present an extension of the algorithm which involves the generation of cutting planes to tighten the linear relaxation of model

(MP). To the best of our knowledge, this is the first specially-tailored exact method for the scheduling task of the multi-period service territory design problem.

5.4.1 Column Generation

In each node of the branch-and-bound tree, we use column generation to solve the corresponding linear relaxation of model (MP), i.e., the linear relaxation of model (MP) extended by the branching decisions and, if applicable, by the cutting planes that are generated in the node. The basic idea of column generation is to work with a restricted master problem (RMP), which contains only a subset of the columns of model (MP) and to add new columns only if they might improve the objective value. Two steps are performed iteratively. (1) The LP relaxation of the RMP is solved to obtain primal and dual solutions. (2) Pricing problems (PP^w) and (PP^d) are solved using the dual multipliers from step 1 to find negative reduced cost columns. If such columns exist, these columns are added to the RMP and the LP relaxation of the RMP is solved again; otherwise the current solution is an optimal solution to the LP relaxation of the RMP. An extensive introduction to column generation can be found in Desrosiers and Lübbecke (2005).

To obtain an initial set of feasible columns for the RMP, we solve the problem at hand with the location-allocation heuristic introduced in Chapter 4. Furthermore, we add one artificial binary variable with high objective function coefficients to each of Constraints (5.18), (5.19), and (5.21) to ensure feasibility when columns that would violate a branching decision are removed from the RMP.

To solve the pricing problems, we proceed as follows. As in Mehrotra et al. (1998), we break problems (PP^w) and (PP^d) down into smaller subproblems by fixing the week or day center $i \in B$. Fixing the week center i in problem (PP^w) yields the following IP, which we denote by (PP_i^w):

$$(PP_i^w) \qquad \sum_{b \in B} \bar{c}_{ib}^w u_b - \pi_0^w \to \min \tag{5.41}$$

$$\text{s.t.} \qquad \sum_{b \in B} t_b u_b \geqslant LB^{week} \tag{5.42}$$

$$\sum_{b \in B} t_b u_b \leqslant UB^{week} \tag{5.43}$$

$$u_b \in \{0, 1\} \qquad b \in B \tag{5.44}$$

Analogously, we obtain problem (PP_i^d) when the day center i is fixed in problem (PP^d):

$$(\mathrm{PP}_i^d) \qquad \sum_{b \in B} \bar{c}_{ib}^d \nu_b - \pi_3^d \to \min \tag{5.45}$$

$$\text{s.t.} \qquad \sum_{b \in B} t_b \nu_b \geqslant LB^{day} \tag{5.46}$$

$$\sum_{b \in B} t_b \nu_b \leqslant UB^{day} \tag{5.47}$$

$$\nu_b \in \{0,1\} \qquad b \in B \tag{5.48}$$

Note that we omitted the subscript i for the variables in both models.

As a result, we obtain $|B| \cdot |W|$ problems to generate promising week clusters and $|B| \cdot |D|$ problems to generate promising day clusters. The problems are similar to knapsack problems with two peculiarities: There can be negative profits for the items, and the weight of each knapsack must exceed a threshold value. We solve problems (PP_i^w) and (PP_i^d) for each center $i \in B$ and pass all columns with negative reduced costs to the RMP. The advantage of this procedure is that we can generate up to $|B| \cdot (|W| + |D|)$ negative reduced cost columns in a single pricing iteration. If we solved problems (PP^w) and (PP^d) instead, we could generate at most $|W| + |D|$ such columns per iteration.

We opted for a two-stage procedure to speed up the algorithm. First, we try to find promising columns by means of a fast greedy heuristic. Only if the heuristic does not find any columns with negative reduced costs, we switch to an exact method to guarantee optimality of the overall algorithm.

The heuristic solves problem (PP_i^w) for a given center $i \in B$ and a given week $w \in W$ as illustrated by the pseudocode of Algorithm 5.1. Obviously, the heuristic has to take into account the fixations in the current node of the branch-and-bound tree. As will be explained in more detail in Section 5.4.2, a fixation may either enforce or forbid the assignment of a customer to a week or a day. For the remainder of this chapter, we denote by $B^{avail}(d,N) \subseteq B$ and $B^{avail}(w,N) \subseteq B$ the subset of customers that are available for being scheduled to day $d \in D$ and week $w \in W$, respectively, in node N of the branch-and-bound tree. A customer is considered available for a day or a week in node N if there is no fixation in the node which prohibits the customer's assignment to that time period, e.g., through a fixation to a week which, in combination with the customer's week rhythm r_b, is not compatible with a visit in the considered time period. In a particular node N, the heuristic proceeds as follows. First, the set $B^{avail}(w,N)$ is determined according to the fixations in node N, and it is sorted in non-decreasing order of parameters $\bar{c}_{ib}^w$. Next, all customers that must be served in week w are added to the cluster. This comprises all weekly customers and all customers that must be served in week w due to fixations in node N. Then, the heuristic iterates over the remaining available customers in non-decreasing order of their parameters $\bar{c}_{ib}^{w}$ and decides for each customer whether it is added to the cluster. Customer b is added to the cluster only if the upper bound UB^{week} on the total service time is not violated and at least one of the two following conditions is met: (1) $\bar{c}_{ib}^{w} < 0$, i.e., the customer has a negative contribution to the overall reduced cost of the cluster, which is beneficial as we look for the cluster with minimum reduced cost. (2) The cluster has not yet reached its minimum service time LB^{week}, i.e., the cluster is currently not feasible and must be augmented by additional customers. After the cluster has been constructed, the heuristic checks if its cumulative service time is greater than or equal to the minimum cumulative service time LB^{week}. Finally, irrespective of whether the exact or the heuristic pricing method has been used, we set the cluster center to the customer $j \in B$ that minimizes the sum of the distances to all customers in the cluster. Thus, the reduced cost of the final cluster $s \subseteq B$ for week w can be computed as

$$\min_{j \in B} \sum_{b \in s} \bar{c}_{jb}^{w} - \pi_0^{w}. \tag{5.49}$$

If this value is negative, the cluster is passed to the RMP. The time complexity of Algorithm 5.1 is dominated by the calculation of the optimal center in step 13, i.e., its complexity is $\mathcal{O}(|B|^2)$.

Algorithm 5.1 Heuristic to solve problem (PP_i^w) for given center $i \in B$ and given week $w \in W$

Input: Center $i \in B$; week $w \in W$; fixations in node N
Output: $s \subseteq B$: A cluster with negative reduced cost if such a cluster can be found
1: determine $B^{avail}(w, N)$ and sort it in non-decreasing order of $\bar{c}_{ib}^{w}$
2: $s \leftarrow \emptyset$
3: **for** $b \in B^{avail}(w, N)$ **do**
4:     **if** (b is fixed to week w) or ($r_b = 1$) **then**
5:         $s \leftarrow s \cup \{b\}$
6:     **end if**
7: **end for**
8: **for** $b \in B^{avail}(w, N) \setminus s$ **do**
9:     **if** ($\sum_{\hat{b} \in s} t_{\hat{b}} + t_b \leqslant UB^{week}$) and ($\bar{c}_{ib}^{w} < 0$ or $\sum_{\hat{b} \in s} t_{\hat{b}} < LB^{week}$) **then**
10:         $s \leftarrow s \cup \{b\}$
11:     **end if**
12: **end for**
13: **if** ($\sum_{\hat{b} \in s} t_{\hat{b}} \geqslant LB^{week}$) and ($\min_{j \in B} \sum_{b \in s} \bar{c}_{jb}^{w} - \pi_0^{w} < 0$) **then**
14:     **return** s
15: **end if**

The heuristic to solve pricing problem (PP_i^d) for a given center $i \in B$ and a given day $d \in D$ is analogous to Algorithm 5.1, therefore, we refrain from giving an explicit explanation. But we want to point out one peculiarity. To this end, we introduce the concept of *day groups*:

Definition 5.5. A day group with respect to node N of the search tree is an equivalence class based on the following equivalence relation on the set of days D: Days $d_1 \in D$ and $d_2 \in D$ are equivalent if and only if they are in the same week, i.e., $\phi(d_1) = \phi(d_2)$, and have identical sets of available customers, i.e., $B^{avail}(d_1, N) = B^{avail}(d_2, N)$.

Note that $\phi(d_1) = \phi(d_2)$ implies $\bar{c}_{ib}^{d_1} = \bar{c}_{ib}^{d_2}$ for all $b \in B$ and $i \in B$. Hence, it follows from this definition that, in a certain node of the search tree, pricing problems $(PP_i^{d_1})$ and $(PP_i^{d_2})$ have the same optimal solutions for any two days d_1 and d_2 that are in the same day group. The reduced costs of the resulting day clusters differ only by the difference in the values of constants π_3^d, $d \in \{d_1, d_2\}$. Thus, to save computation time, we solve problems (PP_i^d) only for one day d^* of each day group explicitly. With $r_i^{d^*}$ denoting the reduced cost for that day, we can calculate the reduced costs for the other days of the same day group as

$$r_i^d = r_i^{d^*} + \pi_3^{d^*} - \pi_3^d. \tag{5.50}$$

As our heuristic pricing method also yields the same solutions for all days of a day group, we proceed the same way in heuristic pricing.

5.4.2 Branching

When we obtain a fractional solution in a node of the branch-and-bound tree, branching is necessary. As other authors have already noted (e.g., Savelsbergh, 1997; Savelsbergh and Sol, 1998), branching on the variables of the master problem changes the structure of the pricing problems and makes them harder to solve as one needs to take care that forbidden columns are not re-generated in the pricing problems. Therefore, our branching is based on the compact formulation (COMP), i.e., we branch on the assignment of customers to time periods. These assignments can easily be derived from the solution to the LP relaxation of the RMP. The assignment of customer $b \in B$ to week $w \in W$ is calculated as

$$u_b^w = \sum_{s \in S^w} a_{sb} \delta_s^w. \tag{5.51}$$

Analogously, the assignment of customer $b \in B$ to day $d \in D$ is given by

$$v_b^d = \sum_{s \in S^d} a_{sb} \delta_s^d. \tag{5.52}$$

Branching is performed hierarchically. As long as there are fractional assignments of customers to weeks, we branch on the week assignments u_b^w. Only if all customers are unambiguously assigned to weeks, we branch on the assignments of customers to days v_b^d. In both cases, we generate two child nodes, with one node forcing the corresponding assignment to take on a value of one and the other forcing it to zero. The fixations must be taken into account in the RMP of the newly generated nodes and in the corresponding pricing problems. In the RMP, we take care of the fixations by removing all clusters from the model that would violate a fixation. In the exact pricing method, we simply adopt the fixations into the IP model, and in the pricing heuristic we consider all fixations in the sets $B^{avail}(d, N)$ and $B^{avail}(w, N)$ as explained in Section 5.4.1.

We implement two different rules to decide which assignments to branch on. We illustrate this in the following using the example of week assignments, but the procedure is analogous for day assignments. Our first branching rule is largest split (LS) branching. In LS branching, we select a fractional customer-week assignment with maximum value, i.e., we select

$$\langle b^\star, w^\star \rangle \in \arg\max_{\langle b,w \rangle} \{u_b^w \mid u_b^w \notin \{0,1\}, w \leqslant r_b\}. \tag{5.53}$$

Since $u_b^w = u_b^{\hat{w}}$ if $w \bmod r_b = \hat{w} \bmod r_b$, we consider only the first r_b weeks for each customer $b \in B$.

Our second branching rule is pseudocost (PSD) branching. This rule is inspired by the works of Achterberg et al. (2005) and Linderoth and Savelsbergh (1999). The basic idea is to estimate the increase of the objective value when a fractional assignment is forced to take on an integer value compared to the objective value of the parent node. Branching priority is given to assignments that are expected to lead to a large deterioration in the objective value. Thus, this rule aims at a quickly rising lower bound.

Consider a particular node N in the branch-and-bound tree. Denote by f^N its objective value, and by f^+ and f^- the objective values of the two child nodes when the branching variable u_b^w is forced to one and zero, respectively. Then, the increase in the objective value per unit change in the branching variable can be calculated as follows:

$$\Delta_{b,w}^+ = \frac{f^+ - f^N}{1 - u_b^w}, \tag{5.54}$$

$$\Delta_{b,w}^- = \frac{f^- - f^N}{u_b^w}. \tag{5.55}$$

We could now calculate scores for each possible branching variable u_b^w. But our preliminary tests have shown that the number of branching decisions is not large enough to derive meaningful scores on such a fine-grained scale. Therefore, we do not calculate customer- and week-specific scores, but aggregate the scores per customer. With $\Delta^+_{b,d}$ being the counterpart of $\Delta^+_{b,w}$ for day assignments, we denote by θ^+_b the sum over all $\Delta^+_{b,w}$ and $\Delta^+_{b,d}$ for all past upward branching decisions on a week or day assignment of customer $b \in B$. θ^-_b is defined analogously for the case of downward branching. Moreover, we denote by n^+_b and n^-_b the number of upward and downward branching decisions, respectively, on a week or day assignment of customer b. Then, two score values Score^+_b and Score^-_b are calculated for each customer b. They represent the average relative increase in the objective value for upward and downward branching on a week or day assignment of the customer.

$$\text{Score}^+_b = \frac{\theta^+_b}{n^+_b} \tag{5.56}$$

$$\text{Score}^-_b = \frac{\theta^-_b}{n^-_b} \tag{5.57}$$

Finally, the score for each customer-week assignment is calculated as

$$\text{Score}^w_b = (1 - u^w_b) \cdot \text{Score}^+_b + u^w_b \cdot \text{Score}^-_b. \tag{5.58}$$

As the customer-week assignment to be branched on we select a fractional assignment with maximum score, i.e., we select

$$\langle b^\star, w^\star \rangle \in \underset{\langle b,w \rangle}{\arg\max} \{\text{Score}^w_b \mid u^w_b \notin \{0,1\}, w \leqslant r_b\}. \tag{5.59}$$

We always use LS branching for the first n_{min} branching decisions to initialize the scores. If, after n_{min} iterations, either all Score^+_b values are uninitialized or all Score^-_b values are uninitialized, we perform additional iterations with LS branching until we obtain at least one initialized Score^+_b value and one initialized Score^-_b value. Afterwards, we switch to PSD branching. During the course of the algorithm, uninitialized scores Score^+_b and Score^-_b are set to the average of the respective initialized scores, i.e.,

$$\text{Score}^+_b = \frac{\sum_{\hat{b} \in B^+} \text{Score}^+_{\hat{b}}}{|B^+|}, \tag{5.60}$$

$$\text{Score}^-_b = \frac{\sum_{\hat{b} \in B^-} \text{Score}^-_{\hat{b}}}{|B^-|}, \tag{5.61}$$

where B^+ and B^- denote the set of customers with initialized values of Score^+_b and Score^-_b, respectively. This way of initializing the scores seems to be more plausible than other alternatives, e.g., taking the maximum or minimum values.

Irrespective of the selected branching rule, we adopt the idea of early branching (see, e.g., Desaulniers et al., 2002) to accelerate our algorithm. The potential benefit of early branching becomes obvious through the following observations: The exact solution of the pricing problems (PP_i^w) and (PP_i^d) is computationally expensive. Moreover, preliminary tests showed that, in many cases, exact pricing does not find any negative reduced cost columns, but is executed only to prove optimality. Even if negative reduced cost columns are found, their impact on the objective value of the node is usually fairly small. Therefore, we skip the exact pricing step under certain conditions. More precisely, when the pricing heuristic does not find any more negative reduced cost columns, we skip exact pricing if the current solution to the LP relaxation of the RMP is fractional and if its objective value is better than that of the current incumbent solution. As a consequence, exact pricing is called less often. Note that when early branching is applied, the objective value of a node might be better than that of its parent node, i.e., the objective value does not provide a valid lower bound any more. Hence, before a node can be pruned, re-optimization with our exact pricing method must be performed. A node is pruned only if the objective value after re-optimization is not better than that of the incumbent solution.

We use a best-first strategy to explore the branch-and-bound tree, i.e., we always select the node with the best initial objective value, which is inherited from the parent node, to be processed next.

5.4.3 Symmetry Reduction

As illustrated in Section 5.3.3, model (MP) contains a lot of symmetry. Thus, efficient symmetry handling is crucial for the design of a successful branch-and-price algorithm. In this section, we propose two techniques to reduce symmetry. In the first technique, we fix a single customer a priori to a particular day of the planning horizon. In the second, more sophisticated technique, we introduce additional variable fixations during the course of the algorithm and prune certain subtrees if we can guarantee that they contain only solutions that are symmetric to solutions in other parts of the search tree. In the following, we explain the techniques in detail.

Fixing a Reference Customer

A simple, yet effective way to eliminate some of the symmetry inherent to the MPSTDP-S* is to fix one service of a particular customer, which is called the *reference customer*, to a particular day of the planning horizon. This approach is similar to the idea presented by

Mourgaya and Vanderbeck (2007) in the context of the periodic vehicle routing problem. In the following, we prove that such a fixation can be done without losing optimality.

Lemma 5.2. A right-shift of week clusters, defined as the week cluster permutation $\sigma : W \mapsto W$ with $\sigma(1) = |W|$ and $\sigma(w) = w - 1$ for each $w > 1$, is a feasible week cluster permutation for any feasible solution.

Proof. Consider the week clusters $C = (C^1, ..., C^{|W|})$ of a feasible solution. Since the solution is feasible, each customer $b \in B$ of the solution is served regularly every r_b weeks with the first service in the first r_b weeks. This means that there exists for each customer $b \in B$ an $n_b \in \{0, ..., r_b - 1\}$ such that each C^w, $w \in W$, contains customer b if and only if $w \bmod r_b = n_b$. Let $\tilde{C} = (\tilde{C}^1, ..., \tilde{C}^{|W|})$ denote the week clusters obtained by a right-shift of C. Since $|W|$ is the least common multiple of the week rhythms $\{r_b\}_{b \in B}$ and, hence, $|W| \bmod r_b = 0$ for each b, $\tilde{C}^w$ contains b if and only if $w \bmod r_b = (n_b + 1) \bmod r_b = \tilde{n}_b$. Thus, a right-shift of the week clusters of a feasible solution yields a feasible solution and, hence, is a feasible week cluster permutation. □

Proposition 5.1. For any arbitrary customer $b^* \in B$ and any day $d^* \in D$ of the planning horizon, there exists an optimal solution with customer b^* being scheduled to day d^*.

Proof. Given any optimal solution, one can, according to Lemma 5.2, obtain a feasible week-symmetric solution in which customer b^* is served in week $\phi(d^*)$ by performing an appropriate number of right-shifts of the week clusters. Afterwards, as there are no restrictions on the re-orderings of the day clusters within a week, a day-symmetric solution with respect to week $\phi(d^*)$ can be obtained in which customer b^* is served on day d^*. Two week- or day-symmetric solutions consist of the same week and day clusters (merely arranged in a different order) and, hence, have the same objective value. Therefore, the resulting solution is optimal. □

Obviously, the extent of symmetry reduction that can be achieved by such a fixation depends on the selected reference customer. The reduction of week symmetry depends on the customer's week rhythm. The greater the week rhythm r_b of a customer $b \in B$, the more possibilities exist to assign the customer to the weeks of the planning horizon. Hence, to achieve maximal week symmetry reduction, we select a reference customer $b^* \in B$ with $r_{b^*} = max_{b \in B} r_b$. Then, we fix customer b^* to a day $d^* \in D(w^*)$ with $w^* \leqslant r_{b^*}$ in the root node of the branch-and-bound tree. Through this simple technique we can already reduce symmetry by factor $m \cdot r_{b^*}$. Clearly, if $r_{b^*} < |W|$, customer b^* can additionally be fixed to an arbitrary day in each of weeks $w \in \{w^* + r_{b^*}, w^* + 2r_{b^*}, ..., |W| + w^* - r_{b^*}\}$.

Symmetry-reduced Branching

In this section, we introduce a technique which we call symmetry-reduced branching. It was developed by Pouls (2016) and is an enhancement of the branching scheme introduced in Section 5.4.2 with the aim of reducing both week and day symmetry.

Reduction of Week Symmetry Suppose that $u_{b^*}^{w^*}$ is the week assignment that is selected to be branched on in a particular node N of the branch-and-bound tree. Recall that, in standard branching, we always create two child nodes N^+ and N^- of N. We fix $u_{b^*}^{w^*} = 1$ in node N^+, and $u_{b^*}^{w^*} = 0$ in node N^-. The idea of symmetry-reduced branching is to add additional week fixations to node N^- if we can guarantee that, to any solution that becomes infeasible in node N^- by such an additional fixation, there is a week-symmetric solution in the other branch.

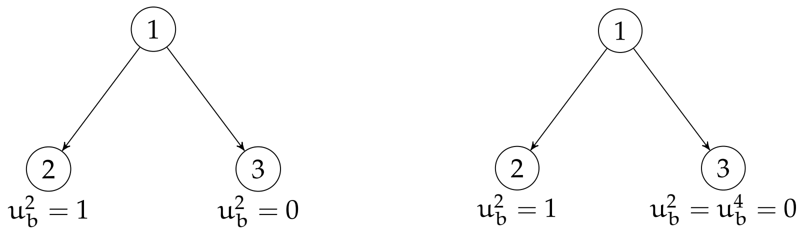
We denote by $\mathcal{S}$ the set of feasible week cluster permutations for a solution that is maximally week-symmetry constrained with respect to the set of week rhythms $R = \{r_b \mid b \in B, 1 < r_b < |W|\}$. Note that customers $b \in B$ with week rhythm $r_b = 1$ or $r_b = |W|$ do not have to be considered since they do not restrict the feasibility of the permutations. By fixing the week or day assignments of customers, as done in the nodes of the branch-and-bound tree, permutations from the set $\mathcal{S}$ are gradually rendered infeasible in the course of the algorithm. We denote by $\mathcal{S}(N) = \{\sigma \in \mathcal{S} \mid \sigma \text{ is feasible with respect to all fixations present in node } N\}$. Week symmetry can be reduced as follows.

Proposition 5.2. If there exists a permutation $\sigma \in \mathcal{S}(N)$ in node N of the search tree and a week $\hat{w} \in W$ with $\hat{w} \neq w^*$, $\hat{w} \leqslant r_{b^*}$ and $((\sigma(\hat{w}) - 1) \bmod r_{b^*}) + 1 = w^*$, the additional fixation $u_{b^*}^{\hat{w}} = 0$ can be added to node N^- without losing optimality.

Proof. If the condition above is fulfilled, then there exists a feasible week cluster permutation which maps the first service of customer b^* from week $\hat{w}$ to week w^*. In this case, we can guarantee to find a solution in the subtree of node N^+ that is week-symmetric to any solution in the subtree of node N^- in which customer b^* is served in week $\hat{w}$. Hence, we cannot forfeit optimality if we introduce the additional fixation $u_{b^*}^{\hat{w}} = 0$ to node N^-. □

If, after the insertion of additional fixations, there are no more feasible week assignments left for customer b^* in node N^-, we immediately prune node N^-.

Consider the following example. Suppose again that the planning horizon consists of $|W| = 4$ weeks and that the week rhythms $r_b \in R = \{1, 2, 4\}$ for all customers $b \in B$. As we can see in Table 5.1, there are at least eight feasible permutations of the week clusters in this setting. Figure 5.1 illustrates the difference between standard branching and symmetry-reduced branching. We assume that a reference customer $b^* \in B$ with week rhythm $r_{b^*} = 4$ has been fixed to the first day and, hence, also to the first week of the planning horizon. This reduces the feasible permutations to permutations no. 1 and 2 from Table 5.1, i.e., to $(1, 2, 3, 4)$ and $(1, 4, 3, 2)$. Moreover, we assume that no other fixations exist in node 1. Suppose that we branch on the week assignment u_b^2 in node 1 and that $r_b = 4$. In standard branching, this would lead to two child nodes, with node 2 fixing the assignment to one, and node 3 fixing it to zero. But for each solution in which the customer is assigned to week $w = 4$ in node 3, permutation no. 2 gives us a week-symmetric solution in which the customer is served in week $w = 2$, which is identical to the situation in node 2. Hence, in symmetry-reduced branching, we add the additional fixation $u_b^4 = 0$ to node 3.



(a) Standard branching **(b)** Symmetry-reduced branching

Figure 5.1: Comparison of standard branching and symmetry-reduced branching for an exemplary week assignment

Reduction of Day Symmetry Suppose that we branch on the day assignment $v_{b^*}^{d^*}$ in node N of the search tree. As in the branching on week assignments, two child nodes N^+ and N^- are generated in standard branching with fixations $v_{b^*}^{d^*} = 1$ in node N^+ and $v_{b^*}^{d^*} = 0$ in node N^-. In symmetry-reduced branching, we add, again, additional fixations to node N^- in order to reduce symmetry.

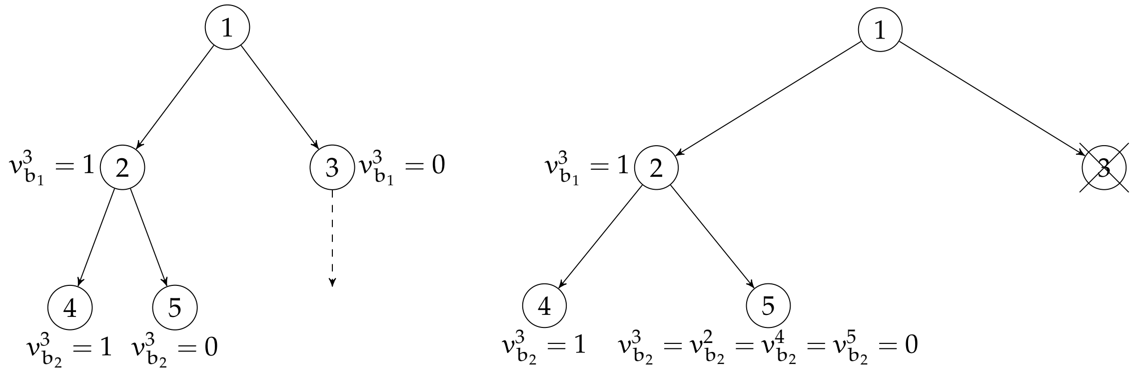
Recall that there are no restrictions with respect to the assignment of customers to days *within* a week. Hence, as long as there are no day fixations in a particular week, the day clusters of the week can be arbitrarily rearranged. But even when some day fixations have already been introduced, day symmetry might still be present. Based on the concept of day groups (see Definition 5.5), day symmetry can be reduced as follows.

Proposition 5.3. Let G be a day group in week $\phi(d^*)$, i.e., $G \subseteq D(\phi(d^*))$, with respect to node N of the search tree. Then, the following fixations can be added to node N^- without losing optimality. If G does not contain the branching day d^*, additional fixations $v_{b^*}^d = 0$ can be added for all days $d \in G$ except one. Otherwise, these fixations can be added for each day $d \in G$, $d \neq d^*$.

Proof. Since all days $d \in G$ have the same set of available customers $B^{avail}(d, N)$, all rearrangements of the corresponding day clusters yield day-symmetric solutions. If G does not contain d^*, we can therefore forbid the assignment of customer b^* to any but one of the days of day group G in node N^-. If G contains d^*, we can forbid any solution in which b^* is served on a day $d \in G$ in node N^- since node N^+ contains a day-symmetric solution. Hence, optimality is guaranteed in both cases. □

Consequently, we check for each day group in week $\phi(d^*)$ if additional fixations can be introduced. If the additional fixations leave no feasible day assignments for customer b^* and week $\phi(d^*)$ in node N^-, we immediately prune node N^-. There is, however, one peculiarity. It might occur that we obtain an integer week assignment for a customer, although the customer is not fixed to a particular week. When we branch on the day assignment of such a customer, the customer's week assignment is implicitly fixed to week $\phi(d^*)$ in node N^+. If, at the same time, the available customers $B^{avail}(d, N)$ are identical for each day $d \in D(\phi(d^*))$, we prune node N^-, and, hence, discard the possibility of the customer being assigned to a different week. Therefore, if this situation occurs, we generate an additional child node, in which we force the customer to be scheduled to a different week, i.e., in which we set $u_{b^*}^{\phi(d^*)} = 0$.

Consider the example shown in Figure 5.2 and assume that there are $m = 5$ days per week. Further, assume that we branch on day assignment $v_{b_1}^3$ in node 1 and that the available customers $B^{avail}(d, N)$ are the same for each day $d \in D(\phi(3))$, i.e., there exists only one day group $G_1 = \{1, 2, 3, 4, 5\}$ consisting of all days of the week. In standard branching, this would again lead to the creation of two child nodes, one with $v_{b_1}^3 = 1$ and the other with $v_{b_1}^3 = 0$. In symmetry-reduced branching, we would add the additional fixations $v_{b_1}^1 = v_{b_1}^2 = v_{b_1}^4 = v_{b_1}^5 = 0$ to node 3, which would leave no feasible day assignments left for customer b_1 in week $\phi(3)$. Hence, node 3 can immediately be pruned. Note that for this example we assume that customer b_1 has previously been fixed to week $\phi(3)$ such that we do not have to create an additional child node which allows the assignment to a different week. Suppose that the next branching is performed on the day assignment $v_{b_2}^3$ in node 2. Due to the fixation $v_{b_1}^3 = 1$ we now have the two day groups $G_2 = \{3\}$ and $G_3 = \{1, 2, 4, 5\}$. Since the branching day 3 is not part of day group G_3, we can forbid the assignment of customer b_2 to any of the days of day group G_3 except one. Hence, we add the additional fixations $v_{b_2}^2 = v_{b_2}^4 = v_{b_2}^5 = 0$ to node 5.



(a) Standard branching **(b)** Symmetry-reduced branching

Figure 5.2: Comparison of standard branching and symmetry-reduced branching for exemplary day assignments

5.4.4 Cut Generation

In an attempt to strengthen the LP relaxation of the RMP, we experimented with an extension of the proposed algorithm by the incorporation of cutting planes. After the column generation phase, we look for valid inequalities that are violated by the current solution to the LP relaxation of the RMP and add them to the RMP.

Note that model (MP) has set-partitioning-like components, e.g., Constraints (5.19) define a set-partitioning polytope. Hence, valid inequalities for the set-packing and set-partitioning polytope, such as the well-known clique inequalities and odd-hole inequalities (see, e.g., Padberg, 1973), could be used to strengthen the LP relaxation of the RMP. For the week clusters in the RMP, we could formulate clique or odd-hole inequalities based on a conflict graph derived from Constraints (5.18)–(5.20) (or a subset of them). However, adding these inequalities to the RMP significantly changes the structure of the pricing problems. While our pricing heuristic could easily be adapted to consider these changes, solving the pricing problems to optimality would become much more complex. The difficulty is to determine whether a column participates in a certain clique or odd-hole inequality of the RMP, and, hence, whether the associated reduced costs must be considered in the pricing problem. Preliminary tests confirmed that the solution of models (PP^w_i) and (PP^d_i), extended to consider the reduced costs of clique cuts, becomes computationally too expensive.

Therefore, we opted to use subset-row (SR) inequalities (Jepsen et al., 2008). They were proposed for a set-partitioning formulation of the vehicle routing problem with time windows and mitigate to some extent the above mentioned disadvantage of clique and odd-hole inequalities. We define an SR inequality q on a subset of Constraints (5.19), and, since each constraint corresponds to a customer, also on a subset $B_q \subseteq B$ of customers. An SR inequality can be stated as

$$\sum_{w \in W} \sum_{s \in S^w} \left\lfloor \frac{1}{k} \sum_{b \in B_q | w \leqslant r_b} a_{sb} \right\rfloor \delta_s^w \leqslant \left\lfloor \frac{|B_q|}{k} \right\rfloor, \tag{5.62}$$

where k is a parameter with $0 < k \leqslant |B_q|$. It can be interpreted as follows. For every k customers $b \in B_q$ that are contained in a week cluster s of a week w with $w \leqslant r_b$, the coefficient of the cluster on the left-hand side of the inequality increases by one. Since each customer b must be served exactly once in the first r_b weeks of the planning horizon, at most $\left\lfloor \frac{|B_q|}{k} \right\rfloor$ such clusters may be selected in an integer solution.

For the separation of SR inequalities, we set parameter k to a fixed value and restrict ourselves to subsets B_q of cardinality n. In preliminary tests, we observed that $n = 3$ and $k = 2$ yield the best results, and, therefore, we use this configuration for the remainder of this chapter. We check for each subset B_q with $|B_q| = n$ if Inequality (5.62) is satisfied in the current solution to the LP relaxation of the RMP and add all violated inequalities to the RMP.

Integrating SR inequalities into pricing problems (PP_i^w) yields the following result:

$$(PP_i^w\text{–SR}) \qquad \sum_{b \in B} \bar{c}_{ib}^w u_b - \pi_0^w - \sum_{q \in Q} \pi_5^q z_q \rightarrow \min \tag{5.63}$$

$$\text{s.t.} \qquad \sum_{b \in B} t_b u_b \geqslant LB^{week} \tag{5.64}$$

$$\sum_{b \in B} t_b u_b \leqslant UB^{week} \tag{5.65}$$

$$z_q \geqslant \left(\frac{1}{k} \sum_{b \in B_q | w \leqslant r_b} u_b \right) - 1 + \epsilon \qquad q \in Q \tag{5.66}$$

$$u_b \in \{0, 1\} \qquad b \in B \tag{5.67}$$

$$z_q \in \mathbb{N}_0 \qquad q \in Q \tag{5.68}$$

In model $(PP_i^w\text{–SR})$, Q denotes the set of SR cuts contained in the RMP, π_5^q denotes the dual variable for SR cut $q \in Q$, and ϵ represents a parameter with value slightly greater than zero. Assuming that $k \in \mathbb{N}^+$, ϵ must be set to a value $0 < \epsilon \leqslant \frac{1}{k}$. This makes sure that Constraints (5.66) in conjunction with the integrality requirements on variables z_q as defined in Constraints (5.68) mimic the floor function of the left-hand side of Inequality (5.62): For each SR cut $q \in Q$, variable z_q is increased by one for every k customers $b \in B_q$ with $w \leqslant r_b$ which are contained in the week cluster that is generated in the pricing problem.

Note that it is not necessary to add constraints which define an upper bound for z_q since all π_5^q are nonpositive and, therefore, z_q implicitly takes on the smallest feasible value.

We adapt our pricing heuristic to reflect the modification of the pricing problems. The mechanism to generate new week clusters remains the same as described in Section 5.4.1, but we need to consider the values of π_5^q in the calculation of the reduced cost of the final cluster. For cluster s and week w the reduced cost is calculated as follows:

$$\min_{j \in B} \sum_{b \in s} \bar{c}_{jb}^{w} - \pi_0^{w} - \sum_{q \in Q} \pi_5^q \left\lfloor \frac{1}{k} \sum_{b \in B_q | w \leqslant r_b} a_{sb} \right\rfloor . \tag{5.69}$$

We pass the corresponding column to the RMP only if this value is smaller than zero.

In addition to SR cuts for week clusters, we also generate SR cuts for day clusters. Recall that Constraints (5.19) enforce that each customer $b \in B$ must be served exactly once in the first r_b weeks of the planning horizon. From this we can derive the following constraints on the level of day clusters:

$$\sum_{w=1}^{r_b} \sum_{d \in D(w)} \sum_{s \in S^d} a_{sb} \delta_s^d = 1 \qquad b \in B \tag{5.70}$$

Based on these constraints, we formulate SR cuts for day clusters. The formulation, separation, and pricing is analogous to the SR cuts for week clusters. Therefore, we do not give any additional explanations.

We found out in preliminary tests that the impact of cutting planes on the optimal objective value of the LP relaxation of the RMP declines rapidly with the number of performed cutting phases. Since the first cutting phase yields by far the largest impact on the objective value, we decided to execute the cutting phase only once in each node of the branch-and-bound tree, namely after the first column generation phase. In all other cases, we proceed from the column generation phase directly to the branching phase. Moreover, we decided that a node does not inherit the cuts from its parent node.

5.5 Computational Evaluation

In the following, we evaluate our algorithm on real-world test instances provided by PTV. The test set comprises 16 service territories of a German manufacturer of paints and coatings. The week rhythms r_b of the customers are from the set $\{1, 2, 4\}$. The total number of visits per territory ranges from 71 to 107, the time to serve a customer, t_b, ranges from ten to 330 minutes. The planning horizon consists of $|W| = 4$ weeks and $m = 5$ days per week. A detailed overview of the test instances is given in Table 5.2.

Table 5.2: Overview of test instances

Instance no.	1	2	3	4	5	6	7	8
Number of customers	31	26	32	25	35	55	36	33
Week rhythms	1, 2, 4	1, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 4	1, 2, 4
Number of visits	80	74	76	71	84	106	72	78
Instance no.	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Number of customers	33	31	32	50	39	42	37	52
Week rhythms	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4
Number of visits	88	89	88	107	86	94	96	88

For all tests, we weight the compactness of week clusters with $\lambda = \frac{1}{3}$ and the compactness of day clusters with $1 - \lambda = \frac{2}{3}$. With $T = \sum_{b \in B} t_b \cdot \frac{|W|}{r_b}$ denoting the total service time over all customers, we limit the total service time of each week to the interval $\left[LB^{week}, UB^{week}\right] = \left[0.9 \cdot \frac{T}{|W|}, 1.1 \cdot \frac{T}{|W|}\right]$ and the total service time of each day to the interval $\left[LB^{day}, UB^{day}\right] = \left[0.8 \cdot \frac{T}{|D|}, 1.2 \cdot \frac{T}{|D|}\right]$. To initialize the scores in the case of PSD branching, we use LS branching for at least the first $n_{min} = 5$ branching decisions. The algorithm was coded in Java. All tests are performed under Ubuntu 16 on a machine with an Intel Xeon E5-2650 v2 CPU at 2.6 GHz and 128 GB of RAM. We use Gurobi 7.0.1 to solve the LP relaxation of the RMP as well as the IPs in the exact pricing step.

We analyze the impact of different features of our algorithm on its running time. In particular, we evaluate the impact of the proposed symmetry reduction techniques, we compare the two branching rules LS and PSD, and we analyze the effect of early branching and cutting planes. If not stated otherwise, the algorithm is configured as follows:

- Full symmetry reduction is applied, i.e., the fixation of a reference customer is combined with symmetry-reduced branching.
- PSD branching in combination with the presented early branching strategy is used.
- The generation of cutting planes is deactivated.

5.5.1 Impact of Symmetry Reduction Techniques

To evaluate the impact of symmetry reduction techniques on running time, we test three different variants of the algorithm: No symmetry reduction at all (NONE), fixing the first visit of a reference customer (FRC), and a combination of reference customer fixing and the symmetry-reduced branching scheme (FRC+SRB). We restrict the experiments in this section to the nine instances with at most 35 customers since for variants NONE and FRC it is not possible to solve larger instances in reasonable time. Furthermore, we set a time limit of ten hours per instance. The running times (T_{total}, in seconds), the number of processed nodes (Num_{Nodes}) as well as the objective values (Obj) for each instance are reported in Table 5.3. Furthermore, we report the percentage deviation in the running time and in the number of processed nodes relative to variant NONE. Negative values indicate an improvement in the respective value. Note that the deviation between FRC and NONE could not be calculated for those test instances for which in both variants no optimal solution could be found within the time limit. In the table we denote these cases by N/A. Additionally, we use bold-faced numbers to indicate the most successful variant on each instance with respect to running time and number of nodes.

The results show the tremendous impact of the proposed symmetry reduction techniques both on the running time and the number of processed nodes. Without any symmetry reduction techniques, two out of nine instances cannot be solved to optimality within the time limit. Although the fixation of a reference customer is a relatively simple technique, its effect is already remarkable. The average reduction in running time amounts to 75.0%, the average reduction in the number of processed nodes to 74.5%. However, one instance (no. 4) can still not be solved to optimality within the time limit, and for one instance (no. 9) the optimal solution is found but optimality cannot be proven within the time limit. When the fixation of a reference customer is combined with symmetry-reduced branching, all nine instances can be solved to proven optimality within the time limit. Moreover, the average reduction in running time and in the number of explored nodes is 89.1% and 90.0%, respectively, compared to the case without symmetry reduction.

5.5.2 Impact of Different Branching Rules

In the following, we compare the performance of the two branching rules LS and PSD. We report in Table 5.4 the running time, the number of processed nodes, and the relative deviation between PSD and LS branching with respect to the two performance figures.

PSD branching clearly outperforms LS branching. While LS branching is not able to solve three out of the 16 test instances to optimality within the time limit, PSD branching solves all instances to proven optimality. On average, the running times of PSD branching are 37.4% below those of LS branching. The average reduction in the number of processed nodes is 45.9%.

Table 5.3: Impact of symmetry reduction techniques: Running time, number of processed nodes, objective values, and deviations with respect to variant NONE

	Absolute values									Deviations relative to NONE			
	NONE			FRC			FRC+SRB			FRC		FRC+SRB	
Instance no.	T_{total}	Num_{Nodes}	Obj	T_{total}	Num_{Nodes}	Obj	T_{total}	Num_{Nodes}	Obj	T_{total}	Num_{Nodes}	T_{total}	Num_{Nodes}
1	606	2,251	1908.5	130	660	1908.5	**54**	**167**	1,908.5	-78.5%	-70.7%	-91.2%	-92.6%
2	8	37	1228.6	2	**6**	1228.6	**2**	7	1,228.6	-77.3%	-83.8%	-78.3%	-81.1%
3	4,284	12,799	1893.7	504	2,172	1893.7	**114**	**429**	1,893.7	-88.2%	-83.0%	-97.3%	-96.6%
4	36,000[1]	669,752	1761.0	36,000[1]	661,801	1710.7	**458**	**6,181**	**1,702.5**	N/A	N/A	-98.7%[2]	-99.1%[2]
5	33,995	140,657	2006.4	5,768	22,862	2006.4	**2,113**	**8,231**	2,006.4	-83.0%	-83.7%	-93.8%	-94.1%
8	171	421	2070.6	103	234	2070.6	**84**	**201**	2,070.6	-39.5%	-44.4%	-50.8%	-52.3%
9	36,000[1]	219,801	1946.8	36,000[1]	189,585	**1946.6**	**416**	**2,031**	**1,946.6**	N/A	N/A	-98.8%[2]	-99.1%[2]
10	1,787	8,191	1714.8	177	782	1714.8	**29**	**97**	1,714.8	-90.1%	-90.5%	-98.4%	-98.8%
11	315	1,539	2067.8	100	532	2067.8	**16**	**60**	2,067.8	-68.4%	-65.4%	-94.9%	-96.1%
Average	12,574	117,272	1,844.2	8,754	97,626	1838.6	**365**	**1,934**	**1,837.7**	-75.0%	-74.5%	-89.1%	-90.0%

[1] No proven optimal solution found within the time limit.
[2] Compared to the values obtained for variant NONE at the time limit.

Table 5.4: Comparison of LS and PSD branching: Running time, number of processed nodes, and deviation of PSD branching relative to LS branching

Instance no.	LS		PSD		Relative deviation	
	T_{total}	Num_{Nodes}	T_{total}	Num_{Nodes}	T_{total}	Num_{Nodes}
1	93	290	**54**	**167**	-42.3%	-42.4%
2	2	7	2	7	-1.4%	0.0%
3	178	738	**114**	**429**	-35.7%	-41.9%
4	2,235	32,590	**458**	**6,181**	-79.5%	-81.0%
5	36,000[1]	189,426	**2,113**	**8,231**	-94.1%[2]	-95.7%[2]
6	36,000[1]	28,725	**17,385**	**6,639**	-51.7%[2]	-76.9%[2]
7	1,181	4,776	**738**	**2,269**	-37.5%	-52.5%
8	125	332	**84**	**201**	-32.9%	-39.5%
9	1,887	9,802	**416**	**2,031**	-78.0%	-79.3%
10	38	137	**29**	**97**	-23.4%	-29.2%
11	18	78	**16**	**60**	-9.8%	-23.1%
12	4,403	4,097	**3,844**	**2,914**	-12.7%	-28.9%
13	4,099	7,290	**1,700**	**3,023**	-58.5%	-58.5%
14	476	639	**435**	**528**	-8.7%	-17.4%
15	40	36	40	36	0.6%	0.0%
16	36,000[1]	23,527	**24,208**	**7,436**	-32.8%[2]	-68.4%[2]
Average	7,673	18,906	**3,227**	**2,516**	-37.4%	-45.9%

[1] No proven optimal solution found within the time limit.
[2] Compared to the values obtained for LS branching at the time limit.

5.5.3 Impact of Early Branching

Next, we analyze the impact of early branching on the performance of the algorithm. Table 5.5 contains the computational results for two variants of the algorithm, namely a variant in which early branching is deactivated (No EB), and a variant in which early branching is enabled (EB). We report again the running times, the number of processed nodes, and the relative deviation between the two variants for each test instance. Additionally, we include the number of times that the exact pricing method was called (Num_{EP}).

The aim of early branching is to reduce the number of times that the exact pricing method is called. As the results show, this effect is achieved for 13 of the 16 test instances with an average reduction of 22.4%. Unfortunately, the reduction in the number of exact pricing calls does not translate into reduced running times. In fact, running time is reduced only on five instances, whereas it is increased on 11 instances. The average increase in running time amounts to 11.8% and is largely caused by an increase in the number of processed nodes by 16.9%. Since the search trees in the two variants of the algorithm might differ greatly on the same test instance, the reason for the increase in the number of nodes cannot

Table 5.5: Impact of early branching: Running time, number of processed nodes, number of exact pricing calls, and deviation of variant EB relative to the variant without early branching

Instance no.	No EB			EB			Relative deviation		
	T_{total}	Num_{Nodes}	Num_{EP}	T_{total}	Num_{Nodes}	Num_{EP}	T_{total}	Num_{Nodes}	Num_{EP}
1	76	233	655	**54**	**167**	**302**	-29.3%	-28.3%	-53.9%
2	2	**4**	17	2	7	**10**	1.6%	75.0%	-41.2%
3	**95**	**308**	742	114	429	**693**	20.3%	39.3%	-6.6%
4	**343**	**3,810**	**5,441**	458	6,181	5,803	33.6%	62.2%	6.7%
5	**1,924**	**6,910**	13,300	2,113	8,231	**10,234**	9.8%	19.1%	-23.1%
6	18,546	**6,628**	38,511	**17,385**	6,639	**25,273**	-6.3%	0.2%	-34.4%
7	757	2,406	4,288	**738**	**2,269**	**2,717**	-2.6%	-5.7%	-36.6%
8	**66**	**156**	361	84	201	**287**	26.4%	28.8%	-20.5%
9	457	**1,870**	3,898	**416**	2,031	**3,152**	-9.0%	8.6%	-19.1%
10	35	113	305	**29**	**97**	**184**	-17.4%	-14.2%	-39.7%
11	**15**	60	99	16	60	**67**	7.7%	0.0%	-32.3%
12	**2,330**	**2,059**	**4,602**	3,844	2,914	5,289	65.0%	41.5%	14.9%
13	**1,105**	**2,192**	**4,744**	1,700	3,023	5,896	53.9%	37.9%	24.3%
14	**392**	**513**	1,107	435	528	**734**	10.8%	2.9%	-33.7%
15	**37**	40	92	40	**36**	**56**	7.1%	-10.0%	-39.1%
16	**20,685**	**6,568**	35,926	24,208	7,436	**27,132**	17.0%	13.2%	-24.5%
Average	**2,929**	**2,117**	7,131	3,227	2,516	**5,489**	11.8%	16.9%	-22.4%

conclusively be explained. We conclude that early branching does, on average, not have the desired effect on the performance of the algorithm, although on specific instances early branching might be beneficial.

5.5.4 Impact of Subset-Row Cuts

In the following, we evaluate the impact of subset-row cuts on running time and on the number of processed nodes. Again, we compare two variants of the algorithm, one with cut generation being disabled (No Cuts), and one with activated cut generation (SR Cuts). The results are shown in Table 5.6.

Table 5.6: Impact of cut generation: Running time, number of processed nodes, and deviation of variant SR Cuts relative to the variant without cutting planes

	No Cuts		SR Cuts		Relative deviation	
Instance no.	T_{total}	Num_{Nodes}	T_{total}	Num_{Nodes}	T_{total}	Num_{Nodes}
1	54	167	**49**	**123**	-8.3%	-26.3%
2	2	7	2	7	12.6%	0.0%
3	**114**	429	155	**288**	36.0%	-32.9%
4	458	6,181	**239**	**2,854**	-47.7%	-53.8%
5	2,113	8,231	**1,989**	**5,119**	-5.9%	-37.8%
6	**17,385**	6,639	36,000[1]	1,424	107.1%[2]	N/A
7	**738**	2,269	805	**1,637**	9.2%	-27.9%
8	**84**	**201**	124	217	47.4%	8.0%
9	**416**	**2,031**	683	3,240	64.2%	59.5%
10	**29**	**97**	107	318	271.2%	227.8%
11	16	60	**12**	**30**	-26.5%	-50.0%
12	3,844	2,914	**1,338**	**561**	-65.2%	-80.7%
13	1,700	3,023	**1,097**	**1,717**	-35.5%	-43.2%
14	435	528	**334**	**361**	-23.2%	-31.6%
15	40	36	**38**	**26**	-4.9%	-27.8%
16	**24,208**	7,436	36,000[1]	5,041	48.7%[2]	N/A
Average	**3,227**	2,516	4,936	1,435	23.7%	-8.3%

[1] No proven optimal solution found within the time limit.
[2] According to the values obtained for variant SR Cuts at the time limit.

There is no clear tendency whether SR cuts improve the performance of the algorithm. On the one hand, the number of processed nodes can be reduced on ten test instances by enabling cut generation, whereas it is increased only on three test instances. Note that for test instances 6 and 16 it is not possible to evaluate the impact of cut generation as the instances could not be solved optimally within the time limit when cut generation is enabled. The average reduction in the number of processed nodes for the remaining test instances amounts to 8.3%. Without the large outlier obtained on test instance 10, this reduction would even amount to 26.5%. On the other hand, the reduction in the number of processed nodes does not consistently translate into shorter running times. SR cuts reduce the running time on eight test instances, and they also increase the running time on eight test instances. This effect can be explained by the results in Table 5.7. When SR cuts are applied, the average number of column generation iterations per node ($\frac{\mathrm{Num}_{\mathrm{Iter}}}{\mathrm{Num}_{\mathrm{Nodes}}}$) increases. At the same time, the LP relaxation of the RMP and the exact pricing problems become more complex by the inclusion of SR cuts, which can be seen by the increase in the average time per column generation iteration for solving the LP relaxation of the RMP ($\frac{\mathrm{Time}_{\mathrm{RMP}}}{\mathrm{Num}_{\mathrm{Iter}}}$, in milliseconds) and by the increase in the average time per call of the exact pricing method ($\frac{\mathrm{Time}_{\mathrm{EP}}}{\mathrm{Num}_{\mathrm{EP}}}$, in milliseconds). The latter effect can be observed particularly on test instances 6 and 16, the two test instances with the highest average number of generated cuts per node ($\frac{\mathrm{Num}_{\mathrm{Cuts}}}{\mathrm{Num}_{\mathrm{Nodes}}}$). Here, the solution times of Gurobi rise dramatically for some exact pricing problems due to the complexity induced by the large number of SR cuts. These results suggest that, in principle, SR cuts have the potential to accelerate the algorithm, but adding too many of them is detrimental. A more successful strategy could be obtained by adding only a subset of the violated SR inequalities to the RMP such that, on the one hand, the size of the LP relaxation of the RMP and the resulting exact pricing problems is manageable and, on the other hand, still a significant improvement in the number of processed nodes is achieved. Further research is required to investigate such an approach.

5.5.5 Comparison with Gurobi

We compare the running time of the proposed branch-and-price algorithm with the running time we obtain when we solve the compact formulation $(COMP)$ using the general purpose MIP solver Gurobi. To ensure a fair comparison, we extend model $(COMP)$ as follows.

We add symmetry breaking constraints to sort the day clusters within each week by the smallest customer index:

$$\sum_{i\in B} v_{ib}^{d} \leqslant \sum_{i\in B} \sum_{b'=1}^{b-1} v_{ib'}^{d-1} \qquad b \in B\setminus\{1\}, w \in W, d \in D(w)\setminus\{(w-1)m+1\} \tag{5.71}$$

Constraints (5.71) imply that variables v_{ib}^{d} can be fixed to zero for all $i \in B$ and $b < ((d-1) \bmod m)+1$.

Table 5.7: Impact of cut generation: In-depth analysis

Instance no.	No Cuts			SR Cuts			
	$\frac{Num_{Iter}}{Num_{Nodes}}$	$\frac{Time_{EP}}{Num_{EP}}$	$\frac{Time_{RMP}}{Num_{Iter}}$	$\frac{Num_{Iter}}{Num_{Nodes}}$	$\frac{Time_{EP}}{Num_{EP}}$	$\frac{Time_{RMP}}{Num_{Iter}}$	$\frac{Num_{Cuts}}{Num_{Nodes}}$
1	8.0	41.7	18.5	10.3	42.7	17.8	89.8
2	7.6	38.5	9.8	7.9	42.5	9.8	0.4
3	8.4	53.4	12.1	11.5	113.5	14.3	58.0
4	3.9	34.4	3.9	3.9	35.0	4.4	2.1
5	8.2	47.5	14.2	10.0	67.1	15.1	78.9
6	19.6	137.5	81.4	23.4	89,963.5	181.0	718.0
7	9.0	56.1	17.2	11.5	82.8	17.3	64.0
8	7.8	48.9	30.2	11.7	84.8	23.4	117.7
9	5.6	54.9	10.7	6.0	55.8	9.1	1.1
10	8.9	44.2	13.7	8.1	45.1	16.4	18.1
11	6.2	42.3	22.3	8.8	42.7	23.2	26.8
12	9.9	81.3	92.0	16.9	132.8	84.9	400.0
13	12.1	69.4	23.5	14.1	70.4	21.3	32.7
14	11.9	61.1	44.7	15.5	65.4	35.7	68.5
15	11.9	73.1	63.3	17.0	79.8	57.5	46.3
16	15.0	107.9	163.3	20.7	20,042.5	228.2	805.0
Average	9.6	62.0	38.8	12.3	6,935.4	47.5	158.0

Based on Proposition 5.1, we fix the service visits of reference customer $b = 1$ as follows:

$$\sum_{i \in B} v_{i1}^{d} = 1 \qquad d \in \{1, mr_1 + 1, 2mr_1 + 1, ..., |D| - mr_1 + 1\} \tag{5.72}$$

Moreover, we warm-start Gurobi with the solution computed by the location-allocation heuristic from Chapter 4 because we use this solution also to obtain an initial set of columns for our branch-and-price algorithm.

Since only very small instances can be solved with Gurobi, we restrict our experiments again to the nine instances with at most 35 customers. We set the time limit for Gurobi to ten hours per instance and its optimality tolerance with respect to the relative MIP gap to 0.01%. Table 5.8 contains for both solution methods their respective running times and objective values. Furthermore, we include the relative MIP gap as reported by Gurobi (Gap) and the relative percentage deviation in running time obtained by using the branch-and-price algorithm instead of Gurobi. A star behind the objective value of Gurobi indicates that Gurobi has found an optimal solution.

While Gurobi is able to solve eight of the nine instances to optimality, it fails to prove optimality on four of these eight instances. The average running time obtained with Gurobi is roughly seven hours, whereas it is only about six minutes for the branch-and-price algorithm. The average relative reduction in the running time amounts to more than 98.1%. These results show the huge benefit of a specially-tailored algorithm over a general purpose MIP solver to solve problem MPSTDP-S*. Our branch-and-price algorithm is able to solve instance sizes to proven optimality that are far out of reach for Gurobi.

Table 5.8: Comparison of the performance of Gurobi and the branch-and-price algorithm

	Gurobi			B&P		Relative deviation
Instance no.	T_{total}	Obj	Gap	T_{total}	Obj	T_{total}
1	1,132	1,908.5*	0.01%	54	1,908.5	-95.27%
2	36,000	1,228.6*	1.91%	2	1,228.6	-100.00%
3	36,000	1,893.7*	0.64%	114	1,893.7	-99.68%
4	12,493	1,702.5*	0.01%	458	1,702.5	-96.34%
5	36,000	2,006.4*	0.17%	2,113	2,006.4	-94.13%
8	24,468	2,070.6*	0.01%	84	2,070.6	-99.66%
9	36,000	1,949.1	2.99%	416	1,946.6	-98.84%
10	36,000	1,714.8*	1.27%	29	1,714.8	-99.92%
11	9,844	2,067.8*	0.00%	16	2,067.8	-99.84%
Average	25,326	1,838.0	0.78%	365	1,837.7	-98.19%

5.6 Conclusions

In this chapter, we studied a highly relevant planning scenario of the scheduling task arising in the context of multi-period service territory design. As far as we are aware, we present the first exact branch-and-price algorithm for this problem. In order to accelerate our algorithm, we introduced a fast heuristic to solve the pricing problems and we presented specially-tailored symmetry reduction techniques. In addition, we adopted well-known techniques from literature, such as pseudocost branching, early branching and subset-row cuts. We performed extensive computational experiments on real-world instances and investigated the impact of the individual techniques. In particular, the symmetry reduction techniques and pseudocost branching have proven to increase the performance of the algorithm significantly. On the contrary, early branching did not show the expected effect and the computational experiments on the subset-row cuts yielded ambivalent results, which necessitates further research. Overall, the results show the effectiveness of our algorithm as all test instances could be solved to proven optimality in reasonable running time. A comparison with the general purpose MIP solver Gurobi revealed that the branch-and-price algorithm reduces running time by over 98.1% on average. This emphasizes the benefit of using a highly specialized algorithm for the problem under study.

6

Districting for Parcel Delivery Services – A Two-Stage Solution Approach and a Real-World Case Study

This chapter studies a real-world problem arising in the context of parcel delivery. Given a heterogeneous set of resources, i.e., different drivers and different vehicles, the problem for each day consists of assigning a driver and a vehicle to each customer requiring service. Two conflicting aspects must be taken into account. On the one hand, service consistency is desirable, meaning that a customer should always be served by the same driver. On the other hand, daily demand fluctuations prohibit fixed resource assignments. With the aim of finding a reasonable compromise between these aspects, we propose a novel two-stage districting approach, which establishes delivery districts in the first stage and adapts them to the daily demand realizations in the second stage. We carry out a case study based on a real-world data set to test the effectiveness of this approach.

6.1 Introduction

Parcel delivery companies, such as UPS, FedEx, and DHL, deliver myriads of packages to customers every working day. In this chapter, we study an important real-world problem arising in this context. Our attention was drawn to this problem in the joint project with our industry partner PTV Group (PTV). We consider a geographical region, which we call the service region. Parcels must be delivered to the customers within the service region on tours starting and ending at a central depot. We assume that a heterogeneous fleet of vehicles with different capacities and a heterogeneous crew of drivers with different contractual working times are available to perform this task. For each working day, the problem consists of deciding which resources, i.e., vehicles and drivers, should be assigned to which customers requiring service on that day. The number of resources to be used is not given in advance, but it is part of the problem to determine their number.

This problem could be treated as a classical vehicle routing problem (VRP), i.e., a VRP could be solved individually for each day. However, there are significant benefits of using a districting approach rather than a vehicle routing approach. Following a districting approach means that customers are grouped into geographically compact delivery districts, which are kept fairly stable over a relatively long period of time, and that the same driver is responsible for serving all customers in a district during this time period (for a general introduction to districting we refer the reader to Kalcsics, 2015). Hence, districting implicitly ensures long-term service consistency, which entails two important advantages. First, drivers become familiar with their delivery districts, which increases their efficiency in providing service to the customers (Smilowitz et al., 2013; Zhong et al., 2007). Second, having always the same driver visit a certain customer improves customer relations by establishing a personal connection between drivers and customers (Groër et al., 2009) and increases customer satisfaction (Jarrah and Bard, 2012). Moreover, as we argue in Chapter 4, geographical compactness provides high flexibility with respect to the sequence in which the customers of a district can be visited without overly increasing the distance traveled. This can be highly advantageous when unplanned events, e.g., traffic congestions, necessitate an ad-hoc modification of a tour.

Since customer demand is fluctuating, there is, however, a trade-off between strictly holding on to the delivery districts and adapting them to the concrete demand realization of a particular day. To account for demand fluctuations, we treat the problem as a two-stage problem as it is common in practice (Wong, 2008), with the two stages corresponding to different planning levels. On both planning levels, we are faced with conflicting objectives, between which a reasonable compromise must be found. (1) On a tactical planning level, the service region must be subdivided into an adequate number of delivery districts, and resources must be assigned to each district. A reasonable trade-off must be found between the number of districts, and, hence, the number of required resources, and the expected workload of the districts. (2) On an operational level, i.e., on a day-to-day basis, districts must be adapted to the concrete demand realization of a day while preserving the resource assignments made at the tactical level. Here, a good compromise between service consistency and working time related objectives, such as compliance with the drivers' contractual working times and workload balance between the drivers, must be found.

The main contributions of this chapter are as follows:

- We deal with a real-world districting problem that involves the determination of the number of districts and the assignment of heterogeneous resources. This combination has, to the best of our knowledge, not been considered in the districting literature before.
- Analogously to the two-stage nature of the problem, we propose a novel and effective two-stage solution approach.
- We present three integer programming (IP) models for the tactical planning problem, which differ in the level of detail of their input data and in their expected compliance with the drivers' contractual working times. Moreover, we present a heuristic solution procedure for the tactical problem.
- For the operational level, we propose a mixed integer programming (MIP) model aiming at the adaptation of the tactical districting solution to the concrete demand realization of a day.
- We perform an extensive case study based on real-world data to test the effectiveness of our approach. In particular, we analyze the suitability of the three tactical planning models and investigate the trade-off between compliance with the drivers' contractual working times and service consistency.

The chapter is organized as follows. In Section 6.2, we describe in detail the problem under study. We review related literature in Section 6.3, and give a brief overview of our two-stage solution approach in Section 6.4. In Sections 6.5 and 6.6, we present the details of our solution approach related to the tactical and the operational problem, respectively. The procedure used to evaluate solutions is explained in Section 6.7. Section 6.8 contains the results of the case study. We provide some concluding remarks in Section 6.9.

6.2 Problem Description

We consider a service region that contains the set of *basic areas* $B = \{1, ..., |B|\}$. Each basic area $b \in B$ represents a geographical area in the service region, e.g., a zip code area. For basic areas $b, i \in B$, $c_{bi} \in \mathbb{R}^+$ denotes the distance between b and i. Moreover, $A_b \subseteq B$ defines the set of basic areas that are adjacent to basic area $b \in B$.

Related to the drivers and the vehicles that are available for serving the customers in the service region, we have the following input data. There is a given set of *driver types* $D = \{1, ..., |D|\}$. Note that we do not consider individual drivers. Instead, we subsume all drivers with identical characteristics under the same driver type. Driver types distinguish themselves by the drivers' contractual daily working times in relation to a full-time driver. The contractual working time of a full-time driver is given by $t^{max} \in \mathbb{R}^+$, and the relative contractual working time of driver type $d \in D$, expressed as the percentage of t^{max}, is denoted as $r_d \in (0, 100]$. The number of available drivers of type $d \in D$ is given by $M_d \in \mathbb{N}^+$. The driver types are totally ordered, i.e., we know for each pair of driver types d_1, $d_2 \in D$ which type is preferred from an economic or operational point of view (e.g., permanent employees might be preferred over contract workers) and, thus, should be used with higher priority. We assume without loss of generality that this preference is reflected by the index, i.e., driver type d_1 is preferred over driver type d_2 if and only if $d_1 < d_2$. Furthermore, we consider a set of *vehicle types* $V = \{1, ..., |V|\}$, each of which represents a distinct vehicle capacity. The capacity of a vehicle of type $v \in V$ is given by $C_v \in \mathbb{R}^+$. $N_v \in \mathbb{N}^+$ denotes the number of available vehicles of type $v \in V$. Analogously to the driver types, we have a total order expressing the preference of vehicle types from an economic or operational point of view (e.g., own vehicles might be preferred over vehicles that are available for leasing). We assume that vehicle type $v_1 \in V$ is preferred over vehicle type $v_2 \in V$ if and only if $v_1 < v_2$.

Moreover, we have demand data in the form of *customer orders* $O = \{1, ..., |O|\}$. Ideally, this data is available as forecast data, i.e., it represents expected future demand. The data is given for the set of days T, which is partitioned into *tactical sample days* $T_1 = \{1, ..., |T_1|\}$, consisting of the first $|T_1|$ days of T, and *operational sample days* $T_2 = \{|T_1| + 1, ..., |T_1| + |T_2|\}$, consisting of the remaining $|T_2|$ days. Data belonging to the tactical sample days will be used for the tactical design of the delivery districts, while data belonging to the operational sample days will act as concrete demand realizations according to which the tactical design must be adapted in day-to-day operations. Under a customer order we subsume all parcels with the same delivery address and the same delivery day. For each customer order $o \in O$ we know the day of delivery $\tau_o \in T$, the total weight of the parcel(s) $l_o \in \mathbb{R}^+$, the delivery location, and the basic area $b_o \in B$ that contains the delivery location. Additionally, we are given the service time $s_o \in \mathbb{R}^+$ required for each customer order $o \in O$.

Each delivery tour starts and ends at a given depot. If the capacity of a vehicle does not suffice to serve all customers that require service by the vehicle on a single tour, additional tours have to be performed, i.e., the vehicle must return to the depot, must be reloaded, and another delivery tour has to be made. For each extra tour, a vehicle reloading time,

$t^{reload} \in \mathbb{R}^+$, is incurred. If more than one tour per vehicle and day is made, we refer to this as a *multi-tour*.

6.2.1 Tactical Design

In the tactical planning problem, the task is to partition the set of basic areas B into *delivery districts*, and to assign a driver type $d \in D$ and a vehicle type $v \in V$ to each district. The data basis for this task is related to the tactical sample days T_1. The maximum number of drivers M_d of each driver type $d \in D$ and the maximum number of vehicles N_v of each vehicle type $v \in V$ must not be exceeded. Moreover, the preference of driver types must be respected, i.e., a driver of type $d \in D$ may only be used if for each driver type $d' \in D$ with $d' < d$ all $M_{d'}$ available drivers are also used. This applies analogously to the vehicle types, i.e., a vehicle of type $v \in V$ may only be assigned to a district if for each vehicle type $v' \in V$ with $v' < v$ all $N_{v'}$ available vehicles are also assigned to a district. The districts are supposed to be contiguous and geographically compact as this facilitates the construction of short delivery tours. The number of delivery districts is not given in advance, but it is part of the problem to determine an adequate number. On the one hand, the number of districts has to be sufficiently high such that the size of the districts allows each driver to serve the customers in his district without working overtime. On the other hand, establishing more districts than required to meet the demand results in a low utilization of the assigned resources and, hence, is inefficient. Consequently, it is important to find a reasonable trade-off between compliance with the drivers' contractual working times and resource efficiency.

6.2.2 Operational Adaptation

On the operational level, we consider each of the operational sample days T_2 individually. The task is now to adapt the district design to the demand on day $\tau \in T_2$. For this purpose, we are allowed to modify the assignments of basic areas to delivery districts. However, we must not change the assignments of driver types to delivery districts as this would eliminate consistency. Furthermore, the vehicle type that is assigned in the tactical decision must be preserved. This means that we have to partition the set of basic areas B into geographically compact and contiguous delivery districts while respecting the decisions that were made at the tactical level with regard to the assignments of driver and vehicle types. Since the tactical district design is given, this can also be viewed as a reassignment decision.

As in the tactical problem, we are faced with conflicting goals: On the one hand, we strive for consistency, which implies that no or only few basic areas should be reassigned. On the other hand, depending on the concrete demand realization, sticking to the tactical district design might result in substantial overtime for the drivers or to a highly unbalanced workload between the drivers, both of which is undesirable in practice. Hence, we must find a reasonable trade-off between consistency and working time related objectives.

6.3 Related Work

We restrict our literature review to districting approaches for vehicle routing applications. These approaches correspond to the well-known cluster-first route-second scheme. Furthermore, we focus on applications in which demand uncertainty plays an important role. In such a setting, districting approaches are particularly attractive as they do not only implicitly provide service consistency, but also entail administrative convenience and facilitate daily route planning (Wong and Beasley, 1984). Note, however, that there also exist vehicle routing approaches that explicitly consider service consistency instead of using a cluster-first route-second procedure (e.g., Coelho et al., 2012; Groër et al., 2009; Kovacs et al., 2014, 2015a,b; Luo et al., 2015; Smilowitz et al., 2013; Sungur et al., 2010).

The review is divided into papers focusing on the tactical design of districts and papers dealing with the operational adaptation of an existing district design.

6.3.1 Tactical Design

Since there exists a lot of literature on the tactical districting task, we categorize it based on the presence of the following planning criteria: the determination of the number of districts (criterion DND) and the assignment of heterogeneous resources, i.e., resources with different capacities, to the districts (criterion RA).

Neither DND nor RA

Wong and Beasley (1984) construct delivery districts with one vehicle being responsible for the customers in each district. The authors present a simple heuristic based on VRP solutions of sample days: Initially, customers are allocated to seed customers, and interchanges are performed subsequently to improve the solution. The evaluation is based on routing solutions obtained for each day. Districts are fixed, i.e., reassignments are not allowed, which can result in demand exceeding vehicle capacity. In contrast to our problem, the number of districts is given in advance and all vehicles have the same capacity, and, hence, their problem does not involve the assignment of heterogeneous resources to districts.

Galvão et al. (2006) present an approach based on Voronoi diagrams to smooth the district contours obtained with an improved version of the ring-radial model of Novaes et al. (2000, see category "DND and RA" on page 116). The selected vehicle type and the number of districts of the initial solution are left unchanged.

Given a density function describing stochastic customer locations, Ouyang (2007) study the problem of partitioning an area into "zones" (analogous to districts) such that the expected total travel distance is minimized and the expected number of customers in each zone equals a predetermined value. To this end, the authors follow continuous approximation guidelines and combine several spatial partitioning techniques. They note that their algorithm can also be used to balance the number of (discrete) customers after information about the actual customers becomes available. In contrast to our problem, the number of districts is implicitly given by the number of customers per zone. Moreover, since the authors try to balance the number of customers evenly between districts, capacities are implicitly assumed to be identical in all districts.

Haughton (2008) investigates the impact of fixed districts on routing efficiency and the drivers' learning burden when daily demand fluctuations are present. The results are compared to those obtained using daily route optimization, i.e., without considering any assignment restrictions imposed by districts. Different variants are considered in which each district is assigned either to a single driver or to a team of drivers. Using simulation experiments, the author analyzes the effect of vehicle capacity, team size, and variability in customer demand. The problem of Haughton is different from the one studied in this chapter since the number of districts is given in advance and capacity is identical for all vehicles.

The design of pickup and delivery districts for a parcel company is studied by González-Ramírez et al. (2011). They aim at creating geographically compact districts that are balanced in terms of workload. They present two solution approaches, the first one being based on two MIP models and the second one being a heuristic that combines elements of tabu search and greedy randomized adaptive search procedure (GRASP). Unlike in our problem, the number of districts is predetermined and district capacities are assumed identical.

Carlsson (2012) and Carlsson and Delage (2013) consider the problem of partitioning a region into a given number of districts, each being served by a single vehicle, such that workload is balanced between the districts. They do not solve the problem using discrete models, but assume that customer locations are distributed according to a continuous probability density. Carlsson (2012) further assume that the probability distribution of the customer locations is known. They propose an algorithm that recursively subdivides the region into districts yielding asymptotically the same workload when many samples from the probability density are drawn. Carlsson and Delage (2013) consider the case that the exact probability distribution of the customer locations is unknown at the time of partitioning. They seek the partition yielding equal worst-case workloads in all districts and present two branch-and-bound algorithms. The main difference to our problem is that the authors of both papers assume the number of districts to be given in advance, whereas in our problem the number of districts is an outcome of optimization. Furthermore, since they try to balance workload evenly between districts, they implicitly assume identical capacities for all districts.

Most closely related to our problem are the works of Zhong et al. (2007) and Schneider et al. (2015), as both try to find a reasonable trade-off between consistency and routing flexibility in the context of parcel delivery. Zhong et al. (2007) introduce the concept of "core areas" corresponding to (partial) districts. All cells (corresponding to basic areas in our terminology) assigned to a core area must be served by the same driver every day. All other cells are free to be assigned to any core area or even to extra drivers, i.e., drivers not associated with a core area, on a day-to-day basis. The authors focus on the effect of driver learning when a cell is repeatedly visited by the same driver. They assume that the average time spent per stop decreases with an increasing number of continued visits to a cell. A two-stage approach is proposed consisting of tabu search for the strategic core area design and a method based on a parallel insertion heuristic for operational cell routing. In the second stage, the authors explicitly consider the drivers' learning effects such that cells are preferably assigned to more familiar drivers, hence, supporting consistency also on the operational level. The major differences to our problem are due to the following reasons. The number of districts is fixed in advance based on historical data, and the authors assume that vehicle capacity is sufficient to deliver all packages on each tour. Even though different maximum working durations for the drivers can be considered, they are associated in a fixed manner with the seed points of the core areas, which are determined in a preprocessing step. In the operational model, the authors further assume that extra drivers can be utilized if needed.

Schneider et al. (2015) also propose a two-stage solution approach. In the first stage, based on vehicle routing solutions computed for historical sample days, they create (partial) districts, each corresponding to one vehicle and containing a certain portion of the customers. More precisely, they iteratively assign customers to one of the selected seed customers until each district contains a certain percentage of the expected customer demand. In the second stage, they compute routes for a concrete day by solving a VRP with time windows using a tabu search heuristic which takes into account the district assignments of the first stage. Districts are considered semi-fixed, i.e., it is allowed to reassign a certain number of customers. The objective in the second stage is the minimization of the number of vehicles and of the distance traveled. This means that consistency is not explicitly considered in the second stage. Contrary to the problem studied in this chapter, Schneider et al. do not consider a heterogeneous fleet of vehicles or drivers with different working times. Moreover, the number of districts is fixed in advance based on the solutions of vehicle routing problems solved for sample days.

DND Only

Daganzo and Erera (1999) and Erera (2000) address the design of delivery districts in the presence of uncertainty by means of continuous approximation models. Besides insights on near-optimal fixed district designs, the authors analyze different strategies to deal with actual demand realizations in day-to-day operations considering load and time constraints. These strategies include "sweeper tours", i.e., secondary tours containing all unserved customers of the initial tours, and more sophisticated schemes involving the dynamic coordination of vehicles in real time. The authors assume identical capacities in terms of load and time.

Haugland et al. (2007) deal with the problem of designing delivery districts. Demand is revealed only after the districting decision has been made. Given an upper bound on the travel cost within each district, the goal is to minimize the expected total travel cost. To this end, the authors propose a tabu search and a multi-start heuristic. As opposed to our problem, the authors assume a homogeneous fleet of vehicles and identical upper bounds on the districts' travel cost.

Bard and Jarrah (2009) deal with a problem in the context of pickup and delivery operations. They aim at partitioning a service area into the minimum number of clusters such that the expected workload and weight in each cluster can be handled by the capacity of a single vehicle within a driver's available working time. To solve this problem, they propose a heuristic, which iteratively subdivides the area into rectangles (corresponding to clusters or districts) with approximately the same number of pickups and deliveries per day. To further improve solution quality, they solve a set-partitioning problem based on the previously generated clusters. Furthermore, they increase computational efficiency by aggregating nearby customers in a preprocessing step. The clusters generated by their method are evaluated in Bard et al. (2010) using a Monte Carlo simulation. In particular, the probability of route failures, i.e., the exceeding of vehicle capacities or tour duration limits, are assessed; reassignments of customers are, however, not considered. Neither drivers with different contractual working times nor a heterogeneous fleet of vehicles are considered. Also, the possibility that a vehicle performs more than one tour on a day is not taken into account.

Lei et al. (2012) and Lei et al. (2016) consider two sets of customers, namely regular (deterministic) and stochastic customers, with the locations and presence of the latter customers being uncertain. The problem they study is to design contiguous districts with one vehicle being responsible for serving all customers within a district. The duration of each tour is limited, and exceeding this limit causes overtime cost. The authors model the problem as a two-stage stochastic program and consider several objectives, including the number of districts and the expected routing costs. Lei et al. (2012) consider a single-period setting and devise a large neighborhood search. In contrast, Lei et al. (2016) study a multi-period setting, in which the regular customers vary dynamically over the time periods, and propose a multi-objective evolutionary algorithm. Unlike in our problem, the authors of both papers do not consider vehicle capacities and assume that the tour duration limit is the same for all districts.

DND and RA

With the objective of minimizing total daily transportation cost, Novaes and Graciolli (1999) and Novaes et al. (2000) study the design of delivery districts in combination with the determination of the vehicle fleet. The authors partition the service region into districts using a ring-radial pattern and approximation formulas to compute expected tour lengths. Although different vehicle capacities and operating costs can be taken into account by their models, only solutions consisting of a homogeneous fleet of vehicles can be obtained with their approaches, which is a restriction compared to the approach presented in this chapter. Moreover, normal working times are assumed to be identical for all drivers. Note that other authors have also studied the problem of determining the best fleet size and mix (see Jabali et al., 2012; Nourinejad and Roorda, 2017), but their works are geared towards strategic decisions related to vehicle acquisition rather than the selection of the best mix of vehicles from a given fleet.

In a follow-up paper of Bard and Jarrah (2009, see category "DND Only" on page 115), Jarrah and Bard (2012) propose a novel approach in which they construct "subregions" of the total area and restrict the customers of a cluster (corresponding to a district) to a subset of the customers within a subregion. Moreover, clusters are required to contain all customers within symmetric rectangles centered at previously selected seed customers. The authors pre-aggregate customers as in Bard and Jarrah (2009) and create clusters through a column generation approach. Instead of solving the entire set of pricing problems in each iteration, they use a tabu list to determine the subproblems to be solved. A variable fixing procedure is employed to find feasible integer solutions. Drivers with different contractual working times or the possibility that a vehicle performs more than one tour on a day are not considered. The approach supports, in principle, clusters of various capacities corresponding to the capacities of different available vehicles, but the authors state that they do not exploit this design feature and consider only test data with a homogeneous fleet.

Summary

Table 6.1 summarizes the main difference between the problem studied in this chapter and the existing literature related to tactical district design. First, there is a stream of literature assuming that the number of districts is given in advance, which is not the case in our problem. Second, some authors regard the determination of the number of districts as part of the optimization problem, but assume identical resource capacities (working times, vehicle capacities) for all districts, and, thus, do not consider the assignment of heterogeneous resources to districts as part of the problem. Third, a few authors basically address different vehicle types, but either they do not further elaborate on this aspect or their approaches can only generate solutions consisting of a homogeneous fleet. Hence, to the best of our knowledge, we are the first to combine the determination of the number of districts with the assignment of heterogeneous resources.

Table 6.1: Overview of planning criteria considered in the related literature on the tactical districting task. The works are categorized according to the combination of considered planning criteria and are sorted chronologically in each category.

Work	DND	RA	Application
Wong and Beasley (1984)	–	–	Distribution operations
Galvão et al. (2006)	–	–	Distribution operations with an example in parcel delivery
Ouyang (2007)	–	–	Distribution operations
Zhong et al. (2007)	–	–	Parcel distribution
Haughton (2008)	–	–	Distribution operations
González-Ramírez et al. (2011)	–	–	Parcel distribution
Carlsson (2012)	–	–	Distribution operations with an example in mail delivery
Carlsson and Delage (2013)	–	–	Distribution operations with an example in parcel delivery
Schneider et al. (2015)	–	–	Parcel distribution
Daganzo and Erera (1999)	✓	–	Distribution operations
Erera (2000)	✓	–	Distribution operations
Haugland et al. (2007)	✓	–	Distribution operations
Bard and Jarrah (2009)	✓	–	Pickup and delivery operations
Lei et al. (2012)	✓	–	Parcel distribution
Lei et al. (2016)	✓	–	Marketing and distribution
Novaes and Graciolli (1999)	✓	(✓)[1]	Distribution operations with an example in parcel delivery
Novaes et al. (2000)	✓	(✓)[1]	Distribution operations with an example in parcel delivery
Jarrah and Bard (2012)	✓	(✓)[2]	Pickup and delivery operations
This chapter	✓	✓	Parcel distribution

[1] Although different vehicle types can be considered, solutions always consist of a homogeneous fleet.
[2] Although different vehicle types are supported in principle, this design feature is not exploited.

6.3.2 Operational Adaptation

As far as we are aware, there is only one paper dedicatedly related to the adaptation of a districting plan on a day-to-day basis. Janssens et al. (2015) address the situation where a tactical plan is available and must be modified to meet the actual demand of a day. They assume that the service region is divided into "microzones" (corresponding to basic areas in our terminology) and that each microzone is assigned to a vehicle in the tactical plan reflecting the preferred assignment. When an estimate of the actual workload is available in the daily business, microzones must be reassigned to vehicles such that balanced and feasible tours are achieved while taking into account the assignments of the tactical plan. The authors consider three objectives, namely total transportation cost, deviation from the tactical plan, and workload imbalance, and propose a multi-neighborhood tabu search heuristic. As opposed to our problem, the authors assume that vehicles are not capacity-constrained, which is an unrealistic assumption for the real-world data set that we consider in this chapter. Other authors who deal with the (infrequent) adaptation of an existing districting plan enforce a certain degree of similarity with the original plan by restricting the number of reassignments (e.g., Caro et al., 2004; Ríos-Mercado and López-Pérez, 2013).

6.4 Overview of Solution Approach and Evaluation Stage

In this section, we provide a brief overview of our solution approach and the way in which we evaluate the quality of the solutions obtained with this approach. The overview is illustrated in Figure 6.1.

Analogously to the two-stage nature of the problem, our solution approach is also divided into two stages. In the first stage, we address the problem of designing delivery districts on the tactical level, i.e., we partition the set of basic areas into delivery districts and assign resources to each district. The data basis for this stage consists of demand data for the tactical sample days. We go into the details of the first stage in the next section.

The tactical solution obtained in the first stage is then used as input for the second stage. In this stage, we adapt the tactical solution to the demand of the operational sample days. More precisely, we treat each operational sample day as a concrete demand realization of a day and reassign basic areas accordingly. The resource assignments of the tactical solution are fixed. We give an in-depth explanation of this stage in Section 6.6.

Finally, the operational solution of stage two is passed to the evaluation stage, in which we assess its quality. Please note that we consider only estimations of the workload in

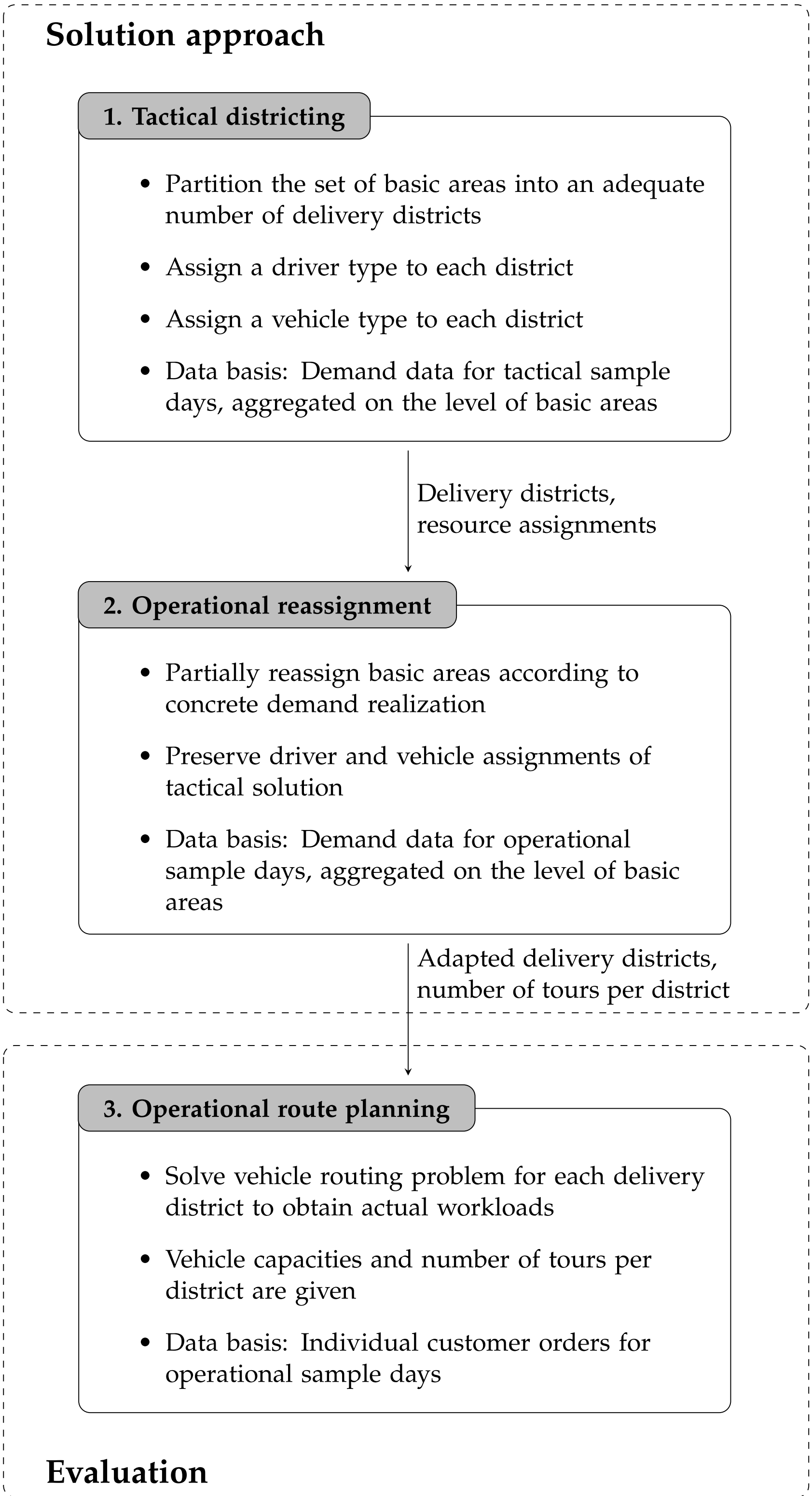


Figure 6.1: Overview of two-stage solution approach and evaluation stage

stages one and two. In order to evaluate a solution based on its actual workload, we solve a vehicle routing problem for each delivery district. The capacity of the vehicle in each district is given by the tactical solution. The number of delivery tours that must be performed by each vehicle follows from the actual weight that must be transported, and is, hence, a result of the operational solution. We further elaborate on the evaluation procedure in Section 6.7.

6.5 Solution Approach Stage 1: Tactical Districting

In this section, we describe in detail the first stage of our solution approach. We propose three different IP models for the tactical design problem and present a heuristic solution procedure. On top of the notation presented in Section 6.2, we start by introducing additional notation that is common to all three tactical models. The entire notation that is used to formulate the models for tactical districting and operational reassignment is summarized in Table 6.3 at the end of Section 6.6.

6.5.1 Notation Common to All Tactical Models

In all three IP models, geographical compactness is measured as the sum of the distances between all basic areas that belong to a certain district and the basic area which is selected as the district center. Such a center-based approach to measure compactness is quite common in the literature on districting (see, e.g., Fleischmann and Paraschis, 1988; Hess et al., 1965; Ríos-Mercado and López-Pérez, 2013; Salazar-Aguilar et al., 2011), and can relatively easily be handled by modern general-purpose MIP solvers.

We introduce the following decision variables:

$$x_{bi} = \begin{cases} 1 & \text{if basic area } b \in B \text{ is assigned to the delivery district represented by} \\ & \text{center } i \in B \\ 0 & \text{otherwise} \end{cases}$$

$$y_{di} = \begin{cases} 1 & \text{if driver type } d \in D \text{ is assigned to the delivery district represented} \\ & \text{by center } i \in B \\ 0 & \text{otherwise} \end{cases}$$

Note that $x_{ii} = 1$ implies that $i \in B$ is selected as a district center. Decision variables describing the assignments of vehicle types to districts will be introduced in the subsequent sections specifically for the individual models.

Additionally, we introduce the following auxiliary variables which are required to incorporate the preference criteria with respect to different driver and vehicle types:

$$e_d = \begin{cases} 1 & \text{if all available drivers of type } d \in D \text{ are assigned to a delivery district} \\ 0 & \text{otherwise} \end{cases}$$

$$f_v = \begin{cases} 1 & \text{if all available vehicles of type } v \in V \text{ are assigned to a delivery district} \\ 0 & \text{otherwise} \end{cases}$$

Besides these variables, we introduce the following additional parameters. We denote by $l_b^\tau = \sum_{o \in O, \tau_o = \tau, b_o = b} l_o$ the total weight of the parcels to be transported to customers in basic area $b \in B$ on day $\tau \in T$. Furthermore, $w_b^\tau \in \mathbb{R}^+$ states the estimated workload of basic area $b \in B$ on day $\tau \in T$ *within* the delivery district: It consists of the total service time $s_b^\tau = \sum_{o \in O, \tau_o = \tau, b_o = b} s_o$ and the estimated total travel time within the district that is required to serve all customers in basic area $b \in B$ on day $\tau \in T$. We will explain in Section 6.8.2 how the estimated total travel time of a basic area can be obtained from demand data that is available on the level of individual customer orders. Note that w_b^τ does not include the travel time between the depot and the delivery district. Instead, we estimate this time by the time required to travel between the depot and the basic area $i \in B$ that represents the center of the district. Recall that due to vehicle capacity limitations it might be necessary to perform several tours to a delivery district to meet customer demand. Hence, when $n \in N = \{1, ..., |N|\}$ tours are performed to the district represented by basic area $i \in B$, the travel time between the depot and the district plus the time required to reload the vehicle at the depot is given by $t_{ni} = 2 \cdot n \cdot t_i^{depot} + (n-1) \cdot t^{reload}$, where $t_i^{depot} \in \mathbb{R}^+$ denotes the time required to travel from the depot to basic area i.

6.5.2 Three Models for Tactical Districting

In the following, we present three IP models for the design of delivery districts on the tactical planning level. The models differ in the following two aspects:

1. The models distinguish themselves by the level of detail of their input data.

2. The models differ in the way in which workload limits are taken into account.

The level of detail of the input data relates to the estimated workload of each basic area and to the weight that must be transported to each basic area, both of which can be considered either as average values over the $|T_1|$ tactical planning days or as individual values for each day.

The workload limits restrict the estimated workload of each district to the interval $[\mathrm{LB}, \mathrm{UB}]$, $\mathrm{LB}, \mathrm{UB} \in \mathbb{R}^+, \mathrm{LB} < \mathrm{UB}$. The lower bound LB is incorporated to prevent the models from creating very small districts resulting in an inefficient utilization of resources. The upper bound UB makes sure that the generated districts are sufficiently small such that there is no need for the drivers to work a lot of overtime. In the three models, we consider two variants of workload limits. Both variants force the average daily workload estimation to be greater than or equal to the specified workload lower bound LB, but the variants differ in the way in which the workload upper bound UB is taken into account. In the first variant of the considered workload limits, the average daily workload estimation must be less than or equal to the upper bound. In the second variant, we require the estimated workload of each individual day not to exceed the upper bound.

The distinct characteristics of the three models are summarized in Table 6.2 and can be described as follows:

- Model AV–AW uses the average daily workloads and weights of the basic areas as input data, and applies the workload limits to the average daily workload estimation of the districts. In this model, the number of tours to a district depends on the average daily weight that must be transported to the district.
- Model A/IV–AW takes into account the average daily workloads and the day-specific weights of the basic areas. It applies the workload limits to the average daily workload estimation of the districts. Since day-specific weights are considered in this model, the number of tours to a district varies from day to day and depends on the weight that must be transported to the district on each day.
- Model IV–A/IW considers day-specific workloads and day-specific weights of the basic areas. The workload lower bound is applied to the average daily workload estimation of the districts, whereas the workload upper bound relates to the individual workload estimation for each day. As in model A/IV–AW, the number of tours to a district varies from day to day depending on the weight to be transported.

Table 6.2: Overview of the three proposed models

Model	Basic area input data	District workload limits
AV–AW	**A**verage **V**alue for workload and weight	LB ⩽ **A**verage **W**orkload ⩽ UB
A/IV–AW	**A**verage **V**alue for workload, **I**ndividual **V**alue of each day for weight	LB ⩽ **A**verage **W**orkload ⩽ UB
IV–A/IW	**I**ndividual **V**alue of each day for workload and weight	LB ⩽ **A**verage **W**orkload, **I**ndividual **W**orkload of each day ⩽ UB

Model IV–A/IW is the most conservative of the three models in the sense that we expect it to yield the fewest overtime hours of all models in the evaluation stage due to the very restrictive workload upper bound which limits the workload of each tactical sample day. Beyond that, model A/IV–AW is expected to be more conservative than model AV–AW since it takes into account day-specific weights, whereas model AV–AW considers only average weights, and, thus, fluctuations in weight are leveled out. The effect that we expect from this is that we obtain more workload peaks in solutions computed with model AV–AW than in solutions obtained with model A/IV–AW due to the necessity to perform multi-tours on days in which the total weight in a district exceeds the vehicle capacity. In Section 6.8.3, we will empirically evaluate if the models behave in the expected way.

Model AV–AW

We introduce the following additional parameters: By $\overline{w}_b = \frac{1}{|T_1|} \cdot \sum_{\tau \in T_1} w_b^\tau$ we denote the average daily workload estimation, and by $\overline{l}_b = \frac{1}{|T_1|} \cdot \sum_{\tau \in T_1} l_b^\tau$ we denote the average daily weight in basic area $b \in B$. Furthermore, we define binary decision variables $z_{n\nu i}$:

$$z_{n\nu i} = \begin{cases} 1 & \text{if vehicle type } \nu \in V \text{ is assigned to the district represented by basic area } i \in B \\ & \text{and } n \in N \text{ tours to the district are performed} \\ 0 & \text{otherwise} \end{cases}$$

Using this notation, model AV–AW can be formulated as follows:

$$\sum_{b \in B} \sum_{i \in B} c_{bi} x_{bi} \to \min \tag{6.1}$$

$$\text{s.t.} \quad \sum_{i \in B} x_{bi} = 1 \qquad b \in B \tag{6.2}$$

$$x_{bi} \leqslant x_{ii} \qquad b, i \in B \tag{6.3}$$

$$\sum_{b \in \bigcup_{b' \in S} (A_{b'} \setminus S)} x_{bi} - \sum_{b \in S} x_{bi} \geqslant 1 - |S| \qquad i \in B, S \subseteq B \setminus (\{i\} \cup A_i), \; S \neq \emptyset \tag{6.4}$$

$$\sum_{d \in D} y_{di} = x_{ii} \qquad i \in B \tag{6.5}$$

$$\sum_{i \in B} y_{1i} \leqslant M_1 \tag{6.6}$$

$$\sum_{i \in B} y_{di} \leqslant M_d e_{d-1} \qquad d \in D, d > 1 \tag{6.7}$$

$$\sum_{i \in B} y_{di} \leqslant M_d \sum_{i \in B} y_{d-1,i} - M_d (M_{d-1} - 1) e_{d-1} \qquad d \in D, d > 1 \tag{6.8}$$

$$\sum_{n \in N} \sum_{\nu \in V} z_{n\nu i} = x_{ii} \qquad i \in B \tag{6.9}$$

$$\sum_{n \in N} \sum_{i \in B} z_{n1i} \leqslant N_1 \tag{6.10}$$

$$\sum_{n \in N} \sum_{i \in B} z_{n\nu i} \leqslant N_\nu f_{\nu-1} \qquad \nu \in V, \nu > 1 \tag{6.11}$$

$$\sum_{n \in N} \sum_{i \in B} z_{n\nu i} \leqslant N_\nu \sum_{n \in N} \sum_{i \in B} z_{n,\nu-1,i} - N_\nu (N_{\nu-1} - 1) f_{\nu-1} \qquad \nu \in V, \nu > 1 \tag{6.12}$$

$$\sum_{b \in B} \bar{l}_b x_{bi} \leqslant \sum_{n \in N} \sum_{\nu \in V} n C_\nu z_{n\nu i} \qquad i \in B \tag{6.13}$$

$$n z_{n\nu i} \leqslant \frac{1}{C_\nu} \sum_{b \in B} \bar{l}_b x_{bi} + 0.999 \qquad n \in N, \nu \in V, i \in B \tag{6.14}$$

$$LB \sum_{d \in D} \frac{r_d}{100} y_{di} \leqslant \sum_{b \in B} \overline{w}_b x_{bi} + \sum_{n \in N} \sum_{\nu \in V} t_{ni} z_{n\nu i} \qquad i \in B \tag{6.15}$$

$$UB \sum_{d \in D} \frac{r_d}{100} y_{di} \geqslant \sum_{b \in B} \overline{w}_b x_{bi} + \sum_{n \in N} \sum_{\nu \in V} t_{ni} z_{n\nu i} \qquad i \in B \tag{6.16}$$

$$e_d \in \{0, 1\} \qquad d \in D, d < |D| \tag{6.17}$$

$$f_\nu \in \{0, 1\} \qquad \nu \in V, \nu < |V| \tag{6.18}$$

$$x_{bi} \in \{0, 1\} \qquad b, i \in B \tag{6.19}$$

$$y_{di} \in \{0, 1\} \qquad d \in D, i \in B \tag{6.20}$$

$$z_{n\nu i} \in \{0, 1\} \qquad n \in N, \nu \in V, i \in B \tag{6.21}$$

In the Objective Function (6.1), we optimize geographical compactness by minimizing the sum of the distances between the district centers and their assigned basic areas. Constraints (6.2) make sure that each basic area is assigned to a delivery district, and Constraints (6.3) state that basic areas can only be assigned to delivery districts that are represented by a basic area which is selected as a district center. Constraints (6.4) were proposed by Drexl and Haase (1999) and ensure the contiguity of the delivery districts. Each of these constraints considers a district center $i \in B$ and a non-empty subset S of basic areas which is not adjacent to the district center. If all basic areas of S are assigned to district center i, it is enforced that at least one basic area that is adjacent to S but not contained in S is also assigned to district center i. By Constraints (6.5), a driver type is assigned to each delivery district. Constraints (6.6) ensure that at most the available number of drivers of type $d = 1$ is used. Through Constraints (6.7) and (6.8) we make sure that we use at most the number of available drivers of each type $d > 1$ and that driver type priorities are respected. Both right-hand sides of the constraints for driver type $d \in D$ equal M_d if and only if $e_{d-1} = 1$ and $\sum_{i \in B} y_{d-1,i} = M_{d-1}$, i.e., if all drivers of type $d-1$ with higher priority are used. Constraints (6.9) assign a vehicle type and a number of tours to each delivery district. Constraints (6.10) guarantee that the available number of vehicles of type $d = 1$ is not exceeded. Analogously to (6.7) and (6.8), Constraints (6.11) and (6.12) ensure that at most the number of available vehicles of each type $\nu > 1$ is used and that vehicle type priorities are taken into account. By Constraints (6.13) we make sure that vehicle capacities are not exceeded. Constraints (6.14) limit the number of tours performed to each delivery district to the number of tours required to transport the average daily weight of the district. The right-hand sides of these constraints correspond to a ceiling function applied to the average daily weight to be transported to the district divided by the vehicle capacity. These constraints are necessary to prevent the model from artificially increasing the workload in a district by making more tours to a district than necessary in order to satisfy the workload lower bound. Constraints (6.15) and (6.16) limit the average daily workload estimation of each district to the interval [LB, UB] for a full-time driver ($r_d = 100$) and to correspondingly less for a part-time driver ($r_d < 100$). The average daily workload estimation of a district consists of the average workload estimations for the assigned basic areas and the time required to travel between the depot and the district (including reloading the vehicle). The latter results from the minimum number of tours needed to transport the average daily weight of the district. Finally, Constraints (6.17)–(6.21) define the binary decision variables.

Model A/IV–AW

For model A/IV–AW, we define time-expanded binary decision variables $z^{\tau}_{n\nu i}$:

$$z^{\tau}_{n\nu i} = \begin{cases} 1 & \text{if vehicle type } \nu \in V \text{ is assigned to the district represented by basic area } i \in B \\ & \text{and } n \in N \text{ tours to the district are performed on day } \tau \in T_1 \\ 0 & \text{otherwise} \end{cases}$$

The model is formulated as follows:

$$\sum_{b \in B} \sum_{i \in B} c_{bi} x_{bi} \rightarrow \min \tag{6.22}$$

s.t. (6.2)–(6.8), (6.17)–(6.20)

$$\sum_{n \in N} \sum_{\nu \in V} z^{\tau}_{n\nu i} = x_{ii} \qquad i \in B, \tau \in T_1 \tag{6.23}$$

$$\sum_{n \in N} z^{\tau}_{n\nu i} = \sum_{n \in N} z^{1}_{n\nu i} \qquad \nu \in V, i \in B, \tau \in T_1, \tau > 1 \tag{6.24}$$

$$\sum_{n \in N} \sum_{i \in B} z^{1}_{n1i} \leqslant N_1 \tag{6.25}$$

$$\sum_{n \in N} \sum_{i \in B} z^{1}_{n\nu i} \leqslant N_\nu f_{\nu-1} \qquad \nu \in V, \nu > 1 \tag{6.26}$$

$$\sum_{n \in N} \sum_{i \in B} z^{1}_{n\nu i} \leqslant N_\nu \sum_{n \in N} \sum_{i \in B} z^{1}_{n,\nu-1,i} - N_\nu (N_{\nu-1} - 1) f_{\nu-1} \qquad \nu \in V, \nu > 1 \tag{6.27}$$

$$\sum_{b \in B} l^{\tau}_{b} x_{bi} \leqslant \sum_{n \in N} \sum_{\nu \in V} n C_\nu z^{\tau}_{n\nu i} \qquad i \in B, \tau \in T_1 \tag{6.28}$$

$$n z^{\tau}_{n\nu i} \leqslant \frac{1}{C_\nu} \sum_{b \in B} l^{\tau}_{b} x_{bi} + 0.999 \qquad n \in N, \nu \in V, i \in B, \tau \in T_1 \qquad (6.29)$$

$$LB \sum_{d \in D} \frac{r_d}{100} y_{di} \leqslant \sum_{b \in B} \overline{w}_b x_{bi} + \frac{1}{|T_1|} \sum_{n \in N} \sum_{\nu \in V} \sum_{\tau \in T_1} t_{ni} z^{\tau}_{n\nu i} \qquad i \in B \qquad (6.30)$$

$$UB \sum_{d \in D} \frac{r_d}{100} y_{di} \geqslant \sum_{b \in B} \overline{w}_b x_{bi} + \frac{1}{|T_1|} \sum_{n \in N} \sum_{\nu \in V} \sum_{\tau \in T_1} t_{ni} z^{\tau}_{n\nu i} \qquad i \in B \qquad (6.31)$$

$$z^{\tau}_{n\nu i} \in \{0, 1\} \qquad n \in N, \nu \in V, i \in B, \tau \in T_1 \qquad (6.32)$$

The Objective Function (6.22) is the same as in model AV–AW. Constraints (6.23) make sure that a vehicle type and a number of tours is assigned to each delivery district on each day. Constraints (6.24) guarantee that for each delivery district the same vehicle type is assigned on each day. Constraints (6.25) restrict the maximum number of assigned vehicles of type $\nu = 1$ to the number of available vehicles of this type. Constraints (6.26) and (6.27) prioritize the assignment of different vehicle types and make sure that the number of available vehicles of each type $\nu > 1$ is not exceeded. Vehicle capacity limits are taken into account through Constraints (6.28), and unnecessary tours are prohibited by Constraints (6.29). Constraints (6.30) and (6.31) limit the average daily workload estimation of each district. This estimation consists of the average estimated workloads for the assigned basic areas and the average time required to travel between the depot and the district (including the time for reloading the vehicle). In contrast to model AV–AW, the number of tours to each district is determined for each day individually based on the total weight to be transported on a day. The time-expanded variables for the assignment of vehicles and the number of tours to each district are defined in Constraints (6.32).

Model IV–A/IW

Model IV–A/IW can be stated as follows:

$$\sum_{b \in B} \sum_{i \in B} c_{bi} x_{bi} \rightarrow \min \qquad (6.33)$$

s.t. (6.2)–(6.8),(6.17)–(6.20),(6.23)–(6.30),(6.32)

$$UB \sum_{d \in D} \frac{r_d}{100} y_{di} \geqslant \sum_{b \in B} w^{\tau}_{b} x_{bi} + \sum_{n \in N} \sum_{\nu \in V} t_{ni} z^{\tau}_{n\nu i} \qquad i \in B, \tau \in T_1 \qquad (6.34)$$

The model differs from model A/IV–AW only in one component: Constraints (6.34) replace Constraints (6.31). In contrast to model A/IV–AW, the estimated workload of the districts on each day is bounded above. For each district and day, this estimation contains the day-specific workload estimations for the assigned basic areas and the day-specific travel times between the depot and the district (including reloading times).

6.5.3 Heuristic Solution Approach

In preliminary tests, we tried to solve the tactical planning models using the MIP solver Gurobi 7.0.2. We succeeded in solving 50-basic area instances to near-optimality, i.e., to an optimality gap not greater than 1%. In three of the four preliminary test runs, we observed running times of just a couple of minutes, whereas one test run took roughly four hours. However, in two test runs on 80-basic area instances, we could not even find a feasible solution within ten hours of running time using model A/IV–AW. Since the instances in our case study comprise over 250 basic areas, we opted to devise a heuristic solution approach.

The heuristic is based on the work of Hess et al. (1965) who tackle a political districting problem. They decompose the problem into two subproblems, which they solve iteratively until the solution converges. The first subproblem consists of locating a given number of district centers, while the second subproblem deals with the allocation of basic areas to the centers. We adopt this iterative approach to improve the computational tractability of our three tactical models. More specifically, we restrict the set of district centers in each iteration to a subset $I \subset B$ of all basic areas with $|I| = p$ being a predetermined number. Since, in contrast to Hess et al., we do not know a priori the number of required districts, our models can decide to use only a subset of the centers and, hence, establish fewer than p districts. Consequently, we refer to the set I as *potential district centers*. If a potential district center is not used, i.e., if $x_{ii} = 0$ for some $i \in I$, the center is said to be a *closed district center*. Otherwise, it is said to be an *open district center*.

The iterative procedure of Hess et al. (1965) adapted to our problem is illustrated in Figure 6.2. In the following, we address each step of the heuristic individually.

0. Determine number of potential district centers The number of potential district centers p is determined in a preprocessing step and should have the following properties: On the one hand, p should be as small as possible to minimize the computational burden needed to solve the resulting IP model. On the other hand, p must be large enough such that the demand in the service region can be accommodated with a corresponding number of delivery districts. Since human planners typically know from experience the rough number of districts needed for a particular service region, a good choice is to set p to a value slightly greater than the human planner's estimate. Another way would be to solve vehicle routing problems for a set of sample days and set p to a value slightly greater than

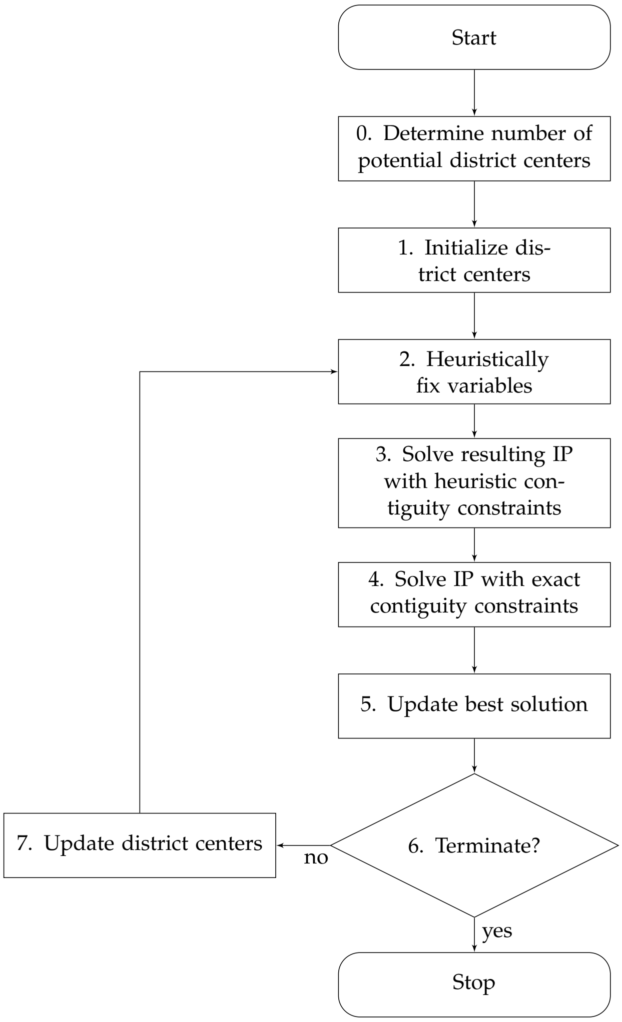


Figure 6.2: Flowchart of heuristic solution approach

the number of vehicles needed to serve the customers on each sample day, which is similar to the procedure proposed by Schneider et al. (2015).

1. Initialize district centers Knowing the number of potential districts, we strive to distribute the p initial district centers over the entire service region. For this purpose, we use the seeding technique proposed by Arthur and Vassilvitskii (2007) in the context of cluster analysis to select p centers from the set of basic areas B. This means that we select the first center uniformly at random and the remaining centers according to the following probabilistic scheme. A basic area is chosen as an additional center with a probability that is proportional to the squared distance between the basic area and the nearest center already selected. We repeat the latter step until p centers have been selected. Models AV–AW, A/IV–AW, and IV–A/IW are then adapted to take into account the set of potential district centers I. This means that Constraints (6.19)–(6.21) and (6.32) are modified as follows:

$$x_{bi} \in \{0,1\} \qquad b \in B, i \in I \tag{6.19a}$$

$$y_{di} \in \{0,1\} \qquad d \in D, i \in I \tag{6.20a}$$

$$z_{nvi} \in \{0,1\} \qquad n \in N, v \in V, i \in I \tag{6.21a}$$

$$z^{\tau}_{nvi} \in \{0,1\} \qquad n \in N, v \in V, i \in I, \tau \in T_1 \tag{6.32a}$$

Furthermore, the domain of index i in all remaining constraints of the three models is also restricted to the set I.

2. Heuristically fix variables We use an approach similar to the one presented in Ríos-Mercado and López-Pérez (2013) to heuristically eliminate some of the x_{bi} variables. The basic idea is to forbid assignments of basic areas to centers that are far away. To this end, we define for each potential center $i \in I$ the set $B_i^{avail} \subseteq B$ of basic areas that are available for being assigned to the center. For a given center i, this procedure is illustrated by the pseudocode of Algorithm 6.1. First, we define a workload threshold $w_{thresh} = \alpha \cdot t^{max}$ with $\alpha \in (0, \infty)$ being a user parameter. Then, we initialize the cumulative workload w_{cum} with the time needed to travel between the depot and the district center if one tour to the district is performed. Next, we sort all basic areas in non-decreasing order of their distances $\{c_{bi}\}$ and iterate one by one over the elements of this sorted set. As long as the current cumulative workload w_{cum} is less than the workload threshold w_{thresh}, we add the basic area to the set of available basic areas B_i^{avail} and increase the cumulative workload w_{cum} by the average estimated workload $\overline{w}_b$ of the basic area. When the cumulative workload w_{cum} exceeds the threshold w_{thresh} for the first time, we stop and return the current set of available basic areas B_i^{avail}. The set B_i^{avail} is then used to fix all variables x_{bi} with $b \notin B_i^{avail}$ to zero. This means that we restrict the basic areas for each district center to the nearest basic areas whose estimated total workload (including the travel time from/to the depot) sums up to approximately α times the daily working time t^{max} of a full-time driver.

3. Solve resulting IP with heuristic contiguity constraints In preliminary tests, we observed that it is in some cases very hard to find a feasible solution when exact contiguity constraints (6.4) are used. Therefore, we opted to first solve the models with the heuristic contiguity constraints proposed by Mehrotra et al. (1998) instead of the exact constraints. To this end, we construct a graph $G = (B, E)$ in which the node set B represents the basic areas and the edge set E connects two nodes b, $b' \in B$ if and only if the basic areas are adjacent, i.e., $b \in A_{b'}$. The length of a path in G is defined as the number of edges in the path. The idea of Mehrotra et al. is to enforce each district to be a subtree of a shortest path

Algorithm 6.1 Function to determine the set of basic areas that are available for being assigned to district center $i \in I$

Input: District center $i \in I$, basic areas B, distances $\{c_{bi}\}$, average workloads $\{\overline{w}_b\}$, travel time t_i^{depot} between depot and i, max. working time of a full-time driver t^{max}, user parameter α

Output: Available basic areas $B_i^{avail} \subseteq B$

function DETERMINE_AVAILABLE_BASIC_AREAS(i, B, $\{c_{bi}\}$, $\{\overline{w}_b\}$, t_i^{depot}, t^{max}, α)
  $w_{thresh} = \alpha \cdot t^{max}$
  $w_{cum} = 2 \cdot t_i^{depot}$
  $B_i^{sorted} \leftarrow B$ sorted in non-decreasing order of distances $\{c_{bi}\}$
  for ($b \in B_i^{sorted}$) **do**
    if ($w_{cum} < w_{thresh}$) **then**
      $B_i^{avail} \leftarrow B_i^{avail} \cup \{b\}$
      $w_{cum} \leftarrow w_{cum} + \overline{w}_b$
    else
      return B_i^{avail}
    end if
  end for
end function

tree of G rooted at the district center. We denote by s_{bi} the length of a shortest path from $i \in I$ to $b \in B$ in G. Further, $S_b = \{b' \in B \mid s_{b'i} = s_{bi} - 1, (b, b') \in E\}$ is defined to be the set of basic areas adjacent to $b \in B$ whose shortest path from the district center $i \in I$ includes one edge less than the shortest path from i to b. Then, contiguity is enforced by adding the following constraints:

$$x_{bi} \leqslant \sum_{b' \in S_b} x_{b'i} \qquad i \in I, b \in B \setminus (\{i\} \cup A_i) \tag{6.35}$$

The constraints make sure that basic area b can only be assigned to the district represented by center i if at least one basic area that is adjacent to b and closer to i is also assigned to the district, hence ensuring that the resulting district is a subtree of a shortest path tree. Note that, as Mehrotra et al. also mention, some contiguous districts are rendered infeasible by these constraints.

4. Solve IP with exact contiguity constraints Next, we remove the heuristic contiguity constraints (6.35) from the models and add the exact contiguity constraints (6.4). Since Constraints (6.4) are a relaxation of Constraints (6.35), we use the solution obtained in step 3 to warm-start the models in step 4. As in Drexl and Haase (1999), we add the exact contiguity constraints (6.4) in a cutting plane fashion due to their exponential number which prohibits the incorporation of the entire set of constraints. This means that we check for each integer solution which we obtain if Constraints (6.4) are fulfilled, and, if applicable, add the violated constraints to the models. For the separation of the violated constraints we proceed as described by Ríos-Mercado and López-Pérez (2013). For each non-empty delivery district $DD_i = \{b \in B \mid x_{bi} = 1\}$, $i \in I$, we construct the subgraph $G_i = (DD_i, E(DD_i))$ of G, where the nodes correspond to the basic areas of the district, and two nodes are connected by an edge if and only if the corresponding basic areas are adjacent. Then, we use breadth-first search to identify the r connected components in G_i and the associated sets of basic areas $\{B_i^1, ..., B_i^r\}$. Clearly, if there exist $r > 1$ connected components, contiguity is violated. In this case, we add an additional constraint (6.4) for each B_i^q, $1 \leqslant q \leqslant r$, which is not connected to the district center, i.e., for which $i \notin B_i^q$, with B_i^q taking the place of set S in (6.4).

As empirically evaluated by Salazar-Aguilar et al. (2011) for different models in the context of commercial territory design, their models can be strengthened by adding valid inequalities that prevent single basic areas from being disconnected from their districts. This observation lead us to add the following constraints, which correspond to the special case of Constraints (6.4) for $|S| = 1$, to our models:

$$x_{bi} \leqslant \sum_{b' \in A_b} x_{b'i} \qquad i \in I, b \in B \setminus (\{i\} \cup A_i), \tag{6.36}$$

Constraints (6.36) enforce for each potential district center $i \in I$ and each basic area $b \in B$, $b \neq i$, which is not adjacent to i that it can only be assigned to i if at least one basic area that is adjacent to b is also assigned to i. Due to their polynomial number, the entire set of constraints can be incorporated into the models, i.e., there is no need of using a cutting plane approach.

5. Update best solution We have no guarantee that the objective value improves from one iteration of the heuristic to the next. In fact, the objective value might even worsen since a solution of a certain iteration is not necessarily feasible in the subsequent iteration due to the relocation of district centers. Hence, we check if the solution of the current iteration is better than the best solution found so far. If this is the case, we update the best solution with the solution of the current iteration.

6. Terminate? The heuristic terminates if one of the following conditions is met. (1) The maximum number of iterations $iter_{max}$ is reached. (2) Cycling is observed, i.e., a solution is obtained in the current iteration that has already been found in a previous iteration.

7. Update district centers We update the district centers according to the solution of the current iteration. For each district whose center is not closed, we select from all basic areas that are assigned to it the one which, when picked as the new district center, yields the smallest contribution to the compactness measure used in the objective functions of the three models. District centers that are closed are left unchanged.

6.6 Solution Approach Stage 2: Operational Reassignment

For each operational sample day, we adapt the tactical solution computed in the first stage to the concrete demand of that day. From the tactical solution we derive the following input data for this operational reassignment: By $\Psi \subset B$ we denote the set of open district centers in the tactical solution and by $\Delta_i \subset B$ the set of basic areas in the district represented by $i \in \Psi$. Further, we use $\delta_i \in D$ and $\nu_i \in V$ to denote the driver type and the vehicle type, respectively, that is assigned to the delivery district represented by center $i \in \Psi$ in the tactical solution. With $\omega \in \mathbb{N}_0$ denoting the maximum number of basic area assignments that are allowed to change compared to the tactical solution, the model for the operational reassignment on day $\tau \in T_2$ can be stated as the following MIP formulation:

$$\frac{1}{\sum\limits_{i \in \Psi} \sum\limits_{b \in \Delta_i} c_{bi}} \sum_{b \in B} \sum_{i \in \Psi} (c_{bi} x_{bi}) + w^{max} \to \min \tag{6.37}$$

s.t.

$$w^{max} \geqslant \frac{100/r_{\delta_i}}{t^{max}} \left(\sum_{b \in B} w_b^\tau x_{bi} + \sum_{n \in N} t_{ni} z_{ni} \right) \qquad i \in \Psi \tag{6.38}$$

$$\sum_{i \in \Psi} x_{bi} = 1 \qquad b \in B \tag{6.39}$$

$$x_{ii} = 1 \qquad i \in \Psi \tag{6.40}$$

$$\sum_{b \in \bigcup\limits_{b' \in S} (A_{b'} \setminus S)} x_{bi} - \sum_{b \in S} x_{bi} \geqslant 1 - |S| \qquad i \in \Psi, S \subseteq B \setminus (\{i\} \cup A_i), S \neq \emptyset \tag{6.41}$$

$$\sum_{n \in N} z_{ni} = 1 \qquad i \in \Psi \tag{6.42}$$

$$\sum_{b \in B} l_b^\tau x_{bi} \leqslant \sum_{n \in N} n C_{\nu_i} z_{ni} \qquad i \in \Psi \tag{6.43}$$

$$n z_{ni} \leqslant \frac{1}{C_{\nu_i}} \sum_{b \in B} l_b^\tau x_{bi} + 0.999 \qquad n \in N, i \in \Psi \tag{6.44}$$

$$|B| - \sum_{i \in \Psi} \sum_{b \in \Delta_i} x_{bi} \leqslant \omega \tag{6.45}$$

$$w^{max} \geqslant 0 \tag{6.46}$$

$$x_{bi} \in \{0, 1\} \qquad b \in B, i \in \Psi \tag{6.47}$$

$$z_{ni} \in \{0, 1\} \qquad n \in N, i \in \Psi \tag{6.48}$$

In Objective Function (6.37), we aim at optimizing the sum of two terms. The first term reflects geographical compactness and is normalized to a value of approximately one by dividing by the sum of the distances between district centers and assigned basic areas in the tactical solution. The second term represents the maximum workload over all districts relative to the contractual working time available in each district, or, for short, the maximum relative workload. The contractual working time that is available in each district is predetermined through the driver type that is assigned to the district in the tactical solution and can be computed as $r_{\delta_i}/100 \cdot t^{max}$. The motivation for the second objective is twofold: First, minimizing the maximum relative workload reduces overtime. Second, it leads to an improvement in the workload balance between the drivers. Note that the maximum relative workload typically takes values of approximately one. Hence, we treat the two objectives as equally important.

The constraints of the model have the following meaning. Constraints (6.38) in conjunction with the minimization objective take care that variable w^{max} is set to the maximum relative workload. Constraints (6.39) require that each basic area is assigned to a district center. Constraints (6.40) make sure that the open district centers of the tactical solution remain open. Contiguity is enforced through Constraints (6.41). As in the three models for the tactical problem, we use a cutting plane approach to add these constraints. Furthermore, we strengthen the formulation again by adding the entire set of constraints for the special case $|S| = 1$. Constraints (6.42) guarantee that the number of tours is determined for each district. Note that we omitted the index $v \in V$ representing the vehicle type in variables z_{ni} as the vehicle type is fixed to the type used in the tactical solution. Vehicle capacity limits are enforced in Constraints (6.43), and Constraints (6.44) restrict the number of tours for each district to the number required for the transportation of the district's total weight. Constraints (6.45) ensure that at most ω basic areas are assigned to a district center different from their center in the tactical solution. Hence, this constraint allows for controlling consistency. Finally, the domain constraints are given by Constraints (6.46)–(6.48).

For a given tactical solution, we solve this model with different values for ω. This way, we obtain several solutions, each with a different emphasis on consistency.

6.7 Evaluation Stage

In the evaluation stage, we asses the quality of the solutions computed in the second stage. For the calculation of the evaluation measures that we propose in this section, it is necessary to determine the actual workload of each district on each operational sample day. While

Table 6.3: Summary of the notation used in the models of solution stages 1 and 2 (the notation below the dotted lines is used only in the model of stage 2)

Sets	
B	Basic areas
$I \subset B$	Potential district centers
$T = T_1 \cup T_2$	Sample days
T_1	Tactical sample days
T_2	Operational sample days
O	Customer orders
D	Driver types
V	Vehicle types
N	Number of tours
$A_b \subseteq B$	Adjacent basic areas to basic area $b \in B$
$\Psi \subset B$	Open district centers in tactical solution
$\Delta_i \subset B$	Basic areas in district of tactical solution represented by center $i \in \Psi$
Parameters	
$c_{bi} \in \mathbb{R}^+$	Distance between basic areas b and i, $b, i \in B$
$t^{max} \in \mathbb{R}^+$	Contractual working time per day of a full-time driver
$r_d \in (0, 100]$	Relative working time of driver type $d \in D$ in percent
$M_d \in \mathbb{N}^+$	Number of available drivers of type $d \in D$
$C_v \in \mathbb{R}^+$	Capacity of vehicle type $v \in V$
$N_v \in \mathbb{N}^+$	Number of available vehicles of type $v \in V$
$l_b^\tau \in \mathbb{R}^+$	Total weight in basic area $b \in B$ on day $\tau \in T$
$\bar{l}_b \in \mathbb{R}^+$	Average total weight in basic area $b \in B$ per day on tactical sample days
$w_b^\tau \in \mathbb{R}^+$	Estimated total workload (service + travel time) of basic area $b \in B$ on day $\tau \in T$
$\bar{w}_b \in \mathbb{R}^+$	Average estimated total workload of basic area $b \in B$ per day on tactical sample days
$t_{ni} \in \mathbb{R}^+$	Travel time between depot and district represented by basic area $i \in B$ plus reloading time at the depot if $n \in N$ tours to the district are made
$LB, UB \in \mathbb{R}^+$	Lower and upper workload limits
$\delta_i \in D$	Driver type assigned to district represented by center $i \in \Psi$ in tactical solution
$\nu_i \in V$	Vehicle type assigned to district represented by center $i \in \Psi$ in tactical solution
$\omega \in \mathbb{N}_0$	Maximum number of basic areas that may be reassigned compared to tactical solution
Variables	
$x_{bi} \in \{0, 1\}$	Takes a value of 1 if and only if basic area $b \in B$ is assigned to the district represented by center $i \in B$
$y_{di} \in \{0, 1\}$	Takes a value of 1 if and only if driver type $d \in D$ is assigned to the district represented by center $i \in B$
$z_{nvi}^{(\tau)} \in \{0, 1\}$	Takes a value of 1 if and only if vehicle type $v \in V$ performs $n \in N$ tours to the district represented by center $i \in B$ (on day $\tau \in T$)
$e_d \in \{0, 1\}$	Takes a value of 1 if and only if all available drivers of type $d \in D$ are assigned to a district
$f_v \in \{0, 1\}$	Takes a value of 1 if and only if all available vehicles of type $v \in V$ are assigned to a district
$w^{max} \in \mathbb{R}^+$	Maximum relative workload of all districts

the total service time in each district can simply be calculated as the sum of the service times over all customer orders in that district, we need to solve a vehicle routing problem for each district in order to obtain the actual travel time. In the following, we present the vehicle routing model and then define our evaluation measures.

6.7.1 Vehicle Routing Model

The vehicle routing problem arising for each district and day is known as a "multi-trip vehicle routing problem" or "vehicle routing problem with multiple use of vehicles" because it is allowed for a vehicle to make several tours on a day (e.g., Brandão and Mercer, 1998; Taillard et al., 1996). But since we do not consider time restrictions, such as customer time windows or maximum driving time per day, we can model the problem for each district and day as a capacitated vehicle routing problem (CVRP). Vehicle capacities correspond to the loading capacity of the assigned vehicle, and the number of vehicles equals the minimum number of tours required to transport the total weight of the district. The formulation introduced in this section is based on Toth and Vigo (2002). Note that we assume travel times to be symmetric and, thus, present a symmetric version of the CVRP.

The CVRP for day $\tau \in T_2$ and the district represented by center $i \in \Psi$ is defined on a complete graph $G = (N_i^\tau, E)$ with edge set E and node set $N_i^\tau = \{0\} \cup O_i^\tau$, where node 0 denotes the depot and $O_i^\tau \subset O$ represents the set of customer orders in district i of the operational solution with delivery day τ. The travel time for an edge $e \in E$ is given by $t_e \in \mathbb{R}^+$. $\eta(n) \subset E$ represents the set of edges that are incident to node $n \in N_i^\tau$. Given a set of nodes $S \subseteq O_i^\tau$ representing customer orders, $\sigma(S) \subset E$ denotes the set of edges that have both endpoints in S. The minimum number of tours required to serve all customer orders of set $S \subseteq O_i^\tau$ is given by $\gamma(S) = \left\lceil \frac{1}{C_{v_i}} \cdot \sum_{o \in S} l_o \right\rceil$.

Using this notation, the CVRP for day $\tau \in T_2$ and district $i \in \Psi$ can be stated as follows:

$$\sum_{e \in E} t_e x_e \to \min \tag{6.49}$$

$$\text{s.t.} \quad \sum_{e \in \eta(o)} x_e = 2 \qquad o \in O_i^\tau \tag{6.50}$$

$$\sum_{e \in \eta(0)} x_e = 2\gamma(O_i^\tau) \tag{6.51}$$

$$\sum_{e \in \sigma(S)} x_e \leqslant |S| - \gamma(S) \qquad S \subseteq O_i^\tau, S \neq \emptyset \tag{6.52}$$

$$x_e \in \{0, 1, 2\} \qquad e \in \eta(0) \tag{6.53}$$

$$x_e \in \{0, 1\} \qquad e \notin \eta(0) \tag{6.54}$$

In the Objective Function (6.49), the total travel time is minimized. For each node that represents a customer order, Constraints (6.50) make sure that exactly two incident edges are selected. The degree constraint for the depot is given by Constraint (6.51), ensuring that the number of selected edges incident to the depot equals two times the number of required tours. Vehicle capacity limits and subtour elimination are enforced by Constraints (6.52). Due to the exponential number of these constraints, we apply a cut generation approach which adds violated constraints iteratively. Constraints (6.53) and (6.54) define the decision variables. They indicate how many times a particular edge is selected. To allow single-customer tours, edges that are incident to the depot can be selected up to two times, whereas all other edges may be selected at most once.

6.7.2 Evaluation Measures

We introduce the following measures to evaluate solutions. The measures are computed for each solution obtained on a particular operational sample day $\tau \in T_2$ after the operational reassignment with a given value of ω has been carried out.

- *Number of districts* (ND). ND equals the number of open district centers, i.e., $ND = |\Psi|$.
- *Driver consistency* (DC). DC reflects the percentage of customer orders that are carried out by the driver who is intended to serve the corresponding basic area according to the tactical solution. With $\Delta_i^* \subset B$ denoting the basic areas that are assigned to the district represented by center $i \in \Psi$ in the operational solution, i.e., *after* the operational reassignment, this measure is computed as

$$DC = \left(1 - \frac{\sum\limits_{i \in \Psi} \sum\limits_{b \in \Delta_i, b \notin \Delta_i^*} \left|\{o \in O \mid \tau_o = \tau, b_o = b\}\right|}{\left|\{o \in O \mid \tau_o = \tau\}\right|}\right) \cdot 100[\%].$$

- *Operational feasibility* (OF). OF measures the percentage of feasible delivery districts – feasible in the sense that the driver in charge does not have to work overtime in order to satisfy the customer demand. Let $t_i^{act,\tau} \in \mathbb{R}^+$ denote the actual workload of the district represented by center $i \in \Psi$, i.e., $t_i^{act,\tau}$ is equal to the total service and vehicle reloading time plus the actual travel time according to the solution of the corresponding CVRP. Then, this measure is calculated as

$$OF = \frac{\left|\{i \in \Psi \mid t_i^{act,\tau} \leqslant \frac{r_{\delta_i}}{100} \cdot t^{max}\}\right|}{\left|\Psi\right|} \cdot 100[\%].$$

- *Workload balance* (WB). WB reflects the extent to which the actual workload is balanced evenly between the drivers. We denote by $R_i = {(100 \cdot t_i^{act,\tau})}/{(r_{\delta_i} \cdot t^{max})}$ the relative workload of the district represented by center $i \in \Psi$. Moreover, we denote by $\mu = \frac{1}{|\Psi|} \cdot \sum_{i \in \Psi} R_i$ the average relative workload over all districts. WB is defined as the maximum absolute deviation between the relative workload of a district and the average relative workload, i.e., it is computed as

$$WB = \max\{\max_{i \in \Psi} R_i - \mu, \mu - \min_{i \in \Psi} R_i\} \cdot 100[\%].$$

Thus, if $WB = 0$, we consider the workload to be perfectly balanced.

6.8 Real-World Case Study

In this section, we perform a case study based on a real-world data set of a European parcel delivery company. First, we briefly describe the underlying data as well as its preparation for the experiments of the case study, and we report the parameterization used in the experiments. Then, we explain how we estimate the travel time within the districts based on the available customer order data. In the subsequent sections, we experimentally investigate the impact of the following aspects: The values of the workload limits used in the three tactical planning models, the presence of homogeneous and heterogeneous resources, the location of the depot, and the length of the tactical planning horizon. We report the running times of the location-allocation heuristic for the three tactical models and, finally, visualize some solutions obtained after the operational adaptation.

6.8.1 Data Preparation and Parameterization

Recall that the data basis for our approach should ideally consist of forecast demand data. However, to rule out any forecast bias, our case study is based on historical data which we treat as forecast data. The data set comprises approximately 67,000 customer orders delivered in a service region in Germany within a time period of four months. For each customer order $o \in O$, the data includes the day of delivery (τ_o), the service time (s_o), the weight (l_o), and the delivery address. We geocoded the delivery addresses using the PTV xLocate Server[1] of our industry partner PTV. Travel times based on the road network have been calculated using the PTV xDima Server[1]. Basic areas B in the case study correspond to sub-zip code areas ("PLZ8 areas") provided by PTV. The adjacency information A_b and

[1]http://xserver.ptvgroup.com/en-uk/home/ptv-xserver-en/

the basic area b_o containing each customer order have been calculated using the free geographic information system QGIS[2]. The service region and its subdivision into 252 sub-zip code areas are depicted in Figure 6.3. The black triangle represents the depot.

Figure 6.3: Depot and sub-zip code areas of the service region under study

We split the data set into two separate test instances to account for seasonal demand fluctuations in the data. The first instance comprises the first two months of the data set with roughly 32,000 customer orders, while the second instance contains the remaining two months with approximately 35,000 customer orders. Each instance is, in turn, subdivided into tactical sample days T_1, consisting of the first month of each instance, and operational sample days T_2, consisting of the second month of each instance.

If not mentioned otherwise, we consider the following experimental setup. We assume that a homogeneous fleet is available with capacity $C_v = 1150$ kg, corresponding to a vehicle of the Mercedes Sprinter class, which is the prevalent vehicle class used at the parcel delivery company. Moreover, we consider only full-time drivers, i.e., drivers with $r_d = 100$, and a maximum contractual working time of $t^{max} = 7.5$ hours. The maximum number of tours performed to a district on the same day is restricted to $|N| = 3$, and reloading a vehicle at the depot takes $t^{reload} = 1/3$ hour. We use $|I| = 16$ potential district centers, which is sufficient to cover the demand of the service region. The number of available drivers M_d and the number of available vehicles N_v are also set to 16.

If we do not state otherwise, we parameterize our solution approach and the evaluation stage as follows. All IP and MIP models presented in this chapter are solved using the MIP

[2] `http://qgis.osgeo.org`

solver Gurobi 7.0.2 with the following tolerances and time limits:

- For the first stage, we set the MIP optimality tolerance to 3%, which we consider as sufficiently small for practical applications, and the time limit for each of steps 3 and 4 of our heuristic to 900 seconds. We perform at most $\mathtt{iter}_{max} = 20$ iterations of the location-allocation procedure and limit the maximum runtime for each instance to 7,200 seconds. Furthermore, we set parameter $\alpha = 3$ for the heuristic variable fixation in step 2 of the heuristic.
- For the second stage, we set the MIP optimality tolerance to 1% and the time limit for the solution of each MIP model to 60 seconds.
- For the evaluation stage, we set the MIP optimality tolerance to 1% and the time limit for each model to 7,200 seconds. Among the several thousand CVRP instances that were solved to obtain the results presented in the subsequent sections, the actual MIP gap exceeded 1% only on three instances with values of 4.9%, 5.0%, and 6.4%. This means that the evaluation measures that we present in the remainder of this chapter were calculated based on near-optimal CVRP solutions.

The implementation was made in Java, and all experiments were performed under Ubuntu 16 on a machine with an Intel Xeon E5-2650 v2 CPU at 2.6 GHz and 128 GB of RAM.

6.8.2 Estimating the Travel Time Within the District

Recall that the estimated workload w_b^τ of basic area $b \in B$ on day $\tau \in T$ within the delivery district consists of the service time s_b^τ plus an estimation for the time required to travel to the customer orders in the basic area on that day. For a given $k \in \mathbb{R}^+$, we compute the travel time estimation for basic area b and day τ as

$$t_b^\tau(k) = \sum_{\substack{o \in O, \\ \tau_o = \tau, b_o = b}} \frac{1}{k} \cdot \left(\sum_{k'=1}^{\lceil k \rceil} t_{o,\kappa(o,k')} - (\lceil k \rceil - k) \cdot t_{o,\kappa(o,k)} \right),$$

where $t_{o_1,o_2} \in \mathbb{R}^+$ denotes the travel time from customer order $o_1 \in O$ to customer order $o_2 \in O$, and $\kappa(o,\epsilon) \in O$ denotes the ϵ-closest customer order from customer order $o \in O$ in terms of travel time (which can be in the given or in another basic area). Thus, for each customer order o, we calculate the average travel time to the k customer orders to which customer order o has the shortest travel time. Note that k does not have to be integer. If k is fractional, we consider the $\lceil k \rceil$-th customer proportionally. We sum up these values over all customer orders within the basic area on the given day to obtain the travel time estimation $t_b^\tau(k)$. This approach is motivated by the observation that it is unlikely that long edges are used in an optimal solution to the vehicle routing problem (see Toth and Vigo, 2003, who develop their granular tabu search based on the same reasoning).

To obtain a value for k which results in a good estimation, we created four test cases. More precisely, we created for each of the two test instances of the preceding section one test case with the original depot and one with a depot centrally located in the service region. We solved the tactical planning problem for each test case with an arbitrary value of $k = 4.5$ using our first-stage solution approach with model A/IV–AW. In the solutions we obtained, we calculated for each $k \in \{1, 1.5, ..., 9.5, 10\}$ the estimated workload for each delivery district (comprising service and reloading time, travel time within the district and between the depot and the district) on each day using the introduced travel time estimation. Additionally, we solved a CVRP for each delivery district on each day to determine the actual workloads. Then, we computed the deviation as well as the absolute deviation between the estimated and the actual workload for each day, district and value of k, and averaged them for each value of k. As Figure 6.4 shows, the results look quite similar for all four solutions. For $k \leqslant 3.5$, the average deviation is negative, i.e., the actual workload is underestimated. On the contrary, the average deviation is positive for $k \geqslant 4.5$, meaning that the actual workload is overestimated. For $k = 4$, the average deviation is closest to zero in all four solutions, thus, yielding the best estimation. $k = 4$ also yields the best estimation with respect to the absolute deviation, with values of roughly 11 minutes on average. Consequently, we use $k = 4$ for all computational experiments in the remainder of this chapter.

6.8.3 Controlling Conservatism

Recall that we expect the three tactical IP models AV–AW, A/IV–AW, and IV–A/IW to be different in terms of how conservative they are. Beyond that, we expect that the degree of conservatism can also be controlled within each model by setting appropriate workload limits LB and UB. In the following, we evaluate both effects. For this purpose, we run experiments on all three models with different workload limits. An overview of all workload limits is given in Table 6.4. We consider for each model the three levels of workload limits (i) LOW, (ii) MEDIUM, and (iii) HIGH, corresponding to upper workload limits UB of 7, 7.5, and 8 hours, respectively. The lower workload limits LB differ, however, between the models. For models AV–AW and A/IV–AW, they correspond to 6, 6.5, and 7 hours, whereas they correspond to only 4.5, 5, and 5.5 hours for model IV–A/IW. As the latter model considers very restrictive workload upper bounds, which prohibit that UB is exceeded on any of the tactical sample days, it is necessary to set relatively low workload limits LB to ensure the feasibility of the model.

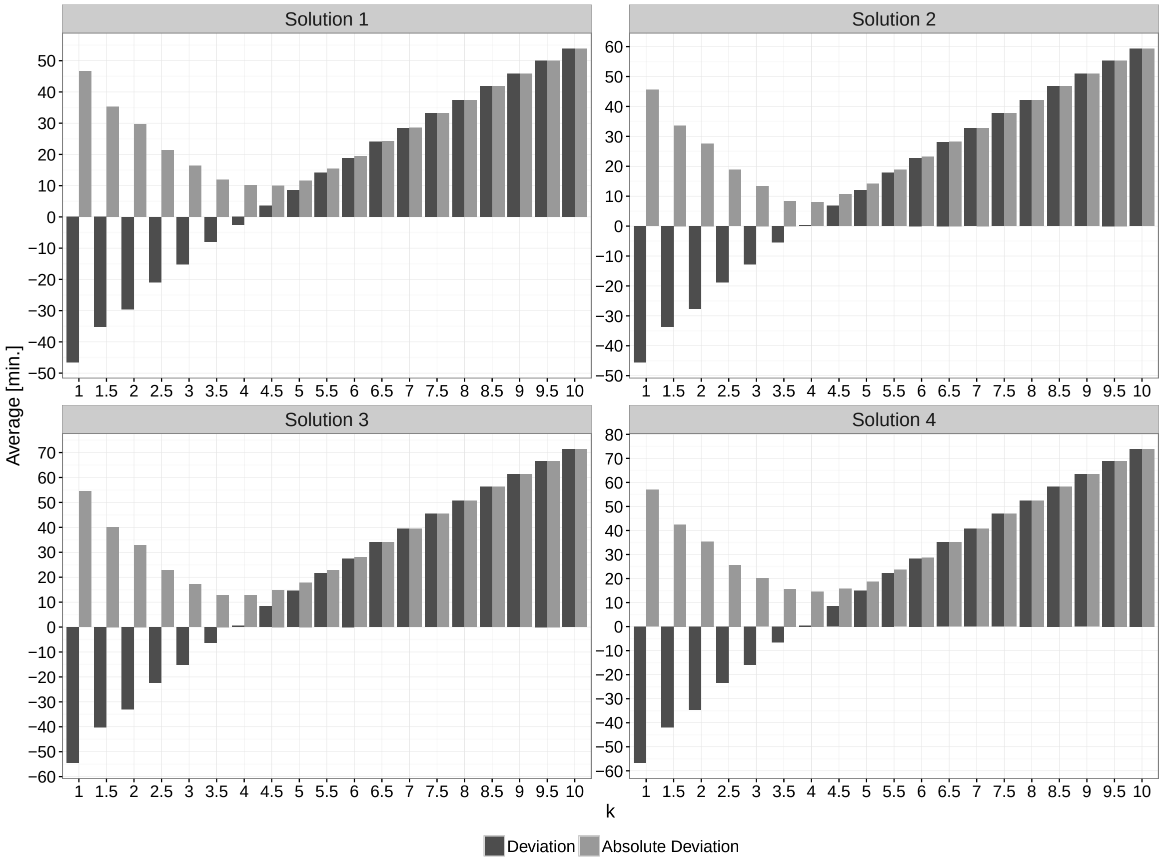


Figure 6.4: Quality of travel time estimations for different values of parameter k measured as deviation and absolute deviation between estimated and actual workload (average values over districts and days)

The values we obtain with the different workload limits for our evaluation measures on the two test instances are illustrated in Figures 6.5 and 6.6. Figures 6.5a and 6.6a depict the number of districts ND. As one would expect, higher workload limits clearly result in the establishment of fewer delivery districts. The results also show that model IV–A/IW establishes the highest number of delivery districts of the three models. Moreover, model A/IV–AW tends to establish more delivery districts than model AV–AW. For a given workload limit, it establishes the same number of districts as model AV–AW on instance 1, and it establishes one more district than model AV–AW on instance 2.

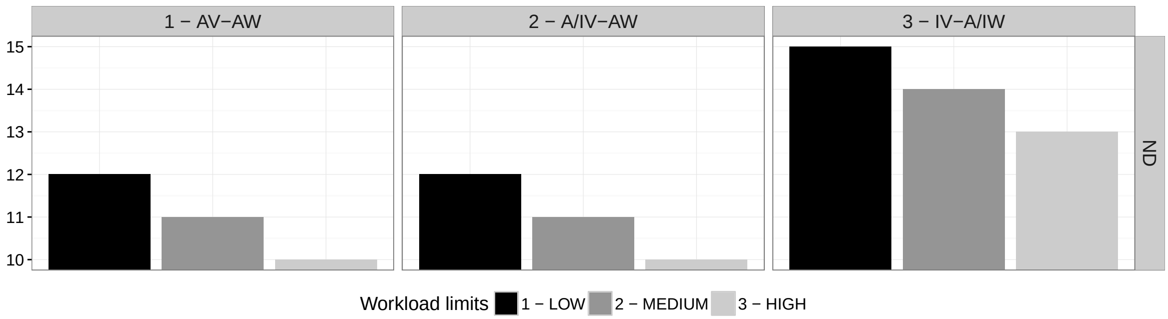
Figures 6.5b and 6.6b show the values for driver consistency DC, operational feasibility OF, and workload balance WB obtained with different values of ω and averaged over all operational sample days. Remember that ω specifies the maximum number of basic areas that may be reassigned in stage 2 of the solution approach. Hence, solving each operational problem for each $\omega \in \{0, ..., 20\}$ allows us to evaluate the trade-off between driver consistency and the other evaluation measures.

Table 6.4: Overview of different workload limits $[\mathrm{LB}; \mathrm{UB}]$ for each of the three tactical models (in hours)

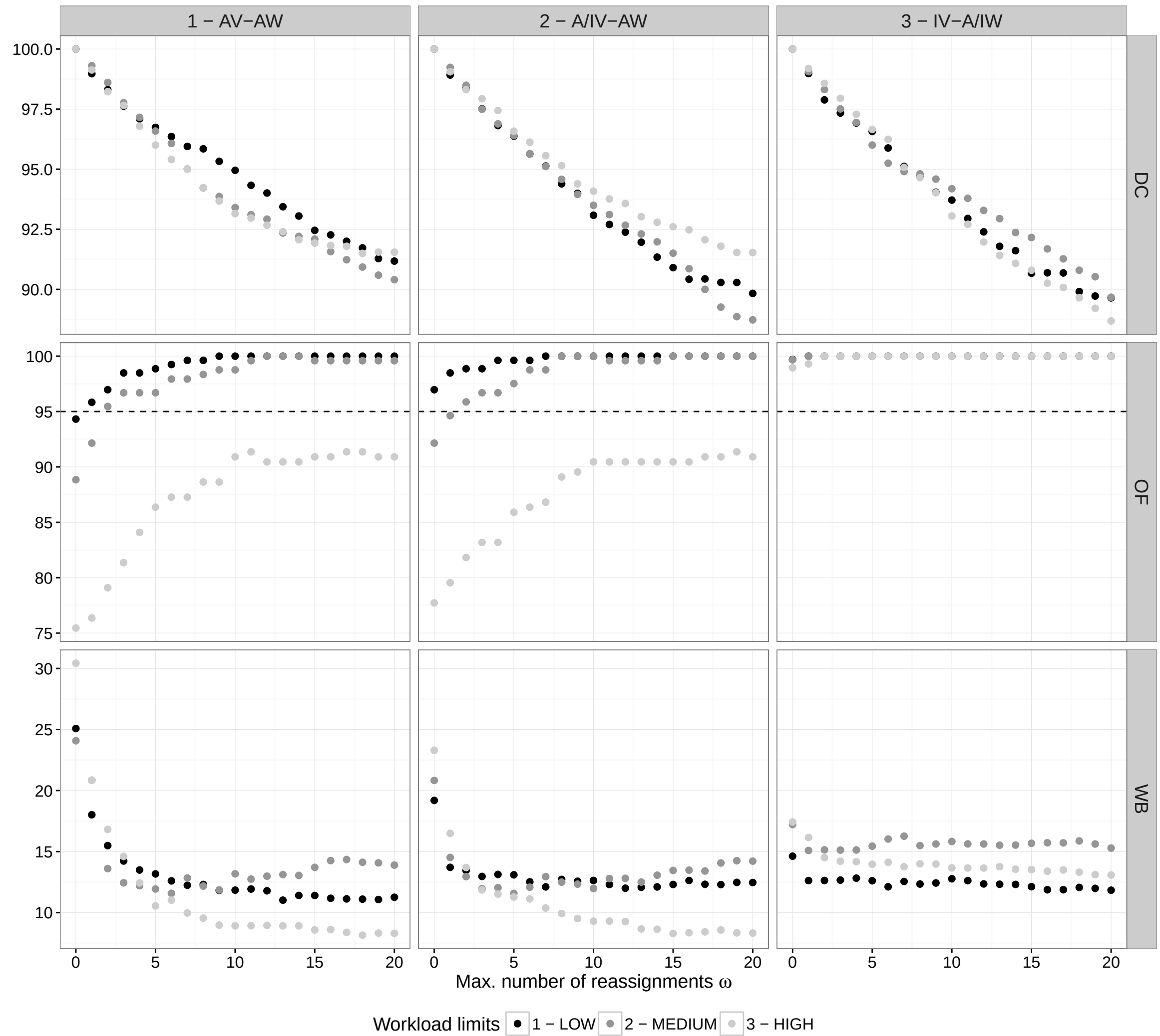
Workload limit	Model		
	AV–AW	A/IV–AW	IV–A/IW
LOW	[6; 7]	[6; 7]	[4.5; 7]
MEDIUM	[6.5; 7.5]	[6.5; 7.5]	[5; 7.5]
HIGH	[7; 8]	[7; 8]	[5.5; 8]

As can be seen from the figures, driver consistency behaves relatively similar for all models and all workload limits. It decreases almost linearly with increasing value of ω and takes values of approximately 90% for $\omega = 20$, with instance 2 yielding slightly lower values than instance 1.

Major differences can be observed with respect to operational feasibility. Model IV–A/IW clearly provides the best operational feasibility. Irrespective of the level of workload limits, only few reassignments are required to attain values close or equal to 100% on both instances. This confirms the expectation that model IV–A/IW is the most conservative of the three tactical models. The other two models yield significantly lower values for operational feasibility. On instance 1, the values obtained with the two models are quite similar with the main difference that model A/IV–AW yields higher values if no reassignments are allowed ($\omega = 0$). On instance 2, the differences between the two models become more obvious. Even for $\omega = 20$, model AV–AW yields an operational feasibility of only approximately 21% and 64% for workload limits HIGH and MEDIUM, respectively, whereas model A/IV–AW attains values of about 62% and 95%, respectively. 100% operational feasibility is achieved only with model A/IV–AW in combination with workload limit LOW, while the maximum value obtained with model AV–AW is roughly 95%. This affirms empirically that model A/IV–AW is more conservative than model AV–AW. Furthermore, the results show that the degree of conservatism can be controlled by an appropriate choice of workload limits. Suppose, for example, that a human planner targets an operational feasibility of roughly 95%. We marked this value in the figures with a dashed horizontal line. Then, on instance 1, the planner should select the workload limits MEDIUM for models AV–AW and A/IV–AW, and the workload limit HIGH for model IV–A/IW, as these limits result in the fewest number of districts, and, thus, also in the minimum number of required resources, with which the planner's target value is attained. Analogously, the planner should select workload limits LOW, MEDIUM, and HIGH for models AV–AW, A/IV–AW, and IV–A/IW, respectively, on instance 2.

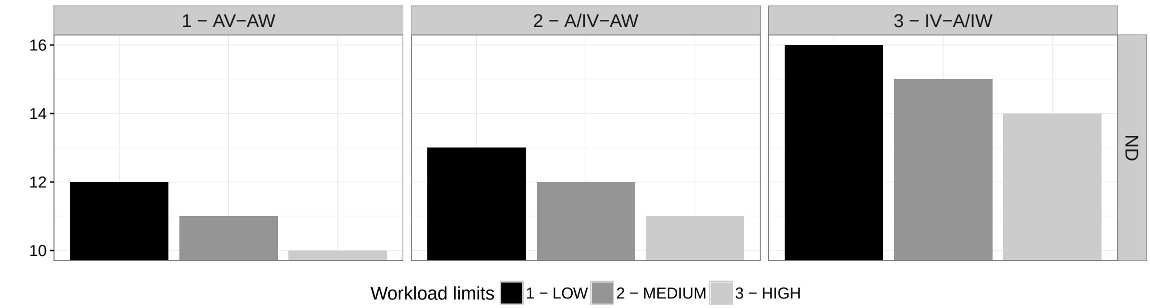


(a) Number of districts

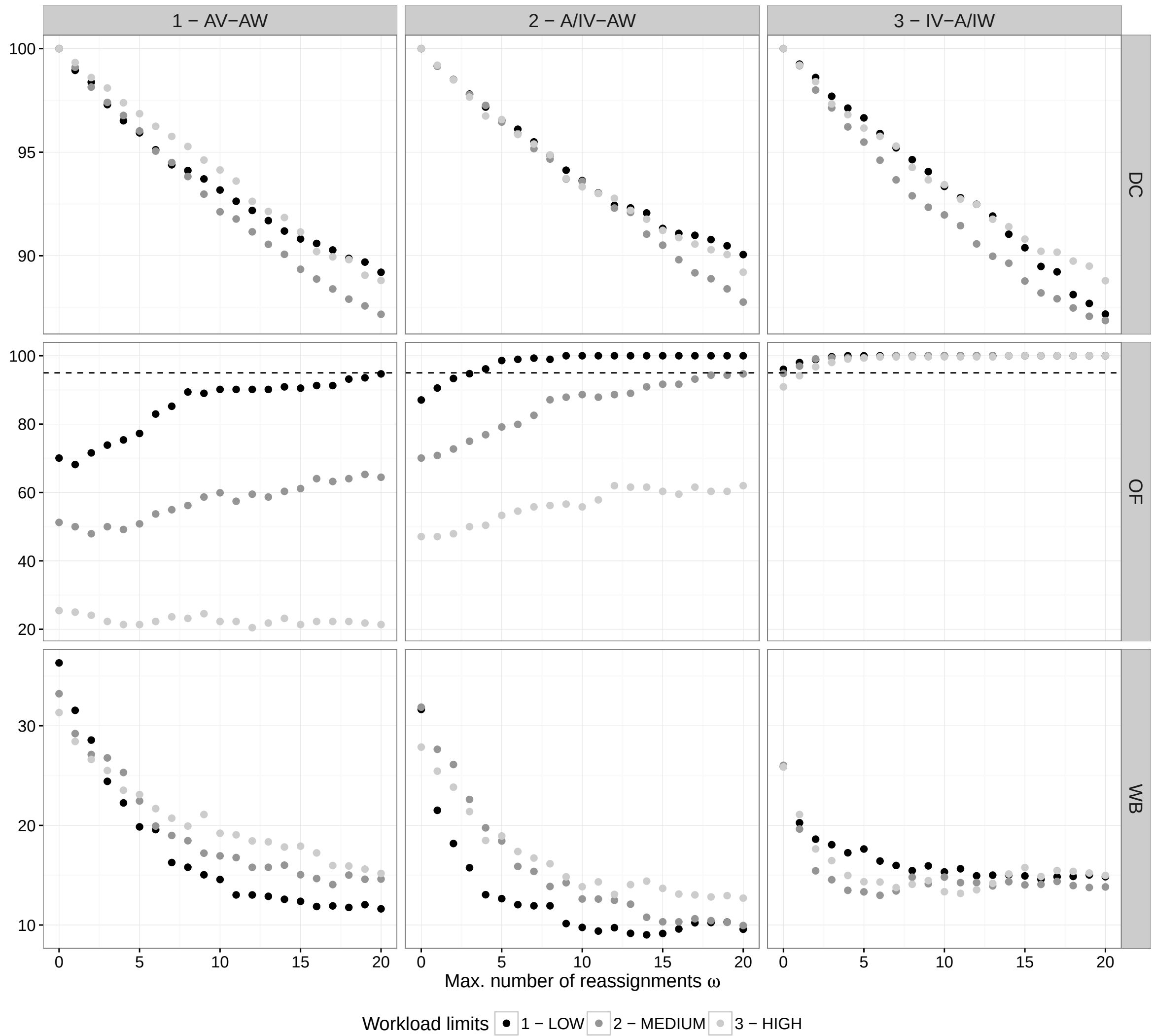


(b) Driver consistency, operational feasibility, and workload balance for different numbers of allowed reassignments (average values over operational sample days)

Figure 6.5: Evaluation measures obtained for the three tactical planning models and different workload ranges on test instance 1



(a) Number of districts



(b) Driver consistency, operational feasibility, and workload balance for different numbers of allowed reassignments (average values over operational sample days)

Figure 6.6: Evaluation measures obtained for the three tactical planning models and different workload ranges on test instance 2

Concerning workload balance, model IV–A/IW yields the best results of the three models for the case that no reassignments are allowed ($\omega = 0$). With increasing value of ω, one can observe a convergence to fairly similar values for all three models and all workload limits, with model IV–A/IW yielding slightly worse values than the other two models.

All in all, we conclude from these experiments that the models behave in the expected way. The results confirm that the degree of conservatism is influenced by the models themselves and by the choice of the workload limits. Furthermore, the travel time estimations in the models work quite well, which can be seen from the results for operational feasibility and workload balance: Increasing the value of ω clearly tends to result in an improvement of the two measures, although, in rare cases, it leads to minor deteriorations due to errors in the estimation. A visual impression of the solutions obtained with the workload limits recommended above on test instance 1 is provided in Figures 6.7–6.9. District boundaries are highlighted by bold lines.

For the remainder of this chapter, we fix the workload limits of the three models according to the presented recommendations for a planner who targets an operational feasibility of 95%. This means that we use workload limits MEDIUM for model A/IV–AW, workload limits HIGH for model IV–A/IW, and workload limits MEDIUM and LOW for model AV–AW on instance 1 and 2, respectively.

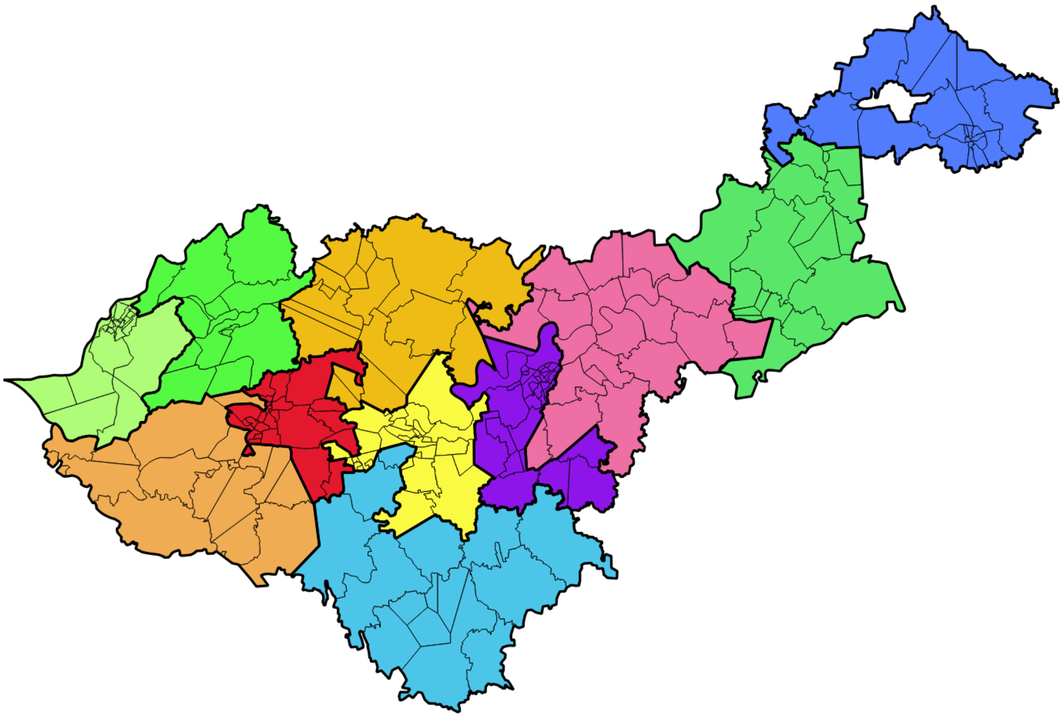


Figure 6.7: Tactical district design obtained with model AV–AW on instance 1

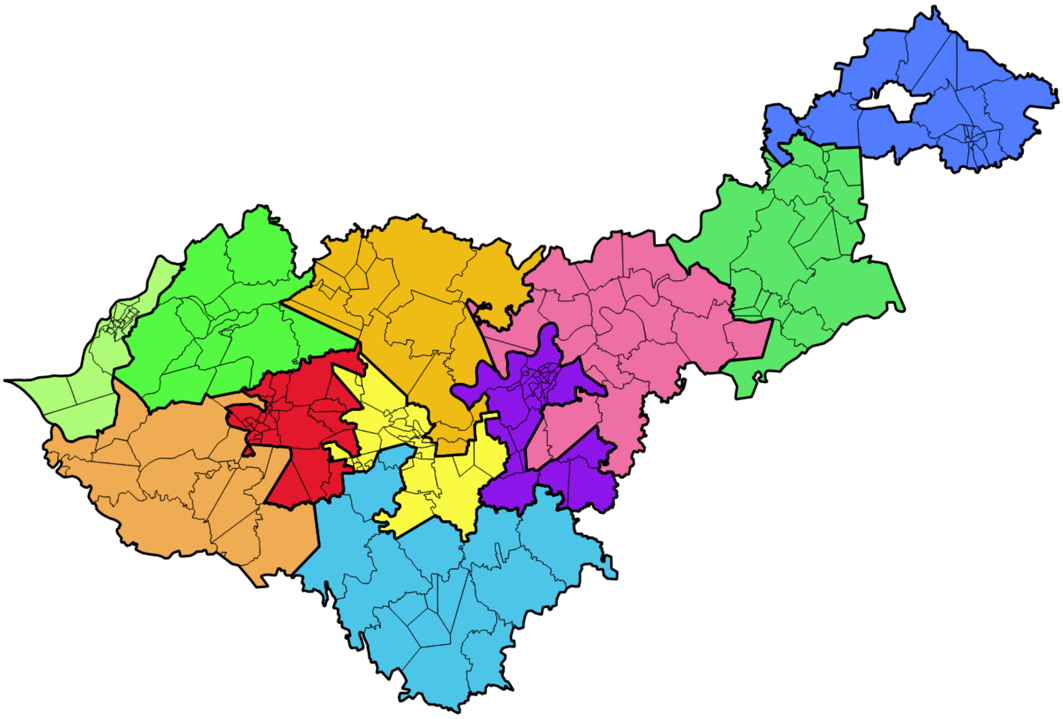


Figure 6.8: Tactical district design obtained with model A/IV–AW on instance 1

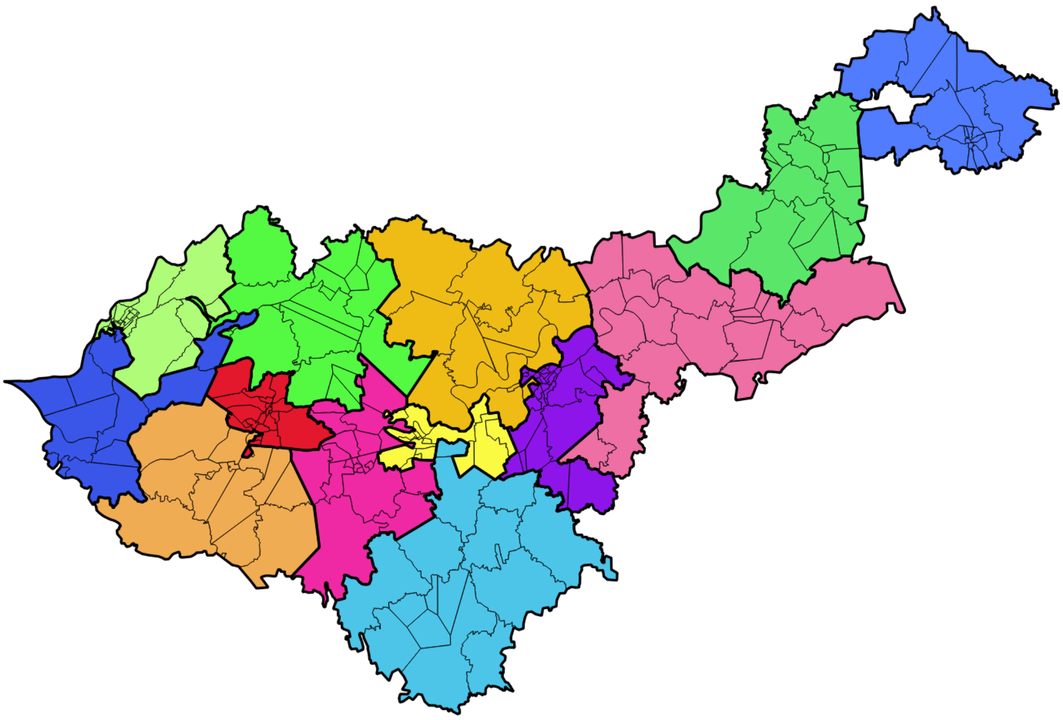


Figure 6.9: Tactical district design obtained with model IV–A/IW on instance 1

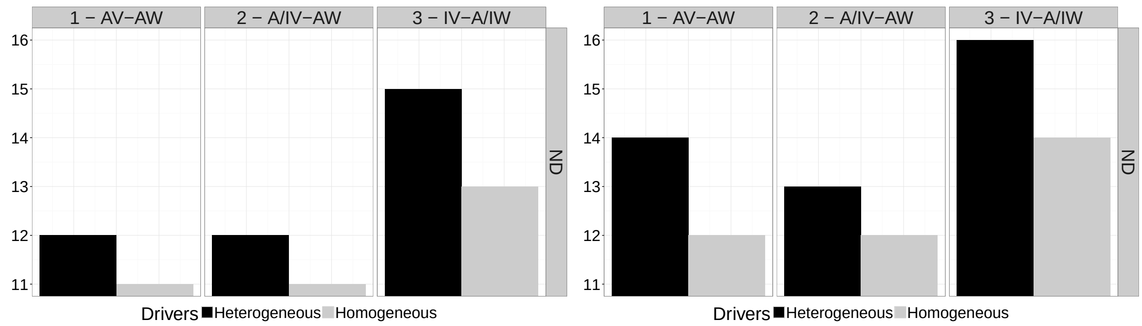
6.8.4 Resources and Depot Configuration

In the following, we examine the effect of different resources and depot configurations. We start by introducing a heterogeneous crew of drivers and a heterogeneous fleet of vehicles. Subsequently, we analyze the impact of having a depot that is centrally located in the service region. Each effect is studied individually, i.e., in each of the following subsections the parameterization changes only in one aspect compared to the parameterization described in Sections 6.8.1–6.8.3.

Different Driver Types

So far, we assumed that the crew of available drivers consists only of full-time drivers. In this section, we extend the crew of available drivers to two different driver types: We consider ten full-time drivers ($M_1 = 10$, $r_1 = 100$) and six drivers with a contractual working time of 75% ($M_2 = 6$, $r_2 = 75$), with full-time drivers being prioritized.

In Figure 6.10, we illustrate the number of districts that we obtain with this crew of drivers (heterogeneous drivers) and compare the results with the number of districts established for the case that only full-time drivers are available (homogeneous drivers). We report these numbers per model and per test instance.



(a) Instance 1 **(b)** Instance 2

Figure 6.10: Number of districts obtained with the three tactical planning models for a heterogeneous and a homogeneous crew of available drivers

The figure shows that the number of districts increases for all models on both instances when a heterogeneous crew of drivers is considered. This is due to the fact that only ten full-time drivers are available and, thus, some districts must be served by part-time drivers, whose contractual working time is 25% below those of the full-time drivers. The largest increase can be observed for model IV–A/IW, where two more districts are established on both instances compared to the case of a homogeneous crew of drivers. On the contrary, model A/IV–AW generates only one additional district on both instances.

Naively, one might think that $\frac{1}{3/4} \approx 1.33$ part-time drivers with a contractual working time of 75% should be sufficient to replace one full-time driver. The results show, however, that between 1.5 and 2 part-time drivers are needed to replace one full-time driver. The reason for this lies in the time required to travel between the depot and the delivery districts, which reduces the working time that is actually available to deliver packages within each district. For part-time drivers, this time constitutes a larger proportion of the contractual working time than for full-time drivers.

Beyond that, we computed for each model, test instance and value of ω the absolute deviation in the average daily values obtained for measures DC, OF, and WB between the case of a homogeneous crew and the case of a heterogeneous crew. The 90%-quantile for the absolute deviations amounts to 1.6%, 4.4%, and 3.7% for measures DC, OF, and WB, respectively. Since the values deviate only by a few percentage points from those reported in Section 6.8.3 for a homogeneous crew of drivers, we omit the figures for these measures.

Figures 6.11–6.13 depict the solutions obtained with a heterogeneous crew on test instance 1. Shaded delivery districts indicate the assignment of a part-time driver.

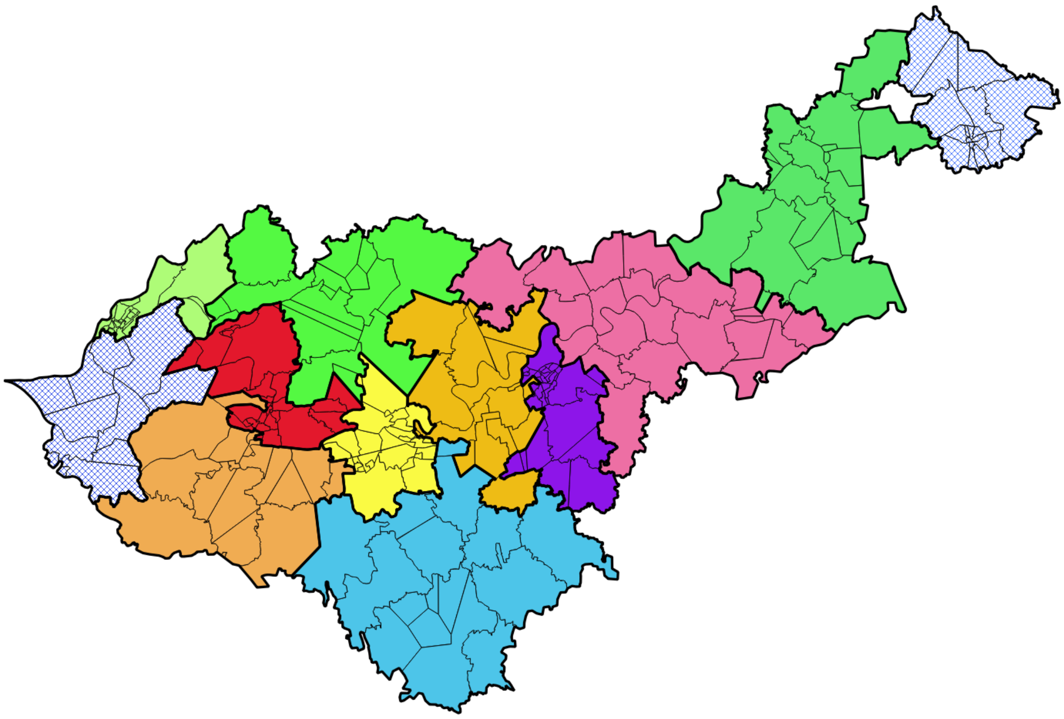


Figure 6.11: Tactical district design obtained with model AV–AW on instance 1 with a heterogeneous crew of drivers

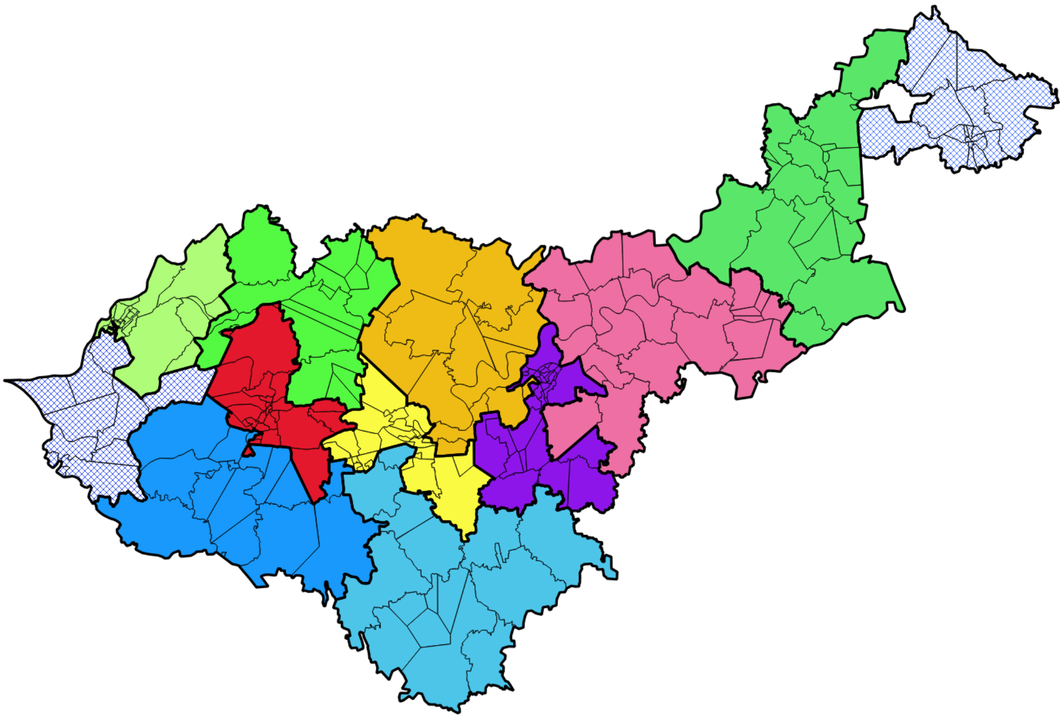


Figure 6.12: Tactical district design obtained with model A/IV–AW on instance 1 with a heterogeneous crew of drivers

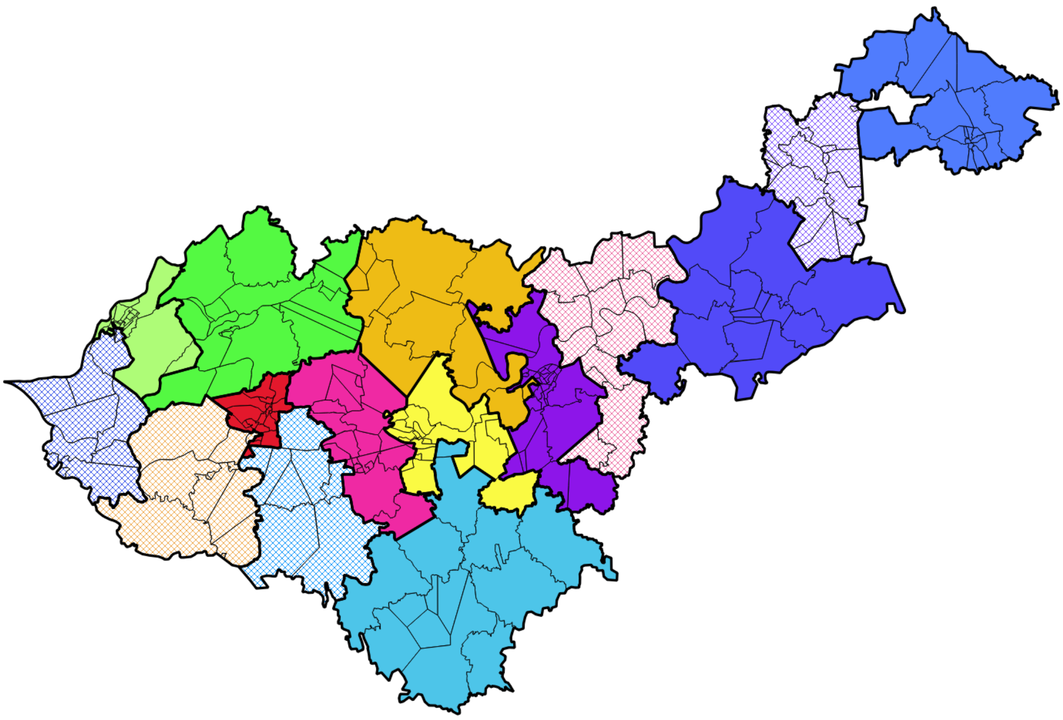
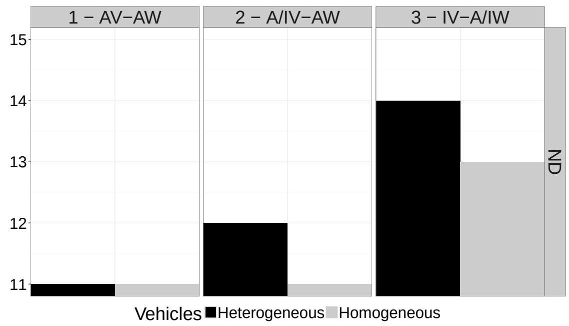


Figure 6.13: Tactical district design obtained with model IV–A/IW on instance 1 with a heterogeneous crew of drivers

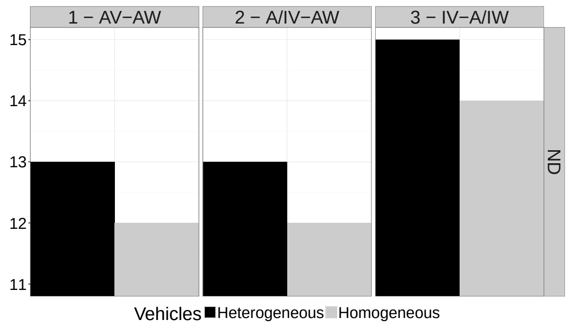
Different Vehicle Types

Now we consider the case that a heterogeneous fleet of vehicles is available. More precisely, we assume that we have $N_1 = 10$ standard vehicles with capacity $C_1 = 1150$ kg and $N_2 = 8$ small vehicles with capacity $C_2 = 800$ kg. Accordingly, the number of potential district centers is set to $|I| = 18$. Due to the higher number of potential districts and, consequently, the greater number of variables in the models, the time limit for each of steps 3 and 4 of the heuristic is increased to 1,800 seconds, and the maximum runtime per instance is raised to 14,400 seconds for the experiments in this section. All other parameters are left unchanged compared to Section 6.8.3.

Figure 6.14 shows the resulting number of delivery districts in comparison to the case of a homogeneous fleet. Although the relative difference in vehicle capacity between a small and a standard vehicle is greater than the relative difference in working time between a part-time and a full-time driver as considered in the preceding section, the increase in the number of districts is smaller. Solving instance 1 with model AV–AW even results in the same number of districts, and only one additional district is established in all other cases. Hence, vehicle capacity seems to be a less restrictive factor than working time on these instances.



(a) Instance 1



(b) Instance 2

Figure 6.14: Number of districts obtained with the three tactical planning models for a heterogeneous and a homogeneous fleet of vehicles

We refrain from reporting the values for measures driver consistency, operational feasibility, and workload balance due to their similarity with those reported in Section 6.8.3 for a homogeneous fleet: The 90%-quantile for the absolute deviations obtained for theses measures amounts to 1.1%, 4.7%, and 3.0%, respectively.

The solutions obtained with a heterogeneous fleet on instance 1 are illustrated in Figures 6.15–6.17. Delivery districts with a small vehicle are shaded.

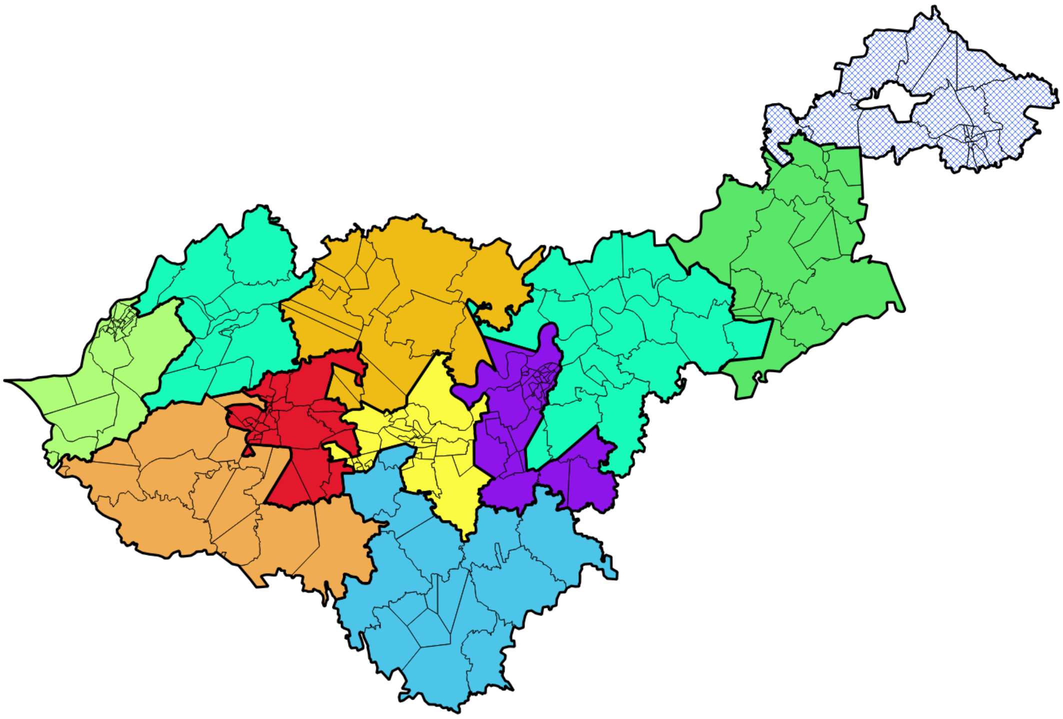


Figure 6.15: Tactical district design obtained with model AV–AW on instance 1 with a heterogeneous fleet of vehicles

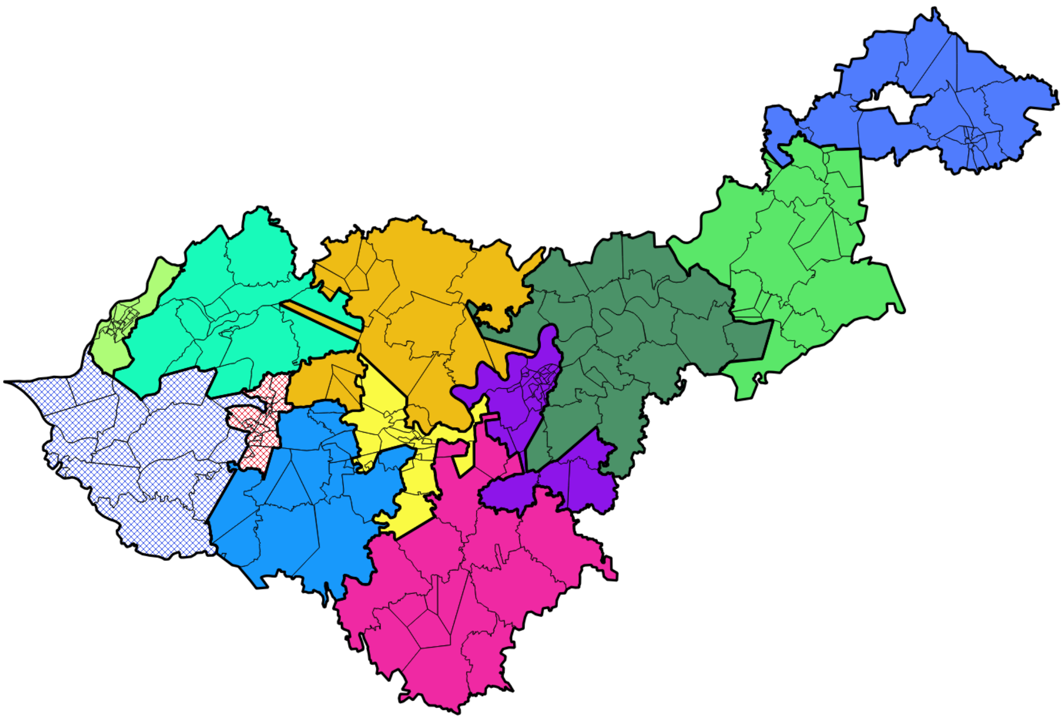


Figure 6.16: Tactical district design obtained with model A/IV–AW on instance 1 with a heterogeneous fleet of vehicles

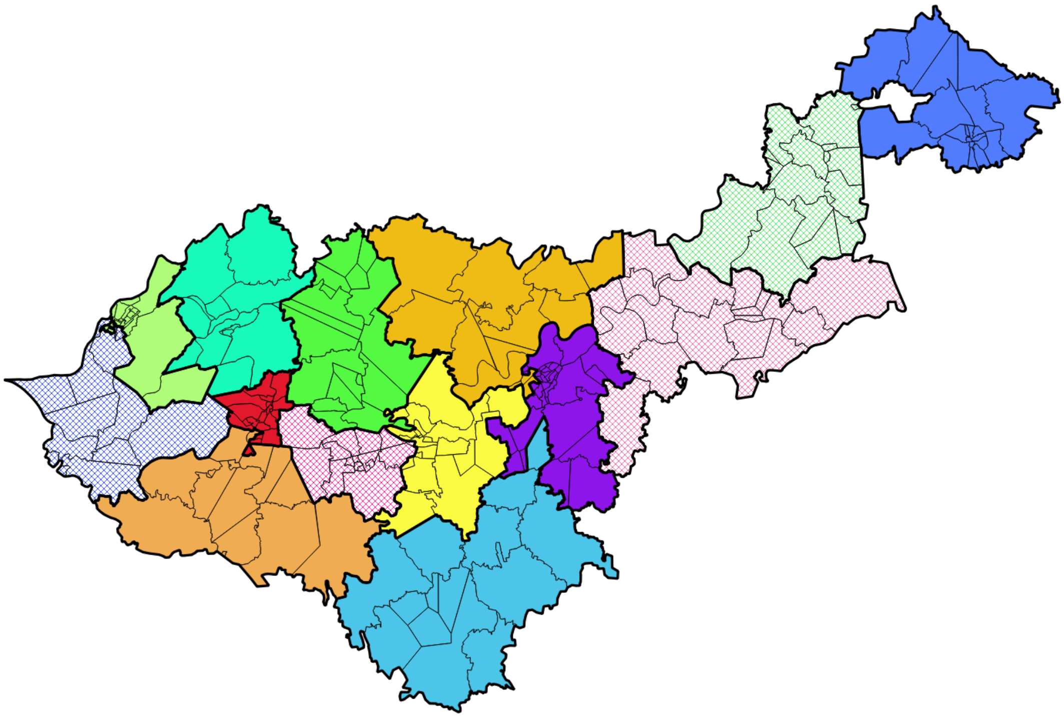


Figure 6.17: Tactical district design obtained with model IV–A/IW on instance 1 with a heterogeneous fleet of vehicles

Location of the Depot

In the following, we investigate the impact of the location of the depot. Recall that the original depot is fairly remote from the service region (see Figure 6.3). We compare this with a setting where the depot is centrally located in the service region as depicted in Figure 6.18. Again, all other parameters are set to the values described in Section 6.8.3.

Figure 6.19 contains the results obtained for the two depot configurations. The values for driver consistency and workload balance deviate only slightly from the numbers of Section 6.8.3: The 90%-quantiles of the absolute deviations with respect to the values obtained for the original depot location are 1.0% and 5.5%, respectively. Hence, we exclude these measures from the figure.

The figures in the upper row show the number of districts generated by the three models. Due to the shorter travel time between the depot and the delivery districts, the number of districts can be reduced with all models if the depot is centrally located. The reduction amounts to two districts in all cases with the exception of model AV–AW on test instance 2. In the latter case, the number of districts can be reduced by only one compared to the original depot configuration.

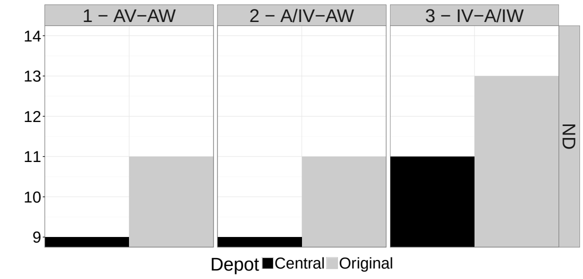
Figure 6.18: Service region with centrally located depot (represented by the black triangle)

The figures in the second and third row contain the values for operational feasibility. There is no clear tendency whether a centrally located depot improves or worsens operational feasibility compared to the case of a remote depot. However, on test instance 2 model A/IV–AW yields remarkably lower values with a centrally located depot and fails to reach the target value of 95% even for $\omega = 20$.

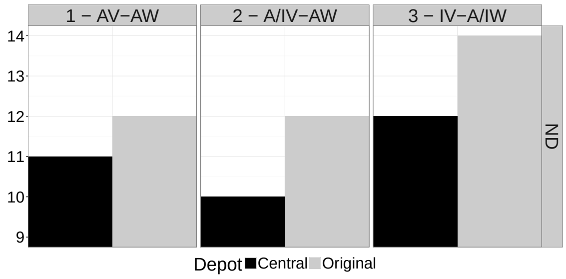
The figures in the fourth row show the total number of multi-tours performed on the operational sample days for $\omega = 20$. Recall that we understand by a multi-tour that a vehicle makes more than one tour to its delivery district on a certain day. With the original depot, the models try to completely avoid multi-tours since the depot's remote location leads to a large increase in workload for each additional tour to a delivery district. However, with a central depot, multi-tours become more attractive since the additional travel time between the depot and the delivery districts is drastically reduced and, thus, the increase in workload is only moderate, in particular for those districts directly surrounding the depot. The fact that more multi-tours are performed for test instance 2 than for test instance 1 can be explained by a considerably higher total weight that must be transported in test instance 2.

6.8.5 Length of the Tactical Planning Horizon $|T_1|$

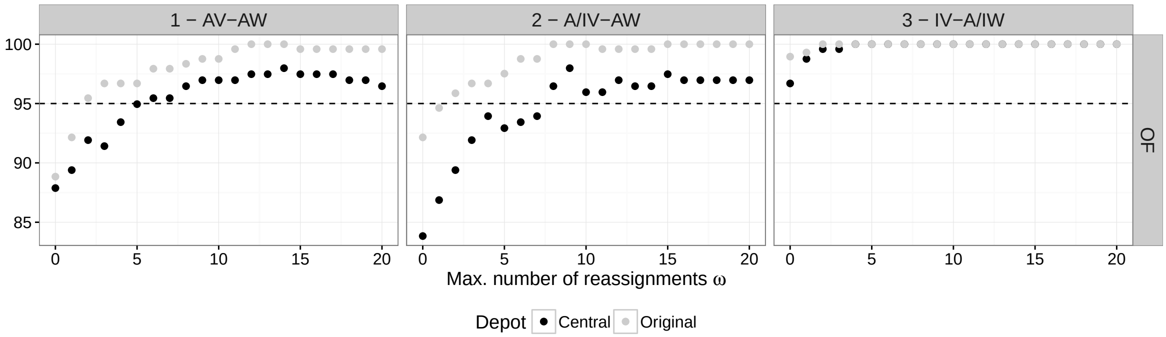
Next, we perform a sensitivity analysis with respect to the length of the planning horizon $|T_1|$ considered in the three tactical planning models. We compare the results we obtain with planning horizons consisting of one week, two weeks, and an entire month. Figure 6.20 contains the number of districts and the values for operational feasibility. We omit again



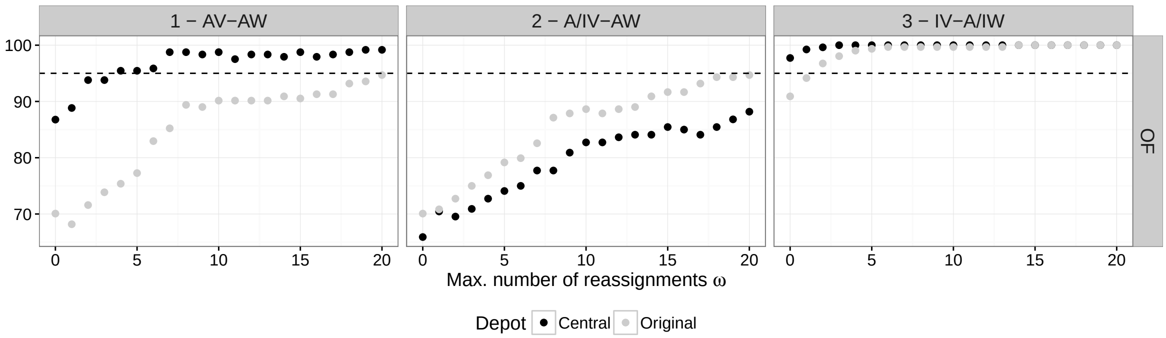
(a) Number of districts on instance 1



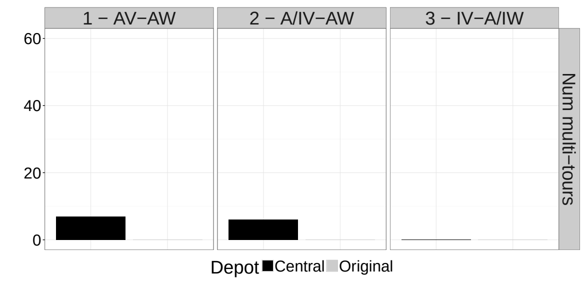
(b) Number of districts on instance 2



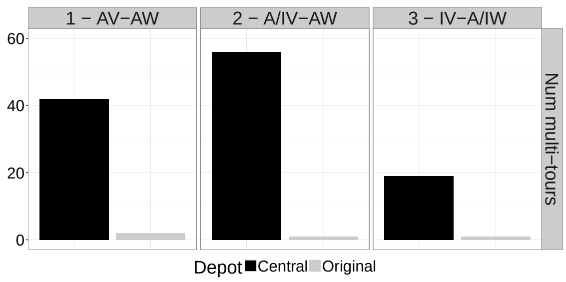
(c) Operational feasibility on instance 1 (average values over operational sample days)



(d) Operational feasibility on instance 2 (average values over operational sample days)

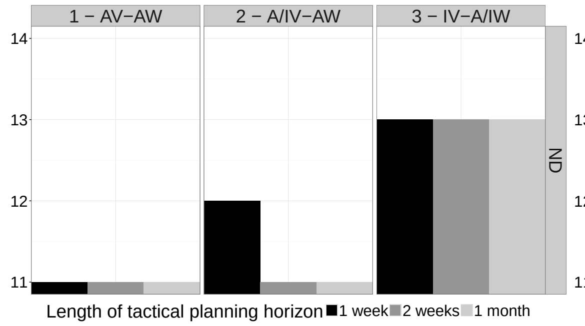


(e) Number of multi-tours on instance 1 for $\omega = 20$ (sum over operational sample days)

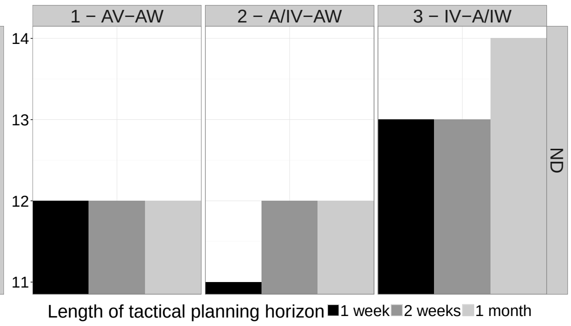


(f) Number of multi-tours on instance 2 for $\omega = 20$ (sum over operational sample days)

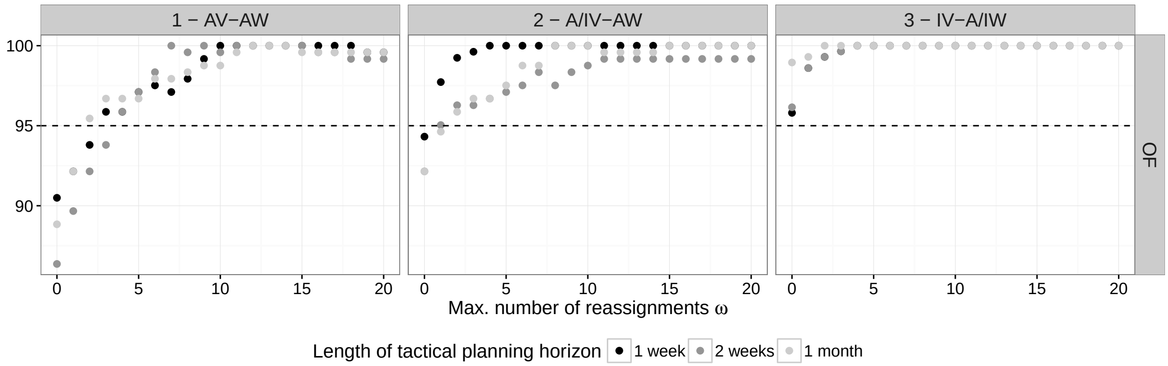
Figure 6.19: Number of districts, operational feasibility and number of multi-tours obtained for the three tactical planning models and different depot locations



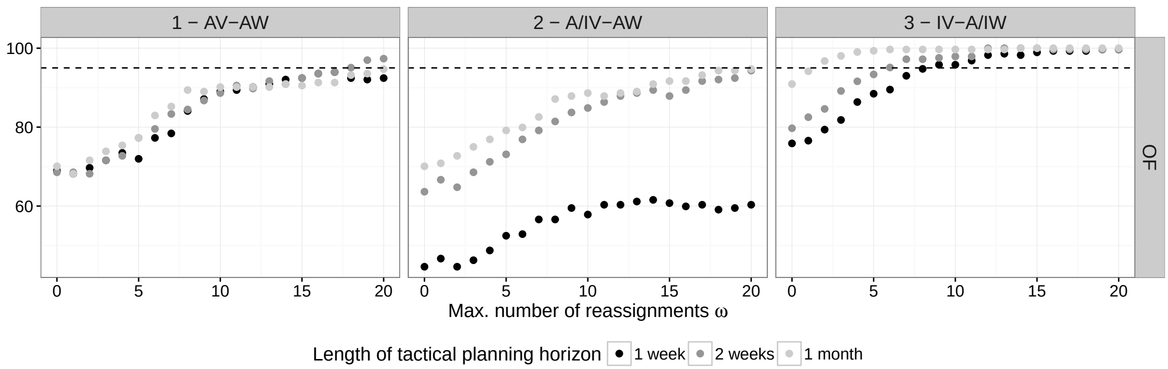
(a) Number of districts on instance 1



(b) Number of districts on instance 2



(c) Operational feasibility for different numbers of allowed reassignments on instance 1 (average values over operational sample days)



(d) Operational feasibility for different numbers of allowed reassignments on instance 2 (average values over operational sample days)

Figure 6.20: Number of districts and operational feasibility obtained for the three tactical planning models and different lengths of the tactical planning horizon $|T_1|$

the values for driver consistency and workload balance. The 90%-quantile for their absolute deviations with respect to the numbers reported in Section 6.8.3 equals 1.3% and 5.1%, respectively.

The results show that the models are fairly robust with respect to different planning horizons. Model AV–AW generates the same number of districts with all planning horizons on both test instances. For model A/IV–AW, a planning horizon of one week seems to be too short as the number of districts deviates on both instances from the values obtained with longer planning horizons. Model IV–A/IW establishes 14 districts on instance 2 if a planning horizon of one month is selected, whereas shorter planning horizons produce one district less. This can be explained by the way in which the model handles workload limits: The estimated workload for every single day must not exceed a given threshold, and, thus, the model tends to create more districts with increasing length of the planning horizon.

Concerning operational feasibility, the results are quite similar. The cases in which major differences between the planning horizons can be observed are due to different numbers of districts. This is, for example, the case for model A/IV–AW on test instance 2, where a planning horizon of only one week yields significantly lower values for operational feasibility than the other planning horizons because of the lower number of established districts.

6.8.6 Running Times of Location-Allocation Heuristic

Table 6.5 contains the running times of the location-allocation heuristic in seconds grouped by the three tactical planning models. We include all experiments presented in the preceding sections with a planning horizon of one month, and report the mean, the minimum and the maximum running time for each model. Furthermore, we report the mean number of location-allocation iterations performed.

Since model AV–AW does not consider day-specific input data, it contains the smallest number of variables of the three models, which results in the shortest running times. Model

Table 6.5: Running times and number of iterations of the location-allocation heuristic for the three tactical planning models

	Running time [s]			
Model	Mean	Min	Max	Iterations
AV–AW	295	15	1,203	8.2
A/IV–AW	4,694	483	14,400	6.6
IV–A/IW	3,343	435	14,400	9.0

A/IV–AW is the computationally most challenging model. It has the highest average running time, even though the fewest iterations are performed when this model is used. Keep in mind that the problem that we tackle with the location-allocation heuristic is a tactical planning problem, which is typically solved only every few months. Hence, the reported running times do not pose a limitation on the suitability of the heuristic for practice, irrespective of the underlying model.

6.8.7 Visualization of Operational Reassignments

Figures 6.21–6.24 exemplarily illustrate some operational reassignments for different values of ω on a particular operational sample day. The underlying tactical district design was computed with model A/IV–AW on test instance 1. Tactical district boundaries are marked by bold lines, the districts resulting from the operational adaptation can be distinguished by different colors. Reassigned basic areas compared to the tactical solution are highlighted by diagonal lines. It can be seen from the figures that not all basic areas that are reassigned for small values of ω are also reassigned for greater values of ω. Obviously, increasing values of ω permit additional combinations of reassignments that are, at least in parts, more attractive than the reassignments that are feasible for smaller values of ω.

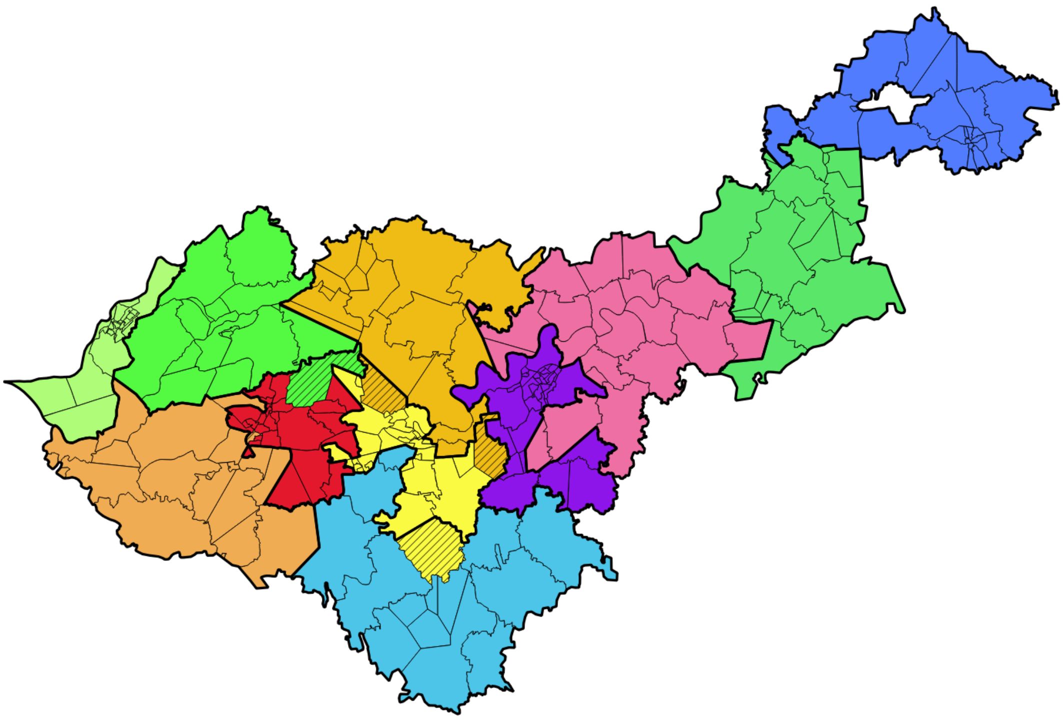


Figure 6.21: Solution obtained using model A/IV–AW on instance 1 after operational reassignment with $\omega = 5$

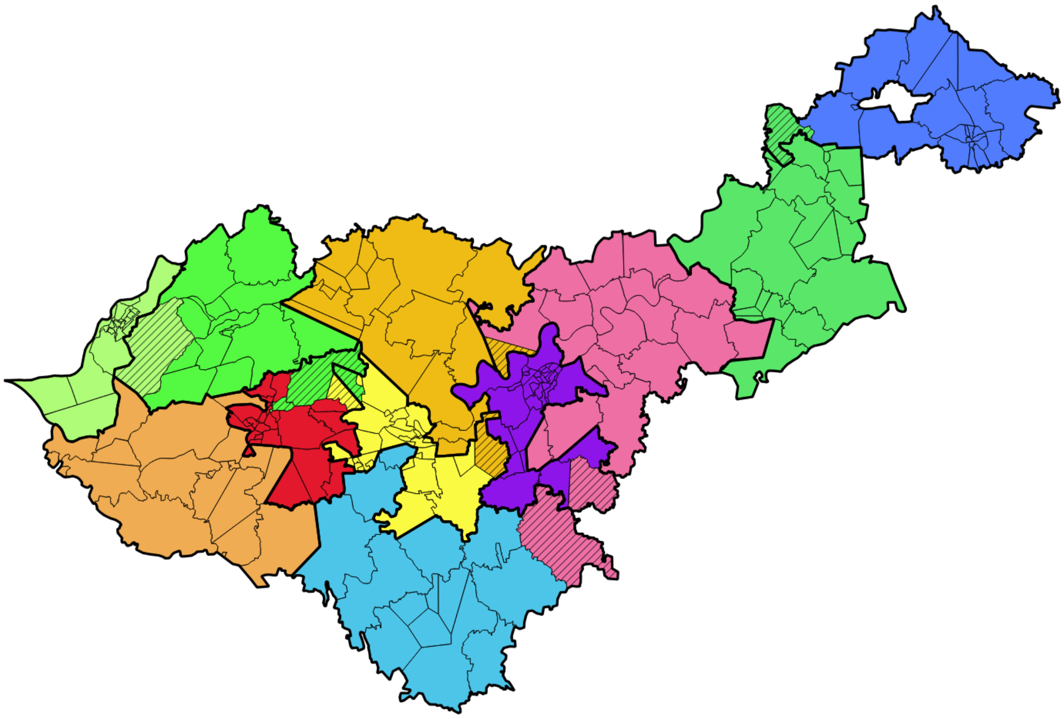


Figure 6.22: Solution obtained using model A/IV–AW on instance 1 after operational reassignment with $\omega = 10$

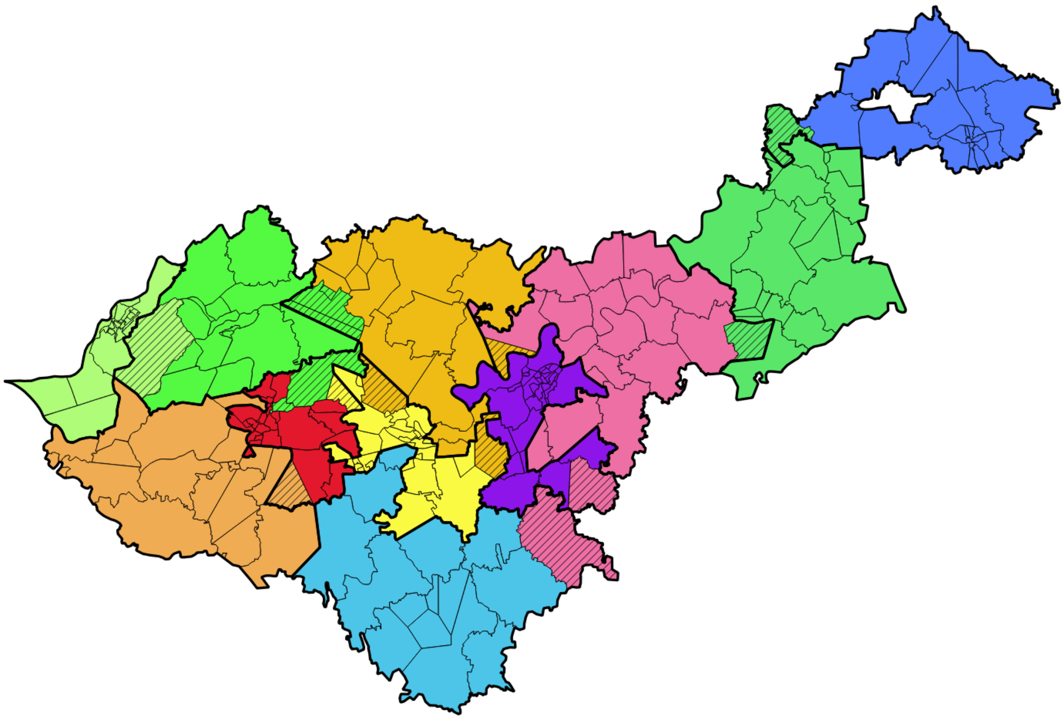


Figure 6.23: Solution obtained using model A/IV–AW on instance 1 after operational reassignment with $\omega = 15$

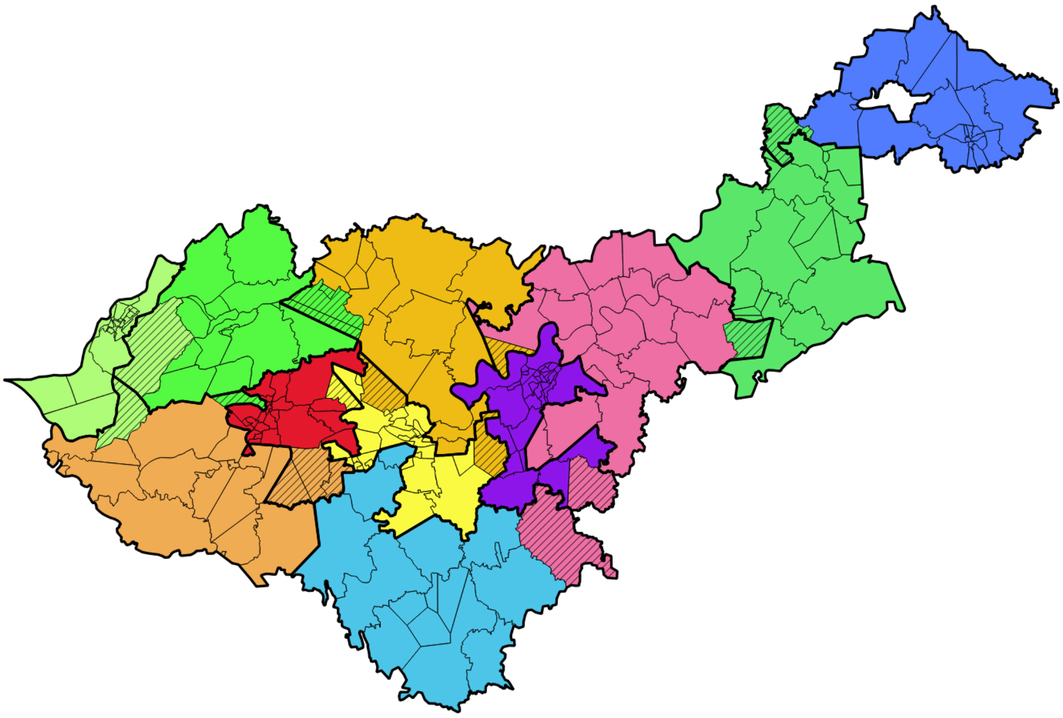


Figure 6.24: Solution obtained using model A/IV–AW on instance 1 after operational reassignment with $\omega = 20$

6.9 Conclusions

In this chapter, we have studied a real-world districting problem arising in parcel delivery. To the best of our knowledge, we are the first to address a districting problem that integrates the determination of the number of districts and the assignment of heterogeneous resources to districts. Corresponding to the two-stage nature of the problem, we have presented a novel two-stage solution approach capable of designing districts on a tactical level and adjusting them in day-to-day operations. Its effectiveness has been shown in an extensive case study on real-world data. The case study revealed that only few adaptations of the tactical district design are necessary to achieve a high degree of operational feasibility, which we believe is an interesting insight for practitioners. Moreover, the case study showed that the three tactical planning models behave as expected. Hence, conservative planners should choose model IV–A/IW since this model produces the best operational feasibility with very good results even if no or only few operational reassignments are allowed. However, the high degree of operational feasibility is achieved at the expense of the highest number of districts of the three models. Less conservative planners and planners willing to accept a slightly higher number of operational reassignments should select model AV–AW or model A/IV–AW, both yielding fairly similar results in the relevant evaluation measures. If computation time is an issue, preference should be given to model AV–AW.

7

Conclusions and Outlook

This thesis deals with territory design for on-site services. Motivated by planning requirements and problems from practice, we developed mathematical models and solution methods and performed extensive experiments on real-world data sets to evaluate their performance. In this chapter, we summarize the main contributions and findings of this thesis and give an outlook on promising topics for future research.

7.1 Conclusions

Starting with a brief introduction to general territory design with an emphasis on typical applications and common planning criteria, we presented in Chapter 2 some real-world planning requirements which have received little attention in the scientific literature on districting so far.

One of these requirements are assignment restrictions, which can either result from interdependencies between customers or from specific customer requirements with respect to the assigned service provider. In particular, requirements related to the skills of service providers have, to the best of our knowledge, not yet been considered in districting problems. In Chapter 3, we have formalized these restrictions and have presented model components that can be used to integrate them into the well-known integer programming model proposed by Hess et al. (1965).

Another planning requirement, namely the scheduling of recurring service visits, gave rise to the introduction of the multi-period service territory design problem (MPSTDP). The problem consists of a partitioning (MPSTDP-P) and a scheduling subproblem (MPSTDP-S). Since the latter has not yet been studied in the scientific literature, we have concentrated on this subproblem. In Chapter 4, we have elaborated the relevant planning criteria of the MPSTDP-S and have formalized it as a mixed integer programming model. Moreover, we have presented a heuristic based on the idea of Hess et al. (1965), which decomposes the MPSTDP-S into a location and an allocation subproblem. Extensive experiments on real-world instances and on instances that were obtained by varying the values of some parameters have shown that the heuristic produces high-quality solutions and clearly beats the software product PTV xCluster version 1.18.1.3 (PTV, 2014). A comparison with lower bounds and known optimal solutions has shown that the solutions computed by the heuristic are only a few percentage points from the optimum. Furthermore, we investigated the cost of weekday regularity, i.e., the increase in travel time when partial or strict weekday regularity is imposed. Our experiments have shown that it is influenced mainly by the variability in the customers' service times and by the compatibilities of their week rhythms. Beyond that, our approach has been integrated into a commercial software product: With the release in December 2016, PTV has replaced the previous algorithm of their xCluster Server with an algorithm based on our location-allocation approach.

A planning scenario of the MPSTDP-S with a particularly high relevance for practice was the object of investigation in Chapter 5. Beginning with a compact integer programming formulation, we have proposed a reformulation of this model consisting of a huge number of variables, which gives rise to a column generation approach. Our branch-and-price algorithm is the first specially tailored exact method for this problem and contains specific acceleration strategies. In particular, we have proposed a fast pricing heuristic and techniques to reduce the symmetry inherent to the model by variable fixations that eliminate symmetric solutions from the search tree. Experiments have shown that real-world problems consisting of up to 55 customers and a four-week planning horizon can be solved to optimality in reasonable running times. Compared to solving the compact formulation with the general-purpose MIP solver Gurobi, our algorithm yields an average reduction in running time of more than 98.1%, which demonstrates the enormous benefit of using a specialized algorithm for this problem.

In Chapter 6, we have addressed a problem in the context of parcel delivery in which, besides grouping basic areas into districts, heterogeneous resources have to be allocated to districts. We have considered two kinds of resources, namely drivers and vehicles. Drivers distinguish themselves by their contractual working times, and vehicles differ in their loading capacities. Hence, resource assignments influence the working time available in each district and the weight that can be transported on a single vehicle tour to the customers in a district. The number of districts is not given in advance, but has to be determined as part of the problem. The combination of resource assignments and a variable number of districts is unique in the existing literature on districting. The problem under study consists of two stages. In the first-stage problem, tactical delivery districts have to be designed. We strive, on the one hand, for efficiency in the sense that as few resources as possible should be employed and, hence, the number of districts should be minimized. On the other hand, the expected workload in each district should be manageable by the assigned driver without working a lot of overtime hours. We have formulated three models to accomplish this task, which differ in the level of detail of their input data and in how conservative they are with respect to the compliance with the drivers' contractual working times. Since the models cannot be solved optimally on the considered real-world instances using the general-purpose MIP solver Gurobi, we have devised a location-allocation heuristic. In the second-stage problem, the tactical district design must be adapted on a day-to-day basis due to demand fluctuations – a planning task that has received very little attention in the literature on districting, which is typically concerned with the design of long-term stable districts. Again, multiple conflicting objectives have to be considered: We would like to maintain the tactical design in order to achieve service consistency; but we also need to make sure that drivers are not overloaded and workload is balanced evenly, which might require adjustments of the tactical solution. We have introduced a model with the aim of finding a reasonable trade-off. Since travel time makes up a large part of the drivers' overall workload, we have incorporated travel time estimates into the proposed models. A case study on real-world data has revealed that slight adaptations of the tactical district design are sufficient to achieve a high degree of operational feasibility. Moreover, the case study has empirically confirmed the expected behavior of the tactical planning models. Hence, decision-makers can select one of the proposed models according to their individual preferences.

7.2 Outlook

The work presented in this thesis provides various directions for future research.

Regarding the MPSTDP-S, we have encountered further real-world planning requirements in our project with PTV that could be integrated into the proposed solution approaches. We report these requirements also in Bender et al. (2016). First, it can be desirable in practice that the day clusters of consecutive days are geographically close to each other since this makes it easier for the service provider to catch up on missed customer visits.

Second, especially if the travel times between the individual day clusters and the depot vary widely, merely balancing service times might lead to high discrepancies in the workloads on different days. Therefore, it could be beneficial to incorporate travel time approximations into the solution approaches, as we have done it in Chapter 6 in the context of parcel shipping. Even though we argued that explicitly determining the service provider's daily routes is only of little use since it can be necessary to reschedule visits in the short term, travel time approximations would allow to balance the expected workload of the service provider and, hence, result in an improved workload balance. The third enhancement, which is highly relevant for practice, is the consideration of overnight stays of the service provider. It raises the question on which days the service provider should stay in a hotel rather than return to the depot after all customers of the day have been served. Several factors may affect this decision. For example, it might be forbidden to schedule overnight stays for particular weekdays, the number of overnight stays per week may be limited, it might be allowed to schedule an overnight stay for a day only if the travel time between the day cluster and the depot exceeds a given threshold, and it may be desirable to schedule overnight stays in a way that maximizes the resulting travel time savings. First prototypical implementations of these three requirements have yielded promising results. Moreover, as we also note in Bender et al. (2016), it could be interesting to investigate the suitability of a stochastic approach to address the uncertainty resulting from short-term customer requests or other uncertain events in day-to-day business.

Especially for our branch-and-price approach, one line of future research could deal with strategies to further accelerate the algorithm such that larger problem instances can be solved. One element for this could be the identification of additional families of valid inequalities to obtain tighter linear relaxations and reduce the number of explored nodes in the search tree. It could also be worthwhile to investigate if a different decomposition into master and pricing problems facilitates the development of a fast exact pricing method. Moreover, the algorithm could be transformed into a column generation-based heuristic, e.g., by omitting the exact pricing step or by heuristically eliminating symmetry. Finally, the individual components of our algorithm can be building blocks for the development of solution methods for similar problems. Besides working on acceleration strategies, our algorithm could also be extended to consider the planning criteria that were introduced for the MPSTDP-S in Chapter 4, but have been neglected in the development of our branch-and-price algorithm. These criteria comprise several visits of a customer per week, customer-specific weekday patterns, differing service times for different visits of a customer, and weekday regularity requirements. Particularly the combination of these criteria poses an interesting challenge for future research.

Concerning the design of districts for parcel delivery companies, we see the following interesting topics to extend the work presented in this thesis. First, long-term structural changes in demand, e.g., due to the growing use of parcel delivery services or changes in population size, might necessitate that the current tactical district design is replaced by a new one. Future research could address the question how significant these changes in demand must be to justify the redesign of tactical districts. Furthermore, it could be investigated how the current tactical district design can gradually be transformed to a new one. In contrast to a major change in the district design from one day to the next, such a gradual transition would break down the learning burden of the drivers, who need to become familiar with their new districts, into smaller pieces. Another point is the integration of additional realistic constraints typically encountered in work contracts, e.g., limiting the number of days with overtime per month to a contractually specified number. Regarding the operational adaptation of districts, the presented approach could be extended to take into account historic reassignments in order to prefer assignments that have frequently been made in the past. Finally, we implicitly assume in this thesis that perfect demand forecasts are available. It would be interesting to evaluate how the proposed solution approach performs depending on different levels of forecast accuracy.

Bibliography

Achterberg, T., Koch, T., and Martin, A. (2005). Branching rules revisited. *Operations Research Letters*, 33(1):42–54.

Archetti, C., Jabali, O., and Speranza, M. G. (2015). Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61:122–134.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of 18th ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. SIAM.

Baker, J. R., Clayton, E. R., and Moore, L. J. (1989). Redesign of primary response areas for county ambulance services. *European Journal of Operational Research*, 41(1):23–32.

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

Bard, J. F., Jarrah, A. I., and Zan, J. (2010). Validating vehicle routing zone construction using Monte Carlo simulation. *European Journal of Operational Research*, 206(1):73–85.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

Bender, M., Meyer, A., Kalcsics, J., and Nickel, S. (2016). The multi-period service territory design problem – An introduction, a model and a heuristic approach. *Transportation Research Part E: Logistics and Transportation Review*, 96:135–157.

Benzarti, E., Sahin, E., and Dallery, Y. (2013). Operations management applied to home care services: Analysis of the districting problem. *Decision Support Systems*, 55(2):587–598.

Bergey, P. K., Ragsdale, C. T., and Hoskote, M. (2003). A simulated annealing genetic algorithm for the electrical power districting problem. *Annals of Operations Research*, 121(1):33–55.

Bertazzi, L., Savelsbergh, M., and Speranza, M. G. (2008). Inventory routing. In Golden, B. L., Raghavan, S., and Wasil, E. A., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 59–72. Springer.

Blais, M., Lapierre, S. D., and Laporte, G. (2003). Solving a home-care districting problem in an urban setting. *Journal of the Operational Research Society*, 54(11):1141–1147.

Blakeley, F., Argüello, B., Cao, B., Hall, W., and Knolmajer, J. (2003). Optimizing periodic maintenance operations for Schindler Elevator Corporation. *Interfaces*, 33(1):67–79.

Bodin, L. and Levy, L. (1991). The arc partitioning problem. *European Journal of Operational Research*, 53(3):393–401.

Bozkaya, B., Erkut, E., Haight, D., and Laporte, G. (2011). Designing new electoral districts for the city of Edmonton. *Interfaces*, 41(6):534–547.

Bozkaya, B., Erkut, E., and Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1):12–26.

Brandão, J. C. S. and Mercer, A. (1998). The multi-trip vehicle routing problem. *The Journal of the Operational Research Society*, 49(8):799–805.

Bundeswahlgesetz (2017). § 3 Wahlkreiskommission und Wahlkreiseinteilung. `https://www.gesetze-im-internet.de/bwahlg/__3.html`. Retrieved May 11, 2017.

Bundeswahlleiter (2017). Wahlkreiseinteilung für die Bundestagswahl 2017. `https://www.bundeswahlleiter.de/bundestagswahlen/2017/wahlkreiseinteilung.html`. Retrieved May 11, 2017.

Butsch, A. (2016). *Districting Problems – New Geometrically Motivated Approaches*. PhD thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany.

Butsch, A., Kalcsics, J., and Laporte, G. (2014). Districting for arc routing. *INFORMS Journal on Computing*, 26(4):809–824.

Caballero-Hernández, S. I., Ríos-Mercado, R. Z., López, F., and Schaeffer, S. E. (2007). Empirical evaluation of a metaheuristic for commercial territory design with joint assignment constraints. In *Proceedings of the 12th Annual International Conference on Industrial Engineering Theory, Applications, and Practice (IJIE)*, pages 422–427.

Camacho-Collados, M., Liberatore, F., and Angulo, J. M. (2015). A multi-criteria police districting problem for the efficient and effective design of patrol sector. *European Journal of Operational Research*, 246(2):674–684.

Campbell, A. M. and Hardin, J. R. (2005). Vehicle minimization for periodic deliveries. *European Journal of Operational Research*, 165(3):668–684.

Carlsson, J. G. (2012). Dividing a territory among several vehicles. *INFORMS Journal on Computing*, 24(4):565–577.

Carlsson, J. G. and Delage, E. (2013). Robust partitioning for stochastic multivehicle routing. *Operational Research*, 61(3):727–744.

Caro, F., Shirabe, T., Guignard, M., and Weintraub, A. (2004). School redistricting: Embedding GIS tools with integer programming. *The Journal of the Operational Research Society*, 55(8):836–849.

Chen, L., Chen, B., Bui, Q., and Hà, M. (2017). Designing service sectors for daily maintenance operations in a road network. *International Journal of Production Research*, 55(8):2251–2265.

Chitsaz, M., Divsalar, A., and Vansteenwegen, P. (2016). A two-phase algorithm for the cyclic inventory routing problem. *European Journal of Operational Research*, 254(2):410–426.

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2012). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24:270–287.

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2014). Thirty years of inventory routing. *Transportation Science*, 48(1):1–19.

Daganzo, C. F. and Erera, A. L. (1999). On planning and design of logistics systems for uncertain environments. In Speranza, M. G. and Stähly, P., editors, *New Trends in Distribution Logistics. Lecture Notes in Economics and Mathematical Systems*, volume 480, pages 3–21. Springer Berlin / Heidelberg.

D'Amico, S. J., Wang, S.-J., Batta, R., and Rump, C. M. (2002). A simulated annealing approach to police district design. *Computers & Operations Research*, 29(6):667–684.

de Fréminville, P. d. l. P., Desaulniers, G., Rousseau, L.-M., and Perron, S. (2015). A column generation heuristic for districting the price of a financial product. *Journal of the Operational Research Society*, 66(6):965–978.

Delgado, C., Laguna, M., and Pacheco, J. (2005). Minimizing labor requirements in a periodic vehicle loading problem. *Computational Optimization and Applications*, 32(3):299–320.

Desaulniers, G., Desrosiers, J., and Solomon, M. M. (2002). Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems. In *Essays and Surveys in Metaheuristics*, pages 309–324. Springer US.

Desrosiers, J. and Lübbecke, M. E. (2005). A primer in column generation. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column generation*, pages 1–32. Springer US, Boston.

Dong, Z. and Turnquist, M. A. (2015). Combining service frequency and vehicle routing for managing supplier shipments. *Transportation Research Part E: Logistics and Transportation Review*, 79:231–243.

Drexl, A. and Haase, K. (1999). Fast approximation methods for sales force deployment. *Management Science*, 45(10):1307–1323.

Ekici, A., Özener, O. Ö., and Kuyzu, G. (2015). Cyclic delivery schedules for an inventory routing problem. *Transportation Science*, 49(4):817–829.

Erera, A. L. (2000). *Design of Large-Scale Logistics Systems for Uncertain Environments*. PhD thesis, University of California, Berkeley, USA.

Ferland, J. A. and Guénette, G. (1990). Decision support system for the school districting problem. *Operations Research*, 38(1):15–21.

Fleischmann, B. and Paraschis, J. N. (1988). Solving a large scale districting problem: A case report. *Computers & Operations Research*, 15(6):521–533.

Forman, S. L. and Yue, Y. (2003). Congressional districting using a TSP-based genetic algorithm. In *Proceedings of the 2003 genetic and evolutionary computation conference: Part II, Chicago, IL, USA, July 12–16, 2003*, pages 2072–2083. Springer, Berlin, Heidelberg.

Francis, P. M., Smilowitz, K. R., and Tzur, M. (2008). The period vehicle routing problem and its extensions. In Golden, B. L., Raghavan, S., and Wasil, E. A., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 73–102. Springer.

Galvão, L. C., Novaes, A. G. N., Souza de Cursi, J. E., and Souza, J. C. (2006). A multiplicatively-weighted Voronoi diagram approach to logistics districting. *Computers & Operations Research*, 33(1):93–114.

Garfinkel, R. S. and Nemhauser, G. L. (1970). Optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B495–B508.

George, J. A., Lamar, B. W., and Wallace, C. A. (1997). Political district determination using large-scale network optimization. *Socio-Economic Planning Sciences*, 31(1):11–28.

Goderbauer, S. (2016). Political districting for elections to the German Bundestag: An optimization-based multi-stage heuristic respecting administrative boundaries. In Lübbecke M., Koster, A., Letmathe, P., Madlener, R., Peis, B., and Walther, G., editors, *Operations Research Proceedings 2014*, pages 181–187. Springer, Cham.

González-Ramírez, R. G., Smith, N. R., Askin, R. G., Miranda, P. A., and Sánchez, J. M. (2011). A hybrid metaheuristic approach to optimize the districting design of a parcel company. *Journal of Applied Research and Technology*, 9(1):19–35.

Groër, C., Golden, B., and Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & Service Operations Management*, 11(4):630–643.

Hanafi, S., Freville, A., and Vaca, P. (1999). Municipal solid waste collection: An effective data structure for solving the sectorization problem with local search methods. *INFOR: Information Systems and Operational Research*, 37(3):236–254.

Haughton, M. A. (2008). The efficacy of exclusive territory assignments to delivery vehicle drivers. *European Journal of Operational Research*, 184(1):24–38.

Haugland, D., Ho, S. C., and Laporte, G. (2007). Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3):997–1010.

Hess, S. W. and Samuels, S. A. (1971). Experiences with a sales districting model: Criteria and implementation. *Management Science*, 18(4):41–54.

Hess, S. W., Weaver, J. B., Siegfeldt, H. J., Whelan, J. N., and Zitlau, P. A. (1965). Nonpartisan political redistricting by computer. *Operations Research*, 13(6):998–1006.

Hojati, M. (1996). Optimal political districting. *Computers & Operations Research*, 23(12):1147–1161.

Irnich, S., Schneider, M., and Vigo, D. (2014). Four variants of the vehicle routing problem. In Toth, P. and Vigo, D., editors, *Vehicle Routing: Problems, Methods, and Applications*, volume 18, pages 241–271. Siam.

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

Janssens, J., Van Den Bergh, J., Sörensen, K., and Cattrysse, D. (2015). Multi-objective microzone-based vehicle routing for courier companies: From tactical to operational planning. *European Journal of Operational Research*, 242(1):222–231.

Jarrah, A. I. and Bard, J. F. (2012). Large-scale pickup and delivery work area design. *Computers & Operations Research*, 39(12):3102–3118.

Jepsen, M., Petersen, B., Spoorendonk, S., and Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497–511.

Kalcsics, J. (2015). Districting problems. In Laporte, G., Nickel, S., and Saldanha da Gama, F., editors, *Location Science*, pages 595–622. Springer International Publishing, 1 edition.

Kalcsics, J., Nickel, S., and Schröder, M. (2005). Towards a unified territorial design approach — applications, algorithms and GIS integration. *Top*, 13(1):1–56.

Kazan, O., Dawande, M., Sriskandarajah, C., and Stecke, K. E. (2012). Balancing perfectly periodic service schedules: An application from recycling and waste management. *Naval Research Logistics*, 59(2):160–171.

Korst, J., Aarts, E., Lenstra, J. K., and Wessels, J. (1991). Periodic multiprocessor scheduling. In *PARLE'91 Parallel Architectures and Languages Europe*, pages 166–178.

Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2015a). The generalized consistent vehicle routing problem. *Transportation Science*, 49(4):796–816.

Kovacs, A. A., Parragh, S. N., and Hartl, R. F. (2014). A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks*, 63(1):60–81.

Kovacs, A. A., Parragh, S. N., and Hartl, R. F. (2015b). The multi-objective generalized consistent vehicle routing problem. *European Journal of Operational Research*, 247(2):441–458.

Lei, H., Laporte, G., and Guo, B. (2012). Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1):67–85.

Lei, H., Laporte, G., Liu, Y., and Zhang, T. (2015). Dynamic design of sales territories. *Computers & Operations Research*, 56:84–92.

Lei, H., Wang, R., and Laporte, G. (2016). Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. *Computers & Operations Research*, 67:12–24.

Li, M., Wang, Z., and Chan, F. T. (2016). A robust inventory routing policy under inventory inaccuracy and replenishment lead-time. *Transportation Research Part E: Logistics and Transportation Review*, 91:290–305.

Linderoth, J. T. and Savelsbergh, M. W. P. (1999). A computational study of search strategies for mixed integer programming. *INFORMS Journal on Computing*, 11(2):173–187.

López-Pérez, J. F. and Ríos-Mercado, R. Z. (2013). Embotelladoras ARCA uses operations research to improve territory design plans. *Interfaces*, 43(3):209–220.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007–1023.

Luo, Z., Qin, H., Che, C., and Lim, A. (2015). On service consistency in multi-period vehicle routing. *European Journal of Operational Research*, 243(3):731–744.

Mehrotra, A., Johnson, E. L., and Nemhauser, G. L. (1998). An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1114.

Minciardi, R., Puliafito, P. P., and Zoppoli, R. (1981). A districting procedure for social organizations. *European Journal of Operational Research*, 8(1):47–57.

Miranda, P. A., Blazquez, C. A., Vergara, R., and Weitzler, S. (2015). A novel methodology for designing a household waste collection system for insular zones. *Transportation Research Part E: Logistics and Transportation Review*, 77:227–247.

Mourão, M. C., Nunes, A. C., and Prins, C. (2009). Heuristic methods for the sectoring arc routing problem. *European Journal of Operational Research*, 196(3):856–868.

Mourgaya, M. and Vanderbeck, F. (2007). Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, 183(3):1028–1041.

Muyldermans, L., Cattrysse, D., and Van Oudheusden, D. (2003). District design for arc-routing applications. *Journal of the Operational Research Society*, 54(11):1209–1221.

Muyldermans, L., Cattrysse, D., Van Oudheusden, D., and Lotan, T. (2002). Districting for salt spreading operations. *European Journal of Operational Research*, 139(3):521–532.

Nemhauser, G. L. and Wolsey, L. A. (1999). *Integer and Combinatorial Optimization*. John Wiley & Sons, New York.

Nourinejad, M. and Roorda, M. J. (2017). A continuous approximation model for the fleet composition problem on the rectangular grid. *OR Spectrum*, 39(2):373–401.

Novaes, A. G. N., de Cursi, J. E. S., and Graciolli, O. D. (2000). A continuous approach to the design of physical distribution systems. *Computers & Operations Research*, 27(9):877–893.

Novaes, A. G. N. and Graciolli, O. D. (1999). Designing multi-vehicle delivery tours in a grid-cell format. *European Journal of Operational Research*, 119(3):613–634.

Ouyang, Y. (2007). Design of vehicle routing zones for large-scale distribution systems. *Transportation Research Part B: Methodological*, 41(10):1079–1093.

Padberg, M. W. (1973). On the facial structure of set packing polyhedra. *Mathematical Programming*, 5(1):199–215.

Perrier, N., Langevin, A., and Campbell, J. F. (2008). The sector design and assignment problem for snow disposal operations. *European Journal of Operational Research*, 189(2):508–525.

Polacek, M., Doerner, K. F., Hartl, R. F., Kiechle, G., and Reimann, M. (2007). Scheduling periodic customer visits for a traveling salesperson. *European Journal of Operational Research*, 179:823–837.

Pouls, M. (2016). A branch-price-and-cut algorithm for customer scheduling in the context of the multi-period service territory design problem. Master's thesis, Institute of Operations Research (IOR) at the Karlsruhe Institute of Technology, Karlsruhe, Germany.

PTV (2014). PTV xCluster Server version 1.18.1.3. `http://xserver.ptvgroup.com/en-uk/products/ptv-xserver/ptv-xcluster-ptv-xterritory/`.

Raa, B. and Aghezzaf, E.-H. (2009). A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research*, 192(2):429–441.

Rahimi-Vahed, A., Crainic, T. G., Gendreau, M., and Rei, W. (2015). Fleet-sizing for multi-depot and periodic vehicle routing problems using a modular heuristic algorithm. *Computers & Operations Research*, 53:9–23.

Ricca, F., Scozzari, A., and Simeone, B. (2008). Weighted Voronoi region algorithms for political districting. *Mathematical and Computer Modelling*, 48(9-10):1468–1477.

Ricca, F., Scozzari, A., and Simeone, B. (2013). Political districting: From classical models to recent approaches. *Annals of Operations Research*, 204(1):271–299.

Ricca, F. and Simeone, B. (2008). Local search algorithms for political districting. *European Journal of Operational Research*, 189(3):1409–1426.

Ríos-Mercado, R. Z. and López-Pérez, J. F. (2013). Commercial territory design planning with realignment and disjoint assignment requirements. *Omega*, 41(3):525–535.

Salazar-Aguilar, M. A., Ríos-Mercado, R. Z., and Cabrera-Ríos, M. (2011). New models for commercial territory design. *Networks and Spatial Economics*, 11(3):487–507.

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

Savelsbergh, M. and Sol, M. (1998). Drive: Dynamic routing of independent vehicles. *Operations Research*, 46(4):474–490.

Schneider, M., Stenger, A., Schwahn, F., and Vigo, D. (2015). Territory-based vehicle routing in the presence of time-window constraints. *Transportation Science*, 49(4):732–751.

Schoepfle, O. B. and Church, R. L. (1989). A fast, network-based, hybrid heuristic for the assignment of students to schools. *The Journal of the Operational Research Society*, 40(11):1029–1040.

Schoepfle, O. B. and Church, R. L. (1991). A new network representation of a "classic" school districting problem. *Socio-Economic Planning Sciences*, 25(3):189–197.

Shanker, R. J., Turner, R. E., and Zoltners, A. A. (1975). Sales territory design: An integrated approach. *Management Science*, 22(3):309–320.

Shirabe, T. (2009). Districting modeling with exact contiguity constraints. *Environment and Planning B: Planning and Design*, 36(6):1053–1066.

Silva de Assis, L., Franca, P. M., and Usberti, F. L. (2014). A redistricting problem applied to meter reading in power distribution networks. *Computers & Operations Research*, 41:65–75.

Smilowitz, K., Nowak, M., and Jiang, T. (2013). Workforce management in periodic delivery operations. *Transportation Science*, 47(2):214–230.

Sungur, I., Ren, Y., Ordóñez, F., Dessouky, M., and Zhong, H. (2010). A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science*, 44(2):193–205.

Taillard, É. D., Laporte, G., and Gendreau, M. (1996). Vehicle routeing with multiple use of vehicles. *The Journal of the Operational Research Society*, 47(8):1065–1070.

Toth, P. and Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(1-3):487–512.

Toth, P. and Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15(4):333–346.

Vickrey, W. (1961). On the prevention of gerrymandering. *Political Science Quarterly*, 76(1):105–110.

Wei, W. D. and Liu, C. L. (1983). On a periodic maintenance problem. *Operations Research Letters*, 2(2):90–93.

Williams, J. C. (1995). Political redistricting: A review. *Papers in Regional Science*, 74(1):13–40.

Wolsey, L. A. (1998). *Integer programming*. John Wiley & Sons, New York.

Wong, K. and Beasley, J. (1984). Vehicle routing using fixed delivery areas. *Omega*, 12(6):591–600.

Wong, R. T. (2008). Vehicle routing for small package delivery and pickup services. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 475–485. Springer US, Boston.

Yamada, T. (2009). A mini–max spanning forest approach to the political districting problem. *International Journal of Systems Science*, 40(5):471–477.

Zhong, H., Hall, R. W., and Dessouky, M. (2007). Territory planning and vehicle dispatching with driver learning. *Transportation Science*, 41(1):74–89.

Zoltners, A. A. and Sinha, P. (1983). Sales territory alignment: A review and model. *Management Science*, 29(11):1237–1256.

Zoltners, A. A. and Sinha, P. (2005). Sales territory design: Thirty years of modeling and implementation. *Marketing Science*, 24(3):313–331.