Districting Problems – New Geometrically Motivated Approaches

Zur Erlangung des akademischen Grades eines Doktors
der Wirtschaftswissenschaften
(Doctor rerum politicarum)

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

genehmigte
DISSERTATION

von

Dipl.-Inform. Alexander Butsch

Tag der mündlichen Prüfung: 13.07.2016
Referent: Prof. Dr. Stefan Nickel
Korreferent: Prof. Dr. Jörg Kalcsics

Acknowledgement

In the following I want to thank some people for their support and guidance during writing this thesis. Without them a successful finish of this thesis would not have been possible.

First of all, I want to thank Prof. Dr. Stefan Nickel for supervising my thesis, for his beneficial ideas throughout this supervision, and for giving me the opportunity to work at his chair.

Second, I want to thank my co-supervisor Prof. Dr. Jörg Kalcsics, in particular, for his support in my first steps in research. In the sequel I benefited from many fruitful discussions and the knowledge that he always lent me a sympathetic ear.

Thank you to my colleagues Marliese Amann, Ines Arnolds, Fabian Dunke, Tanya Gonser, Prof. Dr. Diethard Pallaschke, Melanie Reuter-Oppermann, Brita Rohrbeck, Anne Zander, and Hans-Peter Ziegler for the nice time at the chair. The working atmosphere always was very well. Moreover, I want to thank Matthias Bender for several inspiring discussions about districting.

A special thanks goes to Stefan Bach, Andrea Butsch and to my colleagues Fabian Dunke, Tanya Gonser, and Anne Zander who proofread this thesis and helped to improve the quality of this thesis.

Last but not least I want to thank my family. Thank you to my parents for their endless support you have given to me during all my life. Thank you to my girlfriend Caren and my son Bastian for their love, patience and encouragement, mainly during the last months.

Alexander Butsch

Abstract

Districting is the problem of grouping small basic areas into larger districts, subject to a number of relevant planning criteria. Balance describes the requirement for districts to have approximately the same size with respect to a quantifiable activity measure, e.g., number of inhabitants or working time. A district is said to be geographically compact if it is closely and firmly packed together. Contiguity means that it is possible to travel between the basic areas of a district without leaving the district. The basic areas can be represented by polygons (e.g., cities), lines (e.g., streets), or points (e.g., customers). In the literature, polygonal representations are predominant. However, there are many practical applications for points or lines. Therefore, this thesis mainly focuses on these representations and applications.

The least well defined planning criterion is compactness. Many compactness measures have been proposed, but none of them has proven to be comprehensive. The first part of this thesis summarizes and compares existing measures, and enhances some measuring approaches in order to make them applicable in the case of point or line representations.

Point representations arise for example in the context of sales or service districting. In the second part, this thesis focuses on solution approaches that utilize the problem's underlying geometrical information. It improves the already existing Recursive Partitioning Algorithm (RPA) significantly, especially in terms of compactness. Moreover, it presents an approach based on Power Diagrams that either can be used as a stand-alone algorithm or as a post-processing of the RPA. The Power Diagram based approach improves compactness even further, however, the RPA performs better in terms of balance. Although both approaches are geometrically motivated, distances on a road network can be incorporated.

Line representations occur in districting problems on road networks, for example for mail or leaflet delivery. Usually, the service of a district is provided within one tour. In the third part, this thesis introduces a corresponding algorithm combining features of geometric approaches, tabu search, and adaptive randomized neighborhood search. It is the first approach that includes compactness as well as routing distances explicitly. Computational tests on real-world data confirm the efficiency of this approach and the quality of its solutions.

Contents

1 Introduction

We all come into contact with districts in our daily lives, directly or indirectly. The postman delivering our mails every morning has a delivery area he is responsible for, the team picking up our waste every week has a sector it is responsible for, or in the case of snowfall, each truck is responsible for removing the snow within a district, just to mention some examples. Catholic communities divide their regions into smaller districts in order to organize the carol singers visiting church members, or a supermarket defines districts for each leaflet deliverer. Moreover, when there is an election, e.g., in Germany for the Landtag or the Bundestag, each of us is assigned to an electoral district where we have to vote. In our business lives, we come into contact with districts as well. A company may divide its trading area into smaller sales regions and locate a branch office within each region. In the field of public administration, each public office has an area of responsibility and the inhabitants of that area should go to this office.

All of these examples have something in common: A large geographical area is partitioned into smaller districts. Other terms for *district* are territory, sector, zone, region, or area of responsibility. These sub-divisions usually follow some constraints. Especially, there are small geographic areas that are indivisible, so-called *basic areas*. For example, the border between two regions of responsibility for garbage trucks does not normally lie in in the middle of a street, instead each street is assigned to one district as a whole. In the context of electoral districts, usually whole city quarters are assigned to the same district. Hence, beside the top-down view there is a bottom-up view. A district can be interpreted as a composition of basic areas. Moreover, the process of designing these districts takes some further requirements into account depending on the application. Often, there are fairness requirements on the size according to a quantifiable measure. For example, each postman should have approximately the same expected workload, each electoral district approximately the same number of voters, or each sales region approximately the same sales volume. Moreover, in many contexts for organizational and economic reasons each district should be connected. From an organizational point of view, connected districts induce clearly defined areas of responsibility. From an economical point of view, a connected district prevents unproductive

travel times between the connected components. Furthermore, in general there are further requirements on the shapes of the districts. They should be visually appealing, for example, nearly round-shaped or square. On the one hand, most likely the travel times within a nearly round-shaped or square district are smaller than in a long-shaped district. On the other hand, in the context of electoral districts, the requirement for regularly shaped districts helps to prevent manipulations.

Altogether, these observations lead to the following definition: Districting is the problem of grouping small geographic areas, called basic areas, into larger geographic clusters, called districts, subject to a number of relevant planning criteria. Typical examples for basic areas are cities, zip-code areas, streets, and single customer locations. The most important planning criteria are balance, compactness and contiguity. Balance describes the requirement for districts to have approximately the same size with respect to a quantifiable activity measure, such as number of inhabitants, sales potential, or working time. A district is said to be geographically compact if it is closely and firmly packed together, e.g., nearly round-shaped or square and undistorted. In a contiguous district it is possible to reach every basic area within this district from every other without having to leave the district.

Districting problems also occur as part of other problems in the context of operations research. For example, concerning routing problems many approaches utilize the principle of "cluster first – route second", i.e., in the first step they group the customers into clusters, after that, in the second step they determine a route through this set of customers. In this context, the "cluster first" step can be seen as districting step.
Often, in the context of facility location problems the question of where to locate facilities comes along with the question of how to allocate the customers to these facilities. Especially, if the facilities should be equally sized, the problem can be solved by firstly determining sets of customers served by the same facility and secondly locating the corresponding facilities.

Moreover, there exists a problem looking similar to districting, called clustering. Clustering is the problem of grouping objects such that the objects of the same group, called cluster, are more similar to each other than to those in other clusters. Assume the objects as located in a plane, for example, by interpreting their properties as coordinates. In this case, similarity can be interpreted as spatial closeness. However, there is a main difference between clustering and districting. The general clustering problem does not take the size of the clusters into account. Hence, the obtained clusters are allowed to be very unbalanced.

1.1 Districting Applications

There is a broad range of practical applications for districting. This section outlines the four main categories of districting problems: Political districting, sales districting, service districting, and distribution districting. The categorization is based on Kalcsics [26].

1.1.1 Political Districting

The design of electoral districts is the application that has received most attention in districting literature [6, 7, 11, 17, 18, 23, 32, 38, 39, 40, 45, 46]. Typically, a governmental area has to be divided into a given number of electoral districts and each of these districts elects one political representative in order to send him to a parliament. For example, the voting system for the German Bundestag is known as personalized proportional representation. That is a combination of proportional representation and plurality vote. Each of the 299 districts elects one representative using a first-past-the-post voting. In order to respect the principle of "one man – one vote", i.e., every vote should have the same power, the number of voters should be approximately equal within each electoral district. In other words, the districting plan should be balanced. For example, for the election of the German Bundestag the deviation of a district from the average size should be at most 15%. If the deviation is more than 25% a redistricting is required [35]. Therefore, due to population changes between the elections of 2009 and 2013 there were changes to 21 districts. The deviations for the election of the U.S. congress are noticeably smaller. After the census in 2000 the deviation was at most 0.60% [45].

Often, in the context of political districting, basic areas correspond to cities or quarters, i.e., each city (-quarter) needs to be assigned to one district as a whole. Hence, the basic areas are most commonly represented by polygons. Sometimes, further prescribed borders have to be taken into account during the planning process. For example, for the election of the German Bundestag the borders of the 16 federal states are respected.

In order to prevent *gerrymandering* other important criteria are contiguity and compactness. Gerrymandering is the practice of designing districts in order to prefer a particular party. The term is a combination of the terms "Gerry" and "Salamander". In 1812 governor Elbridge Gerry redistricted Massachusetts for the election of the state senate where one of the electoral districts was said to look like a salamander. Figure 1.1 shows a cartooned illustration of this district [44]. The main idea of Gerrymandering is the usage of the "the winner takes it all" principle. If a party wins the election within a district it does not matter if this party has only a few more votes than another party or if almost everybody votes for this party. In contrast to this, if a party loses the election within a district, actually every vote is useless for

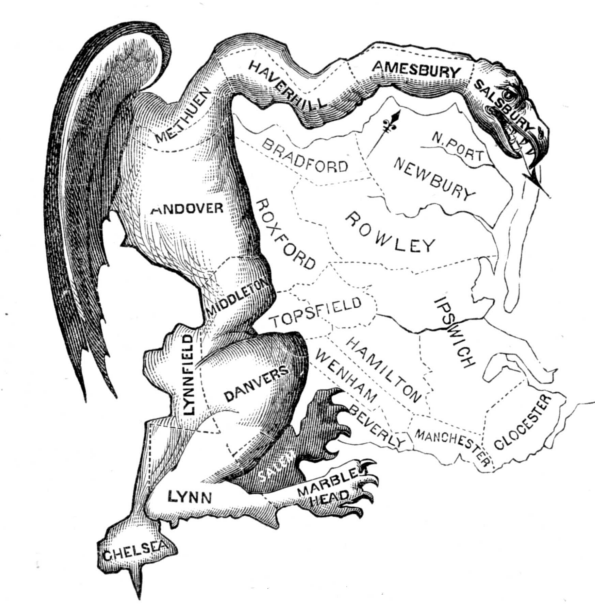


Figure 1.1: Cartoon illustration of gerrymandering [44]

the losing party. Thus, if a party wins many districts barely, but loses some districts clearly, it may obtain a majority in the parliament, even though it has no majority according to the voters in total. Lewyn [31] gives more details about gerrymandering. However, Garfinkel and Nemhauser [17] argue that compactness is of smaller relevance for algorithmic planning. Manipulations are more or less impossible if an algorithm that does not consider political data generates the districting plan.

In contrast to this Puppe and Tasnádi [37] propose taking political data into account explicitly. They define a districting plan as unbiased if for each party the number of seats is proportional to the corresponding share of voters. In other words, the result of a plurality vote should be as close as possible to the result of proportional representation. In order to obtain an unbiased districting plan, the authors consider the problem from a game theoretical point of view.

Some approaches include additional planning criteria. For example, the consideration of socio-economic homogeneity within the generated districts [6], a fair representation of minorities [46], the consideration of geographic obstacles [18], or similarity to an existing districting plan [6].

Williams [46] and Ricca et al. [40] give more details about criteria and approaches in the context of political districting. Webster [45] presents a reflection on current evaluation criteria.

1.1.2 Sales Districting

Zoltners and Sinha [49] report that in the US about 11% of the full-time employees are field or retail salespersons. Companies pay more than a trillion dollars per year to these employees. They argue, that with better planning, there is a potential for improving sales profit by 2% to 7% in many companies. Hence, the planning of sales districts is an economically important field.

The aim of sales districting is the allocation of customers to salespersons. Typically, each district corresponds to the area of responsibility for one salespersons or one team. Hence, the districts should be contiguous and non-overlapping in order to obtain geographically clearly defined sales districts and to avoid competitions between the salespersons of the same company. In general, a salesperson has to visit his customers regularly. Unfortunately, in most cases it is very hard to determine the corresponding travel times explicitly since visit frequencies, time windows, overnight trips, and so on, make the problem extensive. Using compact districts is a good compromise because compactness can be seen as a proxy for the requirement of small travel times.
Usually, balance is a planning criterion in the context of sales districting. First, the salespersons should have approximately the same workload for servicing the customers. Moreover, most commonly each salesperson is rewarded based on the sales volume of his customers. Hence, the salespersons should have approximately the same income opportunities in order to avoid discontent among them. Zoltners and Sinha [48] model these two balance criteria, while Ríos-Mercado and Fernández [41] model actually three balance criteria, namely number of customers, product demand and workload. Hess and Samuels [22] and Fleischmann and Paraschis [16] require that the districts are balanced according to one activity measure, regardless of which one is used. However, in general, the main goal of a company is profit maximization, while other criteria are of minor importance. That is the reason why Drexl and Haase [14] and Haase and Müller [19] do not take balance into account explicitly, but they define a maximum feasible working time for each salesperson.

Some authors present approaches where a customer's sales volume depends on the time the service person invested on this customer [14, 19]. Also the sales force size, i.e., the number of required districts, can be part of the planning [24, 49]. Zoltners and Sinha [48, 49] present comprehensive overviews of the proposed sales districting approaches in the literature.

Most commonly, the presented approaches aggregate the single customers, e.g., according to their zip-codes, and treat these aggregations as basic areas. In contrast to this, Chapters 4 and 5 of this thesis present approaches where the customers are directly treated as basic areas.

1.1.3 Service Districting

Applications in the context of service districting are numerous. These applications can be divided into two sub-classes differing in whether a customer has to visit a fixed service location or the service is provided at a customer's home. The design of school districts is an application where the service is provided at fixed locations, i.e., at the existing schools. In some countries, the school district a student lives in defines the school the student has to go to. Common criteria in the context of school districting are the satisfaction of capacity constraints and contiguity constraints, the minimization of the total distances the students have to travel, the fulfillment of maximal feasible travel times for single students, the consideration of which students have to take a school bus, and the consideration of racial balance [10, 15, 42, 43]. Further examples are regions for hospitals or public utilities, where each inhabitant is allocated to a location.

Due to the aging society the field of home-care services is gaining in importance. A home-care district corresponds to the area of responsibility for one team of health-care staff, such as nurses. These districts should be connected, compact, respect administrative boundaries, have approximately the same workload, and there should be a good accessibility within each district, especially by public transport services [2, 4].

There are some further applications where the service is provided on-site. For example, municipal solid waste collection [20], salt spreading, or road maintenance [33, 34], and snow disposal [36]. Typically, there is one district for each truck. These districts should be compact, contiguous, have approximately the same workload, and allow a good routing. In the context of meter reading [13], mail delivery [5], or leaflet delivery [8] the requirements are similar. The difference is that for these applications the service is generally provided by foot. Chapter 6 of this thesis addresses these applications in more detail.

In addition, the following applications are mentioned in the districting literature: Bergey et al. [3] deal with the problem of dividing a physical power grid into districts for electricity companies. The background of this application is the transformation from a monopolistic governmental company to competitive private companies. D'Amico et al. [12] focus on the planning of command districts for the police. In this case, there are further requirements on the maximal response time to calls for service. In a similar context, Camacho-Collados et al. [9] address the problem of designing patrol sectors.

1.1.4 Distribution Districting

In the field of pickup and delivery planning every requesting costumer has to be serviced. Typically, their demands vary from day to day. In this context, some approaches propose to determine districts on a tactical level and routes on an operational level, i.e., these approaches utilize the principle of "cluster first – route second" [1, 21, 25, 28, 29, 30]. Usually, there is a one-to-one relation between a district and a driver. In this case, the driver becomes familiar with his district and increases his performance. Zhong et al. [47] model this correlation explicitly. Jarrah and Bard [25] argue that the customer also becomes familiar with his driver which results in an enhanced client loyalty. In contrast to this, Zhong et al. [47] propose to allow some customers that have no fixed assignment to a district in order to balance the workload for each day.

The districts should be contiguous and compact in order to allow good routes on the day-to-day basis. Moreover, the districts should be balanced in terms of working time or satisfy a maximal feasible working time within a given time horizon. Actually, the working time contains both the service times and the travel times. Hence, the majority of the approaches includes at least an approximation of the travel times. Chapter 4 of this thesis describes some of these approaches in more detail.

1.2 Scope of this Thesis

This thesis is organized as follows: The first part focuses on modelling of districting problems. Chapter 2 provides an overview of the most common components and planning criteria and points out why it is necessary to apply heuristics in order to solve practical districting problems. The main challenge in this context is the assessment of compactness. It seems to be impossible to define a comprehensive compactness measure. Hence, Chapter 3 describes compactness and measures proposed so far in detail, evaluates these measures, and enhances some measuring approaches. In particular, Chapter 3 analyzes the application of these measures to basic areas that are represented as points.

There are three common representations of basic areas in the districting literature: Polygons, points and (poly-)lines. Polygonal representations mainly occur if the basic areas correspond to cities, quarters or zip-code cial areas. Hence, the majority of the literature concerning political districting utilizes polygonal representations. Thus, this case is well studied in the literature. In contrast to this, the districting literature concerning basic areas represented by points explicitly is rather limited, although point representations are the common case if single customers are considered as basic areas, for example in the context of sales or service districting. Typically, approaches proposed in the literature aggregate these customers and treat these aggregations as basic areas. In contrast to this, the second part of this thesis deals directly with point representations. First, Chapter 4 enhances an approach of Kalcsics et al. [27], called Recursive Partitioning Algorithm (RPA). Especially, in terms of compactness it improves the RPA significantly. Subsequently, Chapter 5 presents an approach based on Power Diagrams. This approach can either be used as a stand-alone algorithm or as a post-processing step applied to the solutions of the RPA. Both approaches have in common that they utilize the districting problem's underlying geometrical information. The main difference is the treatment of balance. The former uses balance both as a soft and a hard criterion, whereas the latter uses it only as a hard criterion.

The literature concerning line representations is also not extensive. In this context, typically each line requires a service and the service of a district is provided within one tour. Typical examples are the delivery of mail or leaflets. In this context, Chapter 6 introduces an algorithm combining features of tabu search and adaptive randomized neighborhood search. In contrast to former approaches, it considers compactness as well as routing distances explicitly.

This thesis concludes with a presentation of our C++ library for solving districting problems, called *"Lizard"* (LIbrary of optimiZation AlgoRithms for Districting), and an outlook to future research.

Bibliography

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

[2] E. Benzarti, E. Sahin, and Y. Dallery. Operations management applied to home care services: Analysis of the districting problem. *Decision Support Systems*, 55:587–598, 2013.

[3] P. K. Bergey, C. T. Ragsdale, and M. Hoskote. A Simulated Annealing Genetic Algorithm for the Electrical Power Districting Problem. *Annals of Operations Research*, 121 (1):33–55, 2003.

[4] M. Blais, S. D. Lapierre, and G. Laporte. Solving a home-care districting problem in an urban setting. *Journal of the Operational Research Society*, 54(11):1141–1147, 2003.

[5] L. D. Bodin and L. Levy. The arc partitioning problem. *European Journal of Operational Research*, 53(3):393–401, 1991.

[6] B. Bozkaya, E. Erkut, and G. Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1): 12–26, 2003.

[7] B. Bozkaya, E. Erkut, D. Haight, and G. Laporte. Designing new electoral districts for the city of Edmonton. *Interfaces*, 41(6):534–547, 2011.

[8] A. Butsch, J. Kalcsics, and G. Laporte. Districting for arc routing. *INFORMS Journal on Computing*, 26(4):809–824, 2014.

[9] M. Camacho-Collados, F. Liberatore, and J. M. Angulo. A multi-criteria Police Districting Problem for the efficient and effective design of patrol sector. *European Journal of Operational Research*, 246(2):674–684, 2015.

[10] F. Caro, T. Shirabe, M. Guignard, and A. Weintraub. School redistricting: embedding GIS tools with integer programming. *The Journal of the Operational Research Society*, 55(8):836–849, 2004.

[11] C.-I. Chou. A Knowledge-based Evolution Algorithm approach to political districting problem. *Computer Physics Communications*, 182(1):209–212, 2011.

[12] S. J. D'Amico, S.-J. Wang, R. Batta, and C. M. Rump. A simulated annealing approach to police district design. *Computers & Operations Research*, 29(6):667–684, 2002.

[13] L. S. de Assis, P. M. Franca, and F. L. Usberti. A redistricting problem applied to meter reading in power distribution networks. *Computers & Operations Research*, 41 (1):65–75, 2014.

[14] A. Drexl and K. Haase. Fast Approximation Methods for Sales Force Deployment. *Management Science*, 45(10):1307–1323, 1999.

[15] J. A. Ferland and G. Guénette. Decision support system for the school districting problem. *Operations Research*, 38:15–21, 1990.

[16] B. Fleischmann and J. N. Paraschis. Solving a large scale districting problem: a case report. *Computers & Operations Research*, 15(6):521–533, 1988.

[17] R. S. Garfinkel and G. L. Nemhauser. Optimal Political Districting by Implicit Enumeration Techniques. *Management Science*, 16(8):495–508, 1970.

[18] J. A. George, B. W. Lamar, and C. A. Wallace. Political district determination using large-scale network optimization. *Socio-Economic Planning Sciences*, 31:11–28, 1997.

[19] K. Haase and S. Müller. Upper and lower bounds for the sales force deployment problem with explicit contiguity constraints. *European Journal of Operational Research*, 237(2): 677–689, 2014.

[20] S. Hanafi, A. Fréville, and P. Vaca. Municipal solid waste collection: An effective data structure for solving the sectorization problem with local search methods. *INFOR*, 37: 236–254, 1999.

[21] D. Haugland, S. C. Ho, and G. Laporte. Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3):997–1010, 2007.

[22] S. W. Hess and S. A. Samuels. Experiences with a Sales Districting Model: Criteria and Implementation. *Management Science*, 18(4-part-ii):41–54, 1971.

[23] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. Nonpartisan Political Redistricting by Computer. *Operations Research*, 13(6):998–1006, 1965.

[24] R. S. Howick and M. Pidd. Sales force deployment models. *European Journal of Operational Research*, 48(3):295–310, 1990.

[25] A. I. Jarrah and J. F. Bard. Large-scale pickup and delivery work area design. *Computers & Operations Research*, 39(12):3102–3118, 2012.

[26] J. Kalcsics. *Location Science*, chapter Districting Problems, pages 595–622. Springer International Publishing, 2015. ISBN 978-3319131115.

[27] J. Kalcsics, S. Nickel, and M. Schröder. Towards a Unified Territorial Design Approach – Applications, Algorithms and GIS Integration. *TOP*, 13(1):1–74, 2005.

[28] H. Lei, G. Laporte, and B. Guo. Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1–2):67–85, 2012.

[29] H. Lei, G. Laporte, Y. Liu, and T. Zhang. Dynamic design of sales territories. *Computers & Operations Research*, 56:84–92, 2015.

[30] H. Lei, R. Wang, and G. Laporte. Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. *Computers & Operations Research*, 67:12–24, 2016.

[31] M. Lewyn. How to limit gerrymandering. *Florida Law Review*, 45:403–486, 1993.

[32] A. Mehrotra, E. L. Johnson, and G. L. Nemhauser. An Optimization based Heuristic for Political Districting. *Management Science*, 44:1100–1114, 1998.

[33] L. Muyldermans, D. Cattrysse, D. Van Oudheusden, and T. Lotan. Districting for salt spreading operations. *European Journal of Operational Research*, 139(3):521–532, 2002.

[34] L. Muyldermans, D. Cattrysse, and D. Van Oudheusden. District design for arc-routing applications. *Journal of the Operational Research Society*, 54(11):1209–1221, 2003.

[35] Outlook Verlag. Bundeswahlgesetz, 2013.

[36] N. Perrier, A. Langevin, and J. F. Campbell. The sector design and assignment problem for snow disposal operations. *European Journal of Operational Research*, 189(2):508–525, 2008.

[37] C. Puppe and A. Tasnádi. A computational approach to unbiased districting. *Mathematical and Computer Modelling*, 48(9–10):1455–1460, 2008.

[38] F. Ricca and B. Simeone. Local search algorithms for political districting. *European Journal of Operational Research*, 189(3):1409–1426, 2008.

[39] F. Ricca, A. Scozzari, and B. Simeone. Weighted Voronoi region algorithms for political districting. *Mathematical and Computer Modelling*, 48:1468–1477, 2008.

[40] F. Ricca, A. Scozzari, and B. Simeone. Political Districting: from classical models to recent approaches. *Annals of Operations Research*, 204(1):271–299, 2013.

[41] R. Z. Ríos-Mercado and E. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776, 2009.

[42] O. B. Schoepfle and R. L. Church. A Fast, Network-Based, Hybrid Heuristic for the Assignment of Students to Schools. *The Journal of the Operational Research Society*, 40(11):1029–1040, 1989.

[43] O. B. Schoepfle and R. L. Church. A New Network representation of a "classic" School districting problem. *Socio-Economic Planning Sciences*, 25(3):189–197, 1991.

[44] E. Tisdale, 1812. Originally published in the Boston Gazette, March 26.

[45] G. R. Webster. Reflections on current criteria to evaluate redistricting plans. *Political Geography*, 32:3–14, 2013.

[46] J. C. Williams. Political Redistricting: A Review. *Papers in Regional Science*, 74(1): 13–40, 1995.

[47] H. Zhong, R. W. Hall, and M. Dessouky. Territory planning and vehicle dispatching with driver learning. *Transportation Science*, 41(1):74–89, 2007.

[48] A. Zoltners and P. Sinha. Sales Territory Alignment: A Review and Model. *Management Science*, 29(11):1237–1256, 1983.

[49] A. Zoltners and P. Sinha. Sales Territory Design: Thirty Years of Modeling and Implementation. *Marketing Science*, 24(3):313–331, 2005.

Part I

Modelling Districting Problems

2 The General Model

Contents

This chapter gives a general overview of our model used for districting problems. Therefore, it summarizes the most common components (cf. Section 2.1) and planning criteria (cf. Section 2.2) of districting problems. Depending on the application or on the used algorithm some additional components or criteria may have to be taken into account or some adaptations are necessary. After that, Section 2.3 states some variations of a mathematical model. This chapter concludes with a short overview of existing solution approaches, mainly focusing on location-allocation approaches.

2.1 Components

At first, the following subsections will introduce the components of our general model.

2.1.1 Basic Areas

A Districting problem comprises a set BA of *basic areas* , sometimes also called *basic units*, (*sales*) *coverage areas*, or *geographical units*. A basic area is the smallest considered geographic area and it is represented by a point (e.g., geo-coded customer location), a (poly-)line (e.g., street) or a polygon (e.g., city or zip-code area). For purposes of a simple notation, this model assumes that each basic area $i \in BA$ can be represented by a point $b_i = (x_i, y_i)$. In the case of non-point objects this can be for example the center of gravity (polygons) or the middle-point (streets). In the following b_i denotes this representative point as well as its basic area.

Moreover, one or more quantifiable *activity measures*, or *activities* for short, are associated with each basic area. Let $w_i^a \in \mathbb{R}_+$ denote the a-th activity and A denote the number of activities. Depending on the application these activities can be the number of people or voters living within the basic area, the (total) sales potential of the people or customers within the basic area, or the time that is necessary to serve the (total) demand of them. In the (most common) case of one activity measure, i.e., $A = 1$, write w_i or $w(i)$ for short.
In general, for a subset $B \subseteq BA$ of basic areas its a-th activity measure is defined as sum of the a-th activity measures of its basic areas, i.e., $w^a(B) := \sum_{i \in B} w_i^a$. However, for certain applications it can be defined as $w^a(B) := \sum_{i \in B} w_i^a + W^a(B)$, with $W^a(B)$ being an additional value depending on the subset B, e.g., the travel time of a TSP-tour visiting all basic areas of B. For short, $w^a(B)$ is called the *size* or *activity* of B. Again, write $w(B)$ if $A = 1$.

2.1.2 Centers

Sometimes a *center* is associated to each district. This can either be a specified location, e.g., an office of a company or a location of a social institution, or a representative point, e.g., the center of gravity. The latter might be helpful for evaluating the district, for example in terms of compactness. The definition or location of a center for each district is often part of the planning process. In this case, the centers are located in a second step after generating the districts.
However, the set CE of centers can also be given in advance, for example by residences of salespersons or by locations of already existing schools. In this case, the model assumes that each center $h \in CE$ can be represented by a point $c_h = (x_h, y_h)$ as well.

Moreover, for certain applications capacities can be associated with each of these centers, for example the maximum number of students for a school. Let $cap_h^a \in \mathbb{R}_+$ denote the capacity of center $h \in CE$ according to the a-th activity.

2.1.3 Distances

The *distance* between two basic areas $i, j \in BA$ is denoted by $d_{i,j} := d(b_i, b_j)$. Depending on the application, $d(\cdot\,,\cdot)$ can be for example the Euclidean distance, or the distance or travel time on a road network. Note that a distance function defined on a road network is not necessarily a metric since the existence of one-way-streets can yield $d_{i,j} \neq d_{j,i}$. However, we assume that each used distance function satisfies the triangle inequality and the coincidence axiom.
Moreover, the distance between a basic area i and a set of basic areas $B \subseteq BA$ is defined as $d(i, B) := \min_{j \in B} d_{i,j}$ or $d(B, i) := \min_{j \in B} d_{j,i}$, respectively. For non-point representations, the distance between two basic areas is either defined as distance between their representative points or as their shortest surface-to-surface distance.
In the case of predefined centers, the distance between a basic area $i \in BA$ and a center $h \in CE$ is denoted by $d_{i,h} := d(b_i, c_h)$. The distance between a basic area i (center h) and a set of centers $C \subseteq CE$ (basic areas $B \subseteq BA$) is defined as: $d(i, C) := \min_{h \in C} d_{i,h}$ ($d(B, h) := \min_{i \in B} d_{i,h}$) or $d(C, i) := \min_{h \in C} d_{h,i}$ ($d(h, B) := \min_{i \in B} d_{h,i}$), respectively.

2.1.4 Districts

A *district* D_g consists of a set of basic areas $B_g \subseteq BA$. Sometimes a district is also called *territory* or *sector*. The district containing basic area i is denoted by $D_{(i)}$, i.e., $D_{(i)} = D_g$ if and only if $i \in B_g$. For short, one can say i is assigned or allocated to D_g.

Depending on the application a district D_g may contain a center $cen_g \in CE$ in addition, i.e., $D_g := (B_g, cen_g)$. In this case, $cen_{(i)}$ denotes the center of the district containing basic area i. Here, one can say i is assigned to $cen_{(i)}$.

Furthermore, the a-th activity of the set of assigned basic areas defines the a-th activity of the district, i.e., $w^a(D_g) := w^a(B_g)$.

2.1.5 Districting Plan

Finally, a *districting plan* consists of a set of districts $S := \{D_1; \ldots; D_p\}$ where p is the number of districts. In most applications p is given in advance, however, it can also be part of the planning process. Other terms for districting plan are *districting layout*, *territory plan*, *territory layout* or *solution* . This work uses the terms districting plan and solution interchangeable.

2.2 Planning Criteria

The districting problem can be described as follows: Partition all basic areas BA into p districts that are balanced, contiguous, and compact. This section describes these criteria in detail and introduces how a district or solution, respectively, is evaluated with respect to them.

2.2.1 Complete and Exclusive Assignment

In general, each basic area must be assigned to exactly one district, i.e., the sets $B_1, \ldots, B_p$ define a partition of the set BA:

$$B_1 \cup \cdots \cup B_p = BA \quad \text{and} \quad B_g \cap B_h = \emptyset, \quad 1 \leq g, h \leq p\,.$$

Sometimes, this criterion is called *integrity*.

2.2.2 Balance

A district D_g is called perfectly *balanced* in terms of the a-th activity measure if its size $w^a(D_g)$ is equal to the average district size $\mu^a := w^a(BA)/p$ according to this activity measure. Without loss of generality, in the following the case of only one activity measure is considered. Since perfectly balanced districts can usually not be achieved, a common way to measure the balance of D_g is to compute the relative percentage deviation of its size from the average size [29], that is

$$bal(D_g) := \frac{|w(D_g) - \mu|}{\mu}\,. \tag{2.1}$$

The larger this deviation the worse the balance. Note that if an additional value depending on the assigned basic areas is included in $w(D_g)$ the balance of a district D_g can be different in different solutions.
Another approach includes a prescribed relative threshold $\tau > 0$ and evaluates each deviation smaller than or equal to this threshold by $bal(D_g) = 0$, while each deviation exceeding this threshold is evaluated as before [6, 7], i.e.,

$$bal_\tau(D_g) := \frac{\max\{w(D_g) - (1+\tau)\cdot\mu; (1-\tau)\cdot\mu - w(D_g); 0\}}{\mu}\,.$$

This threshold can be given for example by law in the context of political districting or by working time restrictions in the context of sales districting.

The balance of a solution is mainly defined as the maximal balance of a single district [21], i.e.,

$$bal_{max}(S) := \max_{g=1,\dots,p} bal(D_g)\,. \tag{2.2}$$

This approach has the drawback that only the worst balanced district is considered and the further districts are not taken into account. Hence, it does not matter if they are perfectly balanced or nearly as worse balanced as the worst one. Therefore, another less common approach is the definition as sum of balances of all districts, for example used by Bozkaya et al. [7],

$$bal_{sum}(S) := \sum_{g=1}^{p} bal(D_g)$$

or as average of them

$$bal_{ave}(S) := \frac{\sum_{g=1}^{p} bal(D_g)}{p}\,. \tag{2.3}$$

In this case a few highly unbalanced districts could be compensated by some well balanced districts. Hence, we suggest combining both approaches using a convex combination of them [8], i.e.,

$$bal_{cc,\alpha}(S) := \alpha \cdot \frac{1}{p} \cdot \sum_{g=1}^{p} bal(D_g) \;+\; (1-\alpha) \cdot \max_{g=1,\dots,p} bal(D_g)\,,$$

where $\alpha \in [0,1]$.

2.2.3 Compactness

A district is *compact* if it is nearly round-shaped or square, undistorted, without holes, and has a smooth boundary. In the context of political districting the main motivation is to prevent gerrymandering. In many other applications such as sales districting or school districting compact districts help to reduce travel distances within the districts. Although, compactness seems to be a very intuitive concept no comprehensive definition exits. Main difficulties are the dependence on the geometrical representation of the basic areas and the consideration of all dimensions of compactness. Many authors have proposed compactness measures in literature, but unfortunately, all of them have some weaknesses. Therefore, the reader is referred to Chapter 3, where these measures are analyzed and developed.

2.2.4 Contiguity

Figuratively spoken, a district is *contiguous* if within a district the travelling from each basic area to every other basic area without leaving this district is possible. The motivation is similar to compactness: Preventing gerrymandering in the context of political districting or reducing travel distances within the districts. In the context of sales or services districting it also helps to obtain clearly defined areas of responsibility since no salesperson has to travel through another district, and thereby may passing customers of other salespersons.

If basic areas are represented by polygons or lines, neighborhood information is implicitly given. Two polygons are neighbored if they share a common border, while two lines are neighbored if they meet in a crossroad. Based on this neighborhood information a neighborhood graph can be derived, and a district is defined straightforwardly as contiguous if its basic areas induce a connected sub-graph of this graph. Thus, this criterion is often also called *connectedness*.

If basic areas are represented by points, there is no straightforward definition for contiguity, not even for neighboring basic areas. Hence, a surrogate how to ensure contiguity is necessary. One idea is the usage of a proximity graph. Then, a district can be defined as contiguous if its basic areas induce a connected sub-graph of this proximity graph. In the literature some different approaches, how to define a proximity graphs, are proposed.

2.2.4.1 Delaunay Triangulation

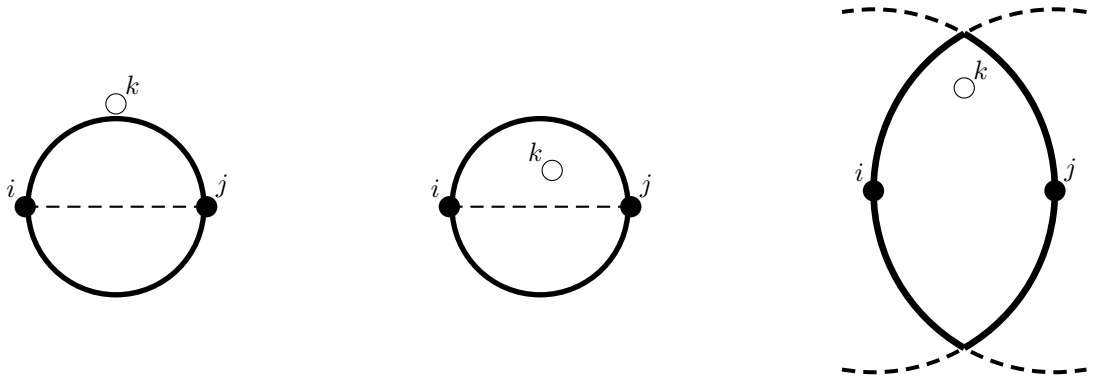
In the Delaunay Triangulation (DT) of BA two basic areas i and j are neighbored if and only if their Voronoi regions have a common border within an enclosing figure of BA. Figure 2.1a depicts an example, for more details about Voronoi regions see Section 5.1. The DT has the property that for each Delaunay triangle there is no further vertex located within its circumscribing circle. Figure 2.1b depicts the circumscribing circles for the example presented in Figure 2.1a. The DT can be computed in $\mathcal{O}(n \cdot \log n)$ [1].

2.2.4.2 Gabriel Graph

The so-called Gabriel Graph (GG) is an undirected proximity graph proposed by Gabriel and Sokal [14]. In the GG of BA two basic areas i and j are neighbored if and only if no other basic area k is located within the closed disc having the line segment between i and j as diameter. For example, i and j depicted in Figure 2.2a are neighbored, whereas i and j in Figure 2.2b are not neighbored since k is located within the illustrated closed disc. The GG is a sub-graph of the DT and it can be computed in $\mathcal{O}(n)$ if the DT is already given [26].

(a) Voronoi regions and DT (b) Circumscribing circles for the triangles

Figure 2.1: Illustrations of the Delaunay Triangulation



(a) GG: i and j are neighbored (b) GG: i and j are not neighbored (c) RNG: i and j are not neighbored

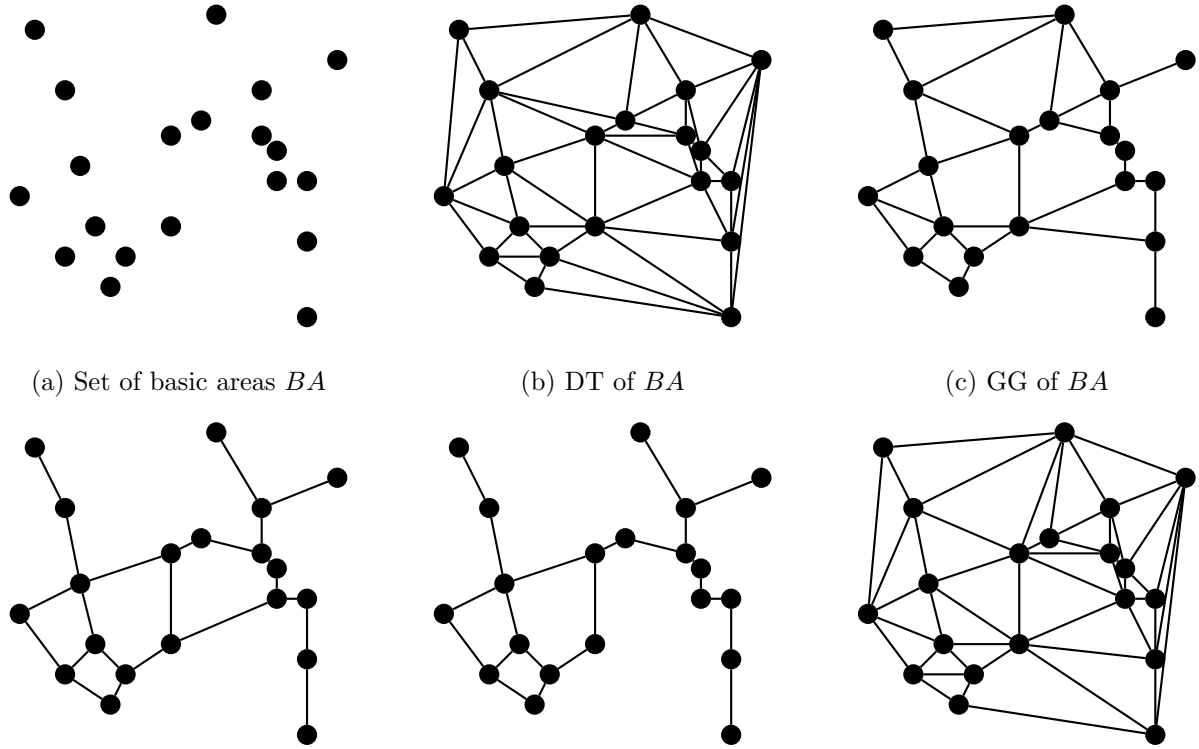
Figure 2.2: Illustrations for proximity graphs

2.2.4.3 Relative Neighborhood Graph

Another approach was proposed by Toussaint [36]. The Relative Neighborhood Graph (RNG) is an undirected proximity graph, where two basic areas i and j of BA are neighbored if and only if no other basic area k exists that is closer to both i and j than they are to each other. For example, in Figure 2.2c k is closer to i and j than they are to each other, so i and j are not neighbored. One can easily see that the RNG is a sub-graph of the GG since no pair of basic areas can be neighbored within the RNG but not within the GG. The RNG can also be computed $\mathcal{O}(n)$ if the DT is already given [24].

2.2.4.4 Urquhart Graph

The so-called Urquhart Graph (UG) is another undirected proximity graph introduced by Urquhart [37]. It is a sub-graph of the DT and obtained by removing the longest edge from each triangle. The computation can be done in $\mathcal{O}(n \cdot \log n)$ as well. The UG is not the same



(a) Set of basic areas BA (b) DT of BA (c) GG of BA

(d) RNG of BA (e) UG of BA (f) HG of BA

Figure 2.3: Proximity graphs

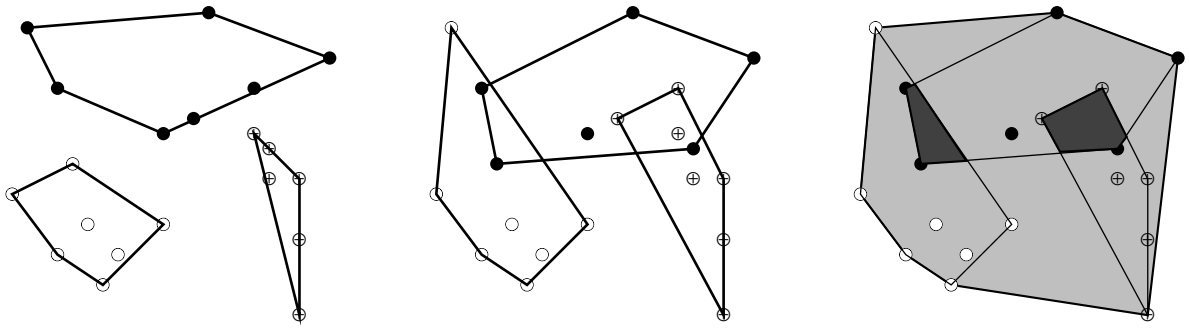
as the RNG, although Urquhart assumed it wrongly while proposing this approach.

2.2.4.5 Haugland Graph

The Haugland Graph (HG) is also an undirected graph and was introduced by Haugland et al. [17]. In contrast to the former approaches it is based on a complete graph. For each pair of intersecting edges in the planar representation of the complete graph the longer (more costly) edge is removed.

Figure 2.3 illustrates the presented proximity graphs for the set of basic areas introduced in Figure 2.3a. Each of these approaches results in a connected planar graph. These illustrations show that the GG depicted in Figure 2.3c, the RNG depicted in Figure 2.3d and the UG depicted in Figure 2.3e are sub-graphs of the DT illustrated in Figure 2.3b. Moreover, comparing Figures 2.3c and 2.3d one can see that the RNG is a sub-graph of the GG. Furthermore, the UG illustrated in Figure 2.3e is obviously not equivalent to the RNG presented in Figure 2.3d.

The RNG and the UG are very thin, i.e., many of their vertices have a very small degree.



(a) Contiguous solution S_1 (b) Non-contiguous solution S_2 (c) Areas of intersection in S_2

Figure 2.4: Illustrations of the contiguity measure

Hence, the number of feasible partitions is very restricted. In contrast to this, within the DT and the HG the degree of their vertices is rather high. Hence, some basic areas are declared as neighbored although they are far away from each other, what is quite counter-intuitive. Thus, the GG seems to be a good compromise since it restricts the number of feasible partitions not as strong as the RNG and the UG, and it contains less counter-intuitive neighboring pairs of basic areas than the DT and the HG.

2.2.4.6 Intersection of Convex Hulls

Another idea to define contiguity if the basic areas are represented by points is the usage of geometrical definitions. Kalcsics et al. [21] and Jarrah and Bard [20] call a district contiguous if the convex hull $ch(B_g)$ of the basic areas comprising district D_g does not intersect the convex hull of the basic areas of any other district D_h. Within a contiguous district for each pair of basic areas there exists a path that does not leave this district. Moreover, using Euclidean distances even the shortest path between them does not leave this district. Hence, no shortest path between two basic areas of one district passes a basic area of another district.
However, this definition is very restrictive. Depending on the application a planner might accept some (small) intersections between basic areas. Therefore, the contiguity measure we suggest determines the contiguity of a solution as sum of the areas of intersection between their convex hulls, normalized by the area of the convex hull of BA, that is

$$ctg(S) := \frac{\sum_{g=1}^{p-1} \sum_{h=g+1}^{p} area(ch(B_g) \cap ch(B_h))}{area(ch(BA))} .$$

Figure 2.4 illustrates this measure. The set of basic areas is partitioned into three districts, illustrated by white circles, black circles and circles filled by a plus. Solution S_1 presented in Figure 2.4a is contiguous according to the restrictive definition since there is no intersection between the convex hulls of the corresponding districts, i.e., $ctg(S_1) = 0$. In contrast to this, solution S_2 illustrated in Figure 2.4b is not contiguous since there are intersections between the corresponding convex hulls. Figure 2.4c shows the corresponding areas of intersection by the two dark gray polygons, whereas the convex hull of BA is illustrated as light gray polygon. In order to obtain $ctg(S_2)$, the total area of the dark gray polygons is set in relation to the area of the light gray polygon.

2.3 Mathematical Modelling

In the literature there are some approaches that model the districting problem as a mathematical program. The first formulation as mixed integer program was proposed by Hess et al. [19] in 1965. They model the districting problem as p-median warehouse location problem. The decision variable x_{ij} states whether basic area i is assigned to the district having basic area j as center or not. A basic area i is defined as center if and only if $x_{ii} = 1$ holds. Hence, there is a one-to-one relation between centers and districts. Recall that p is the given number of districts, μ denotes the average activity measure of a district and τ represents the feasible percentage deviation from μ. The model of Hess et al. [19] is given as follows:

$$\min \sum_{i \in BA} \sum_{j \in BA} w_i \cdot d_{i,j}^2 \cdot x_{ij} \tag{O1}$$

s.t.

$$\sum_{j \in BA} x_{ij} = 1 \qquad \forall\, i \in BA \tag{C2}$$

$$\sum_{i \in BA} x_{ii} = p \tag{C3}$$

$$(1 - \tau) \cdot \mu \cdot x_{jj} \leq \sum_{i \in BA} w_i \cdot x_{ij} \qquad \forall\, j \in BA \tag{C4a}$$

$$(1 + \tau) \cdot \mu \cdot x_{jj} \geq \sum_{i \in BA} w_i \cdot x_{ij} \qquad \forall\, j \in BA \tag{C4b}$$

$$x_{ij} \in \{0; 1\} \qquad \forall\, i, j \in BA \tag{C5}$$

The objective function (O1) minimizes the Weighted Moment of Inertia (cf. Section 3.3.5.1). Hence, the program treats compactness as optimization goal, whereas it treats balance as a hard criterion by defining a lower (C4a) and an upper bound (C4b) for the activity of a district. Moreover, constraints (C4b) guarantee that if $x_{jj} = 0$ holds, x_{ij} equals zero for each basic area j. In other words, if basic area i is assigned to the district having basic area j as center, basic area j has to be defined as center. Constraints (C2) together with the domain restrictions (C5) ensure that each basic area is completely and exclusively assigned to one district. Finally, constraint (C3) guarantees that exactly p basic areas are defined as centers, and, hence, that exactly p districts exist. Surprisingly, the model contains no contiguity constraints.

However, this program is not practical applicable for larger problems. Already a small instance, having 231 basic areas that should be partitioned into 5 districts, is not solvable in reasonable time. CPlex 12.6 shows still a gap of 15.93% after 12 hours on a PC running

Windows 7 with a Pentium $i7-4500U$ processor with 2.80 GHz and 8 GB RAM. Therefore, in order to solve the problem, Hess et al. [19] propose a location-allocation method.

Since Hess et al. [19] do not model contiguity explicitly, some authors have proposed approaches that add contiguity constraints to the original model. These approaches are mainly based on a given neighborhood relation, where N^i denotes the set of basic areas neighbored to basic area i.
One way to model contiguity introduced by Drexl and Haase [12] is the following:

$$\sum_{i \in \bigcup\limits_{k \in B} (N^k \backslash B)} x_{ij} - \sum_{i \in B} x_{ij} \geq 1 - |B| \qquad \forall\, j \in BA, B \subset [BA \backslash (N^j \cup \{j\})] \tag{C6}$$

For each center j, the constraints (C6) consider each subset of basic areas B not containing basic area j and the neighbors of j. If all basic areas of B are assigned to center j, there must be at least one basic area not included in B but neighbored to B that is also assigned to center j. Unfortunately, since each subset has to be taken into account, there is an exponential number of these contiguity constraints. Thus, for example Ríos-Mercado and López-Pérez [30] and Salazar-Aguilar et al. [31] apply a cut generation approach that adds the needed constraints iteratively.

Shirabe [35] presents an approach based on network flows. For each district, each basic area except the center is a source having a supply of one, whereas the center is a sink having the total demand. Figuratively spoken, within a district each basic area sends one unit to the center. Let f_{ikj} the (non-negative) flow from basic area i to basic area k within the district having j as center. Shirabe [35] models the contiguity as follows:

$$\sum_{k \in N(i)} f_{ikj} - \sum_{k \in N(i)} f_{kij} = x_{ij} \qquad \forall\, j \in BA, i \in BA \backslash \{j\} \tag{C7}$$

$$\sum_{k \in N(i)} f_{kij} \leq \left[\left(\sum_{l \in BA} x_{lj} \right) - 2 \right] \cdot x_{ij} \qquad \forall\, j \in BA, i \in BA \backslash \{j\} \tag{C8}$$

$$\sum_{k \in N(j)} f_{kjj} \leq \left[\left(\sum_{l \in BA} x_{lj} \right) - 1 \right] \cdot x_{jj} \qquad \forall\, j \in BA \tag{C9}$$

Constraints (C7) ensure that each basic area has a supply of one within the corresponding district if it is assigned to center j and no supply or demand otherwise. Constraints (C8) guarantee that the corresponding flow into a basic area not assigned to j is zero. Moreover, if j is no center, there is no flow according to the district having j as center at all. Finally, according to constraints (C8) the flow into a center corresponds to the number of assigned

basic areas except the center itself.

There are some variations of the original model of Hess et al. [19] in the literature. Especially the compactness measure used as objective function varies between different proposals. Ríos-Mercado and Fernández [29] use the maximum distance between a center and one of its assigned basic areas as objective function, i.e.,

$$min \max_{i,j \in BA} d_{i,j} \cdot x_{ij} \tag{O2}$$

and Salazar-Aguilar et al. [31] propose the sum of (single) distances from the basic areas to the corresponding centers, i.e.,

$$min \sum_{i \in BA} \sum_{j \in BA} d_{i,j} \cdot x_{ij} \tag{O3}$$

to give two examples.
Moreover, the integration of balance constraints for more than one activity measure is straightforward [29]:

$$(1 - \tau^a) \cdot \mu^a \cdot x_{jj} \leq \sum_{i \in BA} w_i^a \cdot x_{ij} \qquad \forall\, j \in BA, a \in A \tag{C9a}$$

$$(1 + \tau^a) \cdot \mu^a \cdot x_{jj} \geq \sum_{i \in BA} w_i^a \cdot x_{ij} \qquad \forall\, j \in BA, a \in A \tag{C9b}$$

Salazar-Aguilar et al. [32] state the districting problem as bi-objective programming model, where minimizing balance is the second objective beside compactness:

$$\min W \tag{O3}$$

s.t.

$$(\sum_{i \in BA} w_i \cdot x_{ij}) - \mu \cdot x_{jj} \leq W \qquad \forall\, j \in BA \tag{C10a}$$

$$\mu \cdot x_{jj} - (\sum_{i \in BA} w_i \cdot x_{ij}) \leq W \qquad \forall\, j \in BA \tag{C10a}$$

Constraints (C10a) and (C10a) together with the objective function O3 describe the balance measure defined in Equations (2.1) and (2.2). This formulation is necessary in order to replace the absolute value in Equation (2.1). The authors apply a ϵ-constraint method where compactness is used as the primary objective and solve instances up to 150 basic areas in reasonable time. Unfortunately, instances having 150 basic areas are rather small instances and the approach is still not applicable for larger problems.

A quadratic formulation is introduced by Salazar-Aguilar et al. [31]. This model needs two sets of decision variables z_{ig} and y_{jg}. The former represents the assignment of basic area i to district D_g, whereas the latter defines whether basic area j is the center of district D_g or not. Hence, the number of decision variables is $2 \cdot p \cdot |BA|$ instead of $|BA|^2$ for the linear formulation.

$$\min \sum_{g=1}^{p} \sum_{i \in BA} \sum_{j \in BA} d_{i,j} \cdot z_{ig} \cdot y_{jg} \tag{O4}$$

s.t.

$$\sum_{j \in BA} y_{jg} = 1 \qquad \forall\, g = 1, \ldots, p \tag{C11}$$

$$\sum_{g=1}^{p} z_{ig} = 1 \qquad \forall\, i \in BA \tag{C12}$$

$$z_{jg} \geq y_{jg} \qquad \forall\, j \in BA, g = 1, \ldots, p \tag{C13}$$

$$(1 - \tau^a) \cdot \mu^a \leq \sum_{i \in BA} w_i^a \cdot z_{ig} \qquad \forall\, g = 1, \ldots, p, a \in A \tag{C14a}$$

$$(1 + \tau^a) \cdot \mu^a \geq \sum_{i \in BA} w_i^a \cdot z_{ig} \qquad \forall\, g = 1, \ldots, p, a \in A \tag{C14b}$$

$$\sum_{g=1}^{p} \sum_{i \in \bigcup\limits_{k \in B} (N^k \setminus B)} z_{ig} \cdot y_{jg} - \sum_{g=1}^{p} \sum_{i \in B} z_{ig} \cdot y_{jg} \geq 1 - |B| \qquad \forall\, j \in BA, B \subset [BA \setminus (N^j \cup \{j\})] \tag{C15}$$

$$z_{ig} \in \{0; 1\} \qquad \forall\, i \in BA, g = 1, \ldots, p \tag{C16a}$$

$$y_{jg} \in \{0; 1\} \qquad \forall\, j \in BA, g = 1, \ldots, p \tag{C16b}$$

Constraints (C11) ensure that each district has a center. Naturally, if a basic area is defined as district center it has to be assigned to the district (C13). The constraints (C12), (C14a), (C14b) and (C15) correspond to the constraints (C2), (C9a), (C9a) and (C6). Finally, (C16a) and (C16a) are domain restrictions.

The authors solve the problem by an iterative procedure using branch and bound and cut generations. However, in order to solve the quaratic problem in reasonable time, they use a local optimum method.

2.4 Heuristic Solution Approaches

The previous section has shown that the mathematical program formulations are not solvable in reasonable time by exact approaches. Hence, typically heuristics are applied to districting problems.

2.4.1 Location-Allocation

Already Hess et al. [19] have proposed a location-allocation heuristic that splits the problem into two independent problems. The location problem is the problem of determining a set of p centers, whereas the allocation problem is the problem of assigning the basic areas to these centers. Both problems are solved alternatingly until there is no further noticeable improvement. One way to solve the location problem is to determine the center of gravity for each district of the previous allocation phase. For the allocation problem the number of decision variables is only $p \cdot |BA|$ since the centers are prescribed. Recall, that the districting problem needs $|BA|^2$ decision variables. Now, CPlex needs less than one second to solve the described instance having 231 basic areas if the 5 centers are prescribed, while the districting problem is not solvable to optimality within 12 hours. Even for large instances, this problem is solvable in reasonable time.
The location-allocation procedure was applied and enhanced by several authors over the years. For example, Hess and Samuels [18] and Fleischmann and Paraschis [13] solve a relaxed problem in the allocation phase where τ is set to zero and x_{ij} is relaxed, i.e., $x_{ij} \in [0, 1]$. Hence, the solution may contain so-called splits, i.e., basic areas which are assigned partly to different centers. Thus, in order to resolve these splits a subsequent step is necessary.
Moreover, Ríos-Mercado and López-Pérez [30] integrate contiguity constraints and incorporate the similarity to an existent plan. In recent approaches, López et al. [25] apply a location-allocation procedure in the context of territory planning for micro financing institutions, and Yanık et al. [38] to determine sustainable energy regions.

2.4.2 Further Approaches

Over the years many further heuristics have been proposed. For example, Garfinkel and Nemhauser [16] present a set-partitioning approach. Firstly, this approach determines a set of feasible districts. After that, it chooses a subset of these districts in order to obtain a good overall solution.

Seed-growing approaches choose some basic areas as seeds and assign the further basic areas to these seeds taking the required planning criteria into account, i.e., each seed leads to a district. The districts are either treated sequentially or simultaneously. In the context of

districting, many authors propose such approaches, for example Bodin and Levy [5], Bozkaya et al. [6, 7], and Lei et al. [22, 23]. Mainly, these authors use seed-growing approaches in order to generate initial solutions for meta-heuristics.

A broad range of meta-heuristics haven been proposed in the context of districting. Meta-heuristics have in common that they are very flexible for integrating different requirements and planning criteria. D'Amico et al. [10] apply a simulated annealing approach in order to design police districts, Bergey et al. [2] solve an electrical power districting problem by means of simulated annealing and Ricca and Simeone [27] political districting problems. In related works Ríos-Mercado and Fernández [29], Salazar-Aguilar et al. [34], and Ríos-Mercado and Escalante [28] present different variations of GRASP approaches in order to determine sales districts for a beverage company, while de Assis et al. [11] determine districts for meter reading. In the context of the beverage company, Salazar-Aguilar et al. [33] use a scatter search approach. For example, Bozkaya et al. [6, 7] and Ricca and Simeone [27] propose tabu search procedures in the context of political districting, while Blais et al. [4] apply a tabu search approach on a home-care districting problem. Lei et al. [22, 23] apply kinds of (adaptive) large neighborhood search procedures. In different districting contexts, for example Bergey et al. [2, 3] and Chou [9] present evolutionary or genetic algorithms.

Another class of solution approaches are geometric approaches, for example proposed by Kalcsics et al. [21], Galvão et al. [15], or Ricca and Simeone [27]. These approaches utilize the districting problem's underlying geometrical information.

This thesis focuses on geometric approaches in the context of point representations of basic areas. Chapter 4 continues the work of Kalcsics et al. [21]. Chapter 5 proposes an approach based on Power Diagrams. In the context of line representations Chapter 6 introduces an algorithm based on tabu search and adaptive randomized neighborhood search.

Bibliography

[1] F. Aurenhammer, R. Klein, and D. L. Lee. *Voronoi Diagrams and Delaunay Triangulations*. World Scientific, 2013. ISBN 978-9814447638.

[2] P. K. Bergey, C. T. Ragsdale, and M. Hoskote. A Simulated Annealing Genetic Algorithm for the Electrical Power Districting Problem. *Annals of Operations Research*, 121 (1):33–55, 2003.

[3] P. K. Bergey, C. T. Ragsdale, and M. Hoskote. A decision support system for the electrical power districting problem. *Decisison Support Systsems*, 36:1–17, September 2003.

[4] M. Blais, S. D. Lapierre, and G. Laporte. Solving a home-care districting problem in an urban setting. *Journal of the Operational Research Society*, 54(11):1141–1147, 2003.

[5] L. D. Bodin and L. Levy. The arc partitioning problem. *European Journal of Operational Research*, 53(3):393–401, 1991.

[6] B. Bozkaya, E. Erkut, and G. Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1): 12–26, 2003.

[7] B. Bozkaya, E. Erkut, D. Haight, and G. Laporte. Designing new electoral districts for the city of Edmonton. *Interfaces*, 41(6):534–547, 2011.

[8] A. Butsch, J. Kalcsics, and G. Laporte. Districting for arc routing. *INFORMS Journal on Computing*, 26(4):809–824, 2014.

[9] C.-I. Chou. A Knowledge-based Evolution Algorithm approach to political districting problem. *Computer Physics Communications*, 182(1):209–212, 2011.

[10] S. J. D'Amico, S.-J. Wang, R. Batta, and C. M. Rump. A simulated annealing approach to police district design. *Computers & Operations Research*, 29(6):667–684, 2002.

[11] L. S. de Assis, P. M. Franca, and F. L. Usberti. A redistricting problem applied to meter reading in power distribution networks. *Computers & Operations Research*, 41 (1):65–75, 2014.

[12] A. Drexl and K. Haase. Fast Approximation Methods for Sales Force Deployment. *Management Science*, 45(10):1307–1323, 1999.

[13] B. Fleischmann and J. N. Paraschis. Solving a large scale districting problem: a case report. *Computers & Operations Research*, 15(6):521–533, 1988.

[14] K. R. Gabriel and R. R. Sokal. A New Statistical Approach to Geographic Variation Analysis. *Systematic Zoology*, 18(3):259–278, 1969.

[15] L. C. Galvão, A. G. N. Novaes, J. E. Souza de Cursi, and J. C. Souza. A multiplicatively-weighted Voronoi diagram approach to logistics districting. *Computers & Operations Research*, 33:93–114, 2006.

[16] R. S. Garfinkel and G. L. Nemhauser. Optimal Political Districting by Implicit Enumeration Techniques. *Management Science*, 16(8):495–508, 1970.

[17] D. Haugland, S. C. Ho, and G. Laporte. Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3):997–1010, 2007.

[18] S. W. Hess and S. A. Samuels. Experiences with a Sales Districting Model: Criteria and Implementation. *Management Science*, 18(4-part-ii):41–54, 1971.

[19] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. Nonpartisan Political Redistricting by Computer. *Operations Research*, 13(6):998–1006, 1965.

[20] A. I. Jarrah and J. F. Bard. Large-scale pickup and delivery work area design. *Computers & Operations Research*, 39(12):3102–3118, 2012.

[21] J. Kalcsics, S. Nickel, and M. Schröder. Towards a Unified Territorial Design Approach – Applications, Algorithms and GIS Integration. *TOP*, 13(1):1–74, 2005.

[22] H. Lei, G. Laporte, and B. Guo. Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1–2):67–85, 2012.

[23] H. Lei, G. Laporte, Y. Liu, and T. Zhang. Dynamic design of sales territories. *Computers & Operations Research*, 56:84–92, 2015.

[24] A. Lingas. A linear-time construction of the relative neighborhood graph from the Delaunay triangulation. *Computational Geometry*, 4(4):199–208, 1994.

[25] F. López, T. Ekin, F. A. M. Mediavilla, and J. A. Jimenez. Hybrid Heuristic for Dynamic Location-Allocation on Micro-Credit Territory Design. *Computacion y Sistemas*, 19(4): 783–804, 2015.

[26] D. W. Matula and R. R. Sokal. Properties of Gabriel Graphs Relevant to Geographic Variation Research and the Clustering of Points in the Plane. *Geographical Analysis*, 12(3):205–222, 1980.

[27] F. Ricca and B. Simeone. Local search algorithms for political districting. *European Journal of Operational Research*, 189(3):1409–1426, 2008.

[28] R. Z. Ríos-Mercado and H. J. Escalante. GRASP with path relinking for commercial districting. *Expert Systems with Applications*, 44:102–113, 2016.

[29] R. Z. Ríos-Mercado and E. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776, 2009.

[30] R. Z. Ríos-Mercado and J. F. López-Pérez. Commercial territory design planning with realignment and disjoint assignment requirements. *Omega (United Kingdom)*, 41(3): 525–535, 2012.

[31] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and M. Cabrera-Ríos. New Models for Commercial Territory Design. *Networks and Spatial Economics*, 11(3):487–507, 2011.

[32] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and J. L. González-Velarde. A bi-objective programming model for designing compact and balanced territories in commercial districting. *Transportation Research Part C: Emerging Technologies*, 19(5):885–895, 2011.

[33] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, J. L. González-Velarde, and J. Molina. Multiobjective scatter search for a commercial territory design problem. *Annals of Operations Research*, 199(1):343–360, 2012.

[34] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and J. González-Velarde. GRASP strategies for a bi-objective commercial territory design problem. *Journal of Heuristics*, 19(2): 179–200, 2013.

[35] T. Shirabe. A Model of Contiguity for Spatial Unit Allocation. *Geographical Analysis*, 37(1):2–16, 2005.

[36] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.

[37] R. B. Urquhart. Algorithms for computation of relative neighbourhood graph. *Electronics Letters*, 16(14):556–557, 1980.

[38] S. Yanık, Ö. Sürer, and B. Öztayşi. Designing sustainable energy regions using genetic algorithms and location-allocation approach. *Energy*, 97:161–172, 2016.

3 A Current Review on Compactness

Contents

Compactness is an important criterion in the context of districting. Nearly every approach presented in Chapter 1.1 take compactness into account as a planning criterion for different reasons. In the context of political districting the main motivation is the prevention of gerrymandering. In many other applications such as sales districting or school districting compact districts reduce the travel distances within the districts. Although compactness seems to be a very intuitive concept a rigorous definition does not exist. But, why is it so hard or even impossible to define a comprehensive compactness measure? First of all, it is very hard to take all dimensions of compactness into account. Moreover, the definition depends on the geometric representation of the basic areas. Finally, it is often subjective to decide whether a district is more compact than another one or not.

This chapter presents a current review on compactness measures. It starts with a definition of compactness, followed by an analysis of requirements on compactness measures. Then, Section 3.3 presents and discusses a couple of existing measures. Afterwards, Section 3.4 evaluates these measures to their correlation with a visual test. Since most of the proposed measures are based on polygonal representations, Section 3.5 extends these measures in order to make them applicable to further representations of basic areas. The chapter concludes with a short summary.

3.1 Definition

Before discussing requirements on compactness measures as well as proposed measures in detail, this section states some compactness definitions given in the literature:

Young [37] cites 'Webster's Third New International Dictionary' from 1961. It says that

"*a compact figure is homogenous and located within a limited definite space without straggling or rambling over a wide area*".

Niemi et al. [29] cite 'The American Heritage Dictionary'. It defines a figure as compact

"*if it is packed into a relatively small space and if its parts are closely packed together*".

A current online dictionary, 'TheFreeDictionary.com' [35], defines compactness as

"*closely and firmly united or packed together and occupying little space compared with others of its type*".

Bringing these definitions and our intuition together, we conclude that

"*a district is compact if it is nearly round-shaped or square, undistorted, without holes, and has a smooth boundary*".

3.2 Requirements

After defining compactness, this section summarizes and annotates some properties and requirements on compactness measures proposed in the districting literature.

3.2.1 Use of Local and Global Compactness

First of all, *local compactness* and *global compactness* can be distinguished. If a compactness measure is applied to a single district, the literature speaks of local compactness. In contrast to this, if the measure is applied to a districting plan as a whole, the literature speaks of global compactness.

Exclusively considering global compactness may allow some non-compact districts. For example, minimizing the total length of districts' boundaries allows some small non-compact urban districts as long as the large rural districts are compact. See Section 3.3.3.1 for more details. Exclusively considering local compactness can fail due to the fact that the reason for a non-compact district can be an irregular boundary of the overall area. Young [37] proposes that a compactness measure should apply for local compactness as well as for global compactness. However, Horn et al. [21] contradict this conclusion. They remark that one can obtain an evaluation for a districting plan by combining the evaluations of the single districts, for example, by using the average or the minimum of them. But, they agree that the other way around is not possible.

We also think that using global compactness exclusively is not suitable. However, it can be useful to apply a global measure combined with other local measures. In our opinion, combining the evaluations of the single districts is reasonable. However, in order to prevent that a few non-compact districts are compensated by some compact districts, we suggest that the worst evaluated district should be considered as part of the evaluation function in any case.

3.2.2 Use of Multiple Measures

It is very hard or even impossible to define a measure that takes all required dimensions of compactness into account. For example, a comprehensive compactness measure should incorporate the dispersion of a district as well as its perimeter. Moreover, the achieved results should be correlated with the visual impression whether this district is compact or not. As described later, each measure published so far has some drawbacks. Thus, Niemi et al. [29] advise that multiple measures should be used whenever possible.

We agree that for comparing different solutions the usage of different measures is reasonable. However, these measures should cover various dimensions of compactness. Mainly in the context of political districting, this helps preventing gerrymandering since it is very difficult to generate a manipulated districting plan that is not detected by at least one of the proposed measures. Nevertheless, for designing a simple and efficient heuristic it may make sense that the heuristic is restricted to use of only one measure.

3.2.3 Compare Plans

The best possible evaluation of a district or a solution, respectively, depends on the given data set. For example, close to the boundary of the regarded overall area it can be hard or even impossible to achieve a visually compact district. Thus, compactness should be used to compare different solutions, but no single threshold should be used that defines whether a district or a solution is compact or not. This is concluded by Young [37] as well as by Niemi et al. [29]. Nevertheless, Horn et al. [21] see a justification for using a threshold in order to prevent manipulations in the context of political districting. However, they remark that in this case it must be guaranteed that a district does not fail at predefined shapes of basic areas.

We agree with Young and Niemi et al.: An evaluation value is only an indicator and the definition of a threshold is actually impossible since the transition from non-compact to compact is fuzzy. Furthermore, the best reachable evaluation depends on the given data set. Hence, in order to obtain a meaningful compactness evaluation it is necessary to compare a result to other competitive solutions.

3.2.4 Evaluation Between 0 and 1

Niemi et al. [29] propose that evaluation values should vary between 0 and 1, with 1 being most compact. They assume that this property simplifies the interpretation of compactness evaluations.

We agree in principle that it is easier to get an indication whether a district or a solution, respectively, is compact or not if the results are in a prescribed limited range. Nevertheless, one must have in mind the problem described above that a determined evaluation has to be seen in relation to its competitive solutions.

3.2.5 Ignore Shapes of Basic Areas

The boundary and shape of a district depends on the shapes of the basic areas located on the border to neighboring districts. Since these shapes are predefined, it is (nearly) impossible to achieve visually compact districts in some cases. Hence, Young [37] suggests that these shapes should be irrelevant. In contrast to this, Horn et al. [21] assume that these shapes will not affect the ranking anyway if different districting plans are compared.

In our opinion this is only true for the boundary of the overall area. If there is an irregular boundary between two neighboring basic areas assigned to the same district, there is at most only a marginal effect on the compactness evaluation. On the other hand, if these basic areas are assigned to different districts, this boundary is part of the boundaries of both districts, and, hence, the effect is significantly higher. Thus, a districting plan that assigns all basic areas sharing an irregular boundary to the same district is most likely evaluated more compact than one that assigns them to different districts. This result can be quite desired since it correlates with the visual impression. Nevertheless, we agree with Horn et al. that predefined irregular boundaries can result in problems if a fixed threshold defines whether a district is compact or not. In order to overcome this problem, Horn et al. propose to smooth the boundaries of the basic areas. Section 3.3.2.1 will present an approach how this can be done.

3.2.6 Do not Discriminate Rural or Urban Areas

In some real-world instances the regarded overall area contains rural areas as well as urban areas. Usually, urban areas have a higher concentration of people, voters, customers, or students than rural areas. Hence, in a districting plan urban districts are typically noticeably smaller than rural districts. However, Young [37] proposes that a compactness measure should neither prefer nor discriminate urban areas against rural areas. He concludes that a measure should take the shape into account, but not the size of a district. In other words, a measure should be independent of scale. Niemi et al. [29] support this conclusion.

We want to regard this point a bit more differentiated. In the context of political districting a measure that is not independent of scale would either prefer a solution with nearly equally sized districts containing urban and/or rural areas, or a partition into some large rural and few small urban districts. If this decision should not be affected by the applied measure, this measure should fulfill this requirement. However, in the context of sales districting the total travel time of a salesperson within a district depends on the size of this district. Hence, in this context compactness is a kind of proxy for travel times and consequently it might be useful to abstain from this requirement.

3.2.7 Use of Verifiable Data

Young [37] suggests that a compactness measure should be simple and require only data that can be collected and verified easily. Horn et al. [21] complement that a measure should be easy to understand.

We agree with Horn et al. since planners and decision-makers will more likely trust an understandable measure. Furthermore, there is higher transparency if anyone can reproduce the evaluation of a districting plan. For example, in the context of political districting this can result in a higher acceptance of a current districting plan. In terms of required data we agree that they should be easily collectable and verifiable. Fortunately, today the availability of data is noticeably better than in 1988 when Young published his work. For example, today, shapes of cities or distances between different locations can be derived comparatively easily by using geographic information systems. Moreover, a lot of statistical data such as population distributions are freely available on the Internet. Furthermore, current computers can do complex calculations in a fraction of a second. Today's situation is not comparable to that in 1988. Calculations and used data can be noticeably more complex today, but for proposing or applying a measure one should still have in mind the comprehensibility.

3.3 Proposed Measures

So far this chapter has defined compactness and has pointed out requirements on compactness. This section presents and discusses a couple of proposed measures. Many of these measures that have been published in the 60s, 70s and 80s of the last century were set in the context of political districting. Sometimes they are based on ideas published much earlier. It also happened that similar or even identical approaches were published by different authors independently of each other. Especially in the 80s and 90s, some authors have summarized and categorized compactness measures published so far. In 1985, Maceachren [27] described 15 measures and categorized them into four categories:

1. Perimeter-area measures
2. Parameters of related circles
3. Direct comparison to standard shape
4. Dispersion of elements of area

Independently of this work, Young [37] published an overview over eight compactness measures in 1988. In 1990, Niemi et al. [29] continued this work by describing 24 measures and categorizing them into four categories:

1. Dispersion measures
2. Perimeter measures
3. Population measures
4. Other measures

Finally, in 1993 Horn et al. [21] revised this work. Their overview comprises 32 measures categorized in only two main categories, differentiating whether the population of the basic areas and districts is incorporated or not. However, they have defined some sub-categories:

1. Shape-population measures
    1.1. District population compared with population of compact figures
    1.2. Other population measures
2. Shape-only measures
    2.1. Area-only measures
    2.2. Perimeter-only measures
    2.3. Dispersion measures
    2.4. Angular measures
    2.5. Area-perimeter-quotients
    2.6. Relative Moment of Inertia

To the best of our knowledge, no comprehensive overview article has been published since 1993. Hence, in the following this section will summarize previous works and newer approaches. It uses the following notation

- $area(F)$: area of figure F
- $per(F)$: perimeter of figure F
- $radius(C)$: radius of circle C
- $sec(F)$: smallest enclosing circle of figure F
- $cla(F)$: circle having a diameter equal to the longest axis of figure F
- $lic(F)$: largest inscribed circle of figure F
- $serh(F)$: smallest enclosing regular hexagon of figure F
- $ch(F)$: convex hull of figure F
- $er(F)$: enclosing rectangle of figure F
- $le(R)$: length of rectangle R
- $wi(R)$: width of rectangle R

Moreover, for purposes of simplification, D_g denotes a district as well as the shape of this district. If the basic areas are represented by polygons, the shape of a district is straightforwardly defined as union of the corresponding polygons. Otherwise, it is more difficult to define a district's shape. Some approaches for this will be presented in Section 3.5.3.

The remainder of this section presents an overview of existing compactness measures, the classification is inspired by the (sub)-categorizes proposed by Horn et al. [21].

3.3.1 Shape-Only-Dispersion Measures

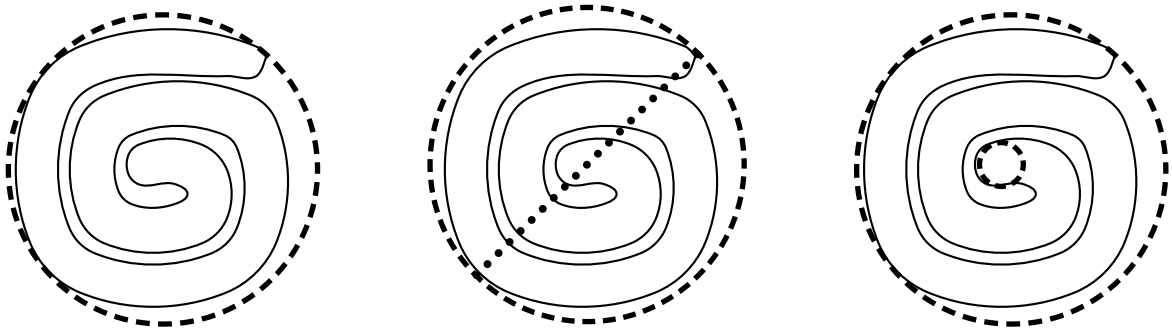
A measure categorized as *shape-only-dispersion measure* mainly takes the dispersion of a district into account. More precisely it focuses on the dispersion of the outer boundary, whereas the dispersion within the district as well as the perimeter length are of minor importance or not taken into account.

3.3.1.1 Reock-Test

The so-called *Reock-Test* is a local compactness measure proposed by Reock [31]. It calculates the ratio of the area of a district and the area of its smallest enclosing circle, i.e.,

$$comp_{reock}(D_g) := \frac{area(D_g)}{area(sec(D_g))}\,. \tag{3.1}$$

Obviously, this ratio is always between 0 and 1. The best possible evaluation of 1 achieves a circle since in this case both areas are equal. Moreover, the Reock-Test is independent of



(a) Reock-Test (b) Gibbs-Test (c) Haggett-Test

Figure 3.1: Different measures for a meandering district looking like a snake

scale. The area of a district is easy to obtain if the basic areas are represented by polygons. Otherwise, an approximation of the shape is necessary, a situation which will be discussed later in more detail.

The Reock-Test considers the smallest enclosing circle and the area of a district, but not the dispersion within this circle. Hence, a non-compact district that nearly fills out the enclosing circle is evaluated favorably. Figure 3.1a illustrates an example: Here, the district's shape looks like a snake. Obviously, this shape is visually non-compact. However, it is evaluated favorably in terms of the Reock-Test since it almost has the same area as its enclosing circle. Nevertheless, Horn et al. [21] conclude that the Reock-Test is better in practice than it seems to be in theory.

In addition, Niemi et al. [29] mention two variants of the Reock-Test:

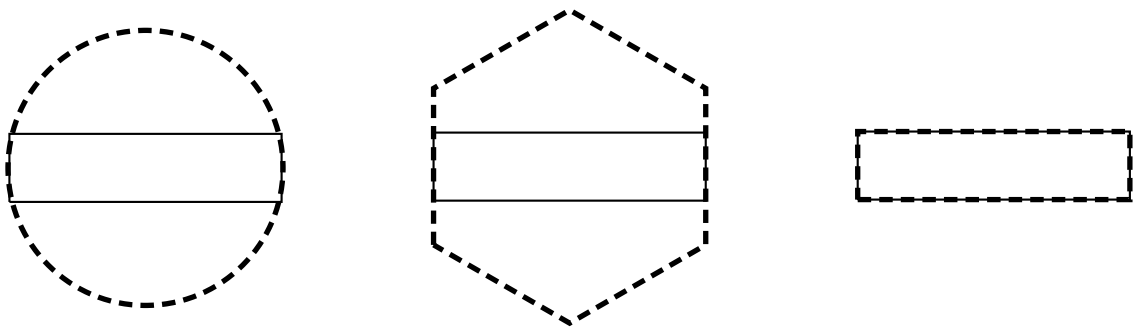
- The first variant uses the smallest enclosing regular hexagon instead of the smallest enclosing circle, i.e.,

$$comp_{reock-hex}(D_g) := \frac{area(D_g)}{area(serh(D_g))}\,.$$

  The idea behind this measure is the fact that it is possible to divide a plane completely into equally sized regular hexagons, whereas it is not possible to do this with circles.

- The second variant uses the smallest enclosing convex figure instead of the smallest enclosing circle. Note that this figure is the convex hull of the polygon points defining the district. This implies

$$comp_{reock-convex}(D_g) := \frac{area(D_g)}{area(ch(D_g))}\,. \tag{3.2}$$



(a) Reock-Test

(b) Variation of the Reock-Test using the smallest enclosing regular hexagon

(c) Variation of the Reock-Test using the convex hull

Figure 3.2: Variations of the Reock-Test applied to an long-shaped rectangular district

This approach has the following drawback: A convex figure is not necessarily visually compact, for example, a long rectangle such as the one illustrated in Figure 3.2 is rather non-compact. However, this test evaluates it with the highest score of 1 since its shape corresponds to its convex hull, as Figure 3.2c shows. The original Reock-Test (variation described above) evaluates this district as bad since the area of this rectangle is considerably smaller than the area of its enclosing circle (hexagon), as Figure 3.2a (3.2b) shows.

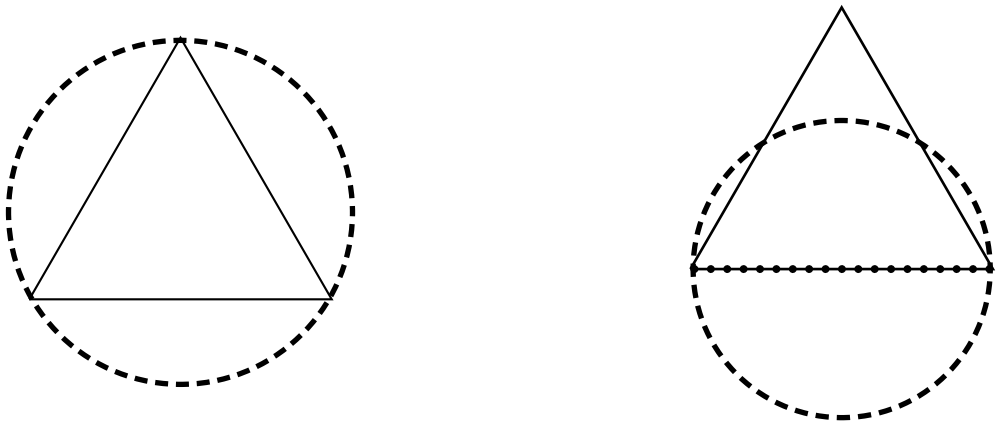
3.3.1.2 Gibbs-Test

Another approach to measure local compactness is the so-called *Gibbs-Test* described by Gibbs [15]. However, Niemi et al. [29] mention that Horton [22] also described this approach before. It determines the ratio of the area of the district and the area of a circle defined by having a diameter equal to the longest axis of the district. It is feasible that this axis leaves the district, i.e.,

$$comp_{gibbs}(D_g) = \frac{area(D_g)}{area(cla(D_g))}\,.$$

Hence, for shapes described by polygons determining the length of this axis is equivalent to determining the largest distance between two polygon points. Again, the results are always between 0 and 1, the best shape according to this measure is a circle, and this test is independent of scale.

The evaluation of an equilateral triangle points out the difference between the Reock-Test and the Gibbs-Test. Figure 3.3a depicts its smallest enclosing circle. Each edge is the longest axis of this triangle. Hence, one possible circle having a diameter equal to the longest axis



(a) Reock-Test (b) Gibbs-Test

Figure 3.3: Illustration of the difference between the Reock-Test and the Gibbs-Test

is the one illustrated in Figure 3.3b. This figure also shows that the achieved circle is not necessarily an enclosing circle. Nevertheless, the area of the district is always smaller than or equal to the area of this circle.

The Gibbs-Test fails at the same example as the Reock-Test: The snake-shaped district shown in Figure 3.1b is evaluated very favorably. Nevertheless, Horn et al. [21] conclude that the Gibbs-Test performs better in practice than expected by theoretical results.

3.3.1.3 Haggett-Test

The *Haggett-Test* is a local compactness measure introduced by Haggett [18]. It computes the ratio of the radii of the largest inscribed circle and the smallest enclosing circle of a district, i.e.,

$$comp_{haggett}(D_g) := \frac{radius(lic(D_g))}{radius(sec(D_g))}\,. \tag{3.3}$$

This ratio is always between 0 and 1 and it is 1 if and only if both circles are identical, i.e., the district itself is also circular. The Haggett-Test is also independent of scale. In order to apply the Haggett-Test to a district only its shape is necessary. In contrast to the Reock-Test and the Gibbs-Test, the evaluation of the district illustrated in Figure 3.1c is poor since its largest inscribed circle is very small in relation to its smallest enclosing circle.

However, the main problem of this test is that the largest inscribed circle is very hard to determine. Moreover, an indentation on the boundary influences the result more than it does for the tests described before. Unfortunately, an indentation on a district's boundary may be prescribed by the shape of a basic area.

Frolov [13] lists a variant of the Haggett-Test. It computes the ratio of the area of the largest inscribed circle and the area of the smallest enclosing circle, i.e.,

$$comp_{frolov}(D_g) := \frac{area(lic(D_g))}{area(sec(D_g))}\,.$$

Since the area of a circle is defined as the square of its radius multiplied with π, it can also be stated as

$$comp_{frolov}(D_g) = \frac{radius(lic(D_g))^2}{radius(sec(D_g))^2}\,.$$

Hence, the values of $comp_{frolov}(\cdot)$ and $comp_{haggett}(\cdot)$ differ, whereas the ranking for a set of districts is equal.

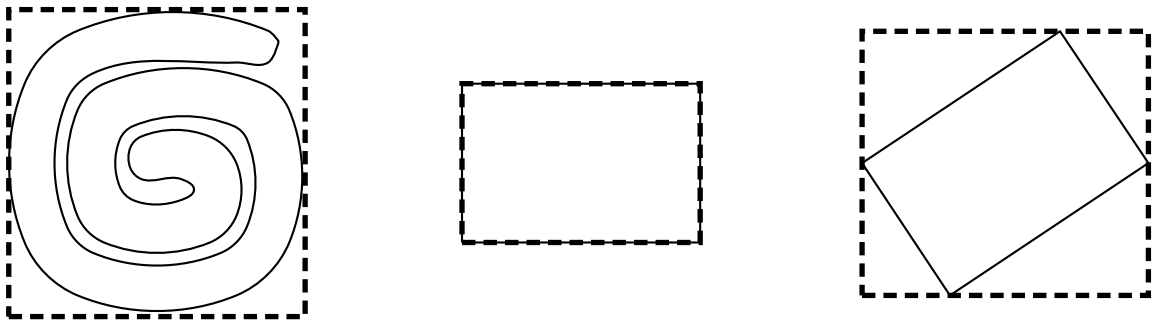
3.3.1.4 Length-Width-Test

Several authors propose *Length-Width-Tests*. All of them have in common that they firstly determine an enclosing rectangle of the evaluated district, and after that measure the compactness based on the length and width of this rectangle. The most compact rectangle is a square since in this case length and width are equal. Hence, the closer the enclosing rectangle is to a square, the better its compactness evaluation. The following overview of Length-Width-Tests starts with tests which define compactness as a ratio of width and length, i.e.,

$$comp_{length-width-ratio}(D_g) := \frac{wi(er(D_g))}{le(er(D_g))}\,. \tag{3.4}$$

These tests are called more specifically *Length-Width-Ratio-Tests*. The difference between the following variations is the kind of enclosing rectangle:

a) Young [37] presents a test that determines the enclosing rectangle such that it touches the district on all four sides and its ratio of length to width is maximal. In other words, this test regards the most non-compact enclosing rectangle that touches the district on all four sides.
b) Niemi et al. [29] mention the idea of defining the rectangle as the one having the minimal perimeter.
c) Niemi et al. [29] list another approach and refer to Harris Jr. [19] as original source. It defines the rectangle such that its length is defined as the length of the longest axis of the district and its width is defined as the maximum length of the district perpendicular to this longest axis.



(a) Snake (b) Axis-parallel rectangle (c) Rotated rectangle

Figure 3.4: Applying various Length-Width-Tests to different shapes

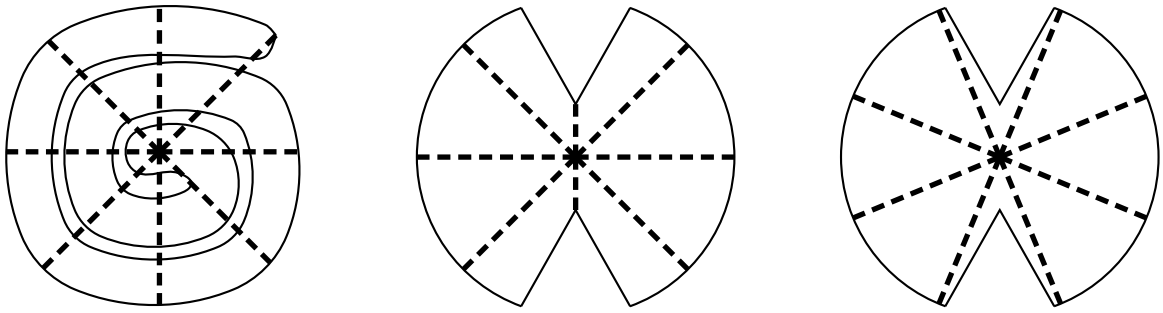
d) Furthermore, Niemi et al. [29] cite an approach by Eig and Seitzinger [12]. Here, the rectangle is defined as the smallest axis-parallel enclosing rectangle. In other words: The edges of the rectangle have to be orientated in north-south and in east-west direction.

Advantages of (all variants of) this test are the independence of scale and the usability for all kind of representations. Actually, polygon points or end-points of lines are sufficient to determine an enclosing rectangle. For variant a) and c) length and width are defined such that the achieved result is between 0 and 1. The best possible evaluation is 1 achieved if and only if the rectangle is a square, i.e., length and width are equal. For variant b) and d), length and width can be defined such that length is greater than or equal to width. In this case, the achieved result is also between 0 and 1.

However, these tests regard only the enclosing rectangle, but no spatial dispersion within this rectangle. So, again the snake-shaped district illustrated in Figure 3.4a shows one weakness of these tests since this visually non-compact shape is evaluated very favorably. Moreover, for variant d) the evaluation depends on the orientation of the district, i.e., two equally shaped districts are evaluated differently if they differ in their orientation. For example, the evaluation of the axis-parallel rectangle depicted in Figure 3.4b is worse compared to that of the rotated rectangle in Figure 3.4c since the enclosing rectangle for the latter is nearly square.

It should also be mentioned that, for example, Papayanopoulos [30] proposes the usage of *Length-Width-Difference-Tests* which determine the difference of length and width instead of the ratio, i.e.,

$$comp_{length-width-ratio}(D_g) := le(er(D_g)) - wi(er(D_g))\,.$$



(a) Snake (b) Original radial lines (c) Rotated radial lines

Figure 3.5: Applying the Boyce-Clark-Test to different shapes

Obviously, this approach can be used for all definitions of enclosing rectangles stated in a) to d). However, it has further drawbacks: First, obtained values are not restricted to be in the range between 0 and 1. Second, its evaluation is not independent of scale.

3.3.1.5 Boyce-Clark-Test

The so-called *Boyce-Clark-Test*, proposed by Boyce and Clark [4], utilizes a set of equally-spaced radial lines from the center of gravity to the outer boundary in order to measure local compactness. It is feasible that such a radial line leaves the district and returns to it. Note that for polygons as well as for a set of points a closed formula to determine the center of gravity exists. At first, this test determines for each line the percentage of its length over the total length of all lines. Thereafter, it determines the absolute differences to the average percentage and adds them up. Let $r_{i,g}$ be the length of the i-th radial line of district D_g and n the total number of considered lines. Then, the Boyce-Clark-Test is defined by the following equation:

$$comp_{boyce-clark}(D_g) := \sum_{i=1}^{n} \left| \frac{r_{i,g}}{\sum_{i=1}^{n} r_{i,g}} \cdot 100 - \frac{100}{n} \right| .$$

For a compact shape such as a circle the lengths of all radial lines are equal, whereas for a non-compact shape such as an elongated rectangle these lengths differ noticeably. Thus, a result close to 0 indicates a compact district. Unfortunately, the results are not limited to be between 0 and 1. Moreover, this measure is independent of scale, but unfortunately number and orientation of the radial lines influence its results highly. Figure 3.5b and Figure 3.5c show the same district, but the radial lines differ. The corresponding evaluation values are 30 and 0, respectively. The same effect occurs for fixing the angles of the radial lines, but rotating the district. In order to increase the probability that the two indentations

are detected in this special case, or more generally spoken, to reduce the effect of different evaluations for identical but rotated shapes, the number of considered radial lines can be increased. A further drawback is that only the outer boundary is taken into account. Hence, the snake illustrated in Figure 3.5a is once more evaluated well, because all radial lines have approximately the same length.

3.3.1.6 Relative Moment of Inertia

Kaiser [23] proposes a local compactness measure that determines the second moment of inertia of a district about its center of mass divided by the second moment of inertia of a circle having the same area. This measure is called *Relative Moment of Inertia* and is defined as follows:

$$comp_{rmoi}(D_g) := \frac{\int_{D_g}\int_{D_g}(x^2+y^2)\mathrm{d}x\mathrm{d}y}{\frac{area(D_g)^2}{2\cdot\pi}}\,. \tag{3.5}$$

It is independent of scale, but unfortunately its results are not limited within a given range. Moreover, Niemi et al. [29] criticize that the Relative Moment of Inertia is more difficult to determine and to understand than other measures. Even Kaiser [23] remarks that in practice Equation (3.5) has to be approximated by numerical integration.

However, for polygons the numerator can be stated as closed formulation. Let district D_g be represented by a polygon. This polygon has the clockwise counted polygon vertices $(x_{g,1}, y_{g,1}), \ldots, (x_{g,n_g}, y_{g,n_g})$. Let $(x_{g,n_g+1}, y_{g,n_g+1}) := (x_{g,1}, y_{g,1})$, then, the polygon's center of gravity $(x_{g,cog}, y_{g,cog})$ results in

$$x_{g,cog} := \frac{1}{6\cdot area(D_g)}\sum_{j=1}^{n_g}(x_{g,j}+x_{g,j+1})\cdot(x_{g,j}\cdot y_{g,j+1}-x_{g,j+1}\cdot y_{g,j})$$

and

$$x_{g,cog} := \frac{1}{6\cdot area(D_g)}\sum_{j=1}^{n_g}(y_{g,j}+y_{g,j+1})\cdot(x_{g,j}\cdot y_{g,j+1}-x_{g,j+1}\cdot y_{g,j})\,.$$

Let $(x'_{g,j}, y'_{g,j})$ be the coordinates of the j-th polygon vertex relative to the polygon's center of gravity. In this case, the second moment of inertia of this district about its center of mass results in

$$\frac{1}{12}\cdot\sum_{j=1}^{n_g}(x'^2_{g,j}+x'_{g,j}x'_{g,j+1}+x'^2_{g,j+1}+y'^2_{g,j}+y'_{g,j}\cdot y'_{g,j+1}+y'^2_{g,j+1})\cdot(y'_{g,j}\cdot x'_{g,j+1}-y'_{g,j+1}\cdot x'_{g,j})\,.$$

Horn et al. [21] conclude that this measure comes close to a theoretically perfect compactness measure. However, as Section 3.4 will present in more detail, in practice this measure does not outperform some other measures.

Note that Section 3.3.5.1 presents another measure that utilizes the moment of inertia. In contrast to this approach it does not normalize the moment of inertia and it only considers a set of discrete points within the district instead of the entire shape.

Finally, Horn et al. [21] lists two correlated measures:

- By using the reciprocal, i.e.,

$$comp_{rmoi-inv}(D_g) := \frac{1}{comp_{rmoi}(D_g)}\,, \tag{3.6}$$

  the values are bounded in the range between 0 and 1.

- Horn et al. [21] refer to Blair and Biss [3] as source of the following version:

$$comp_{blair-biss}(D_g) := \frac{1}{\sqrt{comp_{rmoi}(D_g)}}\,.$$

  However, they do not see an advantage compared to the version before.

3.3.2 Shape-Only-Area-Perimeter Measures

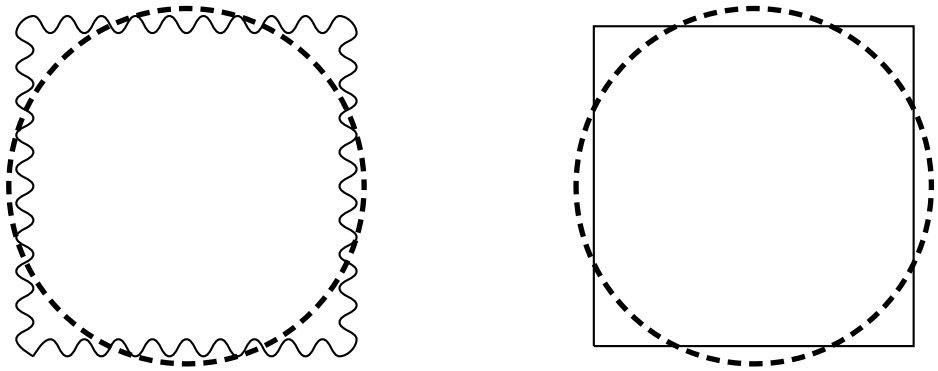
A measure classified as *shape-only-area-perimeter measure* tries to take into account two dimensions of compactness: Dispersion and perimeter. However, these measures do not really take dispersion into account, but the area. In order to be independent of scale, they determine some kind of relation between perimeter and area.

3.3.2.1 Schwartzberg-Test

Schwartzberg [33] describes the first measure of this class. Thus, this compactness measure is most commonly called *Schwartzberg-Test*, although Niemi et al. [29] quote Horton [22] as additional origin. It determines the ratio of the perimeters of the district and of a circle having equal area, i.e.,

$$comp_{schwartzberg}(D_g) := \frac{per(D_g)}{2 \cdot \sqrt{\pi \cdot area(D_g)}}\,. \tag{3.7}$$

This measure is based on the idea that for a given area, regarding all figures having this area, a circle has the smallest perimeter.



(a) Non-smooth boundary (b) Smooth boundary

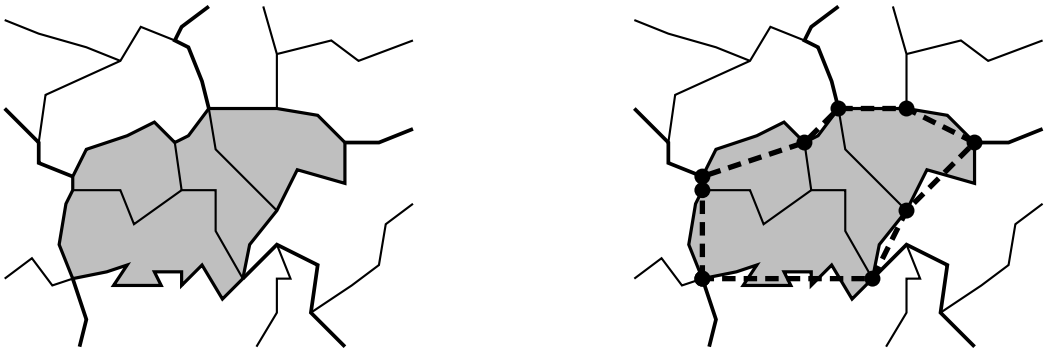
Figure 3.6: Illustration of the Schwartzberg-Test

The Schwartzberg-Test is independent of scale. Moreover, the required data, precisely perimeter and area, are easy to determine if the basic areas are represented by polygons. An evaluation value obtained for applying the Schwartzberg-Test is always greater than or equal to 1, where 1 is the best possible evaluation, reached if and only if the evaluated district is circular. In order to transform the obtained result into the 0 to 1 range we suggest using the reciprocal value, i.e.,

$$comp_{schwartzberg-inv}(D_g) := \frac{2 \cdot \sqrt{\pi \cdot area(D_g)}}{per(D_g)} \,. \tag{3.8}$$

The most noticeable drawback is the fact that the Schwartzberg-Test focuses on the perimeter. A nearly-quadratic district having a winding non-smooth boundary evaluates to a poor result. For example, Figure 3.6a depicts a district having an evaluation value of 1.77 (reciprocal value of 0.56). In contrast to this, Figure 3.6b depicts a quadratic district having an evaluation value of 1.13 (reciprocal value of 0.88).

However, cities or zip-code areas often have non-smooth boundaries. In order to overcome this problem Schwartzberg [33] recommends using an adjusted boundary. He proposes to take the constituent units into account, e.g., basic areas, forming the districts and to identify "trijunctions" on them. These are points on the boundaries of the districts where three or more constituent units of all districts meet. Finally, he defines an adjusted boundary by connecting these trijunctions by straight lines. Figure 3.7 illustrates this approach exemplarily. Take a look on the gray-colored district in Figure 3.7a. Figure 3.7b highlights the corresponding trijunctions. Moreover, the dashed line defines the obtained adjusted boundary. On the one hand, by applying this adjustment the prescribed shapes of the basic areas become more irrelevant. On the other hand, this approach usually reduces the lengths of the



(a) Original boundary (b) Adjusted boundary

Figure 3.7: Approach to determine an adjusted parameter according to [33]

boundaries. Note that other measures can be applied to such an adjusted boundary as well. Especially, the Taylor-Test defined in Section 3.3.4.1 explicitly refers to this adjustment.

Besides the Schwartzberg-Test defined in Equation (3.7), Horn et al. [21] list three similar measures:

- $comp_{schwartzberg-inv-adv}(D_g) := 1 - \frac{2 \cdot \sqrt{\pi \cdot area(D_g)}}{per(D_g)}$
- $comp_{schwartzberg-per}(D_g) := \frac{100 \cdot per(D_g)}{2 \cdot \sqrt{\pi \cdot area(D_g)}}$
- $comp_{grofman}(D_g) := \frac{per(D_g)}{\sqrt{area(D_g)}}$

The first one is simply the reversion of the reciprocal values of the Schwartzberg-Test in the range between 0 and 1, such that 0 becomes the best evaluation. The second measure states the ratio of the perimeters as percentage value. The third one multiplies the result of the Schwartzberg-Test with $2 \cdot \sqrt{\pi}$. Horn et al. [21] refer to Grofman [17] as origin of the latter. Hence, the listed variations are all transformations of the original Schwartzberg-Test stated in Equation (3.7).

3.3.2.2 Cox-Test

Cox [9] suggests determining the ratio of the district's area and the area of a circle having an equal perimeter. Hence, the *Cox-Test* results in

$$comp_{cox}(D_g) := \frac{4 \cdot \pi \cdot area(D_g)}{per(D_g)^2}\,. \tag{3.9}$$

For a given perimeter a circle is the figure having the largest possible area. Thus, the obtained results are always between 0 and 1 and the best possible evaluation of 1 is achieved

by a circle. The Cox-Test is also independent of scale. Analogously to the Schwartzberg-Test, a district shaped such as the one in Figure 3.6a is poorly evaluated since it has a very long boundary. Of course, the same approach for smoothing a boundary as for the Schwartzberg-Test can be applied.

Comparing Equations (3.7) and (3.9) one can observe a relation between the Schwartzberg-Test and the Cox-Test, because $comp_{cox}(D_g) = \frac{1}{comp_{schwartzberg}(D_g)^2}$ holds.

Additionally, Horn et al. [21] list two similar measures:

- $comp_{cox-var1}(D_g) := \frac{area(D_g)}{per(D_g)^2}$
- $comp_{cox-per}(D_g) := \frac{400 \cdot \pi \cdot area(D_g)}{per(D_g)^2}$

The first variant divides the evaluation value of the Cox-Test by $4 \cdot \pi$. The second one states the ratio of the areas as a percentage value. Hence, both variants are correlated with the original Cox-Test introduced in Equation (3.9).

3.3.3 Shape-Only-Perimeter Measures

A *shape-only-perimeter measure* focuses on the districts' perimeters. In contrast to the approaches described before, it takes the entire districting plan into account. To the best of our knowledge, there are only two proposed measures falling under this category, the second measure can be interpreted as a variation of the first one.

3.3.3.1 Perimeter-Test

The *Perimeter-Test* is a global compactness measure that determines the total boundary length of all districts:

$$comp_{perimeter}(S) := \sum_{g=1}^{p} per(D_g) \,.$$

According to Young [37] several authors mention and recommend this measure. The idea behind this approach is that a short total boundary length indicates a compact districting plan.

Unfortunately, this is not always the case. On the one hand, a large total boundary length can be caused by prescribed irregular boundaries of basic areas. In this case, the evaluation is noticeably better if these boundaries are not part of the districts' boundaries, i.e., all basic areas sharing such a border are assigned to the same district. On the other hand, a non-compact urban district can be compensated by some compact rural districts having smooth

(a) 4 compact districts containing rural and urban parts

(b) 1 small non-compact urban district and 3 districts containing mainly rural parts

Figure 3.8: Dividing an overall area having a central urban region into 4 districts

boundaries. Figure 3.8 illustrates an example: The solution depicted in Figure 3.8b has a smaller total boundary length than the one depicted in Figure 3.8a, even though Figure 3.8a is visually more compact than Figure 3.8b.

The boundary length or perimeter, respectively, of a district is easy to compute if the basic areas are represented by polygons. Unfortunately, the Perimeter-Test is not independent of scale and its result is positive, but unlimited. Nevertheless, it can be reasonable to apply this measure in combination with a local compactness measure.

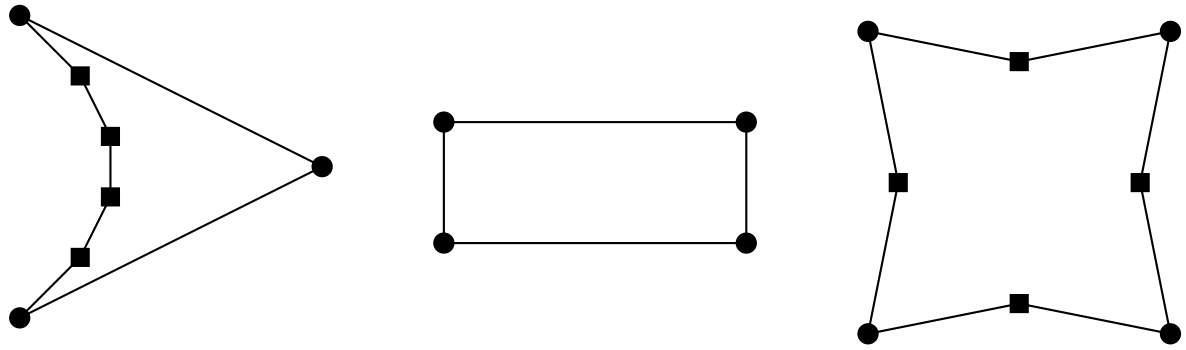
3.3.3.2 Bozkaya-Test

Bozkaya et al. [5] apply a further measure that can be seen as an enhancement of the Perimeter-Test, i.e.,

$$comp_{bozkaya}(S) := \frac{\sum_{g=1}^{p} per(D_g) - per(BA)}{2 \cdot per(BA)},$$

where $per(BA)$ is the perimeter of the regarded overall area. Hence, the *Bozkaya-Test* does not determine the total boundary length. It restricts itself to the common boundaries of two districts and ignores the outer boundary of the overall area. Since the authors use this global compactness as part of an additively weighted multi-criteria function, they normalize this length by the length of the outer boundary of the overall area.

Hence, this variation is independent of scale now. Unfortunately, this normalization does not necessarily limit the results to be between 0 and 1. Moreover, the problem that a shorter boundary does not necessarily indicate a more compact solution is still present. Nevertheless, due to the combination with other evaluation functions, mainly with a local compactness measure, this problem does not occur as strongly as before.



(a) District evaluated negative by the original Taylor-Test

(b) Elongated rectangle

(c) Nearly squared shape

Figure 3.9: Illustration of the Taylor-Test

3.3.4 Convexity Measures

Convexity measures address the question of how convex a district is. This class consists of "Angular measures", a subcategory in the classification of Horn et al. [21], as well as of the comparably new "Bizarreness Measure". These measures have in common that they focus on convexity, whereas dispersion and perimeter are of minor importance.

3.3.4.1 Taylor-Test

Taylor [34] proposes a measure that uses the number of reflexive and non-reflexive interior angles of a district's shape. An angle is reflexive if it has more than 180 degrees. Taylor [34] states that each indentation of a boundary has at least one reflexive angle. He concludes that a small number of reflexive angles indicates a compact district. Let R_g be the set of reflexive angles and N_g be the set of non-reflexive angles for district D_g. Then, the original *Taylor-Test* determines the difference of the numbers of non-reflexive and reflexive angles, normalized by the total number of angles, i.e.,

$$comp_{taylor-original}(D_g) := \frac{|N_g| - |R_g|}{|N_g| + |R_g|}\,.$$

Due to the normalization its result is always smaller than or equal to 1. Taylor [34] states wrongly that it is always greater than or equal to 0. Figure 3.9a illustrates a shape having four reflexive (illustrated by squares) and three non-reflexive (illustrated by points) angles that yields an evaluation value of $-\frac{1}{7}$. In Beth and Taylor [2] the following approach corrects this error: For each angle α it introduces an additional weight $w(\alpha)$ defined as total length of both sides defining the angle. Instead of the number of reflexive and non-reflexive angles

it uses the sum of the corresponding weights, i.e.,

$$comp_{taylor-corrected}(D_g) := \frac{\sum\limits_{\alpha \in N_g} w(\alpha) - \sum\limits_{\alpha \in R_g} w(\alpha)}{\sum\limits_{\alpha \in N_g} w(\alpha) + \sum\limits_{\alpha \in R_g} w(\alpha)} .$$

Both equations are independent of scale and their results are 1 if and only if no reflexive angle exists, i.e., if the shape is convex. This demonstrates the drawback of this approach: It is more a convexity measure than a compactness measure. Each district having a convex shape, e.g., the elongated rectangle depicted in Figure 3.9b, evaluates to the optimal value of 1. Moreover, Figure 3.9c shows a nearly square shape having four reflexive and four non-reflexive angles, where all eight sides have equal length. Hence, this shape evaluates to 0, i.e., according to the Taylor-Test this shape is totally non-compact.

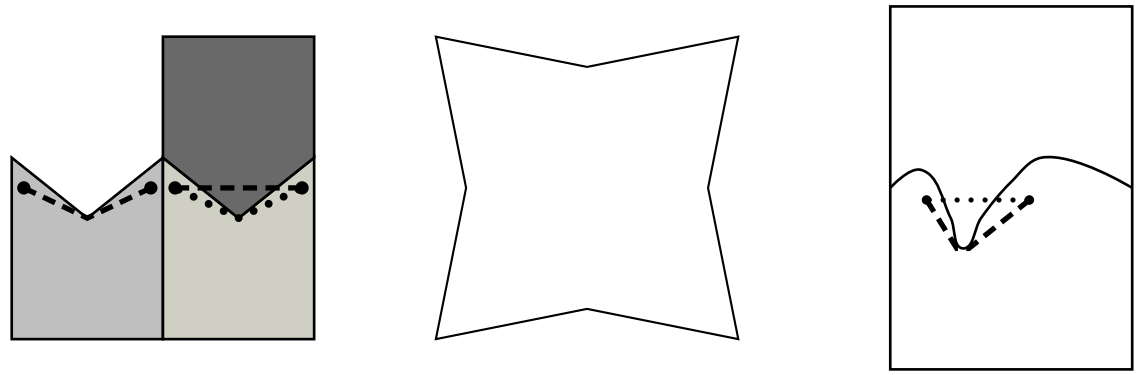
Since a prescribed irregular boundary has many angles that may distort the evaluation, Taylor [34] proposes applying the idea of Schwartzberg [33] in order to smooth the boundary.

3.3.4.2 Bizarreness-Test

Chambers and Miller [6] propose a "measure of Bizarreness". This comparably new local compactness measure has the goal of determining the "convexity" of a district. The authors argue that bizarrely shaped districts such as the famous gerrymander (cf. Section 1.1.1) are highly non-convex. They acknowledge that an elongated convex district is not detected as non-compact by their measure, but this should be evaluated by another measure. The *Bizarreness-Test* calculates the probability that a district contains the whole shortest path between two randomly selected points within this district. Let x and y be two arbitrary points of district D_g. Moreover, let $sp(x, y, D_g)$ be the length of the shortest path between x and y within district D_g and $sp(x, y, BA)$ be the length of the shortest path within the regarded overall area. The following definition states a discrete version using one representative point b_i for each basic area i:

$$comp_{bizarreness}(D_g) := \frac{\sum\limits_{i \in B_g} \sum\limits_{j \in B_g} \chi(\frac{sp(b_i, b_j, BA)}{sp(b_i, b_j, D_g)})}{\sum\limits_{i \in B_g} \sum\limits_{j \in B_g} 1} ,$$

with $\chi(z) = 1$ iff $z = 1$ and 0 otherwise. In other words, for a district D_g it determines the ratio of pairs of its basic areas with the shortest path between them lying completely within this district. In addition, Chambers and Miller [6] propose a continuous version, but in this case an integral has to be computed.



(a) Using Euclidean distances (b) Nearly squared shape (c) Using road distances

Figure 3.10: Illustration of the Bizarreness-Test

Figure 3.10a shows a partition of an overall area into three districts. Both districts located in the south are equally shaped. Nevertheless, the one located in the west is better evaluated than the other. For each pair of points of the first one the shortest path within this district is always equal to the shortest paths within the overall area. Hence, in a wider sense this district is "convex" since the non-convexity is caused by the outer boundary of the overall area. In contrast to this, there are pairs of points of the second district having a shortest path that is not located completely within this district. For example, the dashed line illustrates the shortest path within the overall area between the depicted points. This path leaves the district. In addition, the dotted line illustrates the shortest path between them within the district.

This measure is independent of scale and its results are always between 0 and 1. In contrast to the Taylor-Test a nearly-squared district such as the one depicted in Figure 3.10b is well evaluated. However, an optically non-compact district having an elongated convex shape is evaluated very favorably again.

The non-convexity of a district can be caused by an irregular boundary between two basic areas that are assigned to different districts. This case is not covered by the proposed measure. Thus, Chambers and Miller [7] introduce an extension where road distances instead of Euclidean distances are used to determine the shortest path between two basic areas. That reduces the negative effect of irregular boundaries caused by obstacles such as rivers or mountains on compactness evaluation. Figure 3.10c shows an example where a boundary is prescribed by a river. Concerning the two depicted points, the dashed line illustrates the shortest road path, whereas the dotted line illustrates the shortest Euclidean path. The shortest road path is completely located within the corresponding district, whereas the Euclidean path partly leaves the district. However, the usage of road distances reduces

the comprehensibility of the obtained results. Moreover, the correlation between visual impression and evaluation can decrease.

In addition, Chambers and Miller [6] propose a version that incorporates the activity measures of the basic areas as well:

$$comp_{bizarreness-weighted}(D_g) := \frac{\sum_{i \in B_g} \sum_{j \in B_g} w_i \cdot w_j \cdot \chi(\frac{sp(b_i,b_j,B)}{sp(b_i,b_j,D_g)})}{\sum_{i \in B_g} \sum_{j \in B_g} w_i \cdot w_j} .$$

Figuratively spoken, it determines the probability that a shortest path between two units of activity of the same district is completely located within this district. A unit of activity can be, for instance, a single voter in the context of political districting.

3.3.5 Distance-Based Measures

Distance-based measures focus on the dispersion of basic areas within a district, but they do not take the exact shape into account. The main idea is to add up distances between basic areas or from these basic areas to a specified location, where smaller distances indicate more compact districts. These distances can either be used unweighted or weighted by the corresponding activity measures. Moreover, distance-based measures can be defined as local compactness measures as well as global compactness measures.

3.3.5.1 (Weighted) Moment of Inertia

The first distance-based measure is based on distances between basic areas and centers. For each district this measure adds up the squared distances between all assigned basic areas and its center weighted by the basic areas' activity measures, i.e.,

$$comp_{wmoi}(D_g) := \sum_{i \in B_g} w_i \cdot d^2(b_i, cen_g) . \tag{3.10}$$

The center is chosen such that $comp_{wmoi}(D_g)$ is minimized. Thus, it corresponds to the center of gravity, given by

$$cen_g := \left(\frac{\sum_{i \in B_g} w_i \cdot x_i}{\sum_{i \in B_g} w_i}, \frac{\sum_{i \in B_g} w_i \cdot y_i}{\sum_{i \in B_g} w_i} \right) . \tag{3.11}$$

The global compactness of a districting plan is defined straightforwardly as the sum of the compactness values of its districts, i.e.,

$$comp_{wmoi}(S) := \sum_{g=1}^{p} comp_{wmoi}(D_g)\,.$$

The origin of this measure is in Weaver and Hess [36]. The authors use it as the objective function in the first proposed integer program for districting problems. It is also known as the *Moment of Inertia* or *Population Moment of Inertia* in the literature [21, 29, 37]. It is directly applicable to different kind of geometric representations since only a representative point for each basic area is taken into consideration. For example, for polygons this can be the geographical center, or for lines the middle-point. Another advantage is that the shapes of the assigned basic areas are irrelevant.

However, this measure also has some weaknesses. First of all, it is not independent of scale. Actually, the size of a district strongly affects the obtained result. Moreover, its result is not limited to be in the range between 0 and 1. Recall that the Relative Moment of Inertia described in Section 3.3.1.6 tries to overcome these drawbacks. However, it regards the total area instead of discrete points.

Additionally, setting w_i to 1 for all basic areas defines an unweighted version. In this case, the center results in

$$cen_g^{un} := \left(\frac{\sum\limits_{i \in B_g} x_i}{|B_g|}, \frac{\sum\limits_{i \in B_g} y_i}{|B_g|} \right) \tag{3.12}$$

and the compactness measure leads to

$$comp_{moi}(D_g) := \sum_{i \in B_g} d^2(b_i, cen_g^{un})\,. \tag{3.13}$$

In order to distinguish between these two variants, from now on the term *Weighted Moment of Inertia* denotes the measure defined in Equation (3.10) and *Moment of Inertia* denotes the one defined in Equation (3.13).

In addition, there is a similar measure that normalizes the obtained result such that it is between 0 and 1, called *Normalized Moment of Inertia*. However, this test is more a measure of the activity distribution within a district than a distance-based measure. Hence, it is listed as activity-based-Measure below.

3.3.5.2 (Weighted) Pairwise Distances

Papayanopoulos [30] also proposes a distance-based measure. For each district this measure determines the sum of distances between all pairs of assigned basic areas, weighted by the corresponding activity measure, i.e.,

$$comp_{wpd}(D_g) := \sum_{i \in B_g} \sum_{j \in B_g} w_i \cdot d_{i,j} \,. \tag{3.14}$$

Hence, this measure is called *Weighted Pairwise Distances*. Analogously to the Moment of Inertia, the sum of the compactness values of all districts defines the compactness measure of a districting plan, i.e.,

$$comp_{wpd}(S) := \sum_{g=1}^{p} comp_{wpd}(D_g) \,.$$

The advantages and disadvantages are similar to the ones of the Moment of Inertia. It is applicable to all types of representations of basic areas and the shapes of the basic areas are irrelevant. However, this measure is not independent of scale and its result is not necessarily between 0 and 1.

Finally, this measure can be varied as well:

- The first variation sets w_i to 1 for all basic areas, i.e., it defines an unweighted version, called *Pairwise Distances*. It leads to

$$comp_{pd}(D_g) := \sum_{i \in B_g} \sum_{j \in B_g} d_{i,j} \,. \tag{3.15}$$

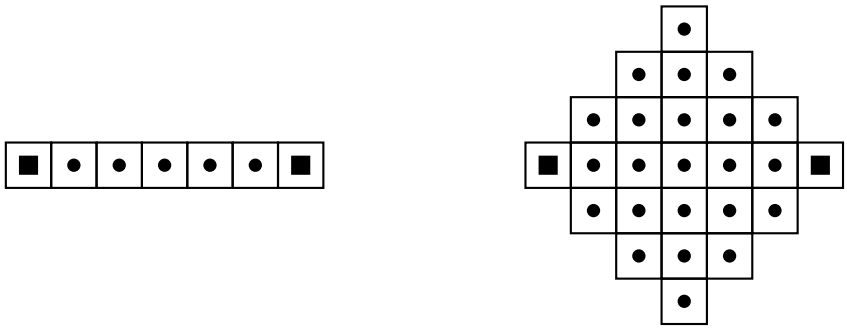
- Another variation, called *Squared Pairwise Distances*, is derived by an approach of Fryer Jr. and Holden [14]. It results in

$$comp_{pd-squared}(D_g) := \sum_{i \in B_g} \sum_{j \in B_g} d_{i,j}^2$$

  and

$$comp_{pd-squared}(S) := \sum_{g=1}^{p} comp_{pd-squared}(D_g) \,.$$

  Note that this variation is correlated with the Moment of Inertia; the proof is given by Fryer Jr. and Holden [14].



(a) Elongated rectangular district (b) Nearly circular district

Figure 3.11: Maximum Distance applied to different districts

In fact, the authors introduce a *Relative Proximity Index*, that is the ratio between the Squared Pairwise Distances of S and S^*, where S^* is the optimal solution in terms of the Squared Pairwise Distances. Unfortunately, this index is greater than or equal to 1. Moreover, finding the optimal solution in terms of the Squared Pairwise Distances is NP-hard [14]. Thus, the authors use an approximation of the optimal solution. Consequently, it is more or less a comparison with the "best known solution" instead of a comparison with the optimal solution.

3.3.5.3 Maximum Distance

Ríos-Mercado and Fernández [32] present another measure that can be used as objective function of an integer program: The *Maximum Distance* between two basic areas of the same district. For a single district this measure is defined as follows:

$$comp_{md}(D_g) := \max_{i,j \in B_g} d_{i,j} \,. \tag{3.16}$$

Consequently, the compactness of a solution is defined as maximum compactness of one single district:

$$comp_{md}(S) := \max_{g=1,\dots,p} comp_{md}(D_g) \,.$$

This measure has the same advantages as the other distance-based measures. It is directly applicable to different representation types of basic areas. Moreover, it does not take shapes of basic areas into account.

The drawbacks are also similar. It is not independent of scale and the obtained result is not bounded in a specified range. Moreover, this measure takes only two extremal basic areas into account, and does not consider the dispersion of the other basic areas. Figure 3.11

illustrates this issue. The plotted point within each basic area is its geographic center. The two points defining the maximum distance are highlighted by squares. Although the district illustrated in Figure 3.11b looks more compact than the one depicted in Figure 3.11a, both districts are measured equally since for both districts the maximum distance between two of its assigned basic areas are equal.

3.3.6 Activity-Based Measures

Niemi et al. [29] argue that there is a difference between geographic dispersion and population dispersion, and, hence, it may make sense to regard the dispersion of the population within a district. The argumentation of Niemi et al. [29] is focused on political districting. In order to be more general the term *activity* substitutes the term *population*. So, *activity-based measures* have in common that they take activities into account. Thus, the weighted versions of distance-based measures described above as well as the weighted version of the Bizarreness-Test can also be categorized as activity-based measures.

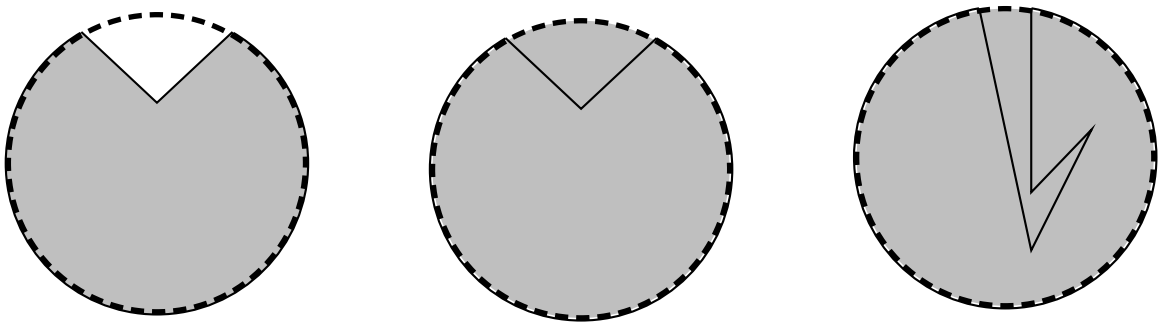
3.3.6.1 Hofeller-Grofman-Test

Hofeller and Grofman [20] propose a measure that has some similarities to the Reock-Test. It computes the ratio of the activities of the district and of the smallest enclosing circle. Hence, the *Hofeller-Grofman-Test* results in

$$comp_{hofeller}(D_g) := \frac{w(B_g)}{w(sec(D_g))}\,.$$

Obviously, this ratio is always between 0 and 1. A district has an evaluation value of 1 if and only if the total existing activity of the enclosing circle is assigned to this district, in other words, if there is no basic area that is located within the enclosing circle of a district, but not assigned to this district. Another advantage of this measure is the independence of scale. Moreover, this measure is applicable even if basic areas are represented by points or lines.

One problem can be the computation of the enclosing circle's activity. In general, the activity for each basic area is given, but its distribution within this basic area is not necessarily given and may hard to derive from the given data. Unfortunately, the Hofeller-Grofman-Test needs activities for parts of basic areas. Actually, for point representations this problem does not occur since a point is either located completely within or completely outside the enclosing circle.



(a) Total activity of the enclosing circle is located within the district

(b) Activity outside the district is concentrated at the border of the enclosing circle

(c) Activity outside the district is meandering within the enclosing circle

Figure 3.12: Illustration of the Hofeller-Grofman-Test

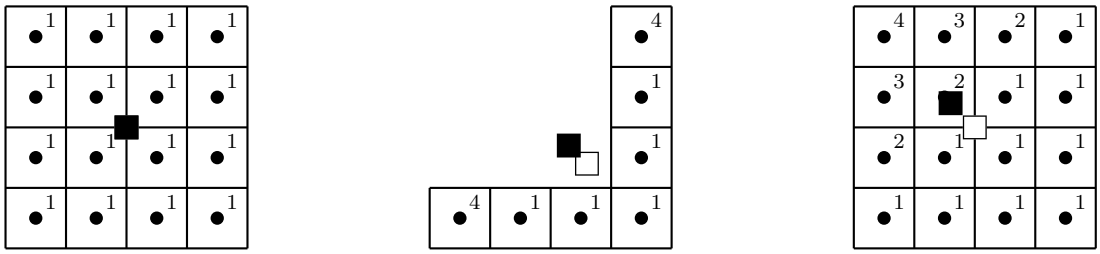
Compared to the Reock-Test, the Hofeller-Grofman-Test has the advantage that it considers areas located within the enclosing circle, but not within the district, more differentiated. For each example depicted in Figure 3.12 the activity is equally distributed within the gray-colored area, whereas no activity exists on the white-colored area. The Hofeller-Grofman-Test evaluates the district depicted in Figure 3.12a as perfectly compact since the total activity of the enclosing circle is assigned to this district. In contrast to this, the district illustrated in Figure 3.12b is worse evaluated compared to the prior since a part of the enclosing circle's activity is not assigned to it. The Reock-Test does not differentiate Figure 3.12a and Figure 3.12b since their respective ratio of district's area and enclosing circle's area are identical.

The Hofeller-Grofman-Test considers only the proportion of activity within the enclosing circle that is not assigned to the district, but not its distribution within this circle. Thus, it makes no difference if this activity is located concentrated at the border of the circle, such as in the example shown in Figure 3.12b, or if it meanders through this circle such as in Figure 3.12c.

In addition, Hofeller and Grofman [20] present a variation of this measure that uses the convex hull instead of the enclosing circle, i.e.,

$$comp_{hofeller-convex}(D_g) = \frac{w(D_g)}{w(ch(D_g))}\,.$$

However, the problem of this approach is again the fact, that an elongated rectangular district is evaluated as perfectly compact.



(a) Uniformly distributed activity within a squared district

(b) L-shaped district with high activity on opposite corners

(c) Squared district with high activity in one corner

Figure 3.13: Illustration of the Normalized Moment of Inertia

3.3.6.2 Normalized Moment of Inertia

Horn et al. [21] list another measure, defined by

$$comp_{nmoi}(D_g) := \frac{\sum\limits_{i \in B_g} w_i \cdot d^2(b_i, cen_g)}{\sum\limits_{i \in B_g} w_i \cdot d^2(b_i, cen_g^{un})} .$$

Hence, it determines the ratio between the Weighted Moment of Inertia (cf. Section 3.3.5.1) and the weighted sum of squared distances to the unweighted center of gravity defined in Equation (3.12). Due to the definition of $comp_{wmoi}(D_g)$ (cf. Equation (3.10)) and cen_g (cf. Equation (3.11)), respectively, this measure results in a value smaller than or equal to 1. In other words, this measure normalizes the Weighted Moment of Inertia such that its result falls between 0 and 1. Thus, it is called *Normalized Moment of Inertia*. Moreover, the measure becomes independent of scale by applying this normalization.

Unfortunately, a district D_g is compact in terms of this measure if cen_g and cen_g^{un}, defined in Equations (3.11) and (3.12), respectively, are close to each other, independently of the shape of the district.

Figure 3.13 illustrates some examples. For each basic area it illustrates the representative point and states its activity. For a district D_g it illustrates cen_g as filled black square and cen_g^{un} as white square. If the activity is more or less uniformly distributed within a district, this district is well evaluated. Figure 3.13a shows an example where cen_g and cen_g^{un} are even identical, i.e., this district is evaluated as perfectly compact. Since cen_g and cen_g^{un} are also close to each other for the district illustrated in Figure 3.13b, the obtained evaluation value is 0.96. That means, the district is well evaluated, although it is optically not compact and its activity is concentrated on opposite corners. In contrast to this, the district depicted in Figure 3.13c is optically compact. However, its activity is concentrated in one corner.

Therefore, its evaluation value is only 0.89, that means according to this test it is less compact than the district regarded before.

In conclusion, this test is more a measure of activity (population) distribution within a district than a compactness measure.

3.4 Evaluation

After presenting a couple of proposed measures, this section analyzes their suitability in theory and practice. First of all, Table 3.1 summarizes the properties of these measures according to the requirements discussed in Section 3.2. Since these results are mainly based on theoretical considerations and on constructed counterexamples, this section studies how these measures work in practice. We have designed our evaluation as a Visual-Test followed by a correlation analysis between this Visual-Test and a few selected measures. These results are based on the Bachelor thesis of Ludwig [26]. In addition, the results presented here contain some more measures, describe some aspects in more detail, and insert some further analysis. The data are the 70 electoral districts for the Landtag of Baden-Württemberg in 2011. Figure 3.14 shows these districts.

Figure 3.14: Electoral districts of Baden-Württemberg in 2011 ©Statistisches Landesamt Baden-Württemberg, Stuttgart, 2011

measure	local or global	0 to 1 range	ignore shapes of basic areas	independent of scale	verifiable data and comprehensibility
Reock-Test	local	yes	no	yes	yes
Gibbs-Test	local	yes	no	yes	yes
Haggett-Test	local	yes	no	yes	average[1]
Length-Width-Ratio-Test	local	yes	no	yes	yes
Length-Width-Difference-Test	local	no	no	no	yes
Boyce-Clark-Test	local	no	no	yes	yes
Relative Moment of Inertia	local	no	no	yes	average[2]
Schwartzberg-Test	local	no[3]	yes[4]	yes	yes
Cox-Test	local	yes	yes[4]	yes	yes
Perimeter-Test	global	no	no	no	yes
Bozkaya-Test	global	no	no	yes	yes
Taylor-Test (original)	local	no	yes[4]	yes	vey
Taylor-Test (corrected)	local	yes	yes[4]	yes	yes
Bizarreness-Test	local	yes	average[5,6]	yes	yes[7]
(Weighted) Moment of Inertia	both	no	yes	no	yes
(Weighted) Pairwise Distances	both	no	yes	no	yes
Maximum Distance	both	no	yes	no	yes
Hofeller-Grofman-Test	local	yes	average[5]	yes	average[8]
(Normalized) Moment of Inertia	local	yes	yes	yes	yes

[1] Largest inscribed circle hard to compute

[2] In general an integral must be calculated; approach might be non-intuitive

[3] Yes if the reciprocal value is used

[4] If adjusted perimeter is used

[5] Shape of overall area is ignored

[6] If road distances are used, shapes are more likely to be ignored

[7] Use of road distances reduces the comprehensibility

[8] Activities of parts of basic areas may be necessary

Table 3.1: Properties of proposed compactness measures

3.4.1 Visual-Test

We have conducted a survey where the participants should evaluate these electoral districts by German school marks between 1 (best) and 6 (worst). In total 185 persons have participated in this survey [26]. For each district our *Visual-Test* defines the evaluation value as the average mark by these participants. The first column of Table 3.2 states the obtained results.

Moreover, Table 3.2 presents the results for applying a few selected compactness measures to the electoral districts as well, namely: The Reock-Test (cf. Equation (3.1)), a variation of the Reock-Test using the convex hull as reference object (cf. Equation (3.2)), the Haggett-Test (cf. Equation (3.3)), the Length-Width-Ratio-Test applied to the smallest axis-parallel enclosing rectangle (cf. Equation (3.4)), a variation of the Relative Moment of Inertia using the reciprocal value (cf. Equation (3.6)), a variation of the Schwartzberg-Test using the reciprocal value (cf. Equation (3.8)), and the Cox-Test (cf. Equation (3.9)). In order to obtain the necessary data we have used ArcGIS 10[1].

As additional information, for each measure the bottom rows of Table 3.2 state the minimum, maximum and average evaluation value over all districts. This points out that the ranges of the evaluations differ noticeably. For example, the variation of the Reock-Test evaluates all districts in the comparatively small range between 0.55 and 0.88. In contrast to this, the Relative Moment of Inertia results lie in the range between 0.35 and 0.96. Moreover, the best evaluated district according to the Cox-Test has an evaluation value of 0.63. The Cox-Test evaluates the worst district by 0.15. The obtained results confirm again that it is very hard to define a threshold for whether a district is compact or not.

A result obtained for applying a compactness measure to a district should be correlated with our visual impression whether this district is compact or not. Hence, the results of a suitable compactness measure should be correlated with the results of this Visual-Test.

[1] ESRI®, www.esri.com

district	Visual-Test	Reock-Test (circle)	Reock-Test (convex hull)	Haggett-Test	Length-Width-Test	Relative Moment of Inertia	Schwartzberg-Test	Cox-Test
1	2.9	0.48	0.71	0.47	0.97	0.80	0.64	0.42
2	3.6	0.35	0.85	0.36	0.55	0.59	0.60	0.36
3	3.9	0.33	0.61	0.27	0.79	0.65	0.57	0.32
4	3.4	0.50	0.75	0.49	1.00	0.78	0.72	0.51
5	3.4	0.55	0.77	0.52	0.87	0.89	0.62	0.39
6	5.2	0.27	0.55	0.21	0.61	0.38	0.42	0.18
7	3.3	0.45	0.78	0.45	0.96	0.76	0.64	0.41
8	3.9	0.40	0.70	0.37	0.71	0.65	0.55	0.30
9	4.0	0.31	0.74	0.26	0.76	0.60	0.58	0.33
10	4.0	0.44	0.73	0.41	0.60	0.74	0.55	0.30
11	3.6	0.43	0.70	0.37	0.79	0.76	0.53	0.28
12	2.9	0.45	0.78	0.37	0.64	0.88	0.67	0.44
13	4.5	0.33	0.67	0.34	0.57	0.48	0.49	0.24
14	3.4	0.41	0.69	0.39	0.65	0.86	0.52	0.27
15	3.4	0.46	0.70	0.45	0.97	0.74	0.66	0.43
16	4.2	0.42	0.66	0.33	0.76	0.69	0.51	0.26
17	4.3	0.47	0.73	0.46	0.87	0.73	0.50	0.25
18	2.7	0.48	0.81	0.48	0.84	0.89	0.72	0.52
19	4.6	0.33	0.57	0.31	0.70	0.51	0.45	0.20
20	4.2	0.44	0.64	0.39	0.81	0.69	0.49	0.24
21	3.4	0.39	0.76	0.43	0.64	0.75	0.52	0.27
22	3.5	0.44	0.74	0.37	0.83	0.75	0.50	0.25
23	4.1	0.30	0.69	0.28	0.75	0.61	0.46	0.21
24	4.0	0.44	0.71	0.44	0.74	0.78	0.56	0.31
25	3.8	0.50	0.70	0.42	0.85	0.83	0.53	0.29
26	2.8	0.58	0.80	0.60	0.89	0.93	0.63	0.40
27	2.8	0.54	0.81	0.51	0.87	0.84	0.71	0.51
28	2.8	0.40	0.83	0.47	0.99	0.81	0.74	0.55
29	3.9	0.44	0.70	0.35	0.78	0.69	0.61	0.37
30	4.7	0.36	0.63	0.27	0.60	0.49	0.49	0.24
31	4.3	0.43	0.64	0.39	0.99	0.69	0.56	0.31
32	4.5	0.33	0.64	0.25	0.63	0.48	0.51	0.26
33	2.4	0.67	0.86	0.59	0.90	0.92	0.76	0.59
34	3.1	0.57	0.76	0.52	0.73	0.83	0.63	0.40
35	2.9	0.49	0.85	0.48	0.96	0.77	0.78	0.62
36	3.5	0.50	0.66	0.44	0.99	0.83	0.66	0.44
37	2.2	0.68	0.88	0.71	0.90	0.96	0.80	0.63
38	3.3	0.50	0.77	0.44	0.90	0.81	0.60	0.36

district	Visual-Test	Reock-Test (circle)	Reock-Test (convex hull)	Haggett-Test	Length-Width-Test	Relative Moment of Inertia	Schwartzberg-Test	Cox-Test
39	3.6	0.47	0.70	0.48	0.76	0.72	0.65	0.42
40	3.1	0.48	0.79	0.47	0.76	0.82	0.67	0.45
41	4.8	0.24	0.62	0.28	0.66	0.61	0.42	0.18
42	3.0	0.56	0.73	0.48	0.89	0.88	0.59	0.35
43	3.0	0.46	0.77	0.52	0.84	0.88	0.60	0.36
44	5.0	0.40	0.61	0.30	0.92	0.60	0.44	0.20
45	3.5	0.48	0.77	0.47	0.91	0.84	0.60	0.36
46	3.4	0.51	0.78	0.46	0.94	0.84	0.58	0.33
47	2.6	0.59	0.80	0.57	0.90	0.87	0.73	0.53
48	4.4	0.34	0.71	0.36	0.53	0.59	0.56	0.31
49	3.1	0.41	0.77	0.37	0.59	0.68	0.62	0.39
50	4.1	0.40	0.69	0.35	0.64	0.56	0.55	0.30
51	4.3	0.44	0.69	0.33	0.77	0.71	0.58	0.34
52	2.7	0.55	0.80	0.48	0.91	0.86	0.71	0.51
53	3.7	0.56	0.70	0.43	0.92	0.83	0.56	0.32
54	4.0	0.51	0.75	0.43	0.84	0.79	0.53	0.28
55	4.4	0.48	0.70	0.41	0.89	0.69	0.50	0.25
56	5.0	0.33	0.63	0.20	0.65	0.52	0.44	0.19
57	3.9	0.43	0.73	0.38	0.75	0.74	0.49	0.24
58	4.5	0.38	0.66	0.37	0.99	0.58	0.54	0.29
59	5.0	0.20	0.56	0.21	0.46	0.35	0.38	0.15
60	3.0	0.51	0.76	0.49	0.94	0.77	0.71	0.51
61	3.4	0.46	0.71	0.44	0.77	0.78	0.62	0.38
62	3.6	0.43	0.74	0.39	0.87	0.78	0.63	0.40
63	2.9	0.51	0.78	0.46	0.88	0.81	0.68	0.46
64	4.5	0.24	0.55	0.25	0.62	0.53	0.51	0.26
65	4.5	0.35	0.65	0.29	0.96	0.54	0.51	0.26
66	3.4	0.45	0.74	0.40	0.61	0.69	0.61	0.37
67	4.2	0.26	0.77	0.30	0.78	0.57	0.62	0.38
68	3.2	0.33	0.65	0.38	0.68	0.71	0.57	0.32
69	3.4	0.42	0.82	0.40	0.64	0.71	0.67	0.45
70	4.0	0.53	0.68	0.50	0.97	0.83	0.55	0.31
min	2.2	0.20	0.55	0.20	0.46	0.35	0.38	0.15
max	5.2	0.68	0.88	0.71	1.00	0.96	0.80	0.63
ave	3.7	0.44	0.72	0.41	0.80	0.72	0.59	0.35

Table 3.2: Compactness measures applied to electoral districts of Baden–Württemberg

3.4.2 Correlation Analysis

This section analyzes the correlation between the stated measures and the Visual-Test by means of the Pearson Correlation Coefficient. Let $comp_1(D_g)$ and $comp_2(D_g)$ be evaluation values of two compactness measures applied to a district D_g. Furthermore, let $\overline{comp_1}$ and $\overline{comp_2}$ be the average values of them over all districts $D_1, \ldots, D_p$. Then, the Pearson Correlation Coefficient is defined by

$$pear(comp_1, comp_2) := \frac{\sum_{g=1}^{p} (comp_1(D_g) - \overline{comp_1}) \cdot (comp_2(D_g) - \overline{comp_2})}{\sqrt{\sum_{g=1}^{p} (comp_1(D_g) - \overline{comp_1})^2} \cdot \sqrt{\sum_{g=1}^{p} (comp_2(D_g) - \overline{comp_2})^2}}.$$

An evaluation value obtained by the Visual-Test is smaller, the more compact the evaluated district is. In contrast to this, for the further measures presented in Table 3.2 a small evaluation value indicates a non-compact district. Hence, a negative Pearson Correlation Coefficient close to -1 between the latter and the Visual-Test indicates a high correlation. The first row of Table 3.3 states for each measure its correlation coefficient with the Visual-Test. In addition, the second row presents the respective correlation coefficients obtained by using ranking positions instead of absolute values. In this case, the best evaluated district according to a measure is ranked on position 1, whereas the worst one is ranked on position 70. Therefore, a high correlation is indicated by a positive coefficient close to 1.

	Reock -Test (circle)	Reock -Test (conv.)	Haggett-Test	Length-Width-Ratio-Test	Relative Moment of Inertia	Schwartz-berg-Test	Cox-Test
absolute values	−0.73	−0.82	−0.82	−0.40	−0.84	−0.87	−0.86
ranking values	0.69	0.80	0.79	0.37	0.82	0.84	0.84

Table 3.3: Correlation between various compactness measures and the Visual-Test

Take a closer look at these coefficients. As expected, the results of the Length-Width-Ratio-Test are not sufficient, compared to other measures they are noticeably worse. Moreover, altogether shape-only-area-perimeter measures are more highly correlated with the Visual-Test than shape-only-dispersion measures. As also expected, the Haggett-Test has a higher correlation with the Visual-Test than the original Reock-Test since the Haggett-Test additionally utilizes the geographic dispersion within the enclosing circle. More surprising is the fact that the Reock-Test using the convex hull as reference object performs better than the original version using the smallest enclosing circle, although a convex figure is not necessarily visually compact. Furthermore, the Relative Moment of Inertia correlates slightly more with

(a) District 9

(b) District 22

(c) Districts 67, 68, and 70

Figure 3.15: Enlarged illustration of some selected districts

the Visual-Test than the other shape-only-dispersion measures. In conclusion, this analysis shows that shape-only-area-perimeter measures as well as shape-only-dispersion measures work better in practice as it might be expected in light of their theoretical drawbacks.

Nevertheless, there are also examples for each measure where a computed evaluation value does not coincide with the visual impression. The districts regarded in the following are illustrated in Figure 3.15. The Reock-Test (Haggett-Test) evaluates district 70 comparatively well, its corresponding ranking position is 11 (9), whereas the Visual-Test ranks this district on position 48. Also the Relative Moment of Inertia ranks this district on position 18 as rather compact. The reason why this district looks less compact than others is its bulge in the northern part.
The survey's participants evaluate district 68 as rather compact, it is ranked on 19th position. However, the applied measures evaluate this district as (rather) non-compact. Especially, its large indentation from north-west to north-east causes a comparatively large enclosing circle or convex hull, respectively. Hence, both versions of the Reock-Test evaluate this district as poor in terms of compactness. The corresponding ranking positions are 60 for using the enclosing circle and 57 for using the convex hull. Moreover, the Haggett-Test ranks it on Position 42, the Schwartzberg-Test (Cox-Test) on position 37 and the Relative Moment of Inertia on position 53.
Due to its elongated shape, district 67 is visually non-compact, and, hence, it is ranked only on position 53 by the Visual-Test. However, its shape is nearly convex and the Reock-Test using the convex hull evaluates it comparatively well and ranks it on position 18. Moreover, since its boundary is smooth, mainly on the southern part, the Schwartzberg-Test (Cox-Test) also evaluates it noticeably better by ranking it on position 26.
According to the Schwartzberg-Test district 22 seems to be non-compact on ranking position 59 since it has a non-smooth boundary. However, the survey's participants evaluate it as average and rank it on position 31.
Finally, consider district 9. Its shape is elongated, and, hence, its largest inscribed circle is

relatively small compared to its smallest enclosing circle. Thus, the Haggett-Test ranks it on position 65 as one of the worst districts in terms of compactness. Its visual impression is not as bad and it is ranked on position 44 by the survey's participants.

In order to overcome the described problems it can be useful to combine different measures. For example, a combination of two compactness measures can be defined as a convex combination of their results:

$$comp(D_g) := \alpha \cdot comp_1(D_g) + (1-\alpha) \cdot comp_2(D_g)\,,$$

with $\alpha \in [0,1]$. Since the combined measures should cover various dimensions of compactness, the next analysis focuses on combinations of one shape-only-area-perimeter measure and one shape-only-dispersion-measure. According to Table 3.3 the former outperforms the latter, so this analysis uses the relations 1:1, 2:1, 3:1, and 4:1, i.e., it sets α to 0.5, 0.67, 0.75, and 0.8. Table 3.4 presents the correlation coefficients between the Visual-Test and the combined measures obtained in the described way. In addition, it states the coefficients for exclusively using $comp_1(\cdot)$ and $comp_2(\cdot)$, respectively, i.e., for setting $\alpha = 1$ and $\alpha = 0$, respectively. Moreover, Table 3.5 shows the correlation coefficients between the corresponding ranking positions.

$comp_1(\cdot)$	$comp_2(\cdot)$	$\alpha = 1$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 0.67$	$\alpha = 0.75$	$\alpha = 0.8$
Schwartzberg	Reock (circle)	−0.87	−0.73	−0.87	−0.89	−0.89	−0.89
Cox	Reock (circle)	−0.86	−0.73	−0.88	−0.89	−0.88	−0.88
Schwartzberg	Reock (convex)	−0.87	−0.82	−0.89	−0.89	−0.89	−0.88
Cox	Reock (convex)	−0.86	−0.82	−0.89	−0.88	−0.88	−0.87
Schwartzberg	Haggett	−0.87	−0.82	−0.90	−0.90	−0.90	−0.90
Cox	Haggett	−0.86	−0.82	−0.90	−0.90	−0.89	−0.88
Schwartzberg	RMoI	−0.87	−0.84	−0.92	−0.93	−0.92	−0.92
Cox	RMoI	−0.86	−0.84	−0.93	−0.93	−0.92	−0.91

Table 3.4: Correlation between absolute values of combined measures and the Visual-Test

Although the Schwartzberg-Test and the Cox-Test are highly correlated with the Visual-Test, the combination of them with shape-only-dispersion-measure yields further improvements. Setting α to 0.67, i.e., combining them with a relation of 2:1 seems to be a good choice. For example, the combined measure of the Schwartzberg-Test and the original Reock-Test again ranks district 67 on position 44, while the Schwartzberg-Test ranks it on position 26; the Visual-Test ranks it on position 53. Thus, the result of the combination is closer to the visual impression than the result of the Schwartzberg-Test.

$comp_1(\cdot)$	$comp_2(\cdot)$	$\alpha = 1$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 0.67$	$\alpha = 0.75$	$\alpha = 0.8$
Schwartzberg	Reock (circle)	0.84	0.69	0.85	0.86	0.87	0.87
Cox	Reock (circle)	0.84	0.69	0.86	0.86	0.87	0.87
Schwartzberg	Reock (convex)	0.84	0.80	0.87	0.86	0.86	0.86
Cox	Reock (convex)	0.84	0.80	0.87	0.87	0.86	0.86
Schwartzberg	Haggett	0.84	0.79	0.88	0.88	0.87	0.87
Cox	Haggett	0.84	0.79	0.88	0.88	0.87	0.87
Schwartzberg	RMoI	0.84	0.82	0.92	0.92	0.91	0.90
Cox	RMoI	0.84	0.82	0.92	0.92	0.90	0.90

Table 3.5: Correlation between ranking positions of combined measures and the Visual-Test

However, there are still examples where a combination also fails. For example, a measure that combines the Cox-Test and the Reock-Test using the convex hull as reference object ranks district 67 still on position 26. Nevertheless, a combination of one shape-only-area-perimeter measure and one shape-only-dispersion-measure is a reasonable way to measure compactness of a district. It takes different dimensions of compactness into account and its result is close to our visual impression. Especially, a measure combining the Relative Moment of Inertia and the Schwartzberg-Test (Cox-Test) results in correlation coefficients with the Visual-Test up to 0.93.

The compactness measures regarded up to now have in common that they are defined such that their computed results fall into the range of 0 to 1, with 1 being the best evaluation. Unfortunately, this is not the case for distance-based measures which are analyzed next. Here, the compactness values are greater than or equal to 0, but no general upper bound can be defined. Moreover, a small result indicates a compact district. Typically, in the context of electoral districting basic areas correspond to cities or communities. However, an electoral district can consist of only one city, sometimes even only of a part of a large city. In the case of Baden-Württemberg the largest city, Stuttgart, is partitioned into four electoral districts (1, 2, 3, 4). Moreover, Karlsruhe (Mannheim) is partitioned into two districts, namely 27, 28 (35, 36). District 34 consists only of the city of Heidelberg. Since distance-based measures are defined on distances between basic areas or between basic areas and specified centers, the corresponding results would be 0. Thus, the following analysis excludes these districts from the set of included districts. Therefore, the number of included electoral districts is reduced to 61.

This analysis uses the Moment of Inertia in a weighted version (cf. Equation (3.10)) as well as in an unweighted version (cf. Equation (3.13)), Pairwise Distance weighted (cf. Equation (3.14)) and unweighted (cf. Equation (3.15)), and the Maximum Distance (cf. Equa-

tion (3.16)). Table 3.6 presents the correlation coefficients between these measures and the Visual-Test, both for comparing absolute values and ranking values.

	Weighted Moment of Inertia	Moment of Inertia	Weighted Pairwise Distances	Pairwise Distances	Maximum Distance
absolute values	0.27	0.18	0.26	0.18	0.34
ranking values	0.34	0.27	0.31	0.22	0.33

Table 3.6: Correlation between various distance-based measures and the Visual-Test

Since a distance-based measure focuses more on the size than on the shape of the district, the correlation with the Visual-Test is expected to be rather small. The coefficients stated in Table 3.6 confirm this expectation. For distance-based measures as well as for the Visual-Test small values indicate compact districts. Hence, according to the absolute values a positive coefficient close to 1 indicates a high correlation. However, the stated correlations are noticeably smaller than for shape-only-area-perimeter measures and shape-only-dispersion measures, respectively. Nevertheless, in the context of sales districting the usage of distance-based measures can be useful since compact districts should help to reduce travel times of salespersons. In this case, the correlation with the visual impression is of minor importance.

The final analysis determines the correlations between pairs of measures based on the results for applying them to the 61 districts. Table 3.7 presents the results achieved for using absolute values, while Table 3.8 presents the results for using relative values. These results confirm some assumptions and statements given in Section 3.3.

Regarding shape-only-dispersion measures, there is a high correlation of 0.9 between the Haggett-Test and the Reock-Test. This can be explained by the fact that both measures set something in relation to the smallest enclosing circle. The Relative Moment of Inertia is also correlated to both of them, having correlation coefficients of 0.86 and 0.87. The correlation to the variation of the Reock-Test using the convex hull as reference object is noticeably smaller for all of them. Concerning the Length-Width-Test no noticeable correlation to any other measure is identifiable.

Regarding shape-only-area-perimeter measures, unsurprisingly, there is a correlation of 1.0 between the Schwartzberg-Test and the Cox-Test since $comp_{cox}(D_g) = \frac{1}{comp_{schwartzberg}(D_g)^2}$ holds.

The stated coefficients also show that shape-only-dispersion measures (expect the Length-Width-Test) and shape-only-area-perimeter measures are rather correlated. Since the measures of both classes also show good results concerning the correlation to the Visual-Test,

	Reock-Test (circle)	Reock-Test (conv.)	Haggett-Test	Length-Width-Test	Relative Moment of Inertia	Schwartzberg-Test	Cox-Test	Weighted Moment of Inertia	Moment of Inertia	Weighted Pairwise Distances	Pairwise Distances	Maximum Distance
Reock (circle)	1.00	0.72	0.90	0.63	0.86	0.71	0.70	-0.38	-0.22	-0.34	-0.15	-0.42
Reock (conv.)	0.72	1.00	0.78	0.37	0.78	0.83	0.82	-0.29	-0.18	-0.22	-0.13	-0.30
Haggett	0.90	0.78	1.00	0.56	0.87	0.76	0.76	-0.30	-0.18	-0.27	-0.15	-0.35
Length-Width	0.63	0.37	0.56	1.00	0.56	0.41	0.40	-0.18	0.04	-0.17	0.08	-0.20
Relative MoI	0.86	0.78	0.87	0.56	1.00	0.71	0.69	-0.36	-0.23	-0.33	-0.20	-0.40
Schwartzberg	0.71	0.83	0.76	0.41	0.71	1.00	1.00	-0.42	-0.26	-0.34	-0.22	-0.46
Cox	0.70	0.82	0.76	0.40	0.69	1.00	1.00	-0.42	-0.27	-0.35	-0.23	-0.47
Weighted MoI	-0.38	-0.29	-0.30	-0.18	-0.36	−0.42	−0.42	1.00	0.83	0.89	0.70	0.91
MoI	-0.22	-0.18	-0.18	0.04	-0.23	−0.26	−0.27	0.83	1.00	0.71	0.95	0.82
Weighted PD	-0.34	-0.22	-0.27	-0.17	-0.33	−0.34	−0.35	0.89	0.71	1.00	0.61	0.81
Pairwise Dist.	-0.15	-0.13	-0.15	0.08	-0.20	−0.22	−0.23	0.70	0.95	0.61	1.00	0.70
Maximum Dist.	-0.46	-0.30	-0.35	-0.20	-0.40	−0.46	−0.47	0.91	0.82	0.81	0.70	1.00

Table 3.7: Correlation between evaluation values of various compactness measures

	Reock-Test (circle)	Reock-Test (conv.)	Haggett-Test	Length-Width-Test	Relative Moment of Inertia	Schwartzberg-Test	Cox-Test	Weighted Moment of Inertia	Moment of Inertia	Weighted Pairwise Distances	Pairwise Distances	Maximum Distance
Reock (circle)	1.00	0.65	0.90	0.62	0.86	0.64	0.64	0.37	0.25	0.37	0.22	0.34
Reock (conv.)	0.65	1.00	0.71	0.30	0.74	0.80	0.80	0.31	0.22	0.24	0.18	0.25
Haggett	0.90	0.71	1.00	0.57	0.86	0.69	0.69	0.33	0.20	0.32	0.19	0.31
Length-Width	0.62	0.30	0.57	1.00	0.50	0.37	0.37	0.16	0.08	0.20	0.07	0.19
Relative MoI	0.86	0.74	0.86	0.50	1.00	0.66	0.66	0.40	0.30	0.39	0.26	0.39
Schwartzberg	0.64	0.80	0.69	0.37	0.66	1.00	1.00	0.45	0.37	0.35	0.34	0.40
Cox	0.64	0.80	0.69	0.37	0.66	1.00	1.00	0.45	0.37	0.35	0.34	0.40
Weighted MoI	0.37	0.31	0.33	0.16	0.40	0.45	0.45	1.00	0.94	0.94	0.89	0.94
MoI	0.25	0.22	0.20	0.08	0.30	0.37	0.37	0.94	1.00	0.89	0.96	0.94
Weighted PD	0.37	0.24	0.32	0.20	0.39	0.35	0.35	0.94	0.89	1.00	0.85	0.88
Pairwise Dist.	0.22	0.18	0.19	0.07	0.26	0.34	0.34	0.89	0.96	0.85	1.00	0.88
Maximum Dist.	0.34	0.25	0.31	0.19	0.39	0.40	0.40	0.94	0.94	0.88	0.88	1.00

Table 3.8: Correlation between ranking positions of various compactness measures

these results are not surprising.
Concerning distance-based measures, there are noticeable differences between the values stated in Table 3.7 on the one side and those stated in Table 3.8 on the other side. Regarding ranking positions there are high correlations of at least 0.85 between pairs of distance-based measures. The coefficients concerning absolute values are noticeably smaller, e.g., the coefficient between the Pairwise Distances and the Weighted Moment of Inertia is 0.61, whereas it is 0.85 concerning ranking positions. Nevertheless, there is a high correlation of 0.89 between the Weighted Moment of Inertia and the Weighted Pairwise Distances. The unweighted versions are correlated with a coefficient of even 0.95.

3.4.3 Summary

The presented theoretical analysis and experimental tests have confirmed that it is hard or even impossible to define a comprehensive compactness measure. Each measure has some weaknesses, for example, its results are not correlated with the visual impression or they are hard to determine or to comprehend. Nevertheless, some shape-only-dispersion measures and some shape-only-area-perimeter measures have large correlations with the Visual-Test. For example, the Reock-Test, Gibbs-Test, Haggett-Test, Schwartzberg-Test, and Cox-Test, are better in practice than they seem to be in theory. The Relative Moment of Inertia also performs very well, but it does not outperform them, although from a theoretical point of view it is close to a perfect compactness measure. Furthermore, there are some further improvements if these measures are combined.

Some measures cover other aspects such as convexity, spatial distribution of the activity, short or smooth boundaries, or total distances for a routing within a district. Hence, depending on the application, it can also be useful to apply one of these measures as part of an overall evaluation function for a solution.

3.5 Extension to Point or Line Representations

Most of the measures described in the section above were published in literature concerning political districting. Thus, it is not surprising that they are based on polygonal representations of basic areas and districts. However, in other applications basic areas can either be represented by points, e.g., in the context of sales districting, or by (poly-) lines, e.g., in the context of districting for mail delivery. This section analyzes which measures are adaptable or even directly usable for point or (poly-) line representations. For purposes of simplification, in the following the terms "point", "line" and "polygon" are used for the geometric representations as well as for the corresponding basic areas. Moreover, the term "line" is used for short, even if a polyline is meant. This section depicts four possible approaches:

1. **Direct use of measures**: In some cases a proposed measure is more or less directly usable for other representations.
2. **Adaptation of measures**: In some cases the underlying idea of a proposed measure can be adapted in order to make it usable for other representations.
3. **Definition of the districts' shapes**: The main idea of this approach is to generate a representative polygon for each district in order to apply an existing measure to this polygon afterwards.
4. **Definition of the basic areas' shapes**: The last approach uses individual basic areas instead of districts. The main idea is to generate a representative polygon for each basic area. Based on these polygons the districts' shapes are determined and existing measure are applied to them.

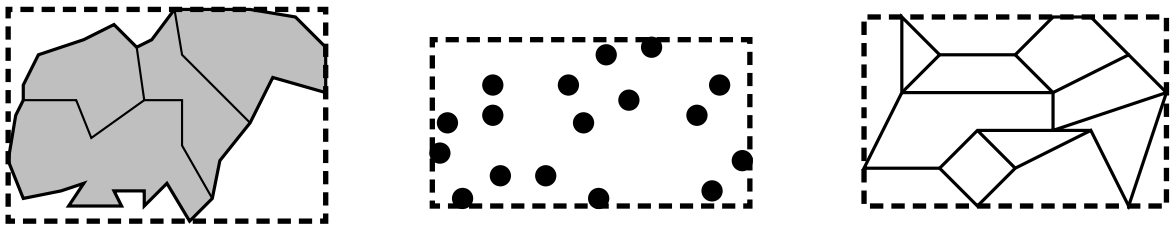
In the following these approaches are examined in more detail.

3.5.1 Direct Use of Measures

Obviously, a measure using area or perimeter of a district is not directly applicable to non-polygonal representations since no straightforward definition of shape for districts consisting of points or lines exists. Unfortunately, the majority of the presented measures use either area or perimeter. Hence, there are only a few measures directly applicable to other geometric representations.

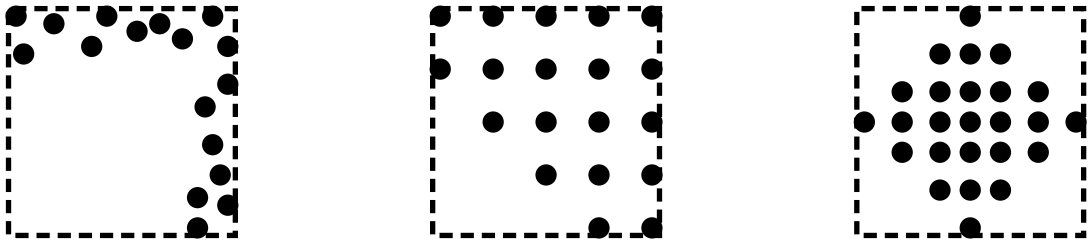
3.5.1.1 Length-Width-Test

Measuring compactness by the Length-Width-Test described in Section 3.3.1.4 is based on an enclosing rectangle of a district. More precisely, it is based on an enclosing rectangle of



(a) Polygon (b) Point (c) Line

Figure 3.16: Axis-parallel enclosing rectangle for different representations



(a) Visually non-compact (b) Visually more compact (c) Visually more compact

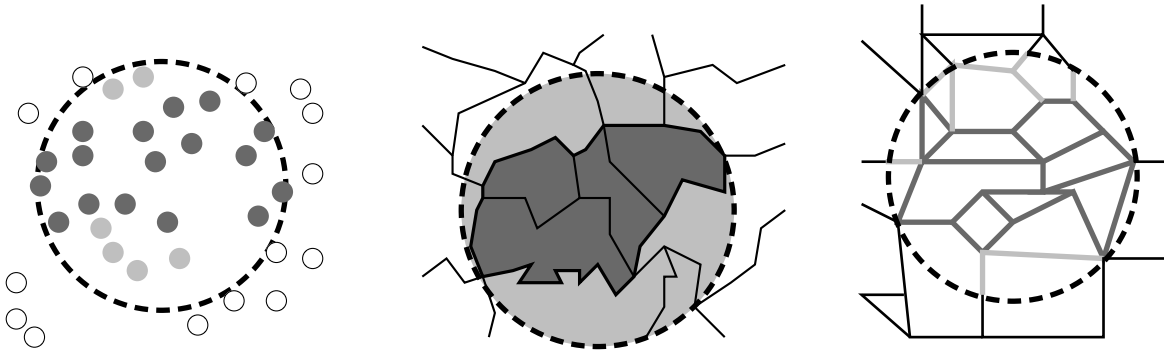
Figure 3.17: Different districts having identical enclosing rectangles

the geometric representations of the basic areas defining this district. Hence, this measure can directly be applied to all regarded kind of representations as Figure 3.16 illustrates.

However, the main drawback is that many districts ranging from visually less compact to visually highly compact have identical enclosing rectangles. Figure 3.17 shows an example where the district depicted in Figure 3.17a is visually less compact than those depicted in Figure 3.17b and Figure 3.17c. The decision which of the latter is more compact is not as clear. Nevertheless, the Length-Width-Test evaluates all of them as perfectly compact since all of their axis-parallel enclosing rectangles are quadratic.

3.5.1.2 Distance-Based Measures

Distance-based measures have in common that they determine a district's compactness by using distances between pairs of basic areas or between basic areas and specified points, as described in detail in Section 3.3.5. Most commonly, the distance between two basic areas is defined as distance between two representative points, one for each basic area. Thus, these measures are directly applicable to points. Note that this simplification of polygons to points can be interpreted as an inversion of the fourth approach. The same idea can be applied to a line, for example, by using the middle-point as representative point.



(a) Point (b) Polygon (c) Line

Figure 3.18: Hofeller-Grofman-Test for different representations of basic areas

3.5.1.3 Normalized Moment of Inertia

The same argumentation as for distance-based measures holds for the Normalized Moment of Inertia explained in Section 3.3.6.2.

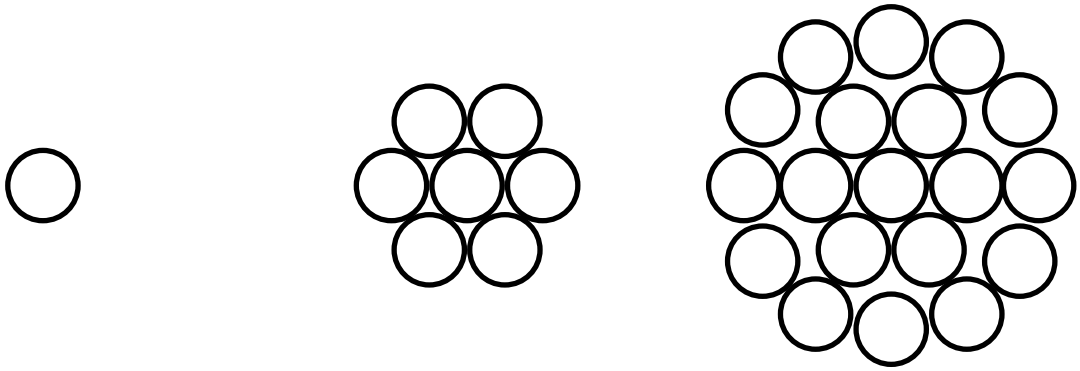
3.5.1.4 Hofeller-Grofman-Test

The Hofeller-Grofman-Test presented in Section 3.3.6.1 computes the ratio between the activities of the evaluated district and of an enclosing figure. This figure can be, for example, the convex hull or the smallest enclosing circle and is readily computable for points or lines. However, the total activity within this figure also has to be computed. This is actually easier for points than for polygons or lines since a point is either located within a figure or not.

Figure 3.18a shows an example: The dark gray points define a district. The light gray points are located within its enclosing circle, whereas the white points are located outside. A line as well as a polygon may only be partly located within an enclosing figure. Hence, the test has to determine the ratio of its activity that is located within this figure. In Figure 3.18c (3.18b) the dark gray polygons (lines) define a district. Parts of other basic areas located within its enclosing circle are colored light gray, whereas parts located outside are colored white (black). Obviously, some basic areas have parts located within this circle as well as outside.

3.5.2 Adaptation of Measures

Each proposed compactness measure has an underlying idea how compactness can be measured. Thus, this idea can be utilized in order to develop a measure based on point representations, even if the proposed measure is based on polygonal representations.



(a) 1 circle (b) 7 circles arranged in 1 sphere (c) 19 circles arranged in 2 spheres

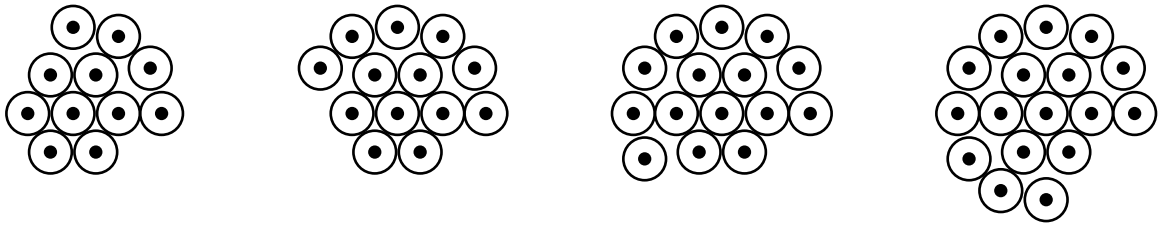
Figure 3.19: Arrangements of circles in spheres

3.5.2.1 Adapted Relative Moment of Inertia

The Relative Moment of Inertia described in Section 3.3.1.6 determines the second moment of inertia of a district about its center of mass divided by the second moment of inertia of a circle having the same area. The idea behind this approach is that a circle is the most compact figure for a given area. In order to adapt this measure on point representations a definition of the most compact spatial distribution of a given number of points is necessary: A set of points seems to be compact if it is embedded in an enclosing circle, i.e., the shape looks like a circle. Moreover, a uniform spatial distribution of the points is an indicator for compactness.

The following idea for using this observation in order to develop a measure has already been presented in the Bachelor thesis of Marquardt [28]. Let n_g be the number of given points. The idea for obtaining a uniform spatial distribution comprises the location of n_g non-overlapping equally sized circles. The middle-points of these circles define a compact spatial distribution of n_g points. This definition yields the questions of how these circles can be arranged and of how the size of these circles can be defined. Concerning the first question, the idea is to arrange the circles in spheres around one circle having its middle-point in the origin. The middle-points of all circles located in the same sphere have the same distance to the origin. The number of circles that can be arranged non-overlapping in the l-th sphere is $6 \cdot l$, i.e., 6 circles can be arranged in the first sphere, 12 in the second sphere, and so on. Hence, the number of circles that can be arranged non-overlapping in the origin and in s spheres in this way results in $3 \cdot (s+1)^2 - 3 \cdot (s+1) + 1$. That means that 7 circles can be arranged in 1 sphere, 19 circles in 2 spheres, and so on. Figure 3.19 illustrates the corresponding arrangements.

Since the number of points is prescribed, the number of required spheres s_g^* is given by means



(a) 11 circle (b) 13 circles (c) 15 circles (d) 17 circles

Figure 3.20: Arrangements of circles in 2 spheres

of the inequation

$$3 \cdot (s_g^* + 1)^2 - 3 \cdot (s_g^* + 1) + 1 \geq n_g > 3 \cdot s_g^{*2} - 3 \cdot s_g^* + 1 \,.$$

Thus, the sphere $s_g(j)$ of the j-th circle is given by

$$3 \cdot (s_g(j) + 1)^2 - 3 \cdot (s_g(j) + 1) + 1 \geq j > 3 \cdot s_g(j)^2 - 3 \cdot s_g(j) + 1 \,.$$

Let di_g be the diameter of the circles, then the distance between the origin and the middle-point of the j-th circle is $di_g \cdot s_g(j)$. Moreover, its position $pos_g(j)$ in the $s_g(j)$-th sphere is

$$pos_g(j) := j - 3 \cdot s_g(j)^2 - 3 \cdot s_g(j) + 1 \,.$$

The first middle-point of each sphere is located on the positive x-axis. Each further point has an angle of

$$\alpha(j) := (pos_g(j) - 1) \cdot \frac{\pi}{3 \cdot s_g(j)}$$

according to the x-axis. Figure 3.20 presents some arrangements exemplarily.

Finally, the question of how to define the diameter di_g of the circles is left. The Relative Moment of Inertia based on polygons compares the district's second moment of inertia with that of a circle having the same area. Now, the idea is to compare the district's Moment of Inertia according to Equation (3.13), with the Moment of Inertia of a compact rearrangement having the same Pairwise Distances (cf. Equation (3.15)). By requiring equal Pairwise Distances, di_g can be computed. Let D_g^* be the rearrangement of the points defining district D_g, then the adapted Relative Moment of Inertia results in

$$comp_{armoi}(D_g) := \frac{comp_{moi}(D_g^*)}{comp_{moi}(D_g)} \,.$$

Note that D_g^* does not necessarily induce a lower bound of $comp_{moi}(\cdot)$ for arrangements of n_g points having the same Pairwise Distances. For example, Figure 3.19c shows an arrangement of 19 circles having open spaces between these circles. Hence, there might be an arrangement where 20 circles are located within the same area. Arrangements having smaller open spaces might be better according to the Moment of Inertia. However, the corresponding middle-points are not distributed uniformly, and, hence, most likely visually rather compact. Consequently, the results of the adapted Relative Moment of Inertia are not limited to be between 0 and 1.

As the previous subsection has shown, a few measures can be applied directly or adapted to non-polygonal representations. However, there remain many measures where other approaches are necessary in order to make them applicable for non-polygonal representation. Since many proposed measures take a district's shape into acoount in some way, an obvious approach is the definition of districts' shapes for non-polygonal representations. In the following, some possible approaches how this can be done are presented and compareed.

3.5.3 Definition of the Districts' Shapes

The following approaches have in common that they define a shape for each district, without taking its neighboring districts into account. Hence, there can be intersections between the shapes of different districts as well as open spaces on the regarded overall area. Thus, the Perimeter-Test and the Bozkaya-Test are not applicable since they use lengths of the common districts' boundaries.

A straightforward approach to define a district's shape is the usage of an enclosing figure such as a circle, a rectangle or the convex hull. However, one should have in mind that the districts' shapes are generated in order to apply existing compactness measures to them. If the shape is specified in advance too much, the result of the subsequent compactness evaluation is already more or less predefined, and, hence, the corresponding evaluation contradicts the visual impression.

3.5.3.1 Enclosing Circle

Using enclosing circles is not recommendable since in this case almost every measure evaluates every district as perfectly compact. Note that the Taylor-Test considers the interior angles, and, hence, strictly spoken, it is not applicable to a circle. However, if the circle is approximated by a regular polygon, the Taylor-Test is applicable and also evaluates this polygon as perfectly compact.

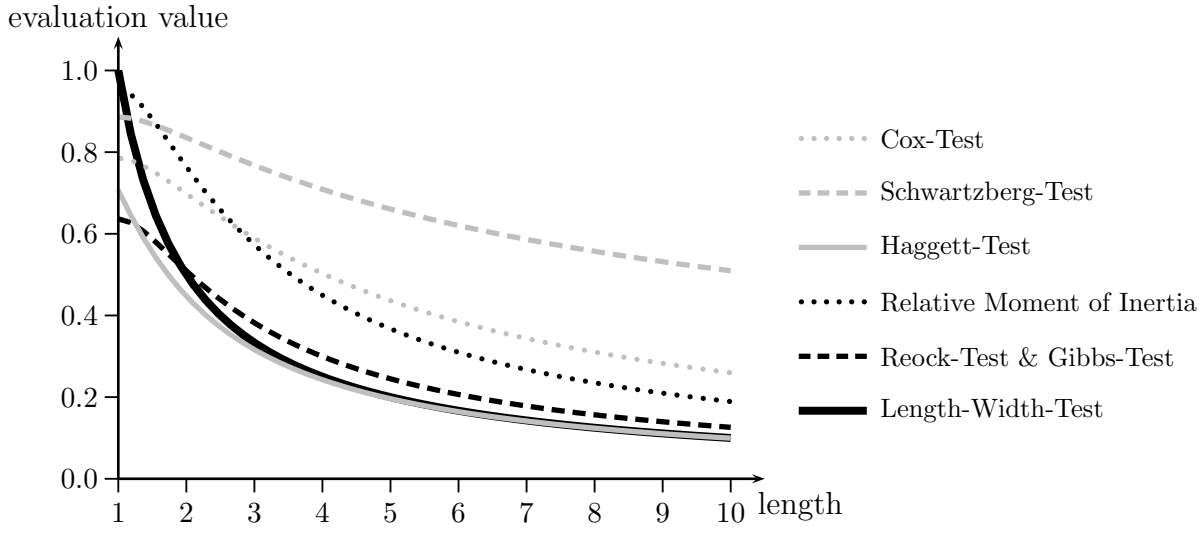


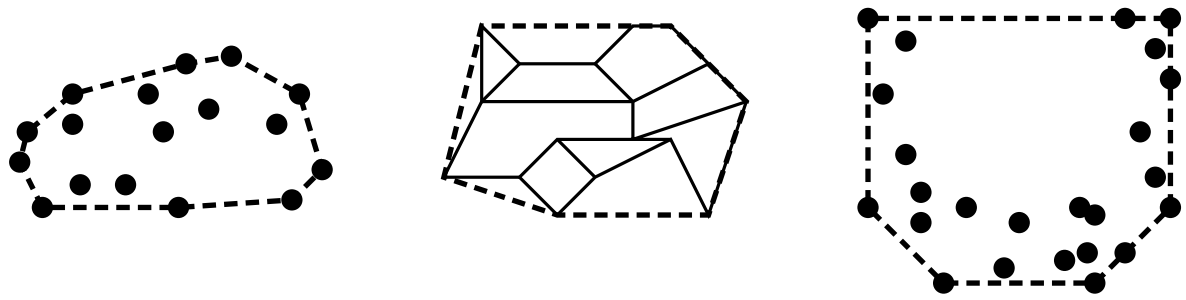
Figure 3.21: Different compactness measures applied to rectangles having a width of 1

3.5.3.2 Enclosing Rectangle

Let $er(D_g)$ be the enclosing rectangle of district D_g. Assume without loss of generality that its length is greater than or equal to its width. In this case, applying selected compactness measures to such a rectangle result in the following equations:

- Reock-Test: $comp_{reock}(er(D_g)) = \frac{area(er(D_g))}{area(sec(er(D_g)))} = \frac{le(er(D_g))\cdot wi(er(D_g))}{0.25\cdot\pi\cdot(le(er(D_g))^2+wi(er(D_g))^2)}$
- Gibbs-Test: $comp_{gibbs}(er(D_g)) = \frac{area(er(D_g))}{area(cla(er(D_g)))} = \frac{le(er(D_g))\cdot wi(er(D_g))}{0.25\cdot\pi\cdot(le(er(D_g))^2+wi(er(D_g))^2)}$
- Haggett-Test: $comp_{haggett}(er(D_g)) = \frac{radius(lic(er(D_g)))}{radius(sec(er(D_g)))} = \frac{wi(er(D_g))}{\sqrt{le(er(D_g))^2+wi(er(D_g))^2}}$
- Relative Moment of Inertia:
  $comp_{rmoi-inv}(er(D_g)) = \frac{\frac{area(D_g)^2}{2\cdot\pi}}{\int_{er(D_g)}\int_{er(D_g)}(x^2+y^2)\mathrm{d}x\mathrm{d}y} = \frac{6\cdot le(er(D_g))\cdot wi(er(D_g))}{\pi\cdot(le(er(D_g))^2+wi(er(D_g))^2)}$
- Schwartzberg-Test:
  $comp_{schwartzberg-inv}(er(D_g)) = \frac{2\cdot\sqrt{\pi\cdot area(er(D_g))}}{per(er(D_g))} = \frac{\sqrt{\pi\cdot le(er(D_g))\cdot wi(er(D_g))}}{le(er(D_g))+wi(er(D_g))}$
- Cox-Test: $comp_{cox}(er(D_g)) = \frac{4\cdot\pi\cdot area(er(D_g))}{per(er(D_g))^2} = \frac{\pi\cdot le(er(D_g))\cdot wi(er(D_g))}{(le(er(D_g))+wi(er(D_g)))^2}$

Figure 3.21 plots the evaluation values for setting the rectangle's width without loss of generality to 1 and varying its length for each of them. In addition, Figure 3.21 plots the evaluation values for the Length-Width-Test. Obviously, the presented measures have in common that the best evaluated rectangle is a square. Moreover, their evaluation values decrease strictly monotonically when increasing the length. This means that for a set of districts these measures differ in their evaluation values for each district, but not in their ranking of them. Hence, no test will give more informative results than those obtained by



(a) Of a set of points (b) Of a set of lines (c) Of a U-shaped set of points

Figure 3.22: Illustration of convex hulls

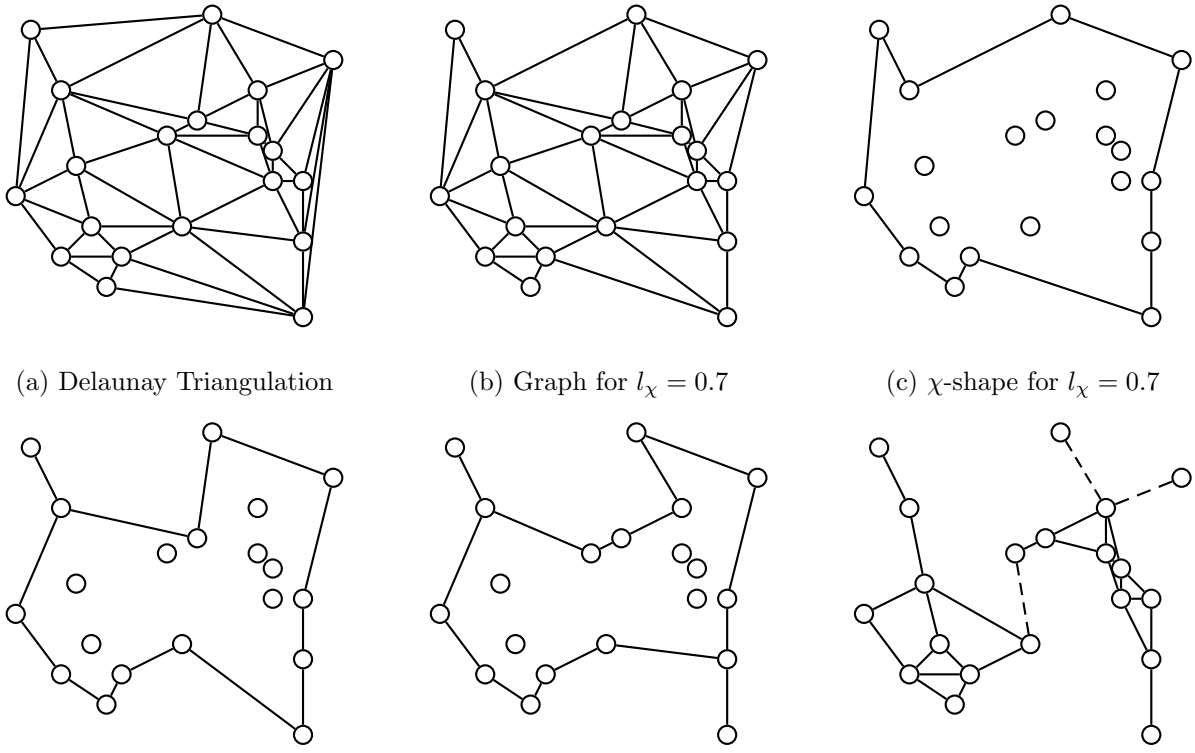
the Length-Width-Test. In addition to the drawbacks of the Length-Width-Test, other tests have the further drawbacks that their evaluation values are not normalized to be between 0 and 1. Moreover, applying convexity measures to an enclosing rectangle is not suitable since each rectangle is convex, and thus perfectly compact according to these measures.

3.5.3.3 Convex Hull

A third approach uses the convex hull as enclosing figure. Figure 3.22a (3.22b) illustrates the convex hull for basic areas represented by points (lines). Kalcsics et al. [24] use this approach to validate whether the districts of a solution are overlapping or not. The exact shapes of convex hulls are not specified in advance as much as shapes of circles or rectangles. Nevertheless, convexity is often an indicator for compactness, some measures even define a convex shape as perfectly compact (cf. Section 3.3.4). Moreover, the convex hull is only a rough approximation. Take a look on the district depicted in Figure 3.22c. The visual impression is that this district is shaped similarly to the letter 'U', and, hence, rather non-compact. However, its convex hull looks significantly more compact. Most measures evaluate this district, and, hence, the corresponding point set, as rather compact.

In general, convex hulls have smooth boundaries and their perimeter lengths are relatively small compared to perimeters of shapes which coincide more with the visual impression. Thus, the Schwartzberg-Test, the Cox-Test or the Boyce-Clark-Test on convex hulls has some weaknesses. Since convex hulls have no indentations, the largest inscribed circle is comparatively large. Hence, applying the Haggett-Test to the convex hull of a point (line) set mostly yields in a good evaluation, even if this point (line) set is visually non-compact. While the perimeter achieved by defining a district's shape in this way is rather too small, the achieved area is generally too large. So, the Reock-Test and the Gibbs-Test evaluate a district as more compact than it seems to be.

Nevertheless, the concept of convex hulls is easy to understand and in many cases the convex



(a) Delaunay Triangulation (b) Graph for $l_\chi = 0.7$ (c) χ-shape for $l_\chi = 0.7$

(d) χ-shape for $l_\chi = 0.6$ (e) χ-shape for $l_\chi = 0.5$ (f) Graph for $l_\chi = 0.3$

Figure 3.23: Illustration of the computation of χ-shapes

hull of a district is close to the visual impression of how the shape of this district looks like. The computation of a convex hull can be done in $\mathcal{O}(n \cdot \log n)$, for example, by Graham-Scan [16] or in $\mathcal{O}(n \cdot \log k)$ by Chans-Algorithm [8], where n is the number of points and k is the number of points on the convex hull. Hence, this approach can be helpful in order to obtain a first impression of how compact a district is.

3.5.3.4 χ-Shapes

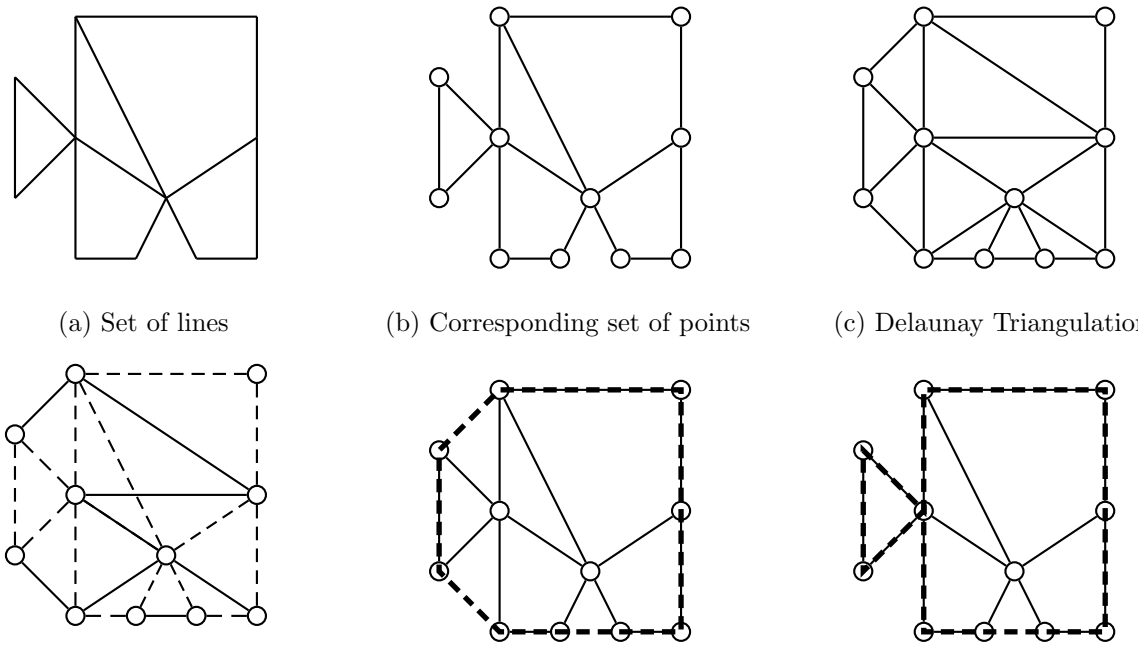
In order to overcome the described drawbacks of convex hulls and to obtain more accurate shapes of the districts the concept of χ-shapes can be used. A χ-shape is a non-convex polygon that describes the shape of a set of points. Duckham et al. [10] present the following algorithm to compute them: At first, the Delaunay Triangulation of the set of points is determined. An example is illustrated in Figure 3.23a. In a Delaunay Triangulation there is no point of the regarded point set that is inside the circumcircle of any other triangle of this Triangulation. Another property is the fact that the Delaunay Triangulation corresponds to the dual graph of the Voronoi Diagram. For more properties and details see, for example, Aurenhammer et al. [1]. Afterwards, this algorithm normalizes the lengths of the triangulations' edges such that the normalized length of the longest edge becomes 1. The next step

depends on a length parameter l_χ between 0 and 1. Each edge having a normalized length greater than l_χ is removed, unless when the corresponding graph becomes disconnected by removing it. Figure 3.23b shows the resulting graph for setting l_χ=0.7. Finally, the outer edges of the obtained graph define the χ-shape. Figure 3.23c illustrates the χ-shape for l_χ=0.7. The complexity of this algorithm is $\mathcal{O}(n \cdot \log n)$.

Obviously, the achieved shape depends on the length parameter. For l_χ=1 the χ-shape coincides with the convex hull. The smaller l_χ, the more indentations has the obtained boundary. Figuratively spoken, by decreasing the length parameter "one lets the air out of the convex hull". Figure 3.23d (3.23e) depicts the obtained χ-shapes for setting l_χ to 0.6 (0.5). Figure 3.23f shows an example where the graph would be disconnected if all edges having a normalized length greater than 0.3 would be removed. Hence, the dashed edges remain in the graph.

In contrast to other approaches such as $\alpha-$shapes [11], χ-shapes have no holes. However, as the χ-shapes presented in Figure 3.23 demonstrate, there can be single points on a χ-shape's boundary which are connected with only one other point. For well-chosen length parameters the visual impression of how the shape of a district looks like comes close to the obtained χ-shape. However, the main difficulty is the choice of this parameter. Moreover, this choice also affects the result of the compactness measure that is applied to the resulting χ-shape. Therefore, area and perimeter depend on this choice as follows: The smaller l_χ is, the smaller the area of the χ-shape and the larger the perimeter of the χ-shape.

In contrast to convex hulls, for different point sets the obtained χ-shapes are more likely not to be identical. Furthermore, a χ-shape can have indentations, and each of them increases the perimeter length compared to the convex hull. Hence, χ-shapes are able to detect indentations or open spaces at the outer area of the point set. Thus, applying the Schwartzberg-Test or Cox-Tests to χ-shapes is more suitable than applying it to the shapes obtained by using one of the approaches discussed earlier. The same holds for the Boyce-Clark-Test since the determined outer χ-shapes' boundaries are not as smooth as for convex hulls. However, χ-shapes have no holes by construction that means that open spaces in the interior of the point set are not represented by them. Thus, the approximation of the largest inscribed circle is maybe inaccurate. Hence, the applicability of the Haggett-Test still has some weaknesses. However, due to the detection of indentations they are smaller than for applying the Haggett-Test to convex hulls. The convex hull is only a rough approximation of a district's shape having no indentations. Thus, the center of mass is often located centrally and a district's area fills out large parts of an enclosing circle. Thus, the Relative Moment of Inertia, the Reock-Test and the Gibbs-Test often evaluate shapes approximated in this way



(a) Set of lines (b) Corresponding set of points (c) Delaunay Triangulation

(d) Corresponding graph including the underlying set of lines (e) χ-shape for $l_\chi = 0.7$ (f) χ-shape for $l_\chi = 0.3$

Figure 3.24: Illustration of the computation of a χ-shape for a set of lines

as compact, even if the underlying set of points is visually non-compact. Applying these measures to χ-shapes is also more suitable since the corresponding approximation of the districts' shapes is more exact.

In summary, there are some reasons for defining the shape of a point set by a χ-shape. However, a computed shape, and, hence, also the result of a compactness evaluation of this shape, highly depends on the choice of the length parameter. In the case of convexity measures this dependency is especially obvious since by reducing the length parameter the non-convexity of a shape increases.

The described construction of χ-shapes is based on point sets. We can also extend this idea in order to construct χ-shapes on sets of lines. At first, for a given set of lines, this extension infers a set of points by looking at the start and end points of each line. Figure 3.24b depicts the obtained set of points for the set of lines shown in Figure 3.24a. Now, it determines the Delaunay Triangulation for this set of points as before. The resulting graph is depicted in Figure 3.24c. The next step incorporates the prescribed set of lines. For each line (segment) it adds an edge between its start- and end-point to this graph and marks it. Figure 3.24d illustrates these edges as dashed lines. Thus, the graph obtained so far consists of two sets of edges: The first set is defined by the prescribed set of lines, the second set is defined by the Delaunay Triangulation. Afterwards, the lengths of the edges of the second set are

normalized as before. Hence, the removal of edges depending on l_χ concerns only the edges of this set. The edges of the first set remain in the graph in any case. Again, each edge of the first set is removed if its normalized length is greater than l_χ and if the graph stays connected after removing it. Finally, the outer edges of the graph achieved after the removing step define the χ-shape of the set of lines. For the presented example, Figure 3.24e shows exemplarily the χ-shape for setting l_χ to 0.7, while Figure 3.24f shows the χ-shape for setting l_χ to 0.3.

3.5.3.5 Summary

There are different approaches for defining an approximated shape based on a set of points or lines. Our aim is to apply a compactness measure to the resulting shape, therefore, the shape should coincide with the visual impression of how this shape looks like. Hence, the usage of prescribed figures such as circles or rectangles is not suitable. Also the usage of convex hulls has some drawbacks, mainly the prescribed restriction on convex shapes. Thus, the usage of more flexible approaches such as χ-shapes is recommendable. Table 3.9 summarizes the theoretical results of the (sub-)sections above, where the applicability is evaluated from good to worse by '++', '+', '0', '-', whereas '↯' denotes that the approach is not applicable at all. This table only lists measures that are not directly applicable to point sets. Section 3.5.5 will present results for using these measures in practice.

measure	enclosing circle	enclosing rectangle	convex hull	χ-shape
Reock-Test	↯ (result always 1)	- (Length-Width-Test)	+	++
Gibbs-Test	↯ (result always 1)	- (Length-Width-Test)	+	++
Haggett-Test	↯ (result always 1)	- (Length-Width-Test)	0	+
Boyce-Clark-Test	↯ (result always 1)	- (Length-Width-Test)	0	++
Relative Moment of Inertia	↯ (result always 1)	- (Length-Width-Test)	0	++
Schwartzberg-Test	↯ (result always 1)	- (Length-Width-Test)	0	++
Cox-Test	↯ (result always 1)	- (Length-Width-Test)	0	++
Perimeter-Test	not applicable			
Bozkaya-Test	not applicable			
Taylor-Test (corrected)	not applicable	↯ (result always 1)	↯ (result always 1)	0
Bizarreness-Test	↯ (result always 1)	↯ (result always 1)	↯ (result always 1)	0

Table 3.9: Compactness measures applied to different definitions of districts' shapes

(a) Enclosing rectangle as overall area

(b) Convex hull as overall area

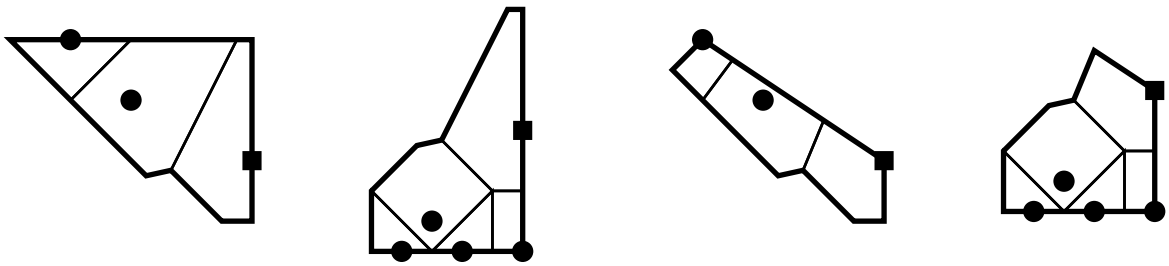
Figure 3.25: Voronoi region for a set of basic area

3.5.4 Definition of the Basic Areas' Shapes

In contrast to the approaches described in the previous section the following approaches treat each basic area independently of its assignment to prescribed or computed districts. Hence, its shape can be determined before a districting algorithm is executed. After computing a polygonal shape for each point or line, each measure based on polygonal representations can be applied. The approaches presented in the following have in common that the complete overall area is taken into account for computing the basic areas' shapes. Each point within this overall area is assigned to the area of exactly one basic area that means that there are no intersections between shapes of different basic areas and no open spaces on the overall area. Hence, the Perimeter-Test and the Bozkaya-Test are applicable in this case.

3.5.4.1 Voronoi Regions

The first approach is based on *Voronoi Diagrams*. For a given set of generator points a Voronoi Diagram partitions an area into so-called *Voronoi regions* such that each region contains all points that are closer to the corresponding generator than to any other generator. For more details about Voronoi regions see Section 5.1. Here, each basic area is defined as a generator. Figuratively spoken, each point of the regarded overall area is assigned to its closest basic area. The approach uses the smallest axis-parallel enclosing rectangle or the convex hull of the set of basic areas as the overall area. Figure 3.25a depicts an example for the former, whereas Figure 3.25b illustrates an example for the latter considering the same basic areas. The main problem of this approach is that the obtained shape for a basic area depends on the locations of its neighboring basic areas and the boundary of the overall area. In regions having a high density of points the obtained polygons are noticeably smaller than in regions having a small density of points. For example, if the points correspond to customer locations, most likely the obtained polygons in rural areas are noticeably larger than in urban regions. Although having in mind that a compactness measure should be



(a) D_1 (encl. rectangle) (b) D_2 (encl. rectangle) (c) D_1 (convex hull) (d) D_2 (convex hull)

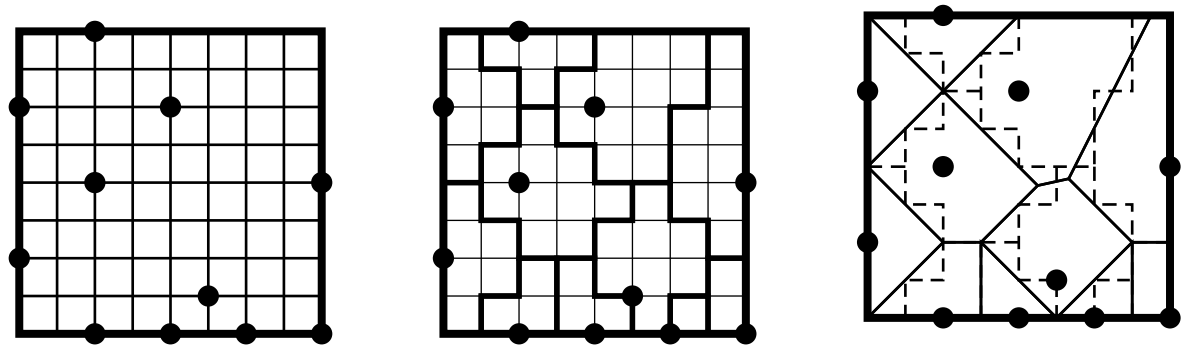
Figure 3.26: Districts varying on the assignment of a basic area and on the overall area

independent of scale, in the context of sales districting the size of an obtained polygon can be interpreted as a proxy for the travel time that is necessary to reach the corresponding customer from a neighbored customer.

For a polaygon, it is not only the size that depends on the neighboring basic areas, but also its boundary line and its number of the vertices. Furthermore, the definition of the overall area highly influences the shape of the outer basic areas. For example, the shapes of the dark gray (light gray) polygons in Figure 3.25a and 3.25b differ noticeably. Especially, the shape of an outer basic area influences the shape of its district, and, hence, it also influences the result of a compactness measure applied to this district. Figure 3.26 shows an example based on the basic areas introduced in Figure 3.25. Take a closer look on the light gray polygon located in the north-west. Figure 3.26 illustrates its corresponding point as square and regards two possible assignments of this basic area to a district: Figure 3.26a and Figure 3.26b illustrate these assignments for the case that the overall area is defined as an enclosing rectangle, whereas Figure 3.26c and Figure 3.26d illustrate them for the case that the overall area is defined as a convex hull. Table 3.10 states the results for applying selected compactness measures to these districts.

measure	enclosing rectangle		convex hull	
	District D_1	District D_2	District D_1	District D_2
Reock-Test (circle)	0.36	0.34	0.27	0.60
Reock-Test (convex)	0.97	0.90	0.95	0.96
Gibbs-Test	0.36	0.34	0.27	0.60
Haggett-Test	0.40	0.37	0.23	0.54
Schwartzberg-Test (recipr.)	0.77	0.74	0.70	0.86
Cox-Test	0.59	0.55	0.49	0.74

Table 3.10: Compactness measures applied to the districts depicted in Figure 3.26



(a) Enclosing rectangle and grid (b) Obtained polygons (c) Grid regions and Voronoi regions

Figure 3.27: Approach of Lei et al. [25]

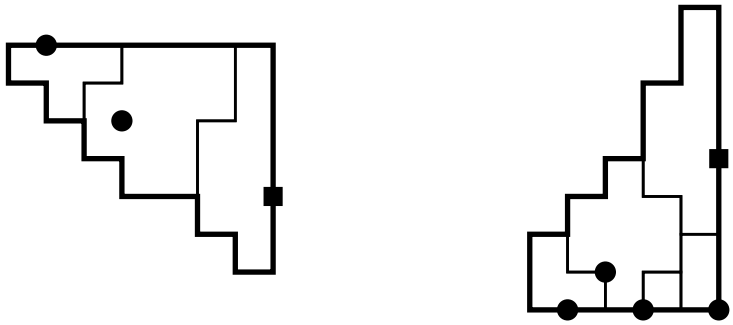
Obviously, the district depicted in Figure 3.26a is evaluated as more compact than the one depicted in Figure 3.26b. In contrast to this, these measures evaluate the district illustrated in Figure 3.26d as more compact than the district illustrated in Figure 3.26c. Note that the original set of points defining the districts in Figure 3.26a (3.26b) and Figure 3.26c (3.26d) are identical. Hence, depending on the definition of the overall area the ranking of a set of districts can differ.

In summary, each compactness measure based on polygons is applicable to basic areas and districts obtained by using Voronoi regions, but the results are influenced by the definition of the overall area and by the spatial distribution of the points. Hence, some results may not coincide with the visual impression.

3.5.4.2 Grid Regions

Lei et al. [25] propose a similar approach. At first, they define the overall area as the smallest axis-parallel enclosing rectangle. Then, they compute d as the minimum of the smallest positive distance between two points in x-direction and y-direction. Afterwards, they partition the enclosing rectangle into quadratic cells having length d. Figure 3.27a exemplarily depicts such a grid. If no points have the same x-value or y-value, there is at most one point located in each cell. Each cell having no point is merged with its closest cell having a point. Details on how to handle the case of two or more cells having the same distance are not given. Figure 3.27b illustrates the obtained polygons of the basic areas introduced in Figure 3.27a. The authors define basic areas and districts in this way and apply the Bozkaya-Test to solutions of their districting algorithm.

Figure 3.27c compares the boundaries obtained by this approach (dashed lines) to Voronoi regions (solid lines). Obviously, the obtained boundaries are mostly larger and non-smoother than those obtained by using Voronoi regions. Each boundary of a district is a subset of



(a) District 1 (grid) (b) District 2 (grid)

Figure 3.28: Shapes of districts obtained by grid regions

these boundaries. Finally, Figure 3.28 shows the grid regions for the examples depicted in Figure 3.10. Applying the Schwartzberg-Test results in 0.68 for district 1 and 0.65 for district 2. As expected, these results are noticeably worse compared to the results stated in Table 3.10, namely 0.77 for district 1 and 0.74 for district 2 since the boundaries are noticeably longer.

Again, each compactness measure based on polygonal representations is applicable to the polygons achieved in this way. However, the non-smooth and long boundaries may influence the results of the applied measures. Hence, this approach has the same disadvantage as the approach before, namely that some results may not coincide with the visual impression.

3.5.5 Evaluations

Finally, this subsection assess the presented approaches of measuring compactness of districts where the basic areas are represented by points. It is based on the Bachelor thesis of Marquardt [28] with the addition of some more measures and extra detail. We have conducted a survey where the participants should evaluate 30 districts depicted in Figure 3.29 by means of German school marks. A total of 170 participants have participated in the survey. Table 3.11 shows the average marks for each district.

district	mark	district	mark	district	mark	district	mark	district	mark
1	3.7	7	4.2	13	3.9	19	4.0	25	4.3
2	3.8	8	3.6	14	3.2	20	3.5	26	2.1
3	3.2	9	1.9	15	2.9	21	3.0	27	3.1
4	1.6	10	3.6	16	3.3	22	2.9	28	2.7
5	2.8	11	3.4	17	2.7	23	4.2	29	3.5
6	3.9	12	4.5	18	2.8	24	4.1	30	3.0

Table 3.11: Results of the Visual-Test applied to the districts depicted in Figure 3.29

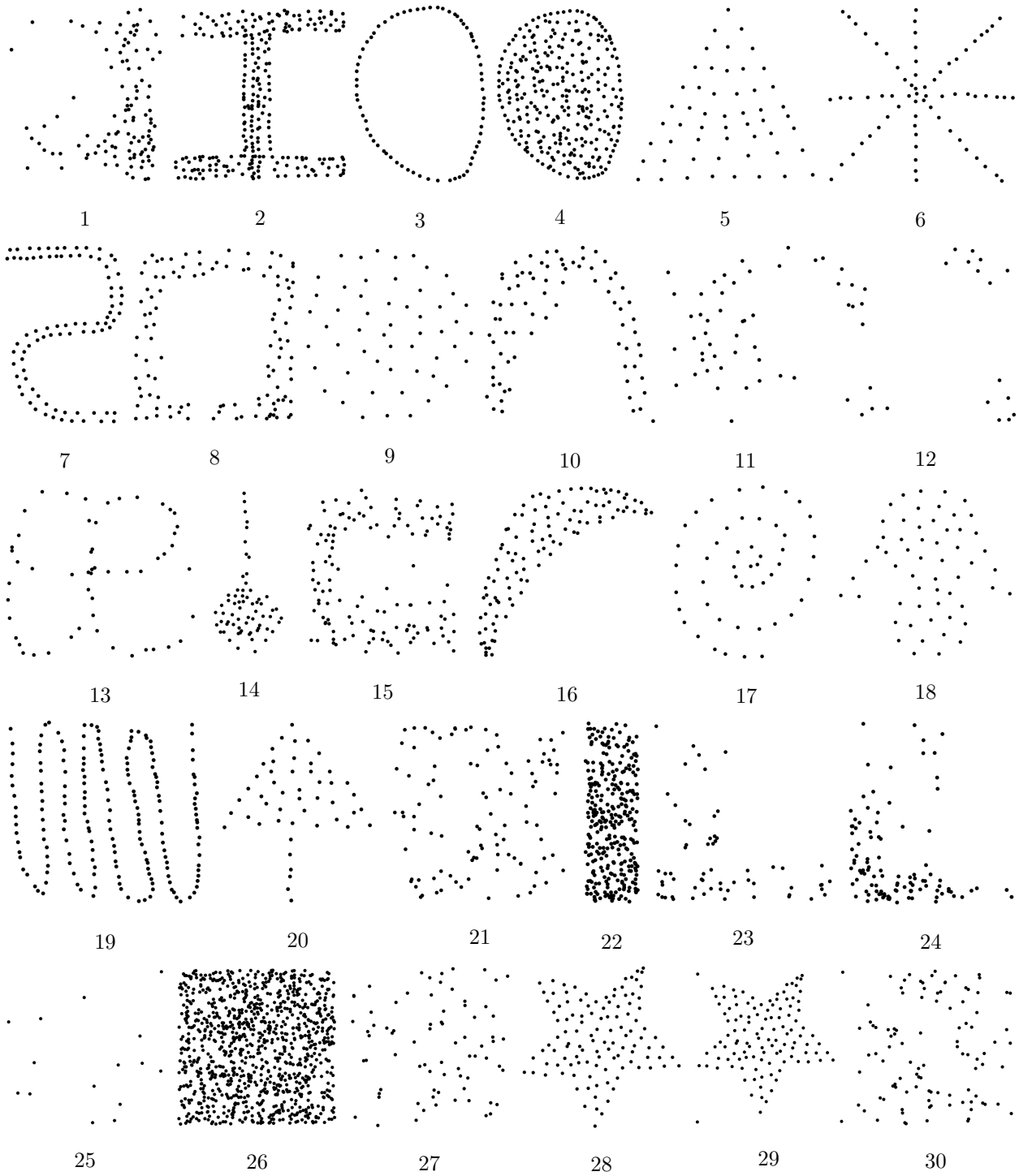


Figure 3.29: Districts used for the presented Visual-Test

Look at some of the results in more detail. Between district 3 and 4, the latter is evaluated noticeably better than the former, with an average mark of 1.6 compared to 3.2. Also district 8 is evaluated as rather non-compact with an average mark of 3.6. Hence, for the survey's participants a large open space in the interior of a set of points seems to be an indicator for non-compactness, even if the corresponding outer boundary is nearly squared.
Table 3.11 also shows an average mark of 2.7 for district 17, whereas district 9 has an average mark of 1.9. The first one looks a bit like a snake, whereas the points of the second one are distributed more or less uniformly. Hence, for the participants uniformly distributed points look more compact than non-uniformly distributed points.
However, the stated results also show that the higher the density of the points is, the higher the visual impression of compactness. For example, the points in district 26 are denser than those in district 27 and district 30. Even the points in district 22 are denser than those of them. In both cases the denser districts have received higher marks in the Visual-Test.

In order to summarize the results of this Visual-Test we can conclude:

> "*A district is compact if its points are distributed uniformly with a high density within a squared or circular hull and if there is no open space within this hull.*"

Unfortunately, distance-based measures add up the distances between all pairs of basic areas or between the basic areas and specified points, or they use only the maximum distance between a pair of basic areas. Hence, for the former the evaluation deteriorates if the number of points increases. For example, district 26 is well evaluated by the Visual-Test; however in terms of Pairwise Distances its evaluation is very poor. For the latter the distribution of the points does not matter at all. In order to overcome the first problem, measures that are independent of the number of points can be formulated by using the average Pairwise Distance

$$comp_{apd}(D_g) := \frac{1}{|B_g|^2} \cdot \sum_{i \in B_g} \sum_{j \in B_g} d_{i,j} \,,$$

or the average squared distance to the center of gravity

$$comp_{amoi}(D_g) := \frac{1}{|B_g|} \cdot \sum_{i \in B_g} d^2(b_i, cen_g) \,,$$

respectively.

Table 3.12 summarizes the correlation coefficients between different distance-based measures and the Visual-Test. Recall that for distance-based measures a coefficient of 1 indicates total correlation. The table shows that the adapted measures using average distances outperform

the original versions. Again, distance-based measures and the Visual-Test are not correlated; unfortunately, they are sometimes even negatively correlated. However, in general distance-based measures are not applied in order to compare single districts but to compare different solutions, i.e., they are used as global compactness measures. Hence, some of the weaknesses pointed out in this analysis do not occur in this case.

	Pairwise Distances	Average Pairwise Distances	Moment of Inertia	Average Moment of Inertia	Maximum Distance	Adapted Relative Moment of Inertia
absolute values	−0.34	0.23	−0.35	0.24	−0.25	−0.31
ranking values	−0.26	0.18	−0.21	0.17	−0.11	0.46

Table 3.12: Correlation between compactness measures and the Visual-Test

Table 3.12 also states the correlation between the Visual-Test and the adapted Relative Moment of Inertia introduced in Section 3.5.2.1. Here, concerning the absolute values a correlation of 1 indicates a total correlation. Hence, this measure outperforms the distance-based measures. However, a correlation coefficient of 0.46 does not really indicate a high correlation.

The approaches presented in Section 3.5.3 focus on the outer boundary. According to their definitions no holes within the determined shapes are possible. Hence, different districts are equally evaluated if their outer basic areas are identical, independently of the distribution of their interior basic areas. For example, the shapes of district 3 and district 4 are best approximated as an ellipse. Actually, both districts evaluate to nearly equal results, although all points of district 3 are located on the boundary of this elliptical shape, whereas the points of district 4 are located all over this elliptical shape. The nearly equal evaluation of them contradicts the visual impression. Thus, as expected the correlation coefficients stated in Tables 3.13 to 3.16 are not very high. Note that for absolute values again -1 indicates a total correlation. Therefore, it may make sense to apply shape approximations allowing holes, for example α-shapes. However, it is even desirable, for point representations open spaces within the overall area can occur, for example, if no customer is located within a region. In general for polygonal representations no open space within the overall area exists. Hence, the open space within district 3 can be caused by the non-existence of points within this region as well. In this case, it is not really a fault if the evaluation values of district 4 and district 3 are nearly equal, it is even desirable. Moreover, the density of the points within a district may also be caused by the spatial distribution of the prescribed set of points. However, without information about an entire solution a measure is not able to detect if a

low density or an open space within a district is given externally or achieved as result of a districting approach.

Now, the different approaches of defining districts' shapes are examined in more detail, namely the smallest enclosing axis-parallel rectangle, the convex hull and different χ-shapes differing in the setting of l_χ (cf. Section 3.5.3). Table 3.13 states the correlation coefficients between the Visual-Test and the original Reock-Test (cf. Equation (3.1)). Table 3.14 shows the coefficients between the Visual-Test and the reciprocal value of the Schwartzberg-Test (cf. Equation (3.8)) and Table 3.15 shows those between the Relative Moment of Inertia (cf. Equation (3.5)) and the Visual-Test.

	Enclosing rectangle	χ-shape 0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Convex hull
absolute values	0.11	−0.36	−0.38	−0.42	−0.54	−0.53	−0.52	−0.47	−0.46	−0.28
ranking values	−0.04	0.36	0.37	0.40	0.53	0.56	0.56	0.49	0.52	0.40

Table 3.13: Correlation between the Reock-Test and the Visual-Test

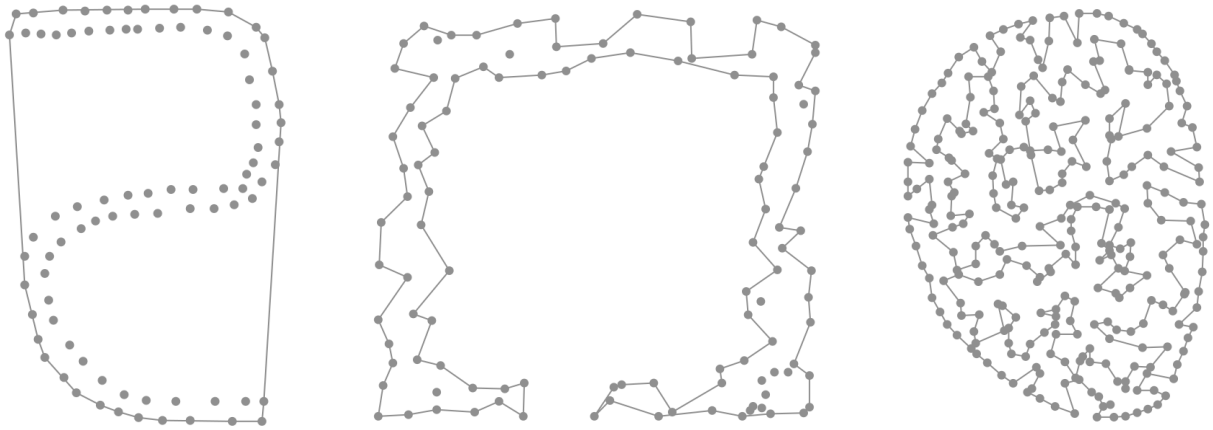
	Enclosing rectangle	χ-shape 0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Convex hull
absolute values	0.11	0.02	−0.01	−0.10	−0.50	−0.53	−0.55	−0.48	−0.47	−0.20
ranking values	−0.04	−0.03	0.03	0.26	0.46	0.53	0.57	0.53	0.55	0.38

Table 3.14: Correlation between the Schwartzberg-Test and the Visual-Test

	Enclosing rectangle	χ-shape 0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Convex hull
absolute values	0.11	−0.19	−0.17	−0.22	−0.41	−0.41	−0.40	−0.36	−0.35	−0.03
ranking values	−0.04	0.21	0.19	0.28	0.46	0.53	0.58	0.53	0.50	0.27

Table 3.15: Correlation between the Relative Moment of Inertia and the Visual-Test

For a variation of the Reock-Test using the convex hull Table 3.16 states the coefficients. In this case, the evaluation value for defining a district's shape by an enclosing rectangle or by the convex hull is 1 in any case. Hence, no correlation coefficient can be determined. As expected, the results for applying compactness measures to enclosing rectangles do not coincide with the visual impression. For χ-shapes the highest correlation is achieved if l_χ is defined in the range between 0.5 and 0.7. For larger values of l_χ the approximated shape is often non-intuitive and too rough. Hence, the obtained result for applying a compactness measure does not coincide with the visual impression.



(a) S-shaped district (b) District 8 (c) District 22

Figure 3.30: χ-shapes of some selected districts

	Enclosing rectangle	χ-shape 0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Convex hull
absolute values	↯	−0.29	−0.29	−0.36	−0.53	−0.52	−0.51	−0.46	−0.46	↯
ranking values	↯	0.29	0.29	0.38	0.55	0.55	0.53	0.52	0.56	↯

Table 3.16: Correlation between the Reock-Test using the convex hull and the Visual-Test

Figure 3.30a depicts an example for setting l_χ=1.0, i.e., for using the convex hull. Especially, applying the Schwartzberg-Test to convex hulls is not suitable, as the corresponding correlation coefficient of only −0.19 shows. On the other side, for small values of l_χ the achieved shapes are often very irregular. This can be positive in some cases, for example the χ-shape for district 8 obtained by setting l_χ=0.25 depicted in Figure 3.30b is most likely evaluated as non-compact. Unfortunately, a small value of l_χ may lead to a poor evaluation of a visual compact district. For example, the χ-shape of district 22 obtained by setting l_χ=0.25 illustrated in Figure 3.30c is also visually non-compact.

Nevertheless, applying one of the presented tests to χ-shapes defined by setting l_χ in the range between 0.5 and 0.7 is the most suitable way to measure the compactness of a district consisting of a set of points without considering the total set of basic areas.

Finally, the following analysis investigates different combinations of selected measures. Table 3.17 presents the correlation coefficients with the Visual-Test for using χ-shapes and setting l_χ=0.6, while Table 3.19 states these coefficients for setting l_χ=0.7. The corresponding coefficients based on the ranking values are stated in Table 3.18 and Table 3.20, respectively. In this case, these results often indicate no benefit for combining two measures compared to the usage of one single measure.

$comp_1(\cdot)$	$comp_2(\cdot)$	α						
		1	0	0.25	0.33	0.5	0.67	0.75
Schwartzberg	Reock (circle)	−0.53	−0.53	−0.55	−0.55	−0.55	−0.55	−0.55
Cox	Reock (circle)	−0.56	−0.53	−0.57	−0.57	−0.56	−0.56	−0.55
Adapted RMoI	Reock (circle)	−0.31	−0.53	−0.47	−0.49	−0.51	−0.52	−0.52
Schwartzberg	Reock (convex)	−0.53	−0.52	−0.54	−0.53	−0.53	−0.53	−0.53
Cox	Reock (convex)	−0.56	−0.52	−0.56	−0.56	−0.55	−0.54	−0.54
Adapted RMoI	Reock (convex)	−0.31	−0.52	−0.50	−0.52	−0.52	−0.52	−0.52
Schwartzberg	RMoI	−0.53	−0.41	−0.51	−0.50	−0.48	−0.46	−0.45
Cox	RMoI	−0.56	−0.41	−0.54	−0.53	−0.51	−0.48	−0.46
Adapted RMoI	RMoI	−0.31	−0.41	−0.42	−0.42	−0.42	−0.42	−0.42
Schwartzberg	Adapted RMoI	−0.53	−0.31	−0.50	−0.52	−0.53	−0.54	−0.54
Cox	Adapted RMoI	−0.56	−0.31	−0.52	−0.54	−0.55	−0.56	−0.56

Table 3.17: Correlation between absolute values of the Visual-Test and combined measures applied to χ-shapes (l_χ=0.6)

$comp_1(\cdot)$	$comp_2(\cdot)$	α						
		1	0	0.25	0.33	0.5	0.67	0.75
Schwartzberg	Reock (circle)	0.53	0.56	0.55	0.55	0.55	0.56	0.56
Cox	Reock (circle)	0.53	0.56	0.54	0.55	0.54	0.56	0.54
Adapted RMoI	Reock (circle)	0.46	0.56	0.58	0.57	0.56	0.56	0.56
Schwartzberg	Reock (convex)	0.53	0.55	0.54	0.54	0.55	0.54	0.54
Cox	Reock (convex)	0.53	0.55	0.55	0.54	0.53	0.54	0.54
Adapted RMoI	Reock (convex)	0.31	0.55	0.57	0.60	0.58	0.56	0.57
Schwartzberg	RMoI	0.53	0.53	0.51	0.51	0.52	0.54	0.55
Cox	RMoI	0.53	0.53	0.52	0.51	0.51	0.52	0.54
Adapted RMoI	RMoI	0.46	0.53	0.51	0.52	0.52	0.51	0.52
Schwartzberg	Adapted RMoI	0.53	0.46	0.53	0.53	0.55	0.56	0.54
Cox	Adapted RMoI	0.53	0.46	0.53	0.54	0.55	0.54	0.53

Table 3.18: Correlation between ranking values of the Visual-Test and combined measures applied to χ-shapes (l_χ=0.6)

$comp_1(\cdot)$	$comp_2(\cdot)$	α						
		1	0	0.25	0.33	0.5	0.67	0.75
Schwartzberg	Reock (circle)	−0.55	−0.52	−0.55	−0.55	−0.55	−0.54	−0.54
Cox	Reock (circle)	−0.58	−0.52	−0.58	−0.57	−0.57	−0.56	−0.55
Adapted RMoI	Reock (circle)	−0.31	−0.52	−0.46	−0.48	−0.50	−0.51	−0.51
Schwartzberg	Reock (convex)	−0.55	−0.51	−0.54	−0.54	−0.53	−0.52	−0.52
Cox	Reock (convex)	−0.58	−0.51	−0.57	−0.57	−0.56	−0.54	−0.53
Adapted RMoI	Reock (convex)	−0.31	−0.51	−0.50	−0.51	−0.51	−0.51	−0.51
Schwartzberg	RMoI	−0.55	−0.40	−0.51	−0.50	−0.47	−0.45	−0.44
Cox	RMoI	−0.58	−0.40	−0.55	−0.54	−0.51	−0.47	−0.46
Adapted RMoI	RMoI	−0.31	−0.40	−0.41	−0.41	−0.41	−0.41	−0.40
Schwartzberg	Adapted RMoI	−0.55	−0.31	−0.51	−0.53	−0.54	−0.55	−0.55
Cox	Adapted RMoI	−0.58	−0.31	−0.53	−0.55	−0.57	−0.57	−0.58

Table 3.19: Correlation between absolute values of the Visual-Test and combined measures applied to χ-shapes (l_χ=0.7)

$comp_1(\cdot)$	$comp_2(\cdot)$	α						
		1	0	0.25	0.33	0.5	0.67	0.75
Schwartzberg	Reock (circle)	0.55	0.52	0.58	0.59	0.60	0.58	0.56
Cox	Reock (circle)	0.58	0.52	0.59	0.58	0.59	0.59	0.58
Adapted RMoI	Reock (circle)	0.31	0.52	0.58	0.57	0.55	0.56	0.56
Schwartzberg	Reock (convex)	0.55	0.51	0.57	0.56	0.56	0.55	0.55
Cox	Reock (convex)	0.58	0.51	0.57	0.57	0.56	0.56	0.56
Adapted RMoI	Reock (convex)	0.31	0.51	0.55	0.56	0.58	0.58	0.58
Schwartzberg	RMoI	0.55	0.40	0.56	0.57	0.56	0.55	0.56
Cox	RMoI	0.58	0.40	0.56	0.56	0.56	0.55	0.56
Adapted RMoI	RMoI	0.31	0.40	0.52	0.52	0.55	0.58	0.58
Schwartzberg	Adapted RMoI	0.55	0.31	0.56	0.57	0.57	0.59	0.58
Cox	Adapted RMoI	0.58	0.31	0.56	0.56	0.58	0.58	0.57

Table 3.20: Correlation between ranking values of the Visual-Test and combined measures applied to χ-shapes (l_χ=0.7)

In order to define a district's shape according to one of the approaches presented in Section 3.5.4, the shapes of its basic areas have to be defined previously. However, to define these shapes the total set of basic areas is necessary. That means, for the examples depicted in Figure 3.29 the definition of the shapes is not possible since no information about the further basic areas within the overall area are given. In order to evaluate these approaches it would be necessary that the survey's participants evaluate solutions instead of single districts. Nevertheless, these approaches overcome one problem described before. The shape of district 3 differs in dependence of whether there are points of other districts located within the large open space or not. Hence, it is expected that in some cases these approaches outperform the approaches described before. However, there is still the problem that obtained results depend on the definition of the overall area.

The presented results show the difficulties for applying measures to point or line representations. At first, distance-based measures do not coincide with the visual impression of compactness. Nevertheless, depending on the application the visual impression is of minor importance compared to other criteria such as travel distances within a district. In this case, distance-based measures are useful. Moreover, distance-based measures are most commonly applied in order to evaluate solutions and not to compare single districts. In this case, the usage is more recommendable. Despite some weaknesses, using a common measure such as the Reock-Test, the Schwartzberg-Test and the Cox-Test on χ-shapes is the most suitable way if only single districts are evaluated. For evaluating entire solutions, the definition of the basic areas' shapes has some advantages.

3.6 Conclusions

This chapter has addressed compactness very much in detail. After defining compactness and listing requirements for compactness measures it has presented the most common measures. The majority of them are based on polygonal representations of basic areas. The theoretical and practical analysis of these measures has confirmed that it is (nearly) impossible to define a comprehensive compactness measure. Nevertheless, some measures mainly shape-only-dispersion measures and shape-only-area-perimeter measures perform very well in practice. Others can be useful depending on the application. Altogether, combining different measures has proven to be a successful strategy.

Finally, this chapter has introduced and summarized some ideas on how compactness for point or line representations can be measured. In this case, it is very hard to define compactness on single districts without considering the solution as a whole. If the districts are non-overlapping, we suggest defining the districts' shapes to which existing compactness measures can be applied. In order to define the districts' shapes χ-shapes seem to be most suitable. If overlapping districts can occur, replacing point representations with polygonal basic areas seems to be useful. Depending on the application it is advisable to apply further measures and combine their results. For example, in the context of sales districting the usage of distance-based measures is suitable.

Bibliography

[1] F. Aurenhammer, R. Klein, and D. L. Lee. *Voronoi Diagrams and Delaunay Triangulations*. World Scientific, 2013. ISBN 978-9814447638.

[2] R. S. Beth and P. J. Taylor. Communications. *The American Political Science Review*, 68:1275–1277, 1974.

[3] D. J. Blair and T. H. Biss. *The measurement of shape in geography: an appraisal of methods and techniques*, volume 11. Department of Geography, Nottingham University, 1967.

[4] R. R. Boyce and W. A. V. Clark. The Concept of Shape in Geography. *Geographical Review*, 54(4):561–572, 1964.

[5] B. Bozkaya, E. Erkut, and G. Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1): 12–26, 2003.

[6] C. Chambers and A. Miller. A measure of Bizarreness. *Quarterly Journal of Political Science*, 5(1):27–44, 2010.

[7] C. P. Chambers and A. D. Miller. Measuring legislative boundaries. *Mathematical Social Sciences*, 66(3):268–275, 2013.

[8] T. M. Chan. Optimal Output-Sensitive Convex Hull Algorithms in Two and Three Dimensions. *Discrete and Computational Geometry*, 16:361–368, 1996.

[9] E. P. Cox. A Method of Assigning Numerical and Percentage Values to the Degree of Roundness of Sand Grains. *Journal of Paleontology*, 1(3):179–183, 1927.

[10] M. Duckham, L. Kulik, M. Worboys, and A. Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10):3224–3236, 2008.

[11] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the Shape of a Set of Points in the Plane. *IEEE Transactions on Information Theory*, 29(4):551–559, July 1983.

[12] L. M. Eig and M. V. Seitzinger. *State constitutional and statutory provisions concerning congressionalstate legislative redistricting*. U.S. Government Printing Office, 1981.

[13] Y. Frolov. Measuring of shape of geographical phenomena: a history of the issue. *Soviet Geography: Review and Translation*, 16:676–687, 1975.

[14] R. G. Fryer Jr. and R. Holden. Measuring the Compactness of Political Districting Plans. *Journal of Law and Economics*, 54(3):493–535, 2011.

[15] J. P. Gibbs. *Urban Research Methods*, chapter A method for comparing the Spatial Shape of Urban Units, pages 99–107. New York: Van Nostrand, 1961.

[16] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132–133, 1972.

[17] B. Grofman. Criteria for Districting: A Social Science Perspective. *UCLA Law Review*, 1(33):77–184, 1985.

[18] P. Haggett. *Location Analysis in Human Geography*. Edward Arnold (Publishers) Limited, 1965.

[19] C. C. Harris Jr. A scientific method of districting. *Behavioral Science*, 9(3):219–225, 1964.

[20] T. Hofeller and B. Grofman. *Toward Fair and Effective Representation*, chapter Comparing the Compactness of California Congressional Districts under Three Different Plans: 1980, 1982 and 1984. New York: Agathon, 1990.

[21] D. L. Horn, C. R. Hampton, and A. J. Vandenberg. Practical application of district compactness. *Political Geography*, 12(2):103–120, 1993.

[22] R. E. Horton. Drainage-basin characteristics. *Eos, Transactions American Geophysical Union*, 13(1):350–361, 1932.

[23] H. F. Kaiser. An Objective Method for Establishing Legislative Districts. *Midwest Journal of Political Science*, 10(2):200–213, 1966.

[24] J. Kalcsics, S. Nickel, and M. Schröder. Towards a Unified Territorial Design Approach – Applications, Algorithms and GIS Integration. *TOP*, 13(1):1–74, 2005.

[25] H. Lei, G. Laporte, and B. Guo. Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1–2):67–85, 2012.

[26] P. Ludwig. Evaluation, Vergleich und Optimierung von Wahlkreisen am Beispiel Baden-Württembergs. Bachelorthesis, Karlsruhe Institute of Technology (KIT), 2013.

[27] A. M. Maceachren. Compactness of Geographic Shape: Comparison and Evaluation of Measures. *Geografiska Annaler. Series B, Human Geography*, 67(1):53–67, 1985.

[28] S. Marquardt. Entwicklung und Evaluation von Kompaktheitsmaßen für punktförmige Basisgebiete. Bachelorthesis, Karlsruhe Institute of Technology (KIT), 2014.

[29] R. G. Niemi, B. Grofman, C. Carlucci, and T. Hofeller. Measuring Compactness and the Role of a Compactness Standard in a Test for Partisan and Racial Gerrymandering. *The Journal of Politics*, 52(04):1155–1181, 1990.

[30] L. Papayanopoulos. Quantitative principles underlying apportionment methods. *Annals of New York Academy of Sciences*, 219:181–191, 1973.

[31] E. C. Reock. A Note: Measuring Compactness as a Requirement of Legislative Apportionment. *Midwest Journal of Political Science*, 5(1):70–74, 1961.

[32] R. Z. Ríos-Mercado and E. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776, 2009.

[33] J. Schwartzberg. Reapportionment gerrymanders and the notion of compactness. *Minnesota Law Review*, 50:443–452, 1966.

[34] P. J. Taylor. A New Shape Measure for Evaluating Electoral District Patterns. *The American Political Science Review*, 67(3):947–950, 1973.

[35] TheFreeDictionary by Farlex. Compactness. `www.TheFreeDictionary.com`, 2015. [Online; accessed 5-November-2015].

[36] J. B. Weaver and S. W. Hess. A Procedure for Nonpartisan Districting Development of Computer Techniques. *Yale Law Journal*, 72:288–308, 1963.

[37] H. P. Young. Measuring the Compactness of Legislative Districts. *Legislative Studies Quarterly*, 13(1):105–115, 1988.

Part II

Geometrically Motivated Approaches

4 Recursive Partitioning Algorithm

Contents

Considering applications such as the design of service districts, sales districts, or pickup and delivery districts a basic area may be interpreted as a single point in the plane. These problems are examined in the current chapter. Usually, each basic area corresponds to a single customer requiring a *service*, e.g., technical support, visits by a salesperson, or delivery of parcels. These customers are for example people, branch offices, supermarkets, but also machines. Most commonly, their locations are given as geo-coded addresses.

Typically, each district corresponds to the area of responsibility for one person, e.g., a single technician, salesperson or driver, or a team of them. Let the term "service person" denote this person or team in the following. Each customer should be assigned to exactly one service person. This increases the familiarity of the service person with his customers or their systems as well as its knowledge of the surrounding areas of the customers, for example to find alternative routes in the case of traffic jams. Moreover, in order to avoid competition between different service persons and to reduce unproductive travel times, these areas should be clearly defined geographically, i.e., they should be contiguous and compact. Furthermore, for the reasons of fairness, each service person should have approximately the same workload and/or income opportunity, i.e., the districts should be balanced.

4.1 Related Literature

The districting literature concerning basic areas represented by points explicitly is rather limited. For example, Kalcsics et al. [16] present a purely point based, application independent, geometric solution approach. It recursively sub-divides the districting problem and results in balanced and non-overlapping districts.

Haugland et al. [11] address the problem of designing districts for stochastic vehicle routing problems. The authors refer to applications such as parcel delivery where demands of the customers vary from day to day. Here, the customers should be grouped into fixed districts, one for each driver. By doing so, the drivers become familiar with their customers and their districts. The aim is to minimize the total expected routing costs. Moreover, the authors include an upper bound for the routing costs within a district. A district is feasible if each realization of the stochastic demands of the customers does not exceed this bound. In order to obtain contiguous districts, the authors use the Haugland-Graph (see Section 2.2.4.5) and ensure that each district is a connected sub-graph of this graph. They propose a tabu search approach in order to solve this districting problem. However, their approach includes balance only implicitly and does not take compactness into account.

In contrast to this, Lei et al. [18] examine the vehicle routing and districting problem with uncertain customer locations. However, these customers are only a subset of the total set of customers. The authors propose an objective function containing the number of districts, the expected routing costs, and compactness. The authors approximate the routing costs using the Beardwood-Halton-Hammersley theorem [2] and an overtime rate. However, they do not consider balance explicitly as well. For each deterministic customer (basic area) they determine a shape according to the approach described in Section 3.5.4. Then, compactness is measured in terms of the Bozkaya-Test (cf. Section 3.3.3.2) and contiguity is ensured by means of these determined shapes as well. The authors apply a large neighborhood search procedure in order to solve this districting problem.
A related work of Lei et al. [19] addresses the multiple traveling salesperson and districting problem with multiple periods and multiple depots, where the set of customers varies dynamically over time. However, at the beginning of each period, the number and the locations of the customers are available. Different planning criteria are merged in the objective function: The minimization of the number of districts, the optimization of the compactness with respect to the Bozkaya-Test, the minimization of the balance in terms of profit, and the minimization of the dissimilarity between the solutions over time. The profit of a salesperson consists of the income by visiting the customers minus the traveling costs approximated by the Beardwood-Halton-Hammersley theorem. Hence, the authors assume a travelling salesperson problem (TSP) within each period. In order to solve the entire problem, the authors propose an adaptive large neighborhood search meta-heuristic.
In addition, Lei et al. [20] include stochastic customers and obtain a multi-objective dynamic stochastic districting and routing problem. The authors present an enhanced multi-objective co-evolutionary algorithm with mating restrictions.

Bard and Jarrah [1] focus on pickup and delivery applications. Their aim is the determination of a minimal set of contiguous districts where each district corresponds to the area of responsibility for one single vehicle. Hence, capacity and time constraints have to be satisfied. In the context considered here practical instances have up to 50.000 customer. Thus, the authors apply a pre-processing step firstly that aggregates some customers in order to reduce the complexity of the problem. After that, they determine a grid of balanced clusters. In order to estimate the routing times of the vehicles, the authors incorporate for each customer the probability that he needs service as well as the probabilities which customer is visited next to this customer. The determination of the grids contains random decisions. Therefore, the authors determine a set of solutions and combine them by using a set covering approach.
Jarrah and Bard [14] continue this work and propose a column-generation approach com-

bined with ideas of tabu search in order to limit the number of considered sub-problems. They introduce a set of geometric constraints that ensure that each cluster spans a symmetric rectangle centered at a predetermined seed. However, the proposed model contains capacity constraints according to the working time and the vehicle capacity, but no balance constraint. Moreover, running times of several hours are reported for instances of some thousand basic areas.

Zhong et al. [25] deal with the driver learning within a region explicitly. However, in contrast to other approaches they differ between core areas and flex areas. The customers of a core area are permanently assigned to a service person, whereas the customers of a flex area are assigned to a service person whenever they require service. By allowing this flexibility for some customers, the workload can be balanced better for every day.

Since a service person has to visit its district regularly, its location, e.g., office, depot or residence, is an important factor according to the obtained travel times. However, there is no consensus in literature whether these locations are predetermined [1, 14, 18, 19, 20, 25] or be subject of the planning process [6, 10].

In the mentioned applications, single customers are often grouped by exogenously given properties such as zip-codes, city quarters or company trading areas, and these groups are treated as basic areas [6, 8, 10, 12, 21, 22, 23, 24, 25]. Hence, in fact, they do not treat the basic areas as points.

In most of the respective applications a service person has to visit the customers and provide the service on-site. Hence, his travel times are a part of his total working time. Many of the described approaches are motivated by the underlying routing problem. They include capacity constraints according to the tour duration or to the vehicle capacity. However, they mainly do not explicitly model balance as planning criteria. Moreover, the presented approaches often assume a TSP tour through all existent customers within a time period. Unfortunately, taking a closer look on possible applications, the travel times may differ noticeably.
In the context of technical support, a technician may solve many problems remotely. Hence, he visits its customers only rarely and typically at most one or two customers per day. Thus, the fraction of the travel times on the total working time is rather small. However, the maximum travel time to an associated customer should not be too large.
In contrast to this, a service person that fills up ticket machines or cigarette machines visits (almost) every customer every day. Hence, his working day mainly consists of travelling. In this case, his daily travel time corresponds to the length of a TSP tour through all assigned customers.

In the context of planning for field staff members, the visit frequency often differs from customer to customer. Some customers have to be visited two or three times per week, whereas others have to be visited only once per quarter. Moreover, some customers want to be visited every week on the same weekday, some others may have time windows, and so on. Thus, for a given district the planning of daily districts or daily tours, respectively, is a problem on its own.
In the context of pickup and delivery planning there is also a significant uncertainty on the daily demand. Hence, the workload of a service person differs from day to day. Therefore, in order to balance the workload of different service persons, a longer time period such as weeks or months have to be taken into account.
In summary, during the planning process of districts it is almost impossible to determine the total travel times in a given time horizon. However, the hope is that geographically compact districts result in smaller travel times on a day-to-day basis compared to non-compact districts.

The goal of this chapter is to present an algorithm that considers the problem in a more generalized way focusing on the districting part of the problem. The aim is to partition the set of customers into a given number of districts such that each district is balanced, compact and contiguous. The presented approach is based on an approach of Kalcsics et al. [16]. It generates contiguous and almost perfectly balanced districts, but in terms of compactness it has some weaknesses. In order to overcome them, this chapter will present some extensions and improvements. Moreover, it introduces a way to integrate prescribed centers into this algorithm. The following description is based on Butsch et al. [4].

The remainder of this chapter is organized as follows. The next section will adapt our general model for point based districting problem. Section 4.3 presents a geometrical divide and conquer heuristic to solve this problem. After that, Section 4.4 presents the results of extensive computational tests that confirm the efficiency and the quality of the obtained solution. Since the residences of the service persons are sometimes prescribed, Section 4.5 shows how to incorporate them into the heuristic. After that, Section 4.6 deals with multiple activity measures. Moreover, Section 4.7 presents a variation of this approach used to determine Emergency Medical Services regions. The chapter concludes with a summary and a short outlook.

4.2 The Model

Chapter 2 already has presented our general model for districting problems. This subsection specifies this model in order to make it applicable in the context of sales or service districting, where the basic areas correspond to single customers. Based on these applications, the following assumptions can be made:

- The number of required districts p is given in advance. If this is not the case, the problem is solved for different values of p and the solutions are compared according to a set of desired criteria.
- Neither an existing districting plan nor prescribed centers need to be taken into account. However, Section 4.5 describes how to incorporate existing centers.
- The planning process contains only one time period, i.e., the assignments should be fixed for this time. For example, a company often plans the visits of its customers for a quarter or a year.
- The customers are deterministic, i.e., their locations and activities are given in advance. For many applications it is difficult or even impossible to determine the daily travel time anyway. Hence, small changes of the set of customers during the time period will most likely not deteriorate the districting plan too much.

4.2.1 Components

The description starts with the specification of the general model (cf. Chapter 2).

4.2.1.1 Basic Areas

Here, each *basic area* $i \in BA$ corresponds to a single (customer) location represented by a point in the plane, e.g., a geo-coded address. For purposes of simplification $b_i = (x_i, y_i)$ denotes this point as well as the corresponding basic area.
Moreover, only one activity measure w_i is associated with each basic area. In most cases this activity is the (estimated) sales potential or the time needed to serve the total demand of the customer within the planning horizon.

4.2.1.2 Districts

A *district* D_g consists of a set of basic areas $B_g \subseteq BA$ that is serviced by a single service person. Hence, in this case, there is a one-to-one relation between D_g and B_g.

The activity of a district is defined as the sum of the activities of its assigned basic areas, i.e.,

$$w(D_g) := \sum_{i \in B_g} w_i \,.$$

Unfortunately, the shape of D_g is not directly defined. Hence, surrogates have to be used if necessary.

4.2.1.3 Districting Plan

A *districting plan* or *solution* is a set of districts $S := \{D_1; \ldots; D_p\}$, where p is the given number of districts.

4.2.1.4 Distances

The distance $d_{i,j} := d(b_i, b_j)$ between two basic areas is either the Euclidean distance, or the distance or travel time on a road network.

4.2.2 Planning Criteria

The aim of the considered districting problem is the following: Partition all basic areas BA into p districts that are balanced, contiguous, and compact.
This model treats complete and exclusive assignment as a hard criterion, whereas it treats compactness and contiguity as a soft criterion. Moreover, it treats balance as a soft and also as a hard criterion.

4.2.2.1 Complete and Exclusive Assignment

The sets $B_1, \ldots, B_p$ define a partition of the set of basic areas BA.

4.2.2.2 Balance

Recall (cf. Section 2.2.2) that one way to measure the balance of a district is to compute the relative percentage deviation of its size from the average size, i.e.,

$$bal(D_g) := \frac{|w(D_g) - \mu|}{\mu} \,.$$

Our model defines the balance of a solution as the maximal balance of a single district, i.e.,

$$bal_{max}(S) := \max_{g=1,\ldots,p} bal(D_g) \,. \tag{4.1}$$

In order to treat balance as a hard criterion as well, this model defines a maximal feasible balance τ, i.e., a solution is feasible if $bal_{max}(S) \leq \tau$ holds. Thus, $L_D := (1 - \tau) \cdot \mu$ defines a lower bound for the size of a district, while $U_D := (1 + \tau) \cdot \mu$ defines an upper bound.

4.2.2.3 Compactness

According to Chapter 3 a district is *compact* if it is nearly round-shaped or square, undistorted, without holes, and has a smooth boundary. There are several compactness measures proposed in the literature. However, most of them are based on polygonal representations of the basic areas. One exception are distance-based measures, which can easily be adapted to point representations (cf. Section 3.5.1.2). Another reason for using distance-based measures is the fact that compactness should be a proxy for expected travel times. Depending on the application's underlying routing problem, the different distance-based measures are more or less recommendable. For each district,

- the *Weighted Moment of Inertia* is the weighted sum of squared distances from all basic areas to the center, i.e.,

$$comp_{wmoi}(D_g) := \sum_{i \in B_g} w_i \cdot d^2(b_i, cen_g)\,. \tag{4.2}$$

- the *Moment of Inertia* is the (unweighted) sum of squared distances from all basic areas to the center, i.e.,

$$comp_{moi}(D_g) := \sum_{i \in B_g} d^2(b_i, cen_g)\,. \tag{4.3}$$

- the *Pairwise Distances* are the distances between all pairs of basic areas added up, i.e.,

$$comp_{pd}(D_g) := \sum_{i \in B_g} \sum_{j \in B_g} d_{i,j}\,. \tag{4.4}$$

- the *Weighted Pairwise Distances* are the weighted distances between all pairs of basic areas added up, i.e.,

$$comp_{wpd}(D_g) := \sum_{i \in B_g} \sum_{j \in B_g} w_i \cdot d_{i,j}\,. \tag{4.5}$$

- the *Maximum Distance* is the maximum distance between two basic areas, i.e.,

$$comp_{md}(D_g) := \max_{i,j \in B_g} d_{i,j}\,. \tag{4.6}$$

Note that these formulations are also usable if $d(\cdot\,,\cdot)$ is not symmetric, and, hence, not a metric.

The (Weighted) Moment of Inertia as well as the (Weighted) Pairwise Distances define the compactness of a solution S straightforwardly as the sum of its districts, i.e.,

$$comp_*(S) := \sum_{g=1}^{p} comp_*(D_g)\,,$$

where $* \in \{wmoi;\, moi;\, pd;\, wpd\}$. In contrast to this, the Maximum Distance defines the compactness of a solution S as the maximal compactness of a single district, i.e.,

$$comp_{md}(S) := \max_{g=1,\ldots,p} comp_{md}(D_g)\,.$$

4.2.2.4 Contiguity

Figuratively spoken, a district is contiguous if it is possible to travel to each basic area within the district from every other basic area within the district without leaving the district. Since the basic areas are represented by points, no implicitly given neighborhood information is available. Moreover, the shape of a district D_g is not defined directly. Hence, in this context another definition is necessary. According to Kalcsics et al. [16] a district is contiguous if the convex hull $ch(B_g)$ of the basic areas comprising district D_g does not intersect with the convex hull of the basic areas of any other district D_h.
Since this model treats contiguity as a soft criterion, it does not forbid these intersections, but it tries to minimize them. In order to do so, the contiguity measure computes the sum of the areas of intersection between their convex hulls, normalized by the area of the convex hull of BA, that is

$$ctg(S) := \frac{\sum_{g=1}^{p-1} \sum_{h=g+1}^{p} area(ch(B_g) \cap ch(B_h))}{area(ch(BA))}\,.$$

4.3 The Algorithm

The so-called Recursive Partitioning Algorithm (RPA) is based on the work of Kalcsics et al. [16] and utilizes the underlying geographical information of the districting problem. Its main idea was already sketched by Forrest [9] without giving details. This main idea is to recursively sub-divide the problem into smaller and smaller sub-problems, until an elementary level is reached where the districting problem can be solved efficiently. Therefore, the basic operation is to divide a subset $B \subseteq BA$ of the basic areas into two "halves" B_l and B_r. In other words, the algorithm splits the districting problem for B into two disjoint sub-problems, one for B_l and one for B_r. Then, it solves these two sub-problems independently in the same way. The solutions to these sub-problems directly yield a solution for the original problem.
Figure 4.1 illustrates an example, where a set B is firstly sub-divided into the subsets B_l and B_r, and afterwards B_l and B_r are sub-divided into the subsets B_{l_l} and B_{l_r}, and, respectively, B_{r_l} and B_{r_r}.

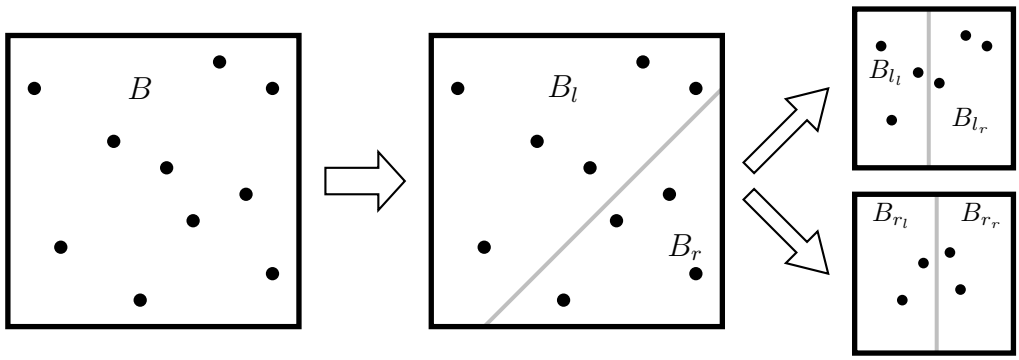


Figure 4.1: Recursive sub-division of a problem

The basic version of Kalcsics et al. [16] generates very fast contiguous, non-overlapping and almost perfectly balanced districts. However, in terms of compactness it has some weaknesses. This section presents some extensions and improvements in order to overcome them. But first, the next subsection will give some basic definitions.

4.3.1 Basic Definitions

Definition 4.3.1 *A* partition problem $PP := (B, q)$ *is the problem of sub-dividing a set of basic areas* $B \subseteq BA$ *into* $1 \leq q \leq p$ *districts.*

PP is called *trivial* if $q = 1$ since in this case B directly defines a district. A partition problem that still has to be sub-divided is called an *unsolved partition problem* and UPP denotes the set of unsolved partition problems.

Definition 4.3.2 *A* bisecting partition $BP := (B_l, B_r, q_l, q_r)$ *of a* PP *is defined by two sets* $B_l, B_r \subset B$ *such that* $B_l \cup B_r = B$ *and* $B_l \cap B_r = \emptyset$*, and two numbers* $1 \leq q_l, q_r < q$ *with* $q_l + q_r = q$*.*

A bisecting partition sub-divides a non-trivial partition problem PP into two smaller partition problems $PP_l := (B_l, q_l)$ and $PP_r := (B_r, q_r)$, where PP_l (PP_r) is called *left* (*right*) *sub-problem* of PP.

4.3.2 Algorithm Overview

Algorithm 4.3.1 outlines and summarizes the RPA. Starting from the original partition problem (BA, p) it chooses an unsolved partition problem $(B,\ q)$ from UPP in each iteration. If $q = 1$ holds, the set of basic areas B already defines a district. Hence, the algorithm adds B to the solution S and deletes the partition problem $(B,\ q)$ from UPP. Otherwise, it divides $(B,\ q)$ into two sub-problems $(B_l,\ q_l)$ and $(B_r,\ q_r)$. Accordingly, it replaces $(B,\ q)$ by $(B_l,\ q_l)$ and $(B_r,\ q_r)$ in UPP. The RPA repeats this procedure until no unsolved partition problem is left.

Algorithm 4.3.1: Recursive Partitioning Algorithm

Input: Set of basic areas BA, number of districts p.
Output: Districting plan S.

1 Set $UPP = \{(BA, p)\}$ and $S = \emptyset$.
2 **while** $UPP \neq \emptyset$ **do**
  Choose $PP = (B,\ q) \in UPP$.
  if $q = 1$ **then**
    set $S = S \cup \{B\}$, $UPP = UPP \backslash \{PP\}$.
  else
    Determine a set FBP of feasible bisecting partitions of PP.
    Choose the best bisecting partition $BP^* := (B_l^*, B_r^*, q_l^*, q_r^*) \in FBP$.
    Set $UPP = UPP \backslash \{PP\} \cup \{(B_l^*, q_l^*); (B_r^*, q_r^*)\}$.
  end
end
3 **return** S.

Until here, there are some questions that remain open:

- How to determine a set of possible bisecting partitions?
- How to decide whether a bisecting partition is feasible or not?
- How to evaluate a bisecting partition?

The objective of the next subsections is to formulate answers to these questions.

4.3.3 Generating Bisecting Partitions

This section addresses the question of how to generate bisecting partitions. Let PP, L_D and L_U be given. Each iteration looks for a bisecting partition $BP := (B_l, B_r, q_l, q_r)$ of (B, q) such that the resulting two sub-problems (B_l, q_l) and (B_r, q_r) are balanced, compact and contiguous. In the following, two approaches to determine bisecting partitions are presented.

4.3.3.1 Line Partitions

One approach, already presented by Kalcsics et al. [16], places a line that divides the set of basic areas B into two subsets B_l and B_r. If a basic area i lies on the line, the approach defines $i \in B_l$. A line $L(z, \alpha)$ is defined by a footpoint $z := (x_z, y_z) \in \mathbb{R}^2$ and an angle $\alpha \in [0, 2\pi)$ of the line with the positive x-axis.

In order to determine a set of bisecting partitions, the algorithm uses $K \in \mathrm{N}^+$ equally spaced line (search) directions having the angles $\alpha_k := k \cdot \frac{\pi}{K}$ $(k = 0, 1, \ldots, K-1)$ with the positive x-axis.

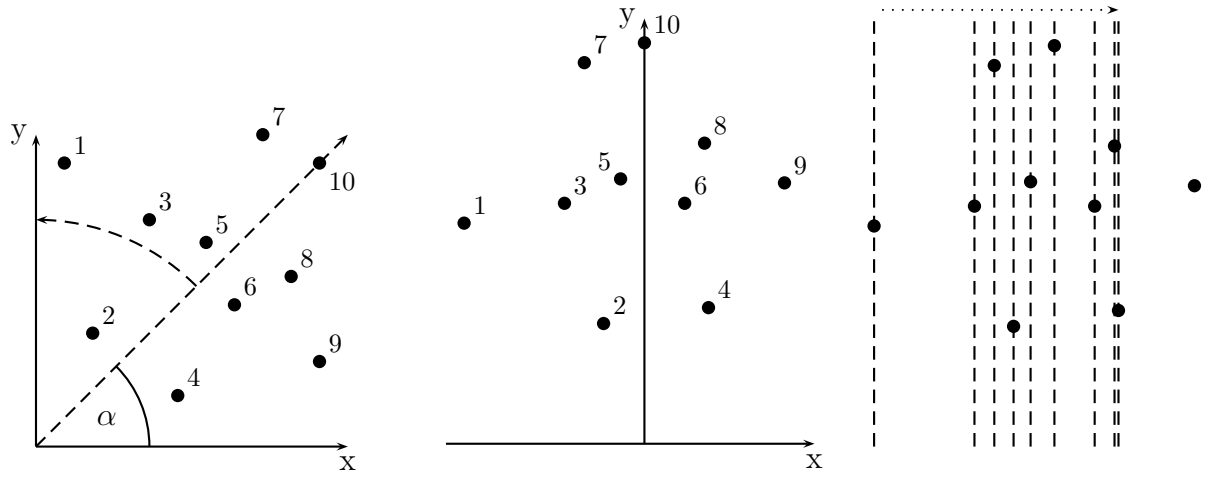
The algorithm rotates the coordinate system for each angle α_k such that the line through the origin with angle α_k becomes the y-axis, i.e.,

$$x_i^k = x_i \cdot \sin \alpha_k - y_i \cdot \cos \alpha_k$$

and

$$y_i^k = x_i \cdot \cos \alpha_k + y_i \cdot \sin \alpha_k.$$

Figure 4.2 illustrates this rotation, where Figure 4.2a shows the (original) set of basic areas, whereas Figure 4.2b depicts the rotated set of basic areas. Note that the dashed line in Figure 4.2a corresponds to the y-axis in Figure 4.2b.



(a) Set of basic areas B (b) Rotated basic areas for $\alpha=\pi/4$ (c) Set of potential lines

Figure 4.2: Generating line partitions

Next, the algorithm sorts the basic areas in B by non-decreasing x^k-values of their representative points. Let $b_1^k, b_2^k, \ldots, b_n^k$, where $n := |B|$, denote the basic areas as well as the representative points of this sorted sequence. Furthermore, without loss of generality, let no two basic areas lie on a common line with respect to α_k. Consider two successive points b_i^k and b_{i+1}^k. Each line, parallel to the y-axis, having an x-value greater than or equal to the x-value of b_i^k, but smaller than the x-value of b_{i+1}^k, divides the set of basic areas B into the same two subsets B_l and B_r. Therefore, between each pair of successive points only one line needs to be examined. Thus, the algorithm restricts itself to the lines through the points $b_1^k, b_2^k, \ldots, b_{n-1}^k$. Figure 4.2c depicts these lines for the rotated points illustrated in Figure 4.2b. Note that a line parallel to the y-axis through b_n^k must not take into account since it implies $B_r = \emptyset$.

Finally, the main idea of this approach is to choose one line of these lines such that the average size of the districts in the left sub-problem is nearly equal to the average size in the right sub-problem. If $\frac{w(B_l)}{q_l}$ equals $\frac{w(B_r)}{q_r}$, the average size is equal for both sub-problems. Since equality usually can not be achieved, the algorithm determines the line that minimizes the difference between the average sizes, i.e., that minimizes

$$sd(B_l, B_r, q_l, q_r) := \left| \frac{w(B_l)}{q_l} - \frac{w(B_r)}{q_r} \right| .$$

The following lemma addresses the question of how to choose $w(B_l)$ such that $sd(B_l, B_r, q_l, q_r)$ is minimized. The corresponding proof and the proofs of the subsequent lemmata are inspired by Kalcsics [15].

Lemma 4.3.1 *Setting* $w(B_l) = w(B) \cdot \frac{q_l}{q_l+q_r}$ *minimizes* $sd(B_l, B_r, q_l, q_r)$.

Proof
The minimum of $sd(B_l, B_r, q_l, q_r)$ is 0 and is obtained if and only if $\frac{w(B_l)}{q_l} = \frac{w(B_r)}{q_r}$:

$$\begin{aligned}
&\frac{w(B_l)}{q_l} = \frac{w(B_r)}{q_r} = \frac{w(B)}{q_r} - \frac{w(B_l)}{q_r} \iff \frac{w(B_l)}{q_l} + \frac{w(B_l)}{q_r} = \frac{w(B)}{q_r} \\
&\overset{\cdot q_l}{\iff} w(B_l) + w(B_l) \cdot \frac{q_l}{q_r} = w(B) \cdot \frac{q_r}{q_l} \iff w(B_l) \cdot \frac{q_l + q_r}{q_r} = w(B) \cdot \frac{q_l}{q_r} \\
&\overset{\cdot \frac{q_r}{q_r+q_l}}{\iff} w(B_l) = w(B) \cdot \frac{q_l}{q_r + q_l}
\end{aligned}$$

□

Let $W_l := w(B)\frac{q_l}{q_r+q_l}$. Moreover, let

$$B_{l_i}^k := \left\{b_1^k; \ldots; b_i^k\right\} \tag{4.7}$$

and

$$B_{r_i}^k := \left\{b_{i+1}^k; \ldots; b_n^k\right\}, \tag{4.8}$$

i.e., $B_{l_i}^k$ ($B_{r_i}^k$) consists of the first i (last $n-i$) elements of the sorted sequence of rotated basic areas. Since $w_i > 0$ holds for all i, the activity of $B_{l_i}^k$ is

$$w(B_{l_i}^k) < w(B_{l_{i+1}}^k) = w(B_{l_i}^k) + w(b_{i+1}^k)$$

and the activity of $B_{r_i}^k$ is

$$w(B_{r_i}^k) > w(B_{r_{i+1}}^k) = w(B_{r_i}^k) - w(b_{i+1}^k)\,. \tag{4.9}$$

Let a' denote the index that satisfies

$$w(B_{l_{a'}}^k) < W_l \;\text{ and }\; w(B_{l_{a'+1}}^k) \geq W_l\,. \tag{4.10}$$

The following lemma shows that $\frac{w(B_l)}{q_l} < \frac{w(B_r)}{q_r}$ only holds for $w(B_l) < W_l$. In addition, this implies $\frac{w(B_l)}{q_l} \geq \frac{w(B_r)}{q_r}$ for $w(B_l) \geq W_l$.

Lemma 4.3.2 *It holds that* $\frac{w(B_l)}{q_l} < \frac{w(B_r)}{q_r}$ *iff* $w(B_l) < W_l$.

Proof
Some transformations lead to the following result:

$$\begin{aligned} & w(B_l) < W_l = w(B) \cdot \frac{q_l}{q_l + q_r} = (w(B_l) + w(B_r)) \cdot \frac{q_l}{q_l + q_r} \\ \overset{\cdot(q_l+q_r)}{\iff} & w(B_l) \cdot q_l + w(B_l) \cdot q_r < w(B_l) \cdot q_l + w(B_r) \cdot q_l \\ \iff & w(B_l) \cdot q_r < w(B_r) \cdot q_l \\ \iff & \frac{w(B_l)}{q_l} < \frac{w(B_r)}{q_r} \end{aligned} \tag{4.11}$$

□

The next three lemmata show that $sd(B_l, B_r, q_l, q_r)$ decreases along the sequence $B^k_{l_1}, \ldots, B^k_{l_{a'}}$, and increases along the sequence $B^k_{l_{a'+1}}, \ldots, B^k_{l_n}$.

Lemma 4.3.3 *This lemma consists of two parts:*

a) For $i \leq a'$, $\left|\frac{w(B^k_{l_i})}{q_l} - \frac{w(B^k_{r_i})}{q_r}\right| > \left|\frac{w(B^k_{l_{i+1}})}{q_l} - \frac{w(B^k_{r_{i+1}})}{q_r}\right|$ *holds.*

b) For $i > a'$, $\left|\frac{w(B^k_{l_i})}{q_l} - \frac{w(B^k_{r_i})}{q_r}\right| < \left|\frac{w(B^k_{l_{i+1}})}{q_l} - \frac{w(B^k_{r_{i+1}})}{q_r}\right|$ *holds.*

Proof
Both parts are based on some transformations:

a)

$$\begin{aligned} & w(B^k_{l_i}) < w(B^k_{l_{i+1}}) \\ \overset{\cdot\frac{-1}{q_l}}{\iff} & -\frac{w(B^k_{l_i})}{q_l} > -\frac{w(B^k_{l_{i+1}})}{q_l} \\ \iff & \frac{w(B^k_{r_i})}{q_r} - \frac{w(B^k_{l_i})}{q_l} > \frac{w(B^k_{r_i})}{q_r} - \frac{w(B^k_{l_{i+1}})}{q_l} \\ \overset{(4.9)}{\Longrightarrow} & \frac{w(B^k_{r_i})}{q_r} - \frac{w(B^k_{l_i})}{q_l} > \frac{w(B^k_{r_i})}{q_r} - \frac{w(B^k_{l_{i+1}})}{q_l} > \frac{w(B^k_{r_{i+1}})}{q_r} - \frac{w(B^k_{l_{i+1}})}{q_l} \\ \overset{\text{Lemma 4.3.2}}{\Longrightarrow} & \left|\frac{w(B^k_{l_i})}{q_l} - \frac{w(B^k_{r_i})}{q_r}\right| > \left|\frac{w(B^k_{l_{i+1}})}{q_l} - \frac{w(B^k_{r_{i+1}})}{q_r}\right| \end{aligned}$$

b)

$$
\begin{aligned}
& w(B_{l_i}^k) < w(B_{l_{i+1}}^k) \overset{\cdot \frac{1}{q_l}}{\iff} \frac{w(B_{l_i}^k)}{q_l} < \frac{w(B_{l_{i+1}}^k)}{q_l} \\
& \iff \frac{w(B_{l_i}^k)}{q_l} - \frac{w(B_{r_i}^k)}{q_r} < \frac{w(B_{l_{i+1}}^k)}{q_l} - \frac{w(B_{r_i}^k)}{q_r} \\
& \overset{(4.9)}{\implies} \frac{w(B_{l_i}^k)}{q_l} - \frac{w(B_{r_i}^k)}{q_r} < \frac{w(B_{l_{i+1}}^k)}{q_l} - \frac{w(B_{r_i}^k)}{q_r} < \frac{w(B_{l_{i+1}}^k)}{q_l} - \frac{w(B_{r_{i+1}}^k)}{q_r} \\
& \overset{\text{Lemma 4.3.2}}{\implies} \left| \frac{w(B_{l_i}^k)}{q_l} - \frac{w(B_{r_i}^k)}{q_r} \right| < \left| \frac{w(B_{l_{i+1}}^k)}{q_l} - \frac{w(B_{r_{i+1}}^k)}{q_r} \right|
\end{aligned}
$$

□

Thus, $sd(B_l, B_r, q_l, q_r)$ is minimized by setting $B_l = B_{l_{a'}}^k$ or $B_l = B_{l_{a'+1}}^k$. Hence, there is one question left: Which of them minimizes $sd(B_l, B_r, q_l, q_r)$?

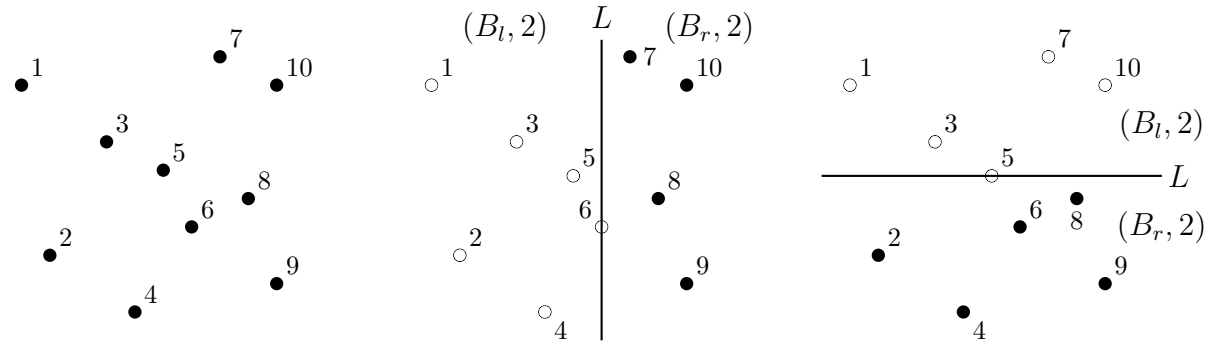
Lemma 4.3.4 *The inequation* $\left| \frac{w(B_{l_{a'}}^k)}{q_l} - \frac{w(B_{r_{a'}}^k)}{q_r} \right| \leq \left| \frac{w(B_{l_{a'+1}}^k)}{q_l} - \frac{w(B_{r_{a'+1}}^k)}{q_r} \right|$ *is satisfied if and only if* $W_l - w(B_{l_{a'}}^k) \leq \frac{1}{2} \cdot w(b_{a'+1}^k)$.

Proof

Some transformations lead to the result:

$$
\begin{aligned}
& \left| \frac{w(B_{l_{a'}}^k)}{q_l} - \frac{w(B_{r_{a'}}^k)}{q_r} \right| \leq \left| \frac{w(B_{l_{a'+1}}^k)}{q_l} - \frac{w(B_{r_{a'+1}}^k)}{q_r} \right| \qquad (4.12) \\
& \overset{\text{Lemmata 4.3.3a) and 4.3.3b)}}{\iff} \frac{w(B_{r_{a'}}^k)}{q_r} - \frac{w(B_{l_{a'}}^k)}{q_l} \leq \frac{w(B_{l_{a'+1}}^k)}{q_l} - \frac{w(B_{r_{a'+1}}^k)}{q_r} \\
& \overset{\cdot q_r}{\iff} w(B_{r_{a'}}^k) - w(B_{l_{a'}}^k) \cdot \frac{q_r}{q_l} \leq w(B_{l_{a'+1}}^k) \cdot \frac{q_r}{q_l} - w(B_{r_{a'+1}}^k) \\
& \iff w(B) - w(B_{l_{a'}}^k) - w(B_{l_{a'}}^k) \frac{q_r}{q_l} \\
& \leq (w(B_{l_{a'}}^k) + w(b_{a'+1}^k)) \cdot \frac{q_r}{q_l} - w(B) + w(B_{l_{a'}}^k) + w(b_{a'+1}^k) \\
& \iff 2 \cdot w(B) - 2 \cdot w(B_{l_{a'}}^k) \cdot (\frac{q_l + q_r}{q_l}) \leq w(b_{a'+1}^k) \cdot (\frac{q_l + q_r}{q_l}) \\
& \overset{\cdot \frac{1}{2} \cdot \frac{q_l}{q_l + q_r}}{\iff} w(B) \cdot \frac{q_l}{q_l + q_r} - w(B_{l_{a'}}^k) \leq \frac{1}{2} \cdot w(b_{a'+1}^k) \\
& \iff W_l - w(B_{l_{a'}}^k) \leq \frac{1}{2} \cdot w(b_{a'+1}^k)
\end{aligned}
$$

□



(a) Partition problem (BA, p) (b) Line partition for $\alpha = \pi/2$ (c) Line partition for $\alpha = 0$

Figure 4.3: Line partitions (Example 4.3.1)

The practical implementation works as follows. After sorting the basic areas, the RPA sums up their activities in order to determine the index a' defined in Equation (4.10). Figuratively spoken, it sweeps a line over the sorted sequence of basic areas until the sum of the activities on the left side of the line is greater than or equal to W_l. After that, it determines

$$a^* := \begin{cases} a' & \text{if } W_l - w(\{b_1^k; \ldots; b_{a'}^k\}) \leq \frac{1}{2} \cdot w(b_{a'+1}^k) \\ a'+1 & \text{otherwise} \end{cases} \tag{4.13}$$

the index of the last basic area that is element of B_l. Altogether, the approach determines the bisecting partition $LP(k, a^*, q_l) := (B_{l_{a^*}}^k, B_{r_{a^*}}^k, q_l, q_r)$. This kind of bisecting partition is called *line partition* .

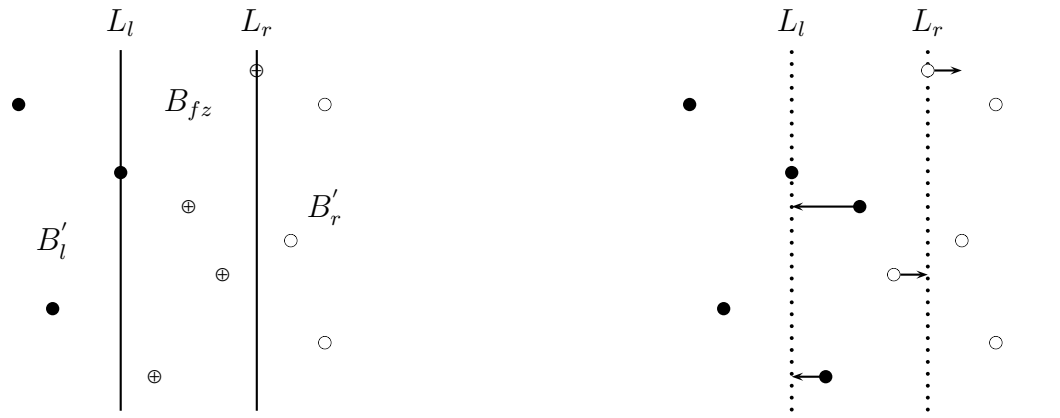
Example 4.3.1 Let the set of basic areas BA specified in Table 4.1 be given. Figure 4.3a illustrates this set.

i	1	2	3	4	5	6	7	8	9	10
x_i	0.5	1	2	2.5	3	3.5	4	4.5	5	5
y_i	5	2	4	1	3.4	2.5	5.5	3	1.5	5
w_i	5	3	4	4	4	6	3	5	7	9

Table 4.1: basic areas BA

Let $q_l = q_r = 2$. This implies $w(B) = 50$ and $W_l = 25$. Exemplarily, let $K = 2$, i.e., $\alpha_0 = 0$ and $\alpha_1 = \pi/2$.

For the vertical line $\alpha = \pi/2$, sorting the basic areas leads to 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. This approach results in the subsets $B_l = \{1; 2; 3; 4; 5; 6\}$ and $B_r = \{7; 8; 9; 10\}$, because $w(\{1; \ldots; 5\}) = 20 < 25$ and $w(\{1; \ldots; 6\}) = 26 > 25$ holds, and, thus, $a^* = 6$ ($b_{a^*}^k = 6$)



(a) Defining L_l and L_r (b) Assigning the basic areas of B_{fz}

Figure 4.4: Illustration of the flex-zone partition approach

holds. Figure 4.3b depicts the resulting line partition $LP(1, 6, 2) = (B_l, B_r, 2, 2)$.
For $\alpha = 0$, i.e., a horizontal line, the sorting results in 7, 1, 10, 3, 5, 8, 6, 2, 9, 4. Then, it holds that $a^* = 5$ ($b^k_{a^*} = 5$), and, hence, $B_l = \{7; 1; 10; 3; 5\}$ and $B_r = \{6; 8; 2; 9; 4\}$. Figure 4.3c illustrates the resulting line partition $LP(0, 5, 2) = (B_l, B_r, 2, 2)$.

Remark 4.3.1 A solution S obtained by using line partitions for each sub-division is a contiguous solution in any case, i.e., $ctg(S) = 0$. Furthermore, an upper bound for the balance of S is given by $\frac{2 \cdot w^{max}}{\mu}$, where $w^{max} := \max_{i \in B} w_i$ (see Kalcsics [15] for details).

Especially in the presence of geographic obstacles such as rivers or mountains, or if BA has a very irregular outer boundary, a sub-division based on lines is sometimes too rigid, resulting in non-compact districts. Therefore, the following subsection presents an approach that allows the border between the left and right sub-problem to be more flexible compared to line partitions. This, however, comes at the expense of contiguity, which can no longer be guaranteed.

4.3.3.2 Flex-Zone Partitions

The main idea of the flex-zone approach is to divide the set of basic areas B into three contiguous zones using lines, where the basic areas of the left (right) zone are directly assigned to the left (right) sub-problem, whereas the basic areas of the third zone, the so-called *flex-zone*, are assigned individually to the sub-problems in a subsequent step. Thus, each zone corresponds to a subset of B. Let B_{ll}, B_{fz} and B_{rr} denote these zones from left to right. Moreover, let L_l and L_r denote the two lines dividing the basic areas into three zones. Figure 4.4a sketches the main idea of this approach.

This subsection explains the flex-zone approach in more detail. For each angle α_k it uses a sorted sequence of basic areas b_1^k, b_2^k, ..., b_n^k defined analogously to the line partition approach and also uses K equally spaced line (search) directions. Furthermore, let $B_{l_i}^k$ and $B_{r_i}^k$ be defined as in Equations (4.7) and (4.8). In addition, let B_{ll} (B_{rl}) denote the set of basic areas to the left (right) of L_l, and, analogously, B_{lr} (B_{rr}) the set of basic areas to the left (right) of L_r. Moreover, let $B_{ll} \subseteq B_{lr}$, i.e., L_l is left to L_r (or both lines are equal). Since balance is treated as a hard criterion as well, the average size of a district in both sub-problems has to be in the interval between L_D and U_D, that means the following four constraints have to be satisfied:

1. $w(B_{ll}) \geq q_l \cdot L_D$
2. $w(B_{rl}) \leq q_r \cdot U_D \Rightarrow w(B) - w(B_{ll}) \leq q_r \cdot U_D \Rightarrow w(B_{ll}) \geq w(B) - q_r \cdot U_D$
3. $w(B_{lr}) \leq q_l \cdot U_D$
4. $w(B_{rr}) \geq q_r \cdot L_D \Rightarrow w(B) - w(B_{lr}) \leq q_r \cdot L_D \Rightarrow w(B_{lr}) \leq w(B) - q_r \cdot L_D$

An obvious approach to define L_l (L_r) is the usage of the first (last) line satisfying these four constraints. Later, this subsection will describe and discuss further approaches. In order to efficiently determine L_l and L_r, let

$$LL := \max\{q_l \cdot L_D; w(B) - q_r \cdot U_D\} \tag{4.14}$$

and

$$LU := \min\{q_l \cdot U_D; w(B) - q_r \cdot L_D\}\,. \tag{4.15}$$

In this case, LU und LL have the property, that LU is smaller than or equal to LL.

Lemma 4.3.5 *If $L_D \leq \frac{w(B)}{q} \leq U_D$ holds, then $LL \leq LU$ holds.*

Proof
First, since $L_D \leq U_D$ holds, the inequalities

$$q_l \cdot L_D \leq q_l \cdot U_D \quad \text{and} \quad w(B) - q_r \cdot U_D \leq w(B) - q_r \cdot L_D$$

are satisfied.
Furthermore, $L_D \leq \frac{w(B)}{q} = \frac{w(B)}{q_l + q_r}$ holds. This implies

$$w(B) \geq L_D \cdot (q_l + q_r) = L_D \cdot q_l + L_D \cdot q_r\,.$$

Thus, $w(B) - L_D \cdot q_r \geq L_D \cdot q_l$ is satisfied as well.
Finally, $U_D \geq \frac{w(B)}{q} = \frac{w(B)}{q_l+q_r}$ holds. Analogously to the latter case, this implies that also $w(B) - q_r \cdot U_D \leq q_l \cdot U_D$ holds. Summarized, $LL \leq LU$ holds. □

After sorting the basic areas, the flex-zone approach determines the lines $L_l := L(b^k_{l^*}, \alpha_k)$ and $L_r := L(b^k_{r^*}, \alpha_k)$ such that

$$w(B^k_{l^*-1}) < LL \text{ and } w(B^k_{l^*}) \geq LL \tag{4.16}$$

and

$$w(B^k_{l_{r^*}}) \leq LU \text{ and } w(B^k_{l_{r^*+1}}) > LU\,. \tag{4.17}$$

As explained above, these lines partition B into three zones. The left zone contains the set of basic areas $B_{ll} := \{b^k_1; \ldots; b^k_{l^*}\}$, the flex-zone (middle zone) contains $B_{fz} := \{b^k_{l^*+1}, \ldots, b^k_{r^*}\}$, and the right zone contains $B_{rr} := \{b^k_{r^*+1}; \ldots; b^k_n\}$. Defining $B_{ll} \subseteq B_l$ and $B_{rr} \subseteq B_r$ implies $w(B_l) \in [q_l \cdot L_D\,,\, q_l \cdot U_D]$ and $w(B_r) \in [q_r \cdot L_D\,,\, q_r \cdot U_D]$, i.e., the average size of a district for both sub-problems is within the feasible interval, independently of the decisions which basic areas of the flex-zone are assigned to which sub-problem. Hence, the bisecting partition will always be feasible in terms of balance. Thus, this approach focuses on compactness while assigning the basic areas of B_{fz} to the sub-problems. A straightforward idea is the assignment of each basic area $i \in B_{fz}$ to the sub-problem that contains its closest basic area j which is not located in the flex-zone, i.e.,

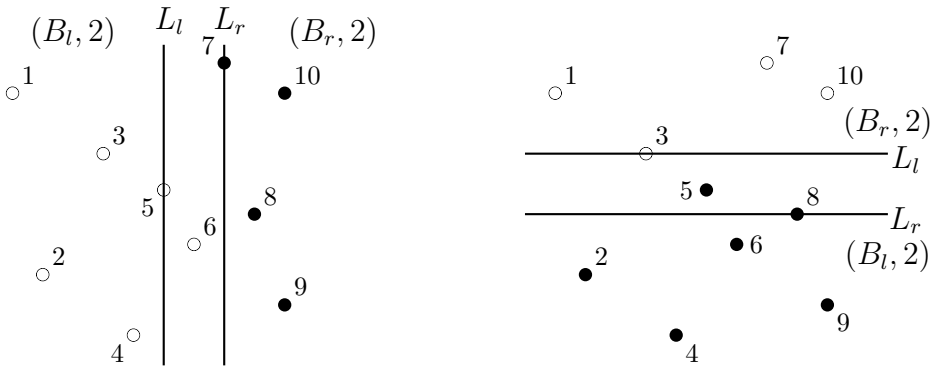
$$B_l \;:=\; B_{ll} \cup \left\{ i \in B_{fz} \;\middle|\; \underset{j \in (B_{ll} \cap B_{rr})}{\arg\min}\; d_{i,j} \in B_{ll} \right\} \tag{4.18}$$

and

$$B_r \;:=\; B_{rr} \cup \left\{ i \in B_{fz} \;\middle|\; \underset{j \in (B_{ll} \cap B_{rr})}{\arg\min}\; d_{i,j} \in B_{rr} \right\}. \tag{4.19}$$

Here, each assignment is based on the initial sets B_{ll} and B_{rr}. This subsection later will present and compare some further ideas for assigning the basic areas of the flex-zone to the sub-problems. Altogether, the flex-zone approach determines the bisecting partition $FZP(k, l^*, r^*, q_l) := (B_l, B_r, q_l, q_r)$, called *flex-zone partition* .

Example 4.3.1 (cont.) Consider the example specified in Table 4.1 and illustrated in Figure 4.3a. Let $\tau = 0.2$. This implies $L_D = 10$, $U_D = 15$, $LL = 20$, and $LU = 30$.



(a) Flex-zone partition for $\alpha = \pi/2$ (b) Flex-zone partition for $\alpha = 0$

Figure 4.5: Flex-zone partitions (Example 4.3.1)

For $\alpha = \pi/2$ (cf. Figure 4.5a), it holds that $l^* = 5$ ($b^k_{l^*} = 5$) and $r^* = 7$ ($b^k_{r^*} = 7$) since $w(\{1;\ldots;4\}) = 16 < 20$, $w(\{1;\ldots;5\}) = 20 \geq 20$, $w(\{1;\ldots;7\}) = 29 \leq 30$, and $w(\{1;\ldots;8\}) = 34 > 30$. The resulting subsets are $B_{ll} = \{1;2;3;4;5\}$, $B_{fz} = \{6;7\}$ and $B_{rr} = \{8;9;10\}$. The flex-zone approach assigns basic area 6 (7) to the left (right) sub-problem since $\arg\min_{j\in(B_{ll}\cap B_{rr})} d_{6,j} = 5 \in B_{ll}$ ($\arg\min_{j\in(B_{ll}\cap B_{rr})} d_{7,j} = 10 \in B_{rr}$) holds. It results in the flex-zone partition $FZP(1,5,7,2) = (\{1;2;3;4;5;6\}, \{7;8;9;10\}, 2, 2)$.
For $\alpha = 0$ (cf. Figure 4.5b), the obtained subsets are $B_{ll} = \{7;1;10;3\}$, $B_{fz} = \{5;8\}$, and $B_{rr} = \{6;2;9;4\}$. The basic areas of the flex-zone are both assigned to the right sub-problem since $\arg\min_{j\in(B_{ll}\cap B_{rr})} d_{5,j} = 6 \in B_{rr}$ and $\arg\min_{j\in(B_{ll}\cap B_{rr})} d_{8,j} = 6 \in B_{rr}$ holds. Thus, the sub-division of the basic areas leads to $BP_l = \{7;1;10;3\}$ and $BP_r = \{5;8;6;2;9;4\}$.

The following subsections will present different definitions of the lines L_l and L_r, and further assignment rules for the basic areas of the flex-zone.

Defining L_l and L_r

As a sub-division that nearly exploits the feasible balance deviation may yield lower flexibility for solving its sub-problems, this subsection introduces two additional approaches of defining the flex-zone a bit more rigorously.

The first approach restricts the feasible balance deviation for each sub-problem (B, q) depending on the number of further sub-divisions and on the maximum activity of one basic area $w^{max} := \max_{i\in B} w_i$. Starting from (BA, p) the number of sub-division levels is given by $\lceil \log_2 p \rceil$. In order to restrict the feasible balance deviation, this approach adds (subtracts) the number of levels multiplied by w^{max} to (from) the minimal (maximal) feasible activity

of the sub-problems induced by L_D (U_D). For example, $q_l \cdot L_d + \lceil \log_2 q_l \rceil \cdot w^{max}$ should be the minimum activity of the left zone.

Unfortunately, the case where the minimal required activity of the left zone is greater than the maximum required activity of the union of left zone and flex-zone may not be excluded, i.e., the sub-problem would be non-solvable. In order to avoid this, this approach defines an upper bound for the minimal activity of the left zone and a lower bound for the maximal activity of the union of the left zone and the flex-zone, such that the former is always smaller than or equal to the latter. Therefore, it uses $\frac{w(B) \cdot q_l}{q_l + q_r} - \frac{w^{max}}{2}$ and $\frac{w(B) \cdot q_l}{q_l + q_r} + \frac{w^{max}}{2}$.

Unfortunately, $\frac{w(B) \cdot q_l}{q_l + q_r} - \frac{w^{max}}{2} < LL$ is possible, i.e., using an upper bound for the minimal required activity of the left zone defined in this way can result in an infeasible sub-problem according to the balance. Analogously, $\frac{w(B) \cdot q_l}{q_l + q_r} + \frac{w^{max}}{2} > LU$ is possible. Therefore, this approach includes LL and LU when defining the described upper and lower bound. Altogether, it defines

$$LL_1 := \min \left\{ \max \left\{ q_l \cdot L_D + lim(q_l); w(B) - q_r \cdot U_D + lim(q_r) \right\}; \max \left\{ \frac{w(B) \cdot q_l}{q_l + q_r} - \frac{w^{max}}{2}; LL \right\} \right\} \quad (4.20)$$

and

$$LU_1 := \max \left\{ \min \left\{ q_l \cdot U_D - lim(q_l); w(B) - q_r \cdot L_D - lim(q_r) \right\}; \min \left\{ \frac{w(B)\, q_l}{q_l + q_r} + \frac{w^{max}}{2}; LU \right\} \right\} \quad (4.21)$$

with $lim(q) = \lceil \log_2 q \rceil \cdot w^{max}$.

Example 4.3.1 (cont.) Continue the example specified in Table 4.1 and illustrated in Figure 4.3a. Here, LL_1 and LU_1 result in $LL_1 = \min \{\max\{29; 29\}, \max\{20.5; 20\}\} = 20.5$ and $LU_1 = \max \{\min\{21; 21\}, \min\{29.5; 30\}\} = 29.5$. In this case, without defining the bounds for the minimal (maximal) activity in the left zone (union of the left zone and flex-zone), $LL_1 > LU_1$ would hold. Figuratively spoken, L_l would be to the right of L_r, i.e., the sub-problem would be non-solvable.

Example 4.3.2 Assume an additional problem, where $w(B) = 200$, $q = 4$, $\tau = 0.2$, and $w^{max} = 9$ holds. This implies $q_l = 2$, $q_r = 2$, $L_D = 40$, $U_D = 60$, $LL = 80$, and $LU = 120$. This leads to $LL_1 = \min \{\max\{89; 89\}, \max\{95.5; 80\}\} = 89$. Moreover, LU_1 results in 111 since $LU_1 = \max \{\min\{111; 111\}, \min\{104.5; 120\}\} = 111$. In this case, the additional definition of an upper (lower) bound for LL_1 (LU_1) is not needed. This also holds for most practical examples since in general the number of basic areas is very large compared to the number of districts and w^{max} is most likely (very) small compared to $w(B)$.

The second approach defines a dynamic feasible deviation that starts with a given start deviation τ_{start} and converges towards the maximum feasible deviation τ. In other words, the higher the number of required sub-problems, the smaller the feasible deviation. Analogously to the previous approach, the approach uses an upper (lower) bound for the minimal (maximal) activity on the left side of L_l (L_r). This implies

$$LL_2 := \min\left\{\max\left\{\mu\cdot(1-\tau(q_l))\cdot q_l; w(B)-\mu\cdot(1+\tau(q_r))\cdot q_r\right\}; \max\left\{\frac{w(B)\cdot q_l}{q_l+q_r}-\frac{w^{max}}{2}; LL\right\}\right\} \quad (4.22)$$

and

$$LU_2 := \max\left\{\min\left\{\mu\cdot(1+\tau(q_l))\cdot q_l; w(B)-\mu\cdot(1-\tau(q_r))\cdot q_r\right\}; \min\left\{\frac{w(B)\cdot q_l}{q_l+q_r}+\frac{w^{max}}{2}; LU\right\}\right\} \quad (4.23)$$

with $\tau(q) = \tau_{start} + (\tau - \tau_{start}) \cdot \frac{\lceil \log_2 p \rceil - \lceil \log_2 q \rceil - 1}{\lceil \log_2 p \rceil - 1}$.

This dynamic deviation requires a consideration in more detail. Assume $\tau = 0.2$, $\tau_{start} = 0.1$ and $p = 10$. Thus, the first sub-division sets $q_l = q_r = 5$, and, hence, as expected it leads to $\tau(q_l) = \tau(q_r) = 0.1 = \tau_{start}$. The next sub-divisions compute $\tau(2) = 0.17$ and $\tau(3) = 0.13$, i.e., the feasible deviations are higher than those of the first sub-division. Finally, for $q_l = 1$ ($q_r = 1$), the deviation results in $\tau(q_l) = 0.2$ ($\tau(q_r) = 0.2$). In this case, no further sub-division is necessary, and, hence, the sub-division can exploit the total feasible deviation.

Example 4.3.1 (cont.) Continue the example depicted in Table 4.1 and Figure 4.3a, and let $\tau_{start} = 0.1$. This implies, $LL_2 = \min\{\max\{22.5; 22.5\}, \max\{20.5; 20\}\} = 20.5$ and $LU_2 = \max\{\min\{27.5; 27.5\}, \min\{29.5; 30\}\} = 29.5$. Again, without defining the upper (lower) bound for the minimal (maximal) activity in the left zone (union of the left zone and flex-zone), L_l would be to the right of L_r.

Example 4.3.2 (cont.) Let $w(B) = 200$, $q = 4$, $\tau = 0.2$ and $w^{max} = 9$, and assume $\tau_{start} = 0.1$. Hence, $LL_2 = \min\{\max\{90; 90\}, \max\{95.5; 80\}\} = 90$ holds. Moreover, it holds that $LU_2 = \max\{\min\{110; 110\}, \min\{104.5; 120\}\} = 110$. So, the flex-zone is a bit more restricted than in the previous approach.

Section 4.4.1 will give some results for the performance of these approaches.

Assigning the Basic Areas of B_{fz}:

The flex-zone approach focuses on compactness while assigning the basic areas of B_{fz} to the two sub-problems. Since there are different compactness measures, there are also different

concepts to assign these basic areas. Note that for the application of the assignment, the usage of road distances instead of Euclidean distances is possible. In this case, obstacles may be regarded implicitly.

1. The first concept assigns each basic area $i \in B_{fz}$ to the sub-problem that contains its closest basic area j which is not located in the flex-zone. See Equations (4.18) and (4.19) for a formal description. Note that this concept always uses the initial sets B_{ll} and B_{rr} to determine the closest basic area. Let $flex_{ca}$ denote this concept.

2. In contrast to the previous concept the second concept updates B_{fz} and B_l or B_r, respectively, after each assignment. It initializes $B_l = B_{ll}$ and $B_r = B_{rr}$. Then, in each iteration, it regards one basic area $i \in B_{fz}$ and assigns it to one sub-problem, i.e., $B_l = B_l \cup \{i\}$ if $\min_{j \in B_l} d_{i,j} \leq \min_{j \in B_r} d_{i,j}$ or $B_r = B_r \cup \{i\}$ otherwise. Furthermore, it deletes i from B_{fz}, i.e., $B_{fz} = B_{fz} \backslash \{i\}$ and alternately chooses the first and the last element of the flex-zone according to the sorted sequence of basic areas. Let $flex_{ca,i}$ denote this concept.

3. The next concept starts with determining the centers of gravity

$$L_{cog} := \left(\frac{\sum\limits_{j \in B_{ll}} w_j \cdot x_j}{\sum\limits_{j \in B_{ll}} w_j}, \frac{\sum\limits_{j \in B_{ll}} w_j \cdot y_j}{\sum\limits_{j \in B_{ll}} w_j} \right) \quad \text{and} \quad R_{cog} := \left(\frac{\sum\limits_{j \in B_{rr}} w_j \cdot x_j}{\sum\limits_{j \in B_{rr}} w_j}, \frac{\sum\limits_{j \in B_{rr}} w_j \cdot y_j}{\sum\limits_{j \in B_{rr}} w_j} \right)$$

for both sub-problems. Then, the assignment decision of a basic area $i \in B_{fz}$ is based on its distance to these centers as well as on the number of districts, the corresponding sub-problem has to be divided into. The latter is necessary in order to prevent that basic areas of the flex-zone are mainly assigned to the sub-problem having the smaller number of districts since usually its center of gravity is located closer to the flex-zone. Formally, the obtained sub-problems are given by

$$B_l := B_{ll} \cup \left\{ i \in B_{fz} \;\middle|\; \frac{d(i, L_{cog})}{q_l} \leq \frac{d(i, R_{cog})}{q_r} \right\} \tag{4.24}$$

and

$$B_r := B_{rr} \cup \left\{ i \in B_{fz} \;\middle|\; \frac{d(i, L_{cog})}{q_l} > \frac{d(i, R_{cog})}{q_r} \right\}. \tag{4.25}$$

Of course, the usage of an unweighted version, i.e., $w_i = 1 \; \forall i$, is possible. If road distances are used, usually the center of gravity is located outside the road network, so

this concept uses the closest basic area (Euclidean distances) to the center of gravity as a proxy. Let $flex_{cog}$ denote this concept.

4. Analogously to the second concept, the next concept updates the center of gravity after each assignment. It alternately assigns the basic areas from the beginning and from the end of the sorted sequence of basic areas. Let $flex_{cog,i}$ denote this concept.

5. The last concept includes the maximum distance to a basic area of each sub-problem as well as the number of districts the sub-problem has to be divided into. The main idea is to assign each basic area $i \in B_{fz}$ to the sub-problem where its furthest basic area is closer. Formally,

$$B_l := B_{ll} \cup \left\{ i \in B_{fz} \;\middle|\; \frac{\max\limits_{j \in B_{ll}} d_{i,j}}{q_l} \leq \frac{\max\limits_{j \in B_{rr}} d_{i,j}}{q_r} \right\} \tag{4.26}$$

and

$$B_r := B_{rr} \cup \left\{ i \in B_{fz} \;\middle|\; \frac{\max\limits_{j \in B_{ll}} d_{i,j}}{q_l} > \frac{\max\limits_{j \in B_{rr}} d_{i,j}}{q_r} \right\} \tag{4.27}$$

describe the subsets. Again, this concept always uses the initial sets B_{ll} and B_{rr}. In this case the usage of the updated sets most likely would result in the same sub-problems since the furthest basic area most likely is not located in the flex-zone. Let $flex_{md}$ denote this concept.

Example 4.3.1 (cont.) Consider the example specified in Table 4.1 again. As described before, the resulting zones are $B_{ll} = \{1; 2; 3; 4; 5\}$, $B_{rr} = \{8; 9; 10\}$ and $B_{fz} = \{6; 7\}$ for $\alpha_k = \pi/2$.

1. The first concept assigns basic area 6 to the left sub-problem and basic area 7 to the right sub-problem (see above).

2. The second concept results in the same sub-division since it firstly assigns 6 to the left sub-problem and afterwards the closest basic area not located in the flex-zone for 7 is still 10, and, hence, this concept assigns 7 to the right sub-problem.

3. The third concept at first determines $L_{cog} = (1.78, 3.23)$ and $R_{cog} = (4.88, 3.36)$. This implies $\frac{d(6,L_{cog})}{2} = 0.94 > \frac{d(6,R_{cog})}{2} = 0.81$ and $\frac{d(7,L_{cog})}{2} = 1.59 > \frac{d(7,R_{cog})}{2} = 1.16$, so both basic areas are assigned to the right sub-problem.

4. The fourth concept determines the same assignment as concept 3 for basic area 6. Then, it updates R_{cog} resulting in $(4.57, 3.17)$. Now $\frac{d(7,L_{cog})}{2} = 1.59 > \frac{d(7,R_{cog})}{2} = 1.20$ holds, and, hence, in this case this concept assigns 7 to the right sub-problem.

5. Finally, for basic area 6 the furthest basic area in B_{ll} is 1 with $d(6,1) = 3.91$ and in B_{rr} it is 10 with $d(6,10) = 2.92$. Since $\frac{3.91}{2} > \frac{2.92}{2}$ holds, the fifth concept again assigns basic area 6 to the right sub-problem. For 7 the furthest basic area in B_{ll} is 4 with $d(7,4) = 4.74$ and in B_{rr} it is 9 with $d(7,9) = 4.12$. Since $\frac{4.74}{2} > \frac{4.12}{2}$ holds, this concept also assigns basic area 7 to the right sub-problem.

These concepts differ in the quality of their solutions in terms of the different compactness measures. Section 4.4.2 will give a further consideration.

4.3.3.3 Set of Bisecting Partitions

For each search direction and each definition of q_l and q_r the algorithm generates a line partition and/or a flex-zone partition. Recall that the parameter K defines the number of search directions. For each search direction $0 \leq k \leq K-1$ the corresponding angle is $\alpha_k := k \cdot \frac{\pi}{K}$.

The question of how to define q_l and q_r is still open. The algorithm objective is to halve the problem, half of the districts should be in the left sub-problem and half of the districts should be in the right sub-problem. Hence, if q is even, the algorithm applies the sub-division $q_l = q_r = \frac{q}{2}$. If q is odd, the algorithm considers the sub-division $q_l = \frac{q+1}{2}$ and $q_r = \frac{q-1}{2}$ as well as the sub-division $q_l = \frac{q-1}{2}$ and $q_r = \frac{q+1}{2}$, i.e., it generates a line partition and/or a flex-zone partition for each search direction for both sub-divisions of q.

4.3.4 Feasibility of Bisecting Partitions

Now, this section addresses the question of how to decide whether a generated bisecting partition is feasible or not.

Definition 4.3.3 *A bisecting partition* (B_l, B_r, q_l, q_r) *is feasible if and only if* (B_l, q_l) *and* (B_r, q_r) *are feasible.*

Since each district needs at least one basic area, for each partition problem, the number of basic areas must be greater than or equal to the number of districts. Moreover, since balance is treated as a hard criterion, each partition problem must be feasible in terms of balance.

A partition problem ist feasible if the average size of the districts is in the interval between L_D and U_D. This leads to the following definition.

Definition 4.3.4 *A partition problem* (B, q) *is feasible if* $|B| \geq q$ *and* $L_D \leq \frac{w(B)}{q} \leq U_D$ *holds.*

4.3.5 Choosing a Bisecting Partition

Finally, this section explains how to evaluate a bisecting partition in terms of the planning criteria and how to choose a bisecting partition out of the set of generated bisecting partitions. In the following, let a partition problem PP and a corresponding bisecting partition BP be given.

4.3.5.1 Evaluating Balance

Following Section 4.2.2.2 the balance of a district is defined as relative percentage deviation of its activity from the average activity μ. Thus, the algorithm straightforwardly defines the balance of a partition problem as

$$bal(B, q) := \frac{|w(B) - q \cdot \mu|}{q \cdot \mu} .$$

Moreover, it defines the balance of a bisecting partition as the maximal balance of one of its sub-problems since the balance of a solution is in defined as maximum balance of one district, see Equation (4.1), i.e.,

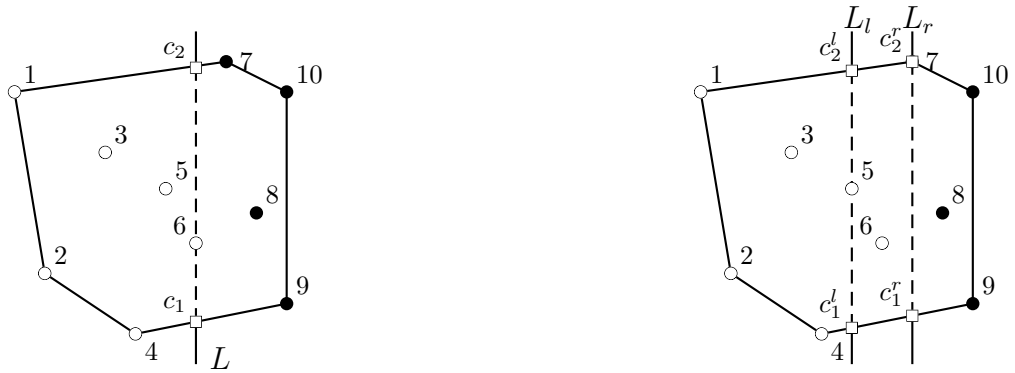
$$bal(BP) := \max\{bal(B_l, q_l); bal(B_r, q_r)\} .$$

4.3.5.2 Evaluating Compactness

The compactness measures stated in Section 4.2.2.3 only work for the final districts. Hence, the algorithm has to adapt them or find surrogates in order to evaluate partition problems.

Length of Intersection

The first approach is the one already proposed by Kalcsics et al. [16]. It is not directly related to any of the measures described in Section 4.2.2.3, but based on the measure proposed by Bozkaya et al. [3]. They use the total length of all boundaries between the districts (cf. Section 3.3.3.2). Here, the basic areas are represented by points and not by polygons. Thus, the districts' boundaries are not given directly. Therefore, this approach uses the



(a) Line partition

(b) Flex-zone partition

Figure 4.6: Measuring the compactness according to the length of intersection

sub-dividing line(s) as a proxy. It determines the length of intersection between this (these) line(s) and the convex hull $ch(B)$ of the basic areas comprising B. By making this (these) intersection(s) short, the approach hopes to end up with a small total border length and therefore with a compact plan. It distinguishes line partitions and flex-zone partitions:

Line Partition: This compactness measure uses the line $L(b^k_{a^*}, \alpha_k)$ in order to evaluate the line partition $LP(k, a^*, q_l)$. Note that by convexity, L intersects $ch(B)$ in at most two points c_1 and c_2, where $c_1 = c_2$ is possible. The Euclidean distance between c_1 and c_2 defines the length of intersection between L and $ch(B)$, and, hence, the compactness of LP:

$$comp_{loi}(LP) := l_2(c_1,\, c_2).$$

Flex-Zone Partition: This compactness measure uses the lines $L_l(b^k_{l^*}, \alpha_k)$ and $L_r(b^k_{r^*}, \alpha_k)$ in order to evaluate the flex-zone partition $FZP(k, l^*, r^*, q_l)$. It determines the points of intersection c_1^l and c_l^2, and c_1^r and c_2^r, respectively, with $ch(B)$. The average length of intersection of $ch(B)$ with L_l and L_r defines the compactness of the flex-zone partition:

$$comp_{loi}(FZP) \;:=\; \frac{1}{2} \cdot \left(l_2(c_1^l,\, c_2^l) + l_2(c_1^r,\, c_2^r) \right).$$

Example 4.3.1 (cont.) Figure 4.6 illustrates this measure for the bisecting partitions depicted in Figure 4.3b and Figure 4.5a. The length of the dashed line in Figure 4.6a corresponds to the compactness of the line partition, whereas the average length of the two dashed lines in Figure 4.6b corresponds to the compactness of the flex-zone partition.

Weighted Moment of Inertia

The next approach is based on the Weighted Moment of Inertia. This compactness measure defined in Equation (4.2) is based on the distances to a center. Since it is too time consuming to approximate q representative points as centers, this approach restricts itself to one representative point as center of a partition problem. This representative point is the basic area cen^a_{PP} that is the closest to the center of gravity of B, i.e.,

$$cen^a_{PP} := \arg\min_{i \in B} \left[l_2 \left(b_i, (\sum_{j \in B} \frac{w_j \cdot x_j}{w(B)}, \sum_{j \in B} \frac{w_j \cdot y_j}{w(B)}) \right) \right].$$

This computation can be done in $\mathcal{O}(|B|)$ time.

As a consequence of this, the compactness of a partition problem is given by

$$comp_{wmoi}(B,\, q) := \sum_{i \in B} w_i \cdot d^2(b_i, b_{cen^a_{PP}}) \,.$$

The compactness of a bisecting partition $BP = (B_l,\, B_r,\, q_l,\, q_r)$ is defined straightforwardly as the sum of the compactness values of its sub-problems, i.e.,

$$comp_{wmoi}(BP) := comp_{wmoi}(B_l,\, q_l) + comp_{wmoi}(B_r,\, q_r) \,.$$

In addition, an unweighted version can be defined by setting $w_i = 1\ \forall i$. This leads to

$$cen^a_{PP} := \arg\min_{i \in B} \left[l_2 \left(b_i, (\sum_{j \in B} \frac{x_j}{|B|}, \sum_{j \in B} \frac{y_j}{|B|}) \right) \right]$$

and

$$comp_{moi}(B,\, q) := \sum_{i \in B} d^2(b_i, b_{cen^a_{PP}}) \,,$$

and

$$comp_{moi}(BP) := comp_{moi}(B_l,\, q_l) + comp_{moi}(B_r,\, q_r) \,.$$

Remark 4.3.2 For Euclidean distances, the center of gravity $(\sum_{j \in B} \frac{w_j \cdot x_j}{w(B)}, \sum_{j \in B} \frac{w_j \cdot y_j}{w(B)})$ minimizes the function $\arg\min_{(x,y) \in \mathbb{R}^2} \left[\sum_{j \in B} w_j \cdot d^2\,(b_j, (x, y)) \right]$.

Example 4.3.1 (cont.) Figure 4.3b shows a bisecting partition for the basic areas specified in Table 4.1, where $B_l = \{1; 2; 3; 4; 5; 6\}$ and $B_r = \{7; 8; 9; 10\}$. There, the center of gravity of the left sub-problem is the point $(2.17, 3.05)$. Obviously, $cen^a_{(B_l,\, q_l)}$ corresponds to basic area 5. For the right sub-problem, the point $(4.77, 3.63)$ is the center of gravity and $cen^a_{(B_r,\, q_r)}$ corresponds to basic area 8. Computing the Weighted Moment of Inertia results in $comp_{wmoi}(B_l,\, B_r,\, q_l,\, q_r) = comp_{wmoi}(B_l, q_l) + comp_{wmoi}(B_r, q_r) = 99.73 + 75.25 = 174.98$.

Figure 4.3c illustrates another bisecting partition for the same set of basic areas. Here, the subsets are $B_l = \{1; 3; 5; 7; 10\}$ and $B_r = \{2; 4; 6; 8; 9\}$. The corresponding centers of gravity are the points $(3.18, 4.64)$ and $(3.66, 2.00)$. Thus, $cen^a_{(B_l,\, q_l)} = 7$ and $cen^a_{(B_r,\, q_r)} = 6$ holds, and, hence, $comp_{wmoi}(B_l,\, B_r,\, q_l,\, q_r) = 120.39 + 62.74 = 183.13$.

Hence, comparing these results, with respect to the Weighted Moment of Inertia, the line partition obtained for the angle $\alpha = \pi/2$ is better than the line partition obtained for the angle $\alpha = 0$.

Pairwise Distances

Another approach is based on the Pairwise Distances between the basic areas of the same district defined in Equations (4.4) and (4.5). Adding up all (weighted) distances for a partition problem requires $\mathcal{O}(|B|^2)$ time. However, if $q > 1$ holds, this sum contains many distances between basic areas which are not in the same district in the final solution.

Therefore, for each basic area i this approach only sums up the distances to a number of its closest basic areas within this partition problem. Figuratively spoken, it determines a "good" district for this basic area with respect to the Pairwise Distances. To that end, $b^i_1, b^i_2, \ldots, b^i_n$ denote the sorted sequence of basic areas of B with respect to their distance to basic area i. Note that the first element of this sequence is the considered basic area itself. A "good" district has a size that is approximately equal to the average size within the partition problem. Based on this idea, this approach includes the closest basic areas such that the corresponding sum of activities is just smaller than or equal to the average activity of a district. Formally,

$$\sum_{j=1}^{\eta(i)} w_{b^i_j} \leq \frac{w(B)}{q} \quad \text{and} \quad \sum_{j=1}^{\eta(i)+1} w_{b^i_j} > \frac{w(B)}{q}$$

defines the number $\eta(i)$ of considered basic areas for i.

Finally, this approach evaluates the compactness of a partition problem by

$$comp_{pd}(B_l,\, q_l) := \sum_{i \in B} \sum_{j=2}^{\eta(i)} d(b_i, b_j^i)$$

or

$$comp_{wpd}(B_l,\, q_l) := \sum_{i \in B} \sum_{j=2}^{\eta(i)} w_i \cdot w_{b_j^i} \cdot d(b_i, b_j^i)\,,$$

respectively. Moreover, it evaluates the compactness of a bisecting partition as the sum of the compactness evaluations of its sub-problems, i.e.,

$$comp_{pd}(BP) := comp_{pd}(B_l,\, q_l) + comp_{pd}(B_r,\, q_r)$$

or

$$comp_{wpd}(BP) := comp_{wpd}(B_l,\, q_l) + comp_{wpd}(B_r,\, q_r)\,,$$

respectively. Since this measure does not have to sort the closest basic areas, $comp_{pd}(PP)$ or $comp_{wd}(PP)$, respectively. can be computed in $\mathcal{O}(|B|^2)$ time, see Hochbaum [13].

Remark 4.3.3 The values of $comp_{pd}(PP)$ and $comp_{wpd}(PP)$ are no lower bounds for the compactness of the final solution for PP.

Example 4.3.1 (cont.) Consider the example illustrated in Figure 4.3b again, where $B_l = \{1; 2; 3; 4; 5; 6\}$. Here, the corresponding average size of a district is $\frac{w(B_l)}{q_l} = \frac{26}{2} = 13$.

For example, for basic area 1 sorting the basic areas according to their distances to basic area 1 leads to the sequence $1, 3, 5, 2, 6, 4$. This implies $\sum_{j=1}^{3} w_{b_j^1} = 5 + 4 + 4 = 13 \leq 13$ and $\sum_{j=1}^{4} w_{b_j^1} = 5 + 4 + 4 + 3 = 16 > 13$, and, hence, $\eta(1) = 3$.
As a further example, sorting the basic areas concerning basic area 5 results in $5, 6, 3, 2, 4, 1$. This leads to $\eta(5) = 2$ since $\sum_{j=1}^{2} w_{b_j^5} = 4+6 = 10 \leq 13$ and $\sum_{j=1}^{3} w_{b_j^5} = 4+6+4 = 14 > 13$ holds.

Finally, the left sub-problem evaluates $comp_{wpd}(B_l,\, q_l) = 335.37$, and the right sub-problem evaluates $comp_{wpd}(B_r,\, q_r) = 140.87$. Hence, the compactness of the bisecting partition results in $comp_{wpd}(B_l,\, B_r,\, q_l,\, q_r) = 335.37 + 140.87 = 476.24$.

Evaluating the bisecting partition depicted in Figure 4.3c results in the Weighted Pairwise Distances 328.72. Thus, with respect to the Weighted Pairwise Distances the line partition obtained for $\alpha = 0$ is better than the one obtained for $\alpha = \pi/2$.

Maximum Distance

Finally, the last approach regards the measure based on the maximum distance between two basic areas of the same district, stated in Equation (4.6). Consequently, in order to define a measure for a partition problem this approach incorporates the maximum distance between two basic areas within this partition problem. In order to approximate the maximum distance within a final district this measure divides this distance by the root of the number of districts the problem has to be divided into, i.e.,

$$comp_{md}(B,\, q) := \frac{\max\limits_{i,j \in B} d_{i,j}}{\sqrt{q}}\,.$$

Then, it evaluates the compactness of a bisecting partition as the maximum compactness evaluation of one sub-problem, i.e.,

$$comp_{md}(BP) := \max\{comp_{md}(B_l,\, q_l); comp_{md}(B_r,\, q_r)\}\,.$$

This computation is time consuming since it can be made in $\mathcal{O}(|B|^2)$ time. For Euclidean distances the running time can be reduced significantly utilizing that the maximal distance between two points of a set B corresponds to the maximal distance between two vertices of the convex hull $ch(B)$. However, the complexity is still $\mathcal{O}(|B|^2)$ since it can occur that $ch(B)$ contains all points of B as vertices.

Example 4.3.1 (cont.) Consider the bisecting partitions depicted in Figure 4.3b and Figure 4.3c once again, where $q_l = q_r = 2$.

First, for the line partition obtained for $\alpha = \pi/2$ the maximum distance between two basic areas in B_l (B_r) is 4.56 (4.50) between basic area 1 (7) and basic area 4 (9). This implies $comp_{md}(B_l,\, B_r,\, q_l,\, q_r) = \max\{\frac{4.56}{\sqrt{2}}; \frac{4.50}{\sqrt{2}}\} = 3.22$.
Next, for $\alpha = 0$ the basic areas 1 (2) and 10 (9) induce the maximum distance between two basic areas in the left (right) sub-problem. This implies $d_{1,10} = 4.50$ and $d_{2,9} = 4.03$. This results in $comp_{md}(B_l,\, B_r,\, q_l,\, q_r) = \max\{\frac{4.50}{\sqrt{2}}; \frac{4.03}{\sqrt{2}}\} = 3.18$. Hence, according to the maximum distance $\alpha = 0$ performs better than $\alpha = \pi/2$.

4.3.5.3 Evaluating Contiguity

The contiguity of a solution is based on the area of intersection between the convex hulls of its districts. Hence, an obvious approach to define the contiguity of a bisecting partition is the usage of the area of intersection between the convex hulls of the sub-problem, i.e.,

$$ctg(BP) := area(ch(B_l) \cap ch(B_r)) \,.$$

A line partition LP generates two non-overlapping sub-problems, i.e., it always holds that $ctg(LP) = 0$. The sub-problems generated by the flex-zone approach may intersect. However, these intersections are always fairly small, as Section 4.4.3 will show. For that reason, the algorithm usually does not explicitly evaluate contiguity.

4.3.5.4 Ranking

Among the feasible bisecting partitions in FBP, the RPA chooses the "best" one and implements it. Since some introduced measures determine absolute values (e.g. compactness), whereas others determine relative values (e.g. balance) the obtained results have to be normalized in order to make them comparable in a ranking function. To that end, for each applied measure $meas_m$ the RPA determines the minimal and maximal values

$$meas_m^{min} := \min_{BP \in FBP} meas_m(BP) \quad \text{and} \quad meas_m^{max} := \max_{BP \in FBP} meas_m(BP)$$

in order to scale the evaluation values. Let MEA denote the set of used measures.

The ranking values of a bisecting partition $BP \in FBP$ is a weighted combination of these scaled values:

$$rk(BP) := \sum_{m=1}^{|MEA|} \beta_m \cdot \frac{meas_m(BP) - meas_m^{min}}{meas_m^{max} - meas_m^{min}} \,, \tag{4.28}$$

where $meas_1, \ldots, meas_{|MEA|}$ are the applied measures and $\beta_1, \ldots, \beta_M$ are user-given weighting factors with $\sum_{m=1}^{|MEA|} \beta_m = 1$ and $\beta_m \geq 0 \ \forall m$. As from a theoretical point of view all bisecting partitions can be evaluated equally in terms of one criterion, the algorithm applies $0/0 =: 0$. Finally, the algorithm sorts the bisecting partitions of FBP in non-decreasing order of their ranking value and implements $BP^* := \arg\min_{BP \in FBP} rk(BP)$, i.e., the best ranked bisecting partition.

4.3.6 Exploring the Set of Partition Problems

The previous section has explained how to generate and rank bisecting partitions. The straightforward "greedy" approach that just chooses the best bisecting partition according to this ranking is, however, sometimes not sufficient. Even though the algorithm only chooses feasible bisecting partitions, there is no guarantee that it does not develop an infeasible sub-problem later. In order to overcome this problem, the RPA includes a backtracking mechanism that allows to revisit an already solved partition problem. There, it revises the sub-division decision and chooses the next best bisecting partition according to the ranking, and continues with it. Thus, each partition problem has a counter $pos(PP)$ that marks the currently implemented bisecting partition in the sorted list of bisecting partitions.

Without backtracking the RPA solves $2p-1$ partition problems until the districting problem is finally solved. However, due to backtracking operations, this number can be much larger since it is exponential in K and p in general. For this reason, it is necessary to limit the search. Unfortunately, there is no guarantee the RPA generates a feasible solution, but instead of reporting no result the RPA reports an infeasible solution in this case. The generation of this solution is based on a relaxation of the balance. After a given number $PPMax$ of examined partition problems, the RPA decreases L_D and increases U_D such that the difference between U_D and L_D is doubled. However, the RPA does not restart at this point, i.e., the relaxed bounds are only applied to solve the currently unsolved or newly generated partition problems. In the worst case there is no solution after $PPMax$ further solved partition problems. The RPA then repeats this relaxation until a given maximal number $RelMax$ of relaxations is reached. At this point, the RPA sets $L_D = 0$ and $U_D = \infty$, i.e., from now on all bisecting partitions are feasible with respect to the balance. Hence, the algorithm performs no more backtracking and terminates quickly. According to Kalcsics [15] $PPMax = 10p$ and $RelMax = 3$ are suitable values. Recall that p denotes the number of required districts.

The RPA does not specify the sequence of solving the problems in UPP. Our practical implementation applies a *first-in first-out* strategy. Nevertheless, further strategies such as a *last-in first-out* strategy or a random based strategy are possible. However, if no backtracking occurs, the solutions are identical. Only if backtracking is necessary, the solutions can differ.

Algorithm 4.3.2 summarizes the Recursive Partitioning Algorithm including the described backtracking mechanism.

Algorithm 4.3.2: The Recursive Partitioning Algorithm

Input: *Set of basic areas BA, number of districts p, a set of measures MEA, a set of approaches to determine bisecting partitions PA, parameters τ, L_D, U_D, K, $\beta_1, \ldots, \beta_{|MEA|}$, $PPMax$, $RelMax$.*

Output: *Districting plan $S = \{D_1; \ldots; D_p\}$.*

1 *Set $UPP = \{(BA, p)\}$, $S = \emptyset$, $pos(BA, p) = 0$, $PPCtr = 0$, and $RelCtr = 0$.*

while $UPP \neq \emptyset$ **do**
  Choose $PP = (B, q) \in UPP$ and set $PPCtr = PPCtr + 1$.
2 **if** $q = 1$ **then**
    Set $S = S \cup \{B\}$, $UPP = UPP \backslash \{PP\}$, and GOTO 5.
3 **if** $pos(PP) = 0$ **then**
    Determine $\mathcal{FBP}$ depending on K and PA.
    Rank the bisecting partitions in $\mathcal{FBP}$ according to Equation (4.28) using $\beta_1, \ldots, \beta_{|MEA|}$ and MEA.
  end
4 **if** $|FBP| > pos(PP)$ **then**
    Set $pos(PP) = pos(PP) + 1$.
    Choose the $pos(PP)$-th ranked bisecting partition $BP^* = (B_l^*, B_r^*, q_l^*, q_r^*)$ in $\mathcal{FBP}$.
    Set $UPP = UPP \backslash \{PP\} \cup \{(B_l^*, q_l^*); (B_r^*, q_r^*)\}$.
  else
    if $PP = (BA, p)$ **then**
      if $relCtr \geq RelMax$ **then**
        Set $L_D = 0$ and $U_D = \infty$.
      else
        Set $L_D = \max\{0; L_D - (U_D - L_D)/2\}$; $U_D = U_D + (U_D - L_D)/2$.
        Set $RelCtr = RelCtr + 1$.
      end
      Set $pos(PP) = 0$ and $PPCtr = 0$.
    else
      Set $UPP = (UPP \backslash Des(PP_f)) \cup \{PP_f\}$.
  end
5 **if** $PPCtr = PPMax$ **then**
    if $relCtr \geq RelMax$ **then**
      Set $L_D = 0$ and $U_D = \infty$.
    else
      Set $L_D = \max\{0; L_D - (U_D - L_D)/2\}$; $U_D = U_D + (U - L_D)/2$.
      Set $RelCtr = RelCtr + 1$.
    Set $PPCtr = 0$.
  end
end
6 **return** S.

4.3.7 Complexity

This subsection closes with an analysis of the complexity of the RPA.

4.3.7.1 Complexity of Determining FBP

The analysis starts with the approaches of generating bisecting partitions. For a partition problem (B, q) the sorted sequence $b_1^k, \ldots, b_n^k$ of basic areas according to an angle α_k can be computed in $\mathcal{O}(|B| \cdot \log |B|)$ time. The further computation depends on the kind of bisecting partitions.

Line Partitions

After sorting the, basic areas computing the line partition $LP(k, a^*, q_l)$ requires $\mathcal{O}(|B|)$ time. So, the total computation of a line partition can be done in $\mathcal{O}(|B| \cdot \log |B|)$ time.

Flex-Zone Partitions

On a sorted sequence of basic areas determining B_{ll}, B_{fz} and B_{rr} requires $\mathcal{O}(|B|)$ time. The assignment of the basic areas of the flex-zone depends on the applied assignment concept, see Section 4.3.3.2.
Computing the closest or furthest basic area within one sub-problem requires $\mathcal{O}(|B|)$ time for each basic area. Since there is no general restriction of the number of basic areas located in the flex-zone, the total computation of a flex-zone partition $FZP(k, l^*, r^*, q_l)$ can be done in $\mathcal{O}(|B|^2)$ time for the concepts $flex_{ca}$, $flex_{ca,i}$ and $flex_{md}$.
Computing the center of gravity of one sub-problem requires $\mathcal{O}(|B|)$ time. Moreover, deciding which center of gravity is closer to a basic area can be done in $\mathcal{O}(1)$ time. Since the concept $flex_{cog}$ determines the center of gravity only once for each sub-problem, the total computation time for this concept is $\mathcal{O}(|B| \cdot \log |B|)$.
In contrast to this, the concept $flex_{cog,i}$ determines a new center of gravity for the corresponding sub-problem after each assignment. Thus, in this case, the total computation time is $\mathcal{O}(|B|^2)$.

Although the worst case complexity for the most concepts of the flex-zone approach is larger than for line partitions, from a practical point of view the running times are still good (see Section 4.4.3). On the one hand, B_{fz} typically only contains a small subset of B. On the other hand, for each basic area the sorted list of basic areas according to the distance to this basic area can be stored after computing it for the first time.

4.3.7.2 Complexity of Choosing a Bisecting Partition

The RPA evaluates each bisecting partition in terms of each applied measure. Hence, the following analysis addresses the complexity of the presented measures.

Balance

Evaluating balance takes $\mathcal{O}(|B|)$ time.

Length of Intersection

Computing the convex hull of B can be done in $\mathcal{O}(|B| \cdot \log |B|)$ time, see Klein [17]. Intersecting the line(s) with the hull requires $\mathcal{O}(|B|)$ time. Hence, evaluating a bisecting partition in terms of the length of intersection takes $\mathcal{O}(|B| \cdot \log |B|)$ time.

Weighted Moment of Inertia

Determining the center of gravity as well as computing its closest basic areas can be done in $\mathcal{O}(|B|)$ time. Moreover, computing the sum of the distances to this center needs $\mathcal{O}(|B|)$ time. Hence, in total, evaluating a bisecting partition in terms of the Weighted Moment of Inertia requires $\mathcal{O}(|B|)$ time.

Pairwise Distances

As described in Section 4.3.5.2, the Pairwise Distances can be computed in $\mathcal{O}(|B|^2)$ time.

Maximum Distance

Determining the Maximum Distance needs $\mathcal{O}(|B|^2)$ time, as described in Section 4.3.5.2.

Contiguity

For each sub-problem, determining the convex hull requires $\mathcal{O}(|B| \cdot \log |B|)$ time. Computing the area of intersection between two convex polygons ch_1 and ch_2 needs $\mathcal{O}(p_1 + p_2)$ time, where p_1 and p_2 are the number of vertices of ch_1 and ch_2. Here, the number of vertices is limited to the number of basic areas, i.e., $ch(B_l)$ ($ch(B_r)$) has at most $|B_l|$ ($|B_r|$) vertices. Since $|B_l| + |B_r| = |B|$, this measure requires $\mathcal{O}(|B| \cdot \log |B|)$ time.

For given results of the single measures, determining all ranking values can be done in $\mathcal{O}(K)$ time. Finally, sorting the bisecting partitions according to their ranking values requires $\mathcal{O}(K \cdot \log K)$ time.

4.3.7.3 Overall Complexity

Finally, this subsection analyzes the complexity of the entire algorithm. The most time consuming operation is the generation and the ranking of all feasible bisecting partitions of a partition problem.
$T(B)$ denotes the complexity of computing and evaluating one bisecting partition for one partition problem. $T(B)$ depends on the approach of generating the bisecting partition as well as on the applied evaluation measures afterwards. Table 4.2 gives an overview for different combinations of generating and evaluating bisecting partitions.

	bal	$comp_{loi}$	$comp_{wmoi}$ $comp_{moi}$	$comp_{pd}$ $comp_{wpd}$	$comp_{md}$	ctg
$flex_*$	$\mathcal{O}(\|B\|^2)$	$\mathcal{O}(\|B\|^2)$	$\mathcal{O}(\|B\|^2)$	$\mathcal{O}(\|B\|^2)$	$\mathcal{O}(\|B\|^2)$	$\mathcal{O}(\|B\|^2)$
$flex_{cog}$	$\mathcal{O}(\|B\| \cdot \log \|B\|)$	$\mathcal{O}(\|B\| \cdot \log \|B\|)$	$\mathcal{O}(\|B\| \cdot \log \|B\|)$	$\mathcal{O}(\|B\|^2)$	$\mathcal{O}(\|B\|^2)$	$\mathcal{O}(\|B\| \cdot \log \|B\|)$
$line$	$\mathcal{O}(\|B\| \cdot \log \|B\|)$	$\mathcal{O}(\|B\| \cdot \log \|B\|)$	$\mathcal{O}(\|B\| \cdot \log \|B\|)$	$\mathcal{O}(\|B\|^2)$	$\mathcal{O}(\|B\|^2)$	$\mathcal{O}(\|B\| \cdot \log \|B\|)$

$* \in \{ca;\ ca,i;\ cog,i;\ md\}$

Table 4.2: Complexity of generating and evaluating a bisecting partition

In general, more than one measure is applied and sometimes more than one kind of bisecting partitions is used. Thus, $T(B)$ is the maximal entry of the corresponding combinations. Hence, for a partition problem generating and ranking all feasible bisecting partitions and choosing the best one requires $\mathcal{O}(K \cdot T(BA) + K \cdot \log K)$ time, where K is the number of different search directions. In order to determine the overall complexity, two cases are distinguished:

1. $L_D = 0$ and $U_D = \infty$: In this case, no backtracking occurs. Let the root problem be on sub-division level 0, its left and right sub-problem on level 1, and so on. For each sub-division level l, the sets of basic areas B_s^l, $1 \leq s \leq S$, of the partition problems $PP_1^i, \ldots, PP_S^i$, $S \leq 2^l$, are pairwise disjoint. Hence, generating the feasible bisecting partitions of all partition problems on level i and determining their ranking value takes $\mathcal{O}(K \cdot T(B_1^l) + \ldots + K \cdot T(B_S^l)) = \mathcal{O}(K \cdot T(BA))$ time. There are at most $\log p$ sub-division levels and at most $2p - 1$ considered partition problems. For each partition problem these bisecting partitions have to be sorted. Thus, the time complexity of the algorithm results in $\mathcal{O}(K \cdot T(BA) \cdot \log p + p \cdot K \cdot \log K)$.

2. $L_D > 0$ and $U_D < \infty$: In this case, backtracking could occur. The complexity depends on the actual number of partition problems explored in the search for a feasible solution. For choosing $PPMax = 10p$ and $RelMax = 3$, the maximal number of examined sub-divisions is linear in p and the time complexity results in $\mathcal{O}(p \cdot K \cdot (T(BA) + \log K))$.

4.4 Computational Results

This section presents the results of our computational tests. First, note the technical conditions: The algorithm was coded in C++ and executed on a PC running Windows 7 with a Pentium(R) $E5500$ processor with 2.80 GHz and 2 GB RAM. The tests are mainly conducted on two datasets. The first one, denoted by *PPS*, is based on real-world data and is provided by a project partner. Here, the basic areas correspond to customer locations and the associated activity measures to the expected service times. Furthermore, the number of required districts is part of the input. This dataset contains 33 test instances where the number of basic areas varies from 284 to 4971 while the number of required districts varies from 2 to 50. Moreover, for 23 of these instances street distances between the basic areas are available and for 12 of them travel times between the basic areas are available. The second dataset, denoted by *ZCA*, contains 50 test instances based on German zip-code areas. For each zip-code area, its center of gravity defines the location of the corresponding basic area and its number of inhabitants defines the activity measure. The number of basic areas varies from 94 to 1036. Here, we have determined 5 to 10 districts for each instance. A distinction between these two data sets is the range of activity measures for each instance, which is much larger for *ZCA* than for *PPS*. For *ZCA* (*PPS*), the average ratio between the largest and the smallest activity measure in a problem instance is 453.7 (13.9).

This section compares the solutions obtained by the usage of different parameter settings in terms of balance, compactness, contiguity and running time. Before evaluating the results, a detailed description of the evaluation parameters is necessary.
Throughout this section, the presented results in terms of balance and compactness are average values over all instances. In terms of balance, bal_{max} denotes the maximum balance defined in Equation (2.2) and bal_{ave} the average balance defined in Equation (2.3). For purposes of readability, the results are stated as percentage values, i.e., an entry of 4.00 describes a balance of 4% or $bal(\cdot) = 0.04$, respectively.
In terms of compactness, $comp_{moi}$ denotes the Moment of Inertia, $comp_{wmoi}$ the Weighted Moment of Inertia, $comp_{pd}$ the Pairwise Distances, and $comp_{wpd}$ the Weighted Pairwise Distances (cf. Section 3.3.5). Since these measures have absolute values as outcomes, the results are stated in relation to a reference solution. This reference solution is the result of the basic version of the RPA using line partitions exclusively (cf. Section 4.3.3) and the length of intersection as compactness measures (cf. Section 4.3.5.2). For example, an entry of -5.00 describes an improvement of 5% compared to the reference solution.
In terms of contiguity, ctg_{ave} denotes the average contiguity over all instances, while ctg_{max} denotes the maximum contiguity of one single instance. The contiguity measure is defined in Equation (2.2.4). For purposes of readability, the results are stated as percentage values

as well.
Finally, in terms of running times, an entry states the total time in seconds necessary to solve all instances.

Unless specified otherwise, we use the following parameter settings: $\tau = 0.05$, $K = 8$, $PPMax = 10p$, and $RelMax = 3$.

4.4.1 Flex-Zone Bounds

This test compares the different approaches to define LL and LU while using flex-zone partitions. The objective is to determine the best approach that implies the best results. The first approach, denoted by $V1$ does not further restrict the bounds induced by the maximum feasible deviation τ (cf. Equations (4.14) and (4.15)). The second approach restricts the bounds depending on the number of further sub-divisions and the maximum weight of one basic area. It is denoted by $V2$ and introduced in Equations (4.20) and (4.21). The third approach, denoted by $V3$, uses a deviation starting with τ_{start} and converging against the maximum feasible deviation τ (cf. Equations (4.22) and (4.23)). This test incorporates two different values of τ_{start}: $\tau_{start} = \frac{1}{2} \cdot \tau = 0.025$ denoted by $V3$-1, and $\tau_{start} = \frac{1}{10} \cdot \tau = 0.005$ denoted by $V3$-2.
The set of bisecting partitions exclusively consists of flex-zone partitions using the $flex_{cog}$ concept (cf. Equations (4.24) and (4.25)). Since the flex-zone approach is mainly constructed in order to improve compactness, the realization of a bisecting partition is only based on compactness. Hence, balance is only a hard criterion in this case.

Tables 4.3 and 4.4 present the results while using different compactness measures for evaluating bisecting partitions, namely, the Moment of Inertia denoted by MoI (rows 1 to 4), the Weighted Moment of Inertia denoted by WMoI (rows 5 to 8), the Pairwise Distances (rows 9 to 12) denoted by PD, and the Weighted Pairwise Distances denoted by WPD (rows 13 to 16).

For *PPS*, the approach $V3$ is slightly better than the others in terms of balance. However, for *ZCA*, there is no clear trend whether $V2$ or $V3$ performs better in terms of balance. Comparing the variations of $V3$ there are small advantages for $V3$-2.
In terms of contiguity, the results are ambiguous. However, $V3$ performs well in any case. Moreover, $V3$-2 performs better than $V3$-1.
Finally, in terms of compactness $V3$ outperforms the competing approaches. Comparing $V3$-1 and $V3$-2, $V3$-1 implies noticeably better results. For example, for *PPS* using the Weighted Moment of Inertia during the execution of the RPA and $V3$-1, the results in terms of the Weighted Moment of Inertia are 6.48% better than the reference solutions, whereas

compactness measure	bound	*bal*		*comp*				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
MoI	$V1$	5.06	4.04	**−0.56**	−0.90	−1.49	−1.47	0.008	0.107
MoI	$V2$	4.97	3.55	**−5.29**	−5.08	−3.55	−3.21	0.006	0.107
MoI	$V3$-1	4.76	3.15	**−6.06**	−6.09	−3.93	−3.70	0.006	0.107
MoI	$V3$-2	4.49	2.92	**−5.19**	−5.28	−3.44	−3.20	0.005	0.020
WMoI	$V1$	5.03	4.05	0.05	**−1.09**	−1.40	−1.75	0.107	0.008
WMoI	$V2$	4.97	3.57	−3.97	**−4.71**	−3.08	−3.17	0.107	0.007
WMoI	$V3$-1	4.72	3.12	−5.87	**−6.48**	−3.99	−3.98	0.107	0.006
WMoI	$V3$-2	4.49	2.94	−5.05	**−5.66**	−3.37	−3.42	0.025	0.003
PD	$V1$	4.81	3.64	−3.95	−4.01	**−3.87**	−3.50	0.573	0.036
PD	$V2$	4.78	3.22	−7.37	−7.43	**−5.45**	−5.02	0.107	0.013
PD	$V3$-1	4.58	2.98	−8.74	−8.82	**−6.05**	−5.55	0.244	0.023
PD	$V3$-2	4.50	2.98	−8.77	−8.40	**−5.83**	−5.26	0.025	0.002
WPD	$V1$	4.94	3.32	−7.45	−8.10	−5.30	**−5.44**	0.204	0.019
WPD	$V2$	4.93	3.33	−7.14	−7.93	−5.22	**−5.48**	0.204	0.017
WPD	$V3$-1	4.76	3.09	−9.26	−10.15	−5.87	**−6.18**	0.295	0.031
WPD	$V3$-2	4.54	3.33	−8.20	−8.32	−5.25	**−5.40**	0.204	0.017

Table 4.3: Dataset *PPS*: Comparing flex-zone bounds

compactness measure	bound	*bal*		*comp*				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
MoI	$V1$	4.79	3.55	**0.72**	1.50	−3.08	−0.15	0.744	0.038
MoI	$V2$	4.57	2.70	**−0.45**	0.59	−3.55	−0.62	1.017	0.029
MoI	$V3$-1	4.59	2.87	**−2.50**	−1.87	−3.96	−1.56	0.504	0.019
MoI	$V3$-2	4.57	2.68	**−2.43**	−1.65	−4.11	−1.47	0.819	0.015
WMoI	$V1$	4.82	3.63	3.34	**0.91**	−1.36	−0.78	0.856	0.027
WMoI	$V2$	4.57	2.73	1.93	**−0.12**	−1.85	−1.22	0.918	0.027
WMoI	$V3$-1	4.64	2.90	−0.53	**−3.02**	−2.59	−2.36	0.647	0.013
WMoI	$V3$-2	4.61	2.72	−0.53	**−2.97**	−2.49	−2.27	0.398	0.010
PD	$V1$	4.75	3.40	7.79	9.41	**−4.50**	2.63	1.331	0.065
PD	$V2$	4.49	2.61	8.83	10.44	**−4.43**	3.06	1.210	0.053
PD	$V3$-1	4.51	2.73	7.55	9.43	**−5.07**	2.67	0.706	0.037
PD	$V3$-2	4.47	2.57	7.62	9.48	**−4.99**	2.68	0.602	0.036
WPD	$V1$	4.79	3.55	2.01	−0.97	−2.27	**−1.91**	1.246	0.042
WPD	$V2$	4.62	2.73	1.03	−2.07	−2.22	**−2.51**	0.918	0.039
WPD	$V3$-1	4.64	2.84	−0.85	−4.39	−2.88	**−3.46**	0.744	0.025
WPD	$V3$-2	4.58	2.70	−0.78	−4.32	−2.67	**−3.37**	0.523	0.017

Table 4.4: Dataset *ZCS*: Comparing flex-zone bounds

using $V3$-2 the results are only 5.66% better. Using $V1$ ($V2$) the improvements are only 1.09% (4.71%).
In summary, $V3$ is the most promising alternative. Since the results of $V3$-1 are noticeably better in terms of compactness, even if $V3$-2 performs slightly better in terms of contiguity and balance, the following tests use this approach whenever flex-zones are used.

4.4.2 Assignment of Basic Areas

This test focuses on the different concepts of assigning the basic areas located in the flex-zone to the sub-problems. The concept $flex_{ca}$ assigns each basic area to the sub-problem of its closest basic area not located in the flex-zone. Equations (4.18) and (4.19) describe this assignment formally. The concept $flex_{ca,i}$ updates the sub-problems after each assignment, whereas $flex_{ca}$ always uses the initial sub-problems. The next concepts assign each basic area to the sub-problem of the closest center of gravity. Equations (4.24) and (4.25) give a formal description. Again, $flex_{cog}$ uses the initial centers of gravity, whereas $flex_{cog_i}$ updates these centers after each assignment. Finally, $flex_{md}$ assigns each basic area to the sub-problem where the furthest basic area is closer. Equations (4.26) and (4.27) provide the corresponding assignment rule.

We apply $V3$-2 to determine the flex-zone bounds, and we set $meas_1 = comp_*$ and $\beta_1 = 1$ again. Tables 4.5 and 4.6 state the corresponding results. First of all, comparing the results for both variants of the assignment to the closest basic area, they are almost identical. Hence, there is no advantage of updating the sub-problems incrementally.

The difference between the concepts $flex_{ca}$ and $flex_{cog}$ is more significantly. The former implies noticeably better results in terms of balance, whereas the latter performs noticeably better in terms of compactness and slightly better in terms of contiguity. For example, for *PPS* using the Moment of Inertia $flex_{ca}$ evaluates 3.51 in terms of balance, whereas $flex_{cog}$ evaluates 4.49. In terms of compactness, more precisely in terms of the Moment of Inertia, the latter results in solutions that perform 5.19% better than reference solutions. In contrast to this, the results of the former concept are only 4.09% better. Table 4.5 shows the same observations for the further compactness measures. The solutions obtained for applying the $flex_{cog}$ concept have an average (maximal) overlap of 0.005% (0.020%) compared to 0.090% (0.428%) for applying the $flex_{ca}$ concept. Hence, there is a trade-off between the different optimization goals. If there is a focus on compactness the usage of $flex_{cog}$ is recommendable. The results of $flex_{cog}$ and $flex_{cog_i}$ are comparable. In terms of balance and contiguity the differences are only minimal and it is ambiguous which one performs better. In terms of compactness $flex_{cog}$ seems to be slightly better. However, for *PPS* using the Moment of

Compactness measure	flex-zone	*bal*		*comp*				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
MoI	$flex_{ca}$	3.51	1.94	**−4.09**	−4.14	−2.78	−2.54	0.428	0.090
MoI	$flex_{ca,i}$	3.51	1.94	**−4.09**	−4.14	−2.78	−2.54	0.428	0.090
MoI	$flex_{cog}$	4.49	2.92	**−5.19**	−5.28	−3.44	−3.20	0.020	0.005
MoI	$flex_{cog,i}$	4.51	2.98	**−5.22**	−5.36	−3.44	−3.23	0.056	0.006
MoI	$flex_{md}$	4.49	3.03	**−5.04**	−5.10	−3.45	−3.10	0.684	0.047
WMoI	$flex_{ca}$	3.82	2.11	−3.97	**−4.75**	−2.68	−2.89	0.718	0.152
WMoI	$flex_{ca,i}$	3.82	2.11	−3.97	**−4.75**	−2.68	−2.89	0.718	0.152
WMoI	$flex_{cog}$	4.72	3.12	−5.87	**−6.48**	−3.99	−3.98	0.107	0.006
WMoI	$flex_{cog,i}$	4.72	3.10	−5.68	**−6.29**	−3.89	−3.90	0.107	0.007
WMoI	$flex_{md}$	4.72	3.12	−4.65	**−5.33**	−3.35	−3.35	0.073	0.012
PD	$flex_{ca}$	3.97	2.09	−8.12	−8.25	**−5.33**	−4.99	0.718	0.172
PD	$flex_{ca,i}$	3.97	2.09	−8.12	−8.25	**−5.33**	−4.99	0.718	0.172
PD	$flex_{cog}$	4.58	2.98	−8.74	−8.82	**−6.05**	−5.55	0.244	0.023
PD	$flex_{cog,i}$	4.65	3.02	−8.33	−8.49	**−5.91**	−5.41	0.300	0.023
PD	$flex_{md}$	4.77	3.18	−7.68	−7.86	**−5.64**	−5.22	0.169	0.025
WPD	$flex_{ca}$	3.77	1.93	−7.87	−8.85	−5.06	**−5.48**	0.665	0.176
WPD	$flex_{ca,i}$	3.77	1.93	−7.87	−8.85	−5.06	**−5.48**	0.665	0.176
WPD	$flex_{cog}$	4.76	3.09	−9.26	−10.15	−5.87	**−6.18**	0.295	0.031
WPD	$flex_{cog,i}$	4.78	3.10	−9.31	−10.22	−5.89	**−6.22**	0.351	0.032
WPD	$flex_{md}$	4.78	3.15	−8.63	−9.32	−5.74	**−5.90**	0.174	0.022

Table 4.5: Dataset *PPS*: Comparing assignment rules

compactness measure	flex-zone	*bal*		*comp*				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
MoI	$flex_{ca}$	3.85	2.05	**−1.53**	−0.84	−2.60	−0.88	1.038	0.162
MoI	$flex_{ca,i}$	3.85	2.05	**−1.53**	−0.84	−2.60	−0.88	1.038	0.162
MoI	$flex_{cog}$	4.59	2.87	**−2.50**	−1.87	−3.96	−1.56	0.504	0.019
MoI	$flex_{cog,i}$	4.59	2.85	**−2.52**	−1.81	−3.97	−1.53	0.504	0.021
MoI	$flex_{md}$	4.60	2.88	**−2.57**	−1.72	−4.20	−1.53	1.193	0.067
WMoI	$flex_{ca}$	3.95	2.08	0.33	**−2.28**	−1.04	−1.72	1.760	0.176
WMoI	$flex_{ca,i}$	3.95	2.08	0.33	**−2.28**	−1.04	−1.72	1.760	0.176
WMoI	$flex_{cog}$	4.64	2.90	−0.53	**−3.02**	−2.59	−2.36	0.647	0.013
WMoI	$flex_{cog,i}$	4.63	2.90	−0.55	**−3.04**	−2.60	−2.37	0.647	0.018
WMoI	$flex_{md}$	4.66	2.92	−0.45	**−2.90**	−2.71	−2.36	0.733	0.041
PD	$flex_{ca}$	3.91	2.09	9.49	11.78	**−4.18**	3.74	1.136	0.183
PD	$flex_{ca,i}$	3.91	2.09	9.49	11.78	**−4.18**	3.74	1.136	0.183
PD	$flex_{cog}$	4.51	2.73	7.55	9.43	**−5.07**	2.67	0.706	0.037
PD	$flex_{cog,i}$	4.52	2.74	7.55	9.43	**−5.04**	2.68	0.637	0.040
PD	$flex_{md}$	4.62	2.86	7.13	9.65	**−5.41**	2.78	1.313	0.046
WPD	$flex_{ca}$	3.91	2.09	0.23	−3.54	−1.54	**−2.83**	1.760	0.202
WPD	$flex_{ca,i}$	3.91	2.09	0.23	−3.54	−1.54	**−2.83**	1.760	0.202
WPD	$flex_{cog}$	4.64	2.84	−0.85	−4.39	−2.88	**−3.46**	0.744	0.025
WPD	$flex_{cog,i}$	4.65	2.85	−0.81	−4.36	−2.87	**−3.44**	0.672	0.027
WPD	$flex_{md}$	4.66	2.92	−0.69	−3.94	−3.08	**−3.23**	1.189	0.047

Table 4.6: Dataset *ZCA*: Comparing assignment rules

Inertia, the solutions of $flex_{cog_i}$ are marginally better. Hence, this test shows no advantage of updating the centers of gravity after each assignment.

Finally, consider the $flex_{md}$ concept. Usually, with respect to the balance its results are slightly worse than the results of the $flex_{cog}$ concept. In terms of contiguity, its solution tends to result in marginally worse evaluations. In total, the results of $flex_{cog}$ also seem to be a little better in terms of compactness, even if there are some counterexamples. For example, for *ZCA* and evaluating bisecting partitions by Pairwise Distances, the solutions obtained by applying $flex_{md}$ are 5.41% better according to the Pairwise Distances than the reference solutions, whereas the solutions achieved by using $flex_{md}$ are only 5.07% better.

In order to obtain compact districts, we advise the usage of the $flex_{cog}$ concept. Hence, the following tests, unless stated otherwise, use this approach whenever flex-zones are used. Nevertheless, the usage of $flex_{md}$ is also possible, its solutions are only slightly worse. If there is a higher focus on balance, we suggest to use the $flex_{ca}$ concept. In principle, it is also possible to combine different approaches. However, by doing so, the set of feasible bisecting partitions in FBP increases, and, thus, the running time for evaluating all bisecting partitions increases, too.

4.4.3 Bisecting Partitions

Now, this test compares the two approaches of generating bisecting partitions introduced in Section 4.3.3. It includes line partitions, flex-zone partitions and a combination of them, i.e., the set of bisecting partitions FBP contains line partitions as well as flex-zone partitions. The flex-zone approach considered here uses the $flex_{cog}$ concept and defines LL and LU according to $V3$-1. Moreover, this test evaluates bisecting partitions only in terms of compactness again.

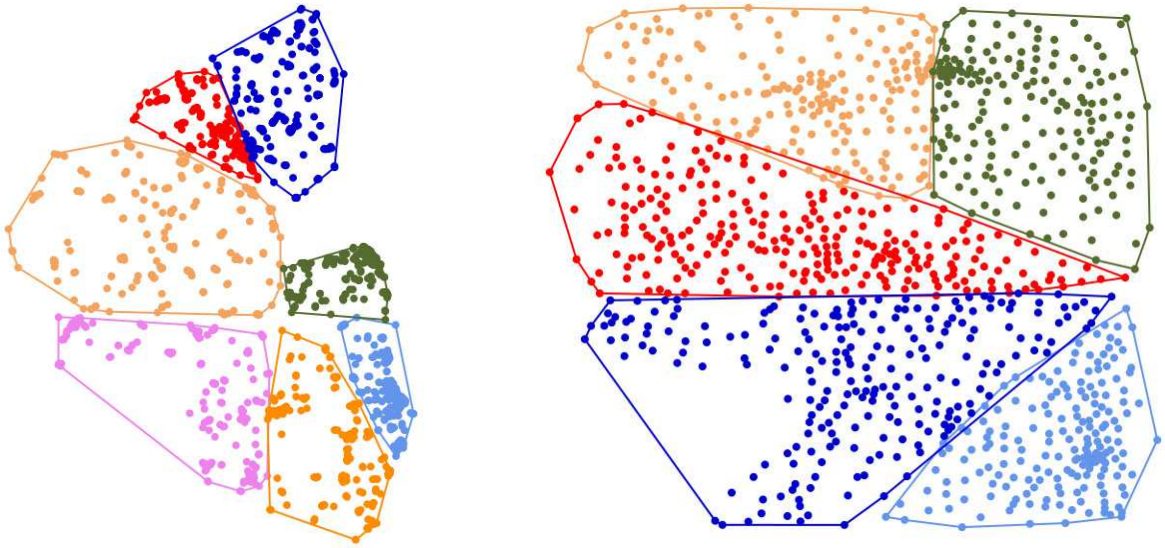
Tables 4.7 and 4.8 report the results. For each compactness measure, the corresponding first row shows the results for using line partitions exclusively, the second row for using flex-zone partitions, while the third row states the results for using the combination of them.
Taking a look at the results, first of all, it is obvious that the balance is noticeably better when using just line partitions. This is not surprising since line partitions focus on balance. More surprising is the observation, that sometimes the compactness is also better when using line partitions instead of flex-zone partitions. For example, for the Moment of Inertia as applied compactness measure Table 4.7 states an improvement in terms of the Moment of Inertia of 5.63% compared to the reference solution for using line partitions, while it depicts an improvement of 5.19% for flex-zone partitions. A possible explanation is that in trying to obtain more compact sub-problems, a flex-zone partition might exploit the allowed balance

bisecting partition	comp. measure	time	*bal*		*comp*				*ctg*	
			max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
line	MoI	58	0.80	0.29	**−5.63**	−5.61	−3.17	−2.94	0.000	0.000
flex-zone	MoI	57	4.49	2.92	**−5.19**	−5.28	−3.44	−3.20	0.020	0.005
both	MoI	58	4.38	2.68	**−6.75**	−6.77	−4.09	−3.92	0.057	0.003
line	WMoI	49	0.72	0.28	−4.82	**−5.57**	−2.91	−3.18	0.000	0.000
flex-zone	WMoI	48	4.72	3.12	−5.87	**−6.48**	−3.99	−3.98	0.107	0.006
both	WMoI	50	4.68	3.00	−6.96	**−7.61**	−4.34	−4.54	0.181	0.006
line	PD	3208	0.81	0.35	−7.82	−8.12	**−5.24**	−4.90	0.000	0.000
flex-zone	PD	3056	4.58	2.98	−8.74	−8.82	**−6.05**	−5.55	0.244	0.023
both	PD	6991	4.53	2.86	−8.33	−8.49	−5.91	**−5.41**	0.107	0.011
line	WPD	3215	0.72	0.29	−9.06	−9.90	−5.27	**−5.73**	0.000	0.000
flex-zone	WPD	2612	4.76	3.09	−9.26	−10.15	−5.87	**−6.18**	0.295	0.031
both	WPD	6739	4.54	2.82	−11.20	−12.13	−6.66	**−7.09**	0.052	0.006

Table 4.7: Dataset *PPS*: Comparing bisecting partitions

bisecting partition	comp. measure	time	*bal*		*comp*				*ctg*	
			max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
line	MoI	55	1.97	0.94	**−3.03**	−2.21	−2.93	−1.30	0.000	0.000
flex-zone	MoI	56	4.59	2.85	**−2.52**	−1.81	−3.97	−1.53	0.504	0.021
both	MoI	92	4.48	2.67	**−3.90**	−3.46	−3.98	−2.16	0.648	0.010
line	WMoI	43	1.91	0.90	−0.68	**−3.50**	−1.20	−2.19	0.000	0.000
flex-zone	WMoI	48	4.64	2.90	−0.53	**−3.02**	−2.59	−2.36	0.013	0.647
both	WMoI	44	4.55	2.72	−1.52	**−4.21**	−2.74	−2.86	0.007	0.647
line	PD	1405	1.91	0.93	9.55	11.46	**−4.21**	3.79	0.000	0.000
flex-zone	PD	1223	4.51	2.73	7.55	9.43	**−5.07**	2.67	0.706	0.037
both	PD	2612	4.41	2.52	7.68	9.10	**−5.47**	2.49	0.637	0.040
line	WPD	1176	1.96	0.92	−0.56	−4.60	−1.43	**−3.06**	0.000	0.000
flex-zone	WPD	1183	4.64	2.84	−0.85	−4.39	−2.88	**−3.46**	0.744	0.025
both	WPD	2699	4.56	2.72	−1.69	−5.36	−3.27	**−3.86**	0.537	0.012

Table 4.8: Dataset *ZCA*: Comparing bisecting partitions



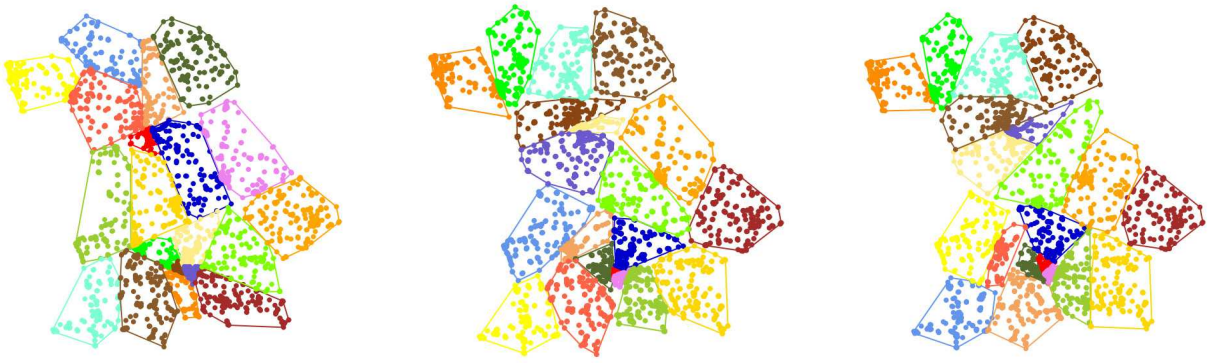
(a) Dataset *PPS* (b) Dataset *ZCA*

Figure 4.7: Worse districts in terms of contiguity

deviation. As a result, usually, in subsequent sub-divisions the set of feasible flex-zone partitions, where the choice is made from, is small and possibly of lower quality. However, in general, the combination of both approaches outperforms line partitions as well as flex-zone partitions in terms of compactness. For example, for *ZCA* and using the Weighted Moment of Inertia as compactness measure, the combination evaluates -4.21 in terms of the Weighted Moment of Inertia, compared to -3.02 for flex-zone partitions.

Solutions obtained by using line partitions exclusively are always contiguous. Comparing the flex-zone approach and the combination, there is no clear trend which of them performs better with respect to contiguity. The solutions are nearly non-overlapping in any case. For example, Figure 4.7a shows the worst solution in terms of contiguity for *PPS*. It is generated by flex-zones and Weighted Pairwise Distances and has an overlap of 0.295%. There is an overlap between the red district and the blue district in the north. Moreover, there are small overlaps in the south-west. Figure 4.7b depicts a bad solution for *ZCA*. Here, there are small overlaps between the light blue and dark blue districts in the south and between the orange and red districts in the north. The combination of flex-zone partitions and line partitions using Pairwise Distances results in this solution. Its contiguity is 0.637%.

The results show similar running times for line partitions and flex-zone partitions. Surprisingly, if the (Weighted) Moment of Inertia measures the compactness of the bisecting partitions, the running times for combining the bisecting partition approaches are more or less identical to them for using one of them exclusively. However, if the algorithm uses the



(a) Basic version (b) Weighted Pairwise Distances (c) Flex-zone approach

Figure 4.8: Improvements compared to the basic version of the RPA

(Weighted) Pairwise Distances the running times for combining the approaches are noticeably larger.

Figure 4.8 illustrates the improvements of the RPA compared to the basic version introduced by Kalcsics et al. [16]. First, Figure 4.8a depicts the solution of the basic version of the RPA using line partitions and the length of intersection as compactness measure. It contains some long-shaped districts like the blue one in the central region or the green one in the west. Replacing the compactness measure by the Weighted Pairwise Distances results in the solution depicted in Figure 4.8b. In terms of the Weighted Pairwise Distances this solution is 21.13% better than the previous solution. The visual impression confirms this result since the shapes are more squared. Combining line partitions and flex-zone partitions improves the compactness again, but only slightly. Figure 4.8c presents a solution that is 0.5% better than the solution before.

This test points out that using flex-zone partitions exclusively is not advisable. Compared to line partitions, the results are noticeably worse in terms of balance, slightly worse in terms of contiguity and not clearly improved with respect to the compactness. However, combining both approaches improves the compactness values noticeably, and, hence, justifies the usage of flex-zone partitions. Thus, we recommend the usage of the combination.

4.4.4 Compactness Measures

This experiment compares the different approaches for measuring the compactness of a bisecting partition introduced in Section 4.3.5.2. These measures are the Length of Intersection, the (Weighted) Moment of Inertia, the (Weighted) Pairwise Distances, and the

Maximum Distance. Let S_{loi}, S_{moi}, S_{wmoi}, S_{pd}, S_{wpd}, S_{md} denote the corresponding solutions.

Here these solutions are compared among each other in terms of different distance-based compactness measures $comp_*(S)$. As $comp_*(S)$ are absolute measures, the test determines the relative percentage deviation between two values, i.e., $\frac{comp_*(S_1)-comp_*(S_2)}{comp_*(S_2)}$. Hence, for a positive (negative) deviation, the first solution S_1 is less (more) compact than the second solution S_2. According to the results of the previous section this test combines line partitions and flex-zone partitions.

	S_{loi}	$\mathbf{S_{moi}}$	S_{wmoi}	S_{pd}	S_{wpd}
$\mathbf{S_{moi}}$	**−6.75**	**0.00**			
S_{wmoi}	−6.96	**0.99**	0.00		
S_{pd}	−8.33	**−2.45**	−3.28	0.0	
S_{wpd}	−11.20	**−4.59**	−5.33	−2.00	0.00
S_{md}	1.31	**8.74**	7.88	11.65	14.19

(a) Instances *PPS* and $comp_{moi}(S)$

	S_{loi}	S_{moi}	$\mathbf{S_{wmoi}}$	S_{pd}	S_{wpd}
S_{moi}	−6.77	0.00			
$\mathbf{S_{wmoi}}$	**−6.48**	**0.32**	**0.00**		
S_{pd}	−9.20	−2.62	**−2.81**	0.0	
S_{wpd}	−12.13	−5.57	**−5.70**	−2.76	0.00
S_{md}	1.26	8.65	**8.51**	11.78	15.47

(b) Instances *PPS* and $comp_{wmoi}(S)$

	S_{loi}	S_{moi}	S_{wmoi}	$\mathbf{S_{pd}}$	S_{wpd}
S_{moi}	−4.09	0.00			
S_{wmoi}	−3.99	0.11	0.00		
$\mathbf{S_{pd}}$	**−6.31**	**−2.29**	**−2.38**	**0.0**	
S_{wpd}	−6.66	−2.63	−2.71	**−0.33**	0.00
S_{md}	0.08	4.40	4.32	**6.89**	7.26

(c) Instances *PPS* and $comp_{pd}(S)$

	S_{loi}	S_{moi}	S_{wmoi}	S_{pd}	$\mathbf{S_{wpd}}$
S_{moi}	−3.92	0.00			
S_{wmoi}	−3.98	−0.06	0.00		
S_{pd}	−5.94	−2.09	−2.02	0.0	
$\mathbf{S_{wpd}}$	**−7.09**	**−3.25**	**−3.17**	**−1.13**	**0.00**
S_{md}	0.20	4.34	4.43	6.62	**7.94**

(d) Instances *PPS* and $comp_{wpd}(S)$

	S_{loi}	S_{moi}	S_{wmoi}	S_{pd}	S_{wpd}
S_{moi}	3.33	0.00			
S_{wmoi}	6.47	3.70	0.00		
S_{pd}	4.25	1.50	−1.03	0.0	
S_{wpd}	3.63	0.67	−1.84	−0.42	0.00
$\mathbf{S_{md}}$	**−4.36**	**−6.55**	**−8.89**	**−7.55**	**−6.62**

(e) Instances *PPS* and $comp_{md}(S)$

Table 4.9: Dataset *PPS*: Average relative percentage deviations in terms of compactness

Tables 4.9 and 4.10 show the results, where the rows correspond to S_1 and the columns to S_2. The entries are percentage values. For example, the entry −6.75 in the first row and first column of Table 4.9a states that according to the Moment of Inertia the solution obtained for using the Moment of Inertia as compactness measure for bisecting partitions is on average 6.75% better than the solution obtained for using the Length of Intersection.

Table 4.9a depicts that using (Weighted) Pairwise Distances leads to better results with

	S_{loi}	$\mathbf{S_{moi}}$	S_{wmoi}	S_{pd}	S_{wpd}
$\mathbf{S_{moi}}$	**−3.90**	**0.00**			
S_{wmoi}	−2.06	**2.54**	0.00		
S_{pd}	7.68	**12.17**	9.74	0.0	
S_{wpd}	−1.69	**2.57**	0.21	−6.47	0.00
S_{md}	2.06	**6.57**	4.24	−2.53	4.61

(a) Instances *ZCA* and $comp_{moi}(S)$

	S_{loi}	S_{moi}	$\mathbf{S_{wmoi}}$	S_{pd}	S_{wpd}
S_{moi}	−3.46	0.00			
$\mathbf{S_{wmoi}}$	**−4.21**	**−0.66**	**0.00**		
S_{pd}	9.10	13.18	**14.15**	0.0	
S_{wpd}	−5.36	−1.75	**−0.87**	−10.98	0.00
S_{md}	3.43	7.53	**8.60**	−2.22	10.29

(b) Instances *ZCA* and $comp_{wmoi}(S)$

	S_{loi}	S_{moi}	S_{wmoi}	$\mathbf{S_{pd}}$	S_{wpd}
S_{moi}	−3.98	0.00			
S_{wmoi}	−2.74	1.34	0.00		
$\mathbf{S_{pd}}$	**−5.47**	**−1.48**	**−2.38**	**0.0**	
S_{wpd}	−3.27	0.85	−2.71	**2.57**	0.00
S_{md}	−2.29	1.92	4.32	**3.64**	5.55

(c) Instances *ZCA* and $comp_{pd}(S)$

	S_{loi}	S_{moi}	S_{wmoi}	S_{pd}	$\mathbf{S_{wpd}}$
S_{moi}	−2.16	0.00			
S_{wmoi}	−2.86	−0.69	0.00		
S_{pd}	2.49	4.79	5.56	0.0	
$\mathbf{S_{wpd}}$	**−3.86**	**−1.69**	**−0.97**	**−5.79**	**0.00**
S_{md}	1.31	3.63	4.42	−0.63	**5.55**

(d) Instances *ZCA* and $comp_{wpd}(S)$

	S_{loi}	S_{moi}	S_{wmoi}	S_{pd}	S_{wpd}
S_{moi}	2.70	0.00			
S_{wmoi}	4.89	2.46	0.00		
S_{pd}	13.42	11.08	8.90	0.0	
S_{wpd}	6.97	4.66	2.63	−4.14	0.00
$\mathbf{S_{md}}$	**−2.24**	**−4.24**	**−6.01**	**−12.17**	**−7.69**

(e) Instances *ZCA* and $comp_{md}(S)$

Table 4.10: Dataset *ZCA*: Average relative percentage deviations in terms of compactness

respect to $comp_{moi}$ than using the Moment of Inertia. Table 4.9b reports similar results in terms of $comp_{moi}$ for using the Weighted Moment of Inertia. A possible explanation for this fact is that the (Weighted) Moment of Inertia simplifies the computation of a bisecting partition's compactness by using just one representative center for each partition problem instead of determining q centers. Unfortunately, the advantage in terms of compactness comes along with noticeably larger running times. Hence, there is a trade-off between the quality of a solution and the corresponding running time. However, Table 4.9a does not report this effect. Table 4.9b depicts this effect for using Weighted Pairwise Distances, however, it is significantly smaller. Possibly, the considerably higher range of the activities for the *ZCA* instances influences this effect. Moreover, Tables 4.9a to 4.9d state similar results for S_{moi} and S_{wmoi}, whereas Tables 4.10a to 4.10d report larger differences. Furthermore, they show a similar effect for S_{pd} and S_{wpd}, where the differences for *ZCA* are more significant. For example, in terms of $comp_{wmoi}$ the solution S_{pd} is 14.15% worse compared to S_{wmoi}, whereas S_{pd} is 0.87% better. Again, the higher range of the activities for *ZCA* is the probable reason for this observation. As expected, the results for S_{md} are well in terms of $comp_{md}$ (see Tables 4.9e and 4.10e). However, they are poor in terms of the other compactness

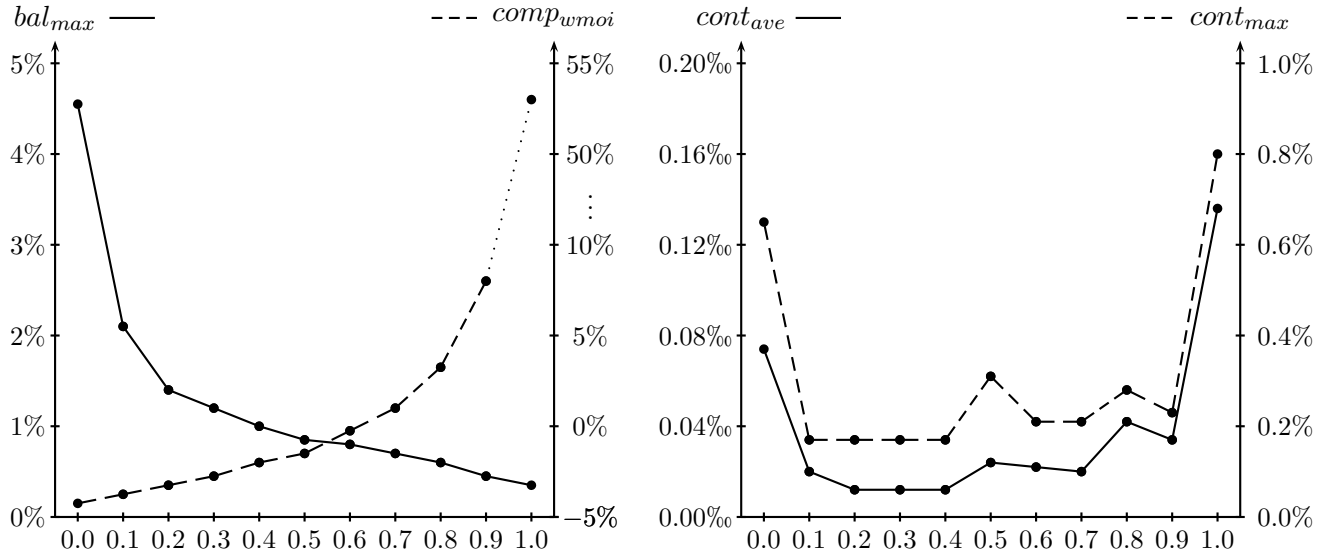
measures. The results show that no compactness measure leads to good results in terms of all compactness measures. Hence, the user should select the compactness measure depending on the data set and on his preferences.

4.4.5 Varying Criteria Weights

The next test discusses the influence of the criteria weights of the ranking function rk defined in Equation (4.28). It restricts itself to two criteria, namely balance and compactness. In this case, the ranking function reduces to

$$rk(BP) := \beta \cdot \frac{bal(BP) - bal^{min}}{bal^{max} - bal^{min}} + (1 - \beta) \cdot \frac{comp_*(BP) - comp_*^{min}}{comp_*^{max} - comp_*^{min}} . \tag{4.29}$$

This test evaluates the changes of the evaluations in terms of the planning criteria for changing β in steps of 0.1. It chooses the combination of line partitions and flex-zone partitions on *ZCA* exemplarily.

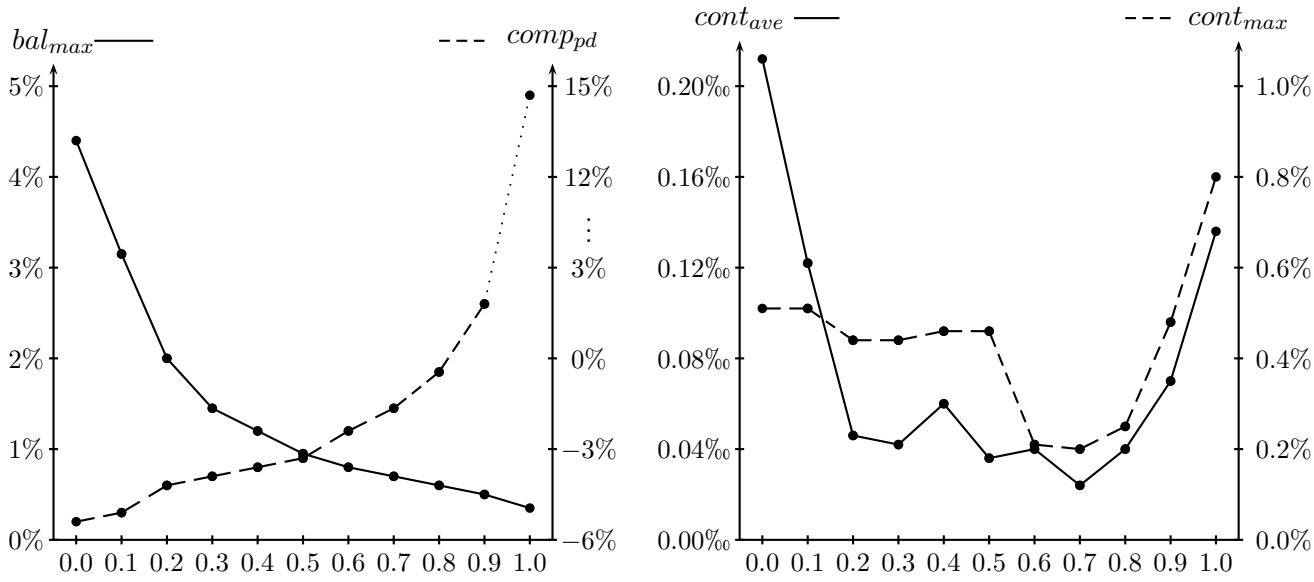


(a) Balance and compactness

(b) Contiguity

Figure 4.9: *ZCA*: Using the Weighted Moment of Inertia: Varying β

Figure 4.9 illustrates the results in terms of balance, compactness and contiguity for using the Weighted Moment of Inertia as compactness measure in rk. The solid line in Figure 4.9a depicts the trend of the balance. For $\beta = 0$, i.e., evaluating the bisecting partitions only in terms of compactness, as expected, the balance is close to the maximal feasible balance of 5%. In the following, the balance decreases (becomes better) if β increases. The evolution of the compactness with respect to β behaves inversely. The dashed line depicts this



(a) Balance and compactness

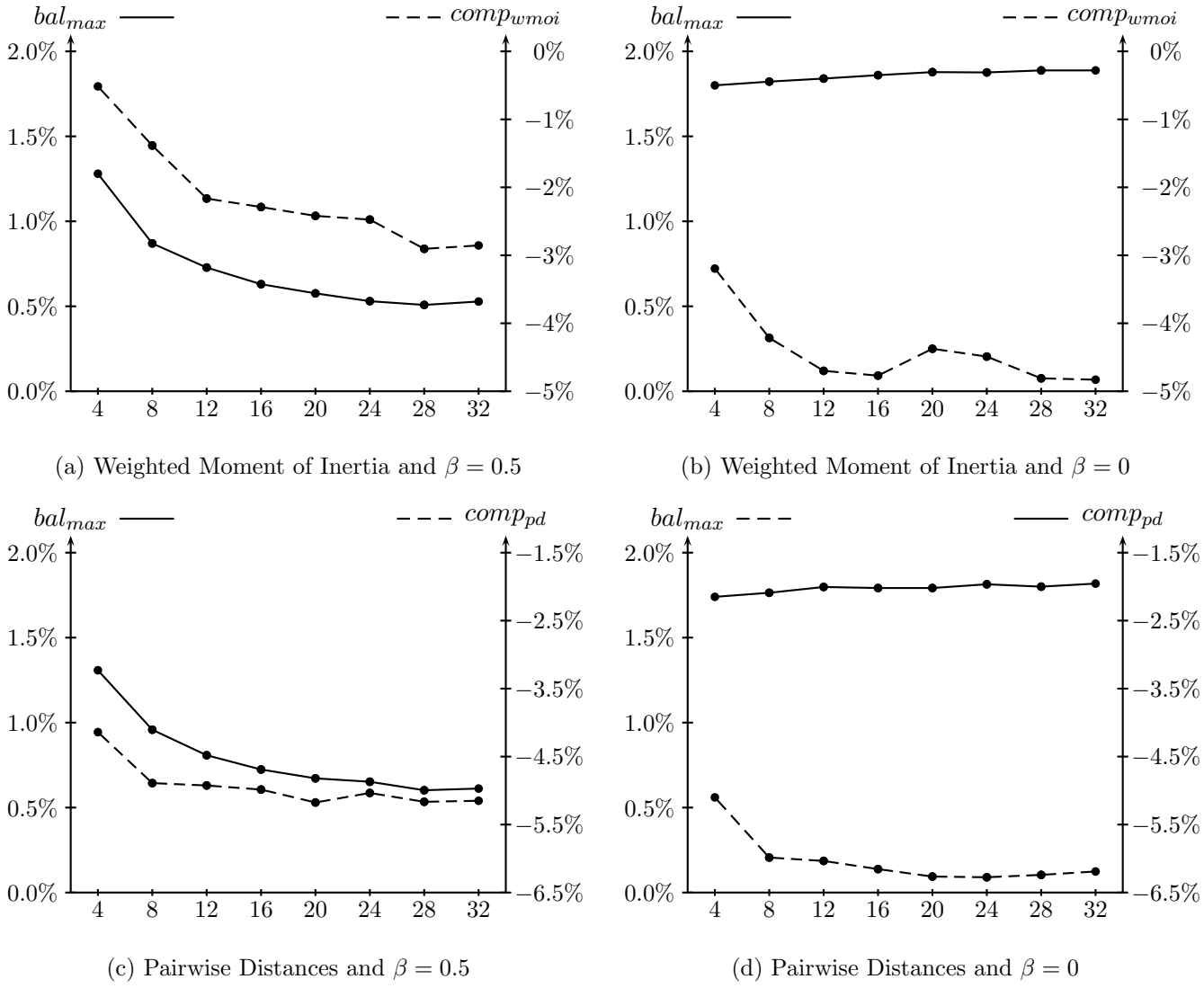
(b) Contiguity

Figure 4.10: *ZCA*: Using Pairwise Distances: Varying β

evolution. For $\beta = 1$ compactness is more than 50% worse compared to the reference solution, and, hence, very poor. For purposes of presentability the scale on the compactness axis varies. Figure 4.9b shows the corresponding evaluations in terms of contiguity. The solid line illustrates the average contiguity, whereas the dashed line depicts the maximum contiguity. Again, note the different scales, per mill for the average contiguity and percent for the maximum contiguity. Mainly, the solutions taking only one criterion into account are comparatively poor. Nevertheless, even these solutions are almost non-overlapping having overlaps smaller than 1%.

Figure 4.10 shows the results for using the Pairwise Distances as compactness measure in rk. Figure 4.10a shows the same trends in terms of balance and compactness as the example before. Figure 4.10b depicts the evaluations in terms of contiguity. The solutions for $\beta = 0$ and $\beta = 1$ are comparatively poor again. However, there is no clear trend for varying β between 0.1 and 0.9.

As expected, the choice of β influences the solutions in terms of balance, compactness, and contiguity. Balance is decreasing in β, whereas compactness is increasing. In terms of contiguity the extremal settings of β lead to comparatively bad results, whereas the trend for the further settings is ambiguous. Setting β to about 0.5 appears to be a good compromise between balance, compactness and contiguity.



(a) Weighted Moment of Inertia and $\beta = 0.5$

(b) Weighted Moment of Inertia and $\beta = 0$

(c) Pairwise Distances and $\beta = 0.5$

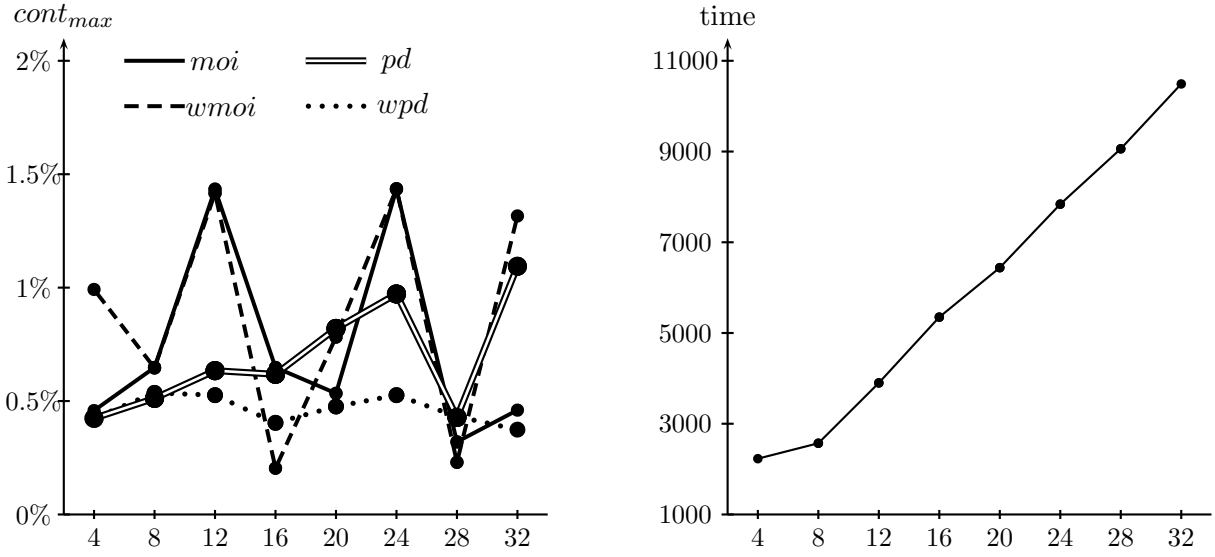
(d) Pairwise Distances and $\beta = 0$

Figure 4.11: *ZCA*: Varying K

4.4.6 Varying Number of Search Directions

This test evaluates the influence of the number of search directions on quality and running time. It chooses the combination of line partitions and flex-zone partitions on *ZCA* exemplarily and compares the results for setting $K \in \{4; 8; 12; 16; 20; 24; 28; 32\}$.

Figure 4.11 illustrates the evaluation in terms of compactness and balance for different compactness measures and different criteria weights. For example, Figure 4.11a shows the results for using the Weighted Moment of Inertia in order to evaluate bisecting partitions and setting $\beta = 0.5$ according to Equation (4.29), i.e., balance and compactness are weighted equally. Both balance (solid lines) and compactness (dashed lines) improve in $K \in [4, 28]$. Using 32 search directions leads to slightly worse results than using 28 directions. Unsurprisingly,



(a) Contiguity

(b) Running times for Pairwise Distances

Figure 4.12: *ZCA*: Varying K for $\beta = 0$

Figure 4.11c shows the same trend for using pairwise distances in order to measure the compactness of bisecting partitions. Figure 4.11b and 4.11d illustrate the corresponding results for $\beta = 0$, i.e., for evaluating bisecting partitions only in terms of compactness. In this case, balance marginally increases (becomes worse) in K, whereas compactness decreases in $K \in [4, 16]$. The trend in terms of compactness for more than 16 search directions is ambiguous.

In terms of contiguity there is no obvious correlation between the number of search directions and the maximal contiguity. Figure 4.12a shows the corresponding trends for using different compactness measures for bisecting partitions and setting $\beta = 0$. Finally, Figure 4.12b states the running time for using the Pairwise Distances. The running time increases (approximately) linearly in the number of search directions. This result confirms the theoretical complexity analysis in Section 4.3.7.3.

This test shows that usually the quality of a solution increases in the number of search directions up to 16. However, for larger settings of K the quality improves only marginally whereas the running times increases linearly. Hence, there is a trade-off between quality and running time.

4.4.7 Running Times

Table 4.11 shows the running times for the largest instances from *PPS*. It consists of 4971 basic areas that have to be divided in 46 districts.

	LoI	MoI	WMoI	PD	WPD
line	1	6	6	542	538
flex-zone	5	6	5	521	417
both	6	6	6	1267	1156

Table 4.11: Running times partitioning 4971 into 46 districts

Surprisingly, the running times of using flex-zone partitions exclusively are better than the ones of using line partitions exclusively. Combining both approaches leads to larger sets of feasible bisecting partitions, and, hence, to larger running times. As already reported in Section 4.4.3 the running time of the RPA highly depends on the used compactness measure. Using the (Weighted) Pairwise Distances needs noticeably more time to solve the districting problem than using the (Weighted) Moment of Inertia. However, the results are also noticeably better (cf. Section 4.4.4). Thus, there is a trade-off between running time and solution quality, again.

However, since the RPA solves a strategical (or a tactical) problem, running times of about 20 minutes are still acceptable. Moreover, using parallelization techniques for solving the different sub-problems may lead to a further reduction of the running times.

4.4.8 Network Distances

This section focuses on the integration of distances or travel times on a road network into the RPA. For purposes of readability, the term "network distances" describes both distances and travel times on a road network. The main advantage of network distances is that they reflect geographic obstacles like rivers or mountains implicitly.

This test compares solutions generated by using network distances to those generated by using Euclidean distances. Note that the used distance function has effects on different parts of the RPA. First, the assignment decision for a basic area located in the flex-zone depends on the distances to the basic areas of the left and right sub-problem. Second, the evaluation of a bisecting partition in terms of compactness depends on the distances to the basic areas of the same sub-problem.
Further note the technical details. This test combines line partitions and flex-zone partitions and uses only compactness in order to evaluate a bisecting partition. The number of search directions is 8. A bisecting partition's compactness evaluation is based on network distances, where the basic version of the RPA provides the reference solution.

Table 4.12 shows the results for road distances. The underlying set of instances consists of

comp. measure	distances	*bal*		*comp* (road distances)				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
MoI	road	4.31	2.52	**−9.63**	−9.40	−4.83	−4.50	0.101	0.021
MoI	Euclidean	4.58	2.76	**−9.73**	−9.46	−5.83	−5.72	0.003	0.000
WMoI	road	4.34	2.48	−8.48	**−8.85**	−3.97	−4.34	0.351	0.032
WMoI	Euclidean	4.77	2.94	−9.47	**−9.55**	−5.55	−5.64	0.003	0.000
PD	road	4.39	2.64	−15.19	−14.99	**−8.48**	−8.23	0.236	0.036
PD	Euclidean	4.64	2.83	−13.53	−13.58	**−7.90**	−7.85	0.107	0.012
WPD	road	4.42	2.74	−13.39	−13.59	−8.00	**−8.14**	0.236	0.042
WPD	Euclidean	4.56	2.79	−12.80	−13.62	−7.72	**−8.01**	0.052	0.005

Table 4.12: The RPA applied with road distances

comp. measure	distances	*bal*		*comp* (times)				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
MoI	times	4.48	2.51	**−4.43**	−4.67	−2.27	−2.13	0.206	0.035
MoI	Euclidean	4.37	2.55	**−5.98**	−6.23	−4.06	−4.29	0.002	0.000
WMoI	times	4.39	2.43	−1.86	**−3.10**	−0.74	−1.04	0.043	0.007
WMoI	Euclidean	4.73	2.89	−7.08	**−7.97**	−4.41	−5.06	0.002	0.000
PD	times	3.79	2.14	−9.26	−10.49	**−5.45**	−5.23	0.131	0.016
PD	Euclidean	4.45	2.71	−4.36	−4.73	**−4.08**	−3.84	0.107	0.018
WPD	times	4.16	2.51	−11.11	−12.78	−6.18	**−6.84**	0.106	0.015
WPD	Euclidean	4.32	2.66	−7.74	−9.59	−5.15	**−5.93**	0.020	0.020

Table 4.13: The RPA applied with travel times

23 instances from *PPS*. Table 4.13 presents the results for travel times on a road network. The use 12 instances are also from *PPS*. In addition, both tables show the results for using Euclidean distances during the execution of the RPA, but evaluating the final solutions in terms of network distances.

Both tables show similar results. Unfortunately, using Euclidean distances implies better results than using network distances when the (Weighted) Moment of Inertia measures the compactness of bisecting partitions. For example, the first and second row of Table 4.12 present the results for using the Moment of Inertia as compactness measure for bisecting partitions. In terms of the Moment of Inertia, the first row states an improvement of 4.43% compared to the reference solution for using road distances during the execution of the RPA. However, for using Euclidean distances the second row depicts an improvement of 5.98%. Most likely, the approximation of a sub-problem's center is too rough. The corresponding measure uses the basic area closest to the center of gravity as center. Unfortunately, this

center is not necessarily close to the center based on network distances.
In terms of (Weighted) Pairwise Distances network distances outperform Euclidean distances. However, Euclidean distances are a good approximation for road distances. For example, in terms of Weighted Pairwise Distances Table 4.12 states a compactness value of 8.01 for Euclidean distances and of a compactness value 8.14 for road distances.

In terms of balance road distances lead to slightly better results than Euclidean distances. In contrast to this, the solutions for road distances are slightly worse with respect to the contiguity, but still very well. Table 4.12 states a maximal contiguity of 0.351%.

In summary, this test shows the following: Although the RPA is a geometric approach, it is able to handle network distances.

4.5 Incorporating Prescribed Centers

In contrast to Section 4.3, sometimes, the planning process has to take a given set of fixed districts' centers CE into account, e.g., because they correspond to residences of service persons or to locations of already existing branch offices which may not be changed.

In order to incorporate these centers into the RPA, this section addresses the districting problem with prescribed centers. In this context, each district $D_g := (B_g, cen_g)$ contains a set of basic areas B_g and exactly one center $cen_g \in CE$. A possible interpretation of this problem is that the basic areas must be allocated to the centers.

In this context, the most common way to measure compactness is to apply the (Weighted) Moment of Inertia (cf. Equations (4.2) and (4.3)), where a district's center corresponds to its prescribed center. This leads to

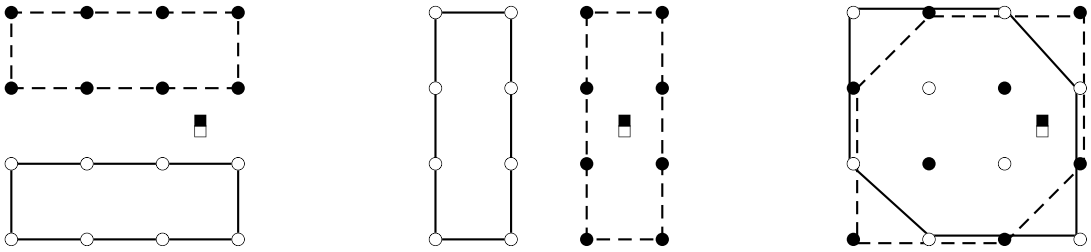
$$comp_{moi}(D_g) := \sum_{i \in B_g} d^2(b_i, cen_g) \tag{4.30}$$

and

$$comp_{wmoi}(D_g) := \sum_{i \in B_g} w_i \cdot d^2(b_i, cen_g^{un}) \,. \tag{4.31}$$

Moreover, in the applications considered here, there are interactions between the basic areas and the center within each district. For example, a service person has to visit its customers periodically. Hence, this extension additionally incorporates the locations of the centers with respect to their districts. However, often, there are several centers packed in a small area. For example, in many real-world examples the service persons' residences are concentrated in urban regions, sometimes even at the same address, whereas there are only few residences in rural regions. Hence, requiring each center to be located within its district would be too prohibitive. Nevertheless, at best each center should be located within its district, but at least closely to its district.

A compact solution is not necessarily a good solution with respect to this criterion. Figure 4.13 depicts an example where two centers (illustrated by squares) are (approximately) located at the same address and three possible districting layouts. In terms of compactness these solutions are (approximately) equivalent. Figure 4.13a shows an example where both centers are located very closely to their corresponding districts. In contrast to this, the black center in Figure 4.13b lies within its district, whereas the white center is (far) away from



(a) Both centers are located closely to their districts (b) One center is located far away from its district (c) Both centers are located within their districts

Figure 4.13: Different solutions for two centers located at the same address

its district. Figure 4.13c depicts an example where both centers are located within their districts. However, this solution is very poor in terms of contiguity.

The aim of the districting problem with prescribed centers is the following: Partition all basic areas BA into $p := |CE|$ districts that are balanced, contiguous, non-overlapping, compact, and assign exactly one center $g \in CE$ to each district such that the center is located within (or close to) the district. The extension presented here treats the latter as a soft and not as a hard criterion. However, Section 4.5.6.1 presents how to handle this criterion as a hard criterion.

4.5.1 Basic Definitions

First, this subsection adapts the definitions given in Section 4.3.1 to prescribed centers.

Definition 4.5.1 *A* partition problem $PP_c := (B, C)$ *is the problem of sub-dividing a set of basic areas* $B \subseteq BA$ *and a set of centers* $C \subseteq CE$*, into* $1 \leq q = |C| \leq p$ *districts.*

Definition 4.5.2 *A bisecting partition* $BP_c := (B_l, B_r, C_l, C_r)$ *of a partition problem is defined by two sets* $B_l, B_r \subset B$ *such that* $B_l \cup B_r = B$ *and* $B_l \cap B_r = \emptyset$*, and two sets* $C_l, C_r \subset C$ *such that* $C_l \cup C_r = C$ *and* $C_l \cap C_r = \emptyset$ *as well as* $q_l = |C_l|$ *and* $q_r = |C_r|$*.*

4.5.2 Generating Bisecting Partitions

Second, this subsection explains how to generate bisecting partitions in the case of prescribed centers. Let $PP_c = (B, C)$, q_l, q_r, L_D, L_U, and α_k be given. Analogously to Section 4.3.3 $b_1^k, b_2^k, \ldots, b_n^k$ denote the sorted sequence of basic areas according to the angle α_k. Moreover, $c_1^k, c_2^k, \ldots, c_q^k$ denote the sorted sequence of centers according to α_k. For purposes of simplification, b_i^k (c_h^k) denotes the basic area (center) as well as its representative point. Without

loss of generality, do not let two centers lie on a common line with respect to α_k. In the following, this subsection presents necessary modifications of line partitions and flex-zone partitions in order to incorporate prescribed centers.

4.5.2.1 Line Partition

In order to derive a line partition, the RPA firstly determines a^* according to Equation (4.13) and sub-divides the set of basic areas into the subsets $B^k_{l_{a^*}}$ and $B^k_{r_{a^*}}$ (cf. Equations (4.7) and (4.8)). Secondly, it assigns the first q_l elements of the sorted sequence of centers to the left sub-problem and the remaining centers to the right sub-problem, i.e.,

$$C_{l_{q_l}} := \left\{c^k_1; \ldots; c^k_{q_l}\right\}$$

and

$$C_{r_{q_l}} := \left\{c^k_{q_l+1}; \ldots; c^k_q\right\} .$$

Altogether, the RPA generates the line partition $LP_c(k, a^*, q_l) := (B^k_{l_{a^*}}, B^k_{r_{a^*}}, C_{l_{q_l}}, C_{r_{q_l}})$.

Example 4.5.1 Let the basic areas BA and centers CE be given as specified in Table 4.14.

i	1	2	3	4	5	6	7	8	9	10
x_i	0.5	1	2	2.5	3	3.5	4	4.5	5	5
y_i	5	2	4	1	3.5	2.5	5.5	3	1.5	5
w_i	5	3	4	4	4	6	3	5	7	9

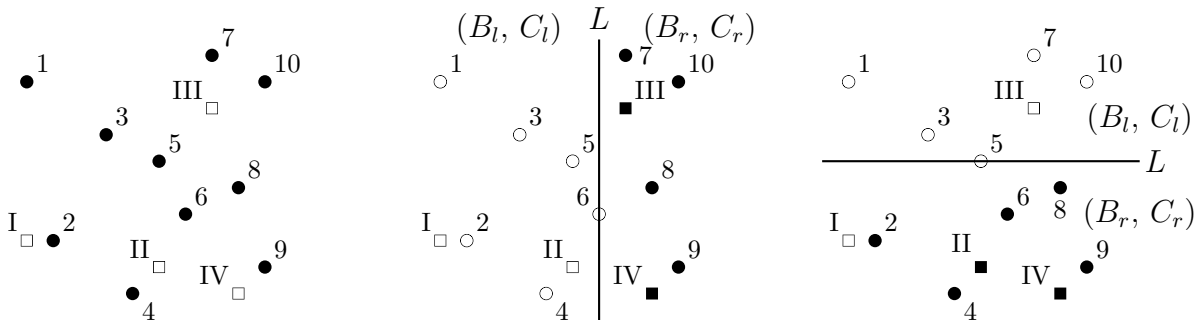
(a) Basic areas BA

h	I	II	III	IV
x_h	0.5	3	4	4.5
y_h	2	1.5	4.5	1

(b) Centers CE

Table 4.14: Specification of the example depicted in Figure 4.14a

Figure 4.14a illustrates this example. Moreover, let $q_l = q_r = 2$. This implies $w(B) = 50$ and $W_l = 25$.
For $\alpha = \pi/2$, i.e., a vertical line, sorting the basic areas leads to 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. The sub-division of the basic areas results in $B_l = \{1; 2; 3; 4; 5; 6\}$ and $B_r = \{7; 8; 9; 10\}$ since $a^* = 6$ ($b^k_{a^*} = 6$). The sorting of the centers results in I, II, III, IV. Since $q_l = 2$ holds, the RPA assigns the first two centers of this sequence to the left sub-problem. This implies $C_{l_{q_l}} = \{I; II\}$ and $C_{r_{q_l}} = \{III; IV\}$. Figure 4.14b depicts the resulting line partition.
For $\alpha = 0$, sorting the basic areas results in 7, 1, 10, 3, 5, 8, 6, 2, 9, 4. Since $a^* = 5$ ($b^k_{a^*} = 5$), the RPA sub-divides the basic areas into the subsets $B_l = \{7; 1; 10; 3; 5\}$ and



(a) Partition problem (BA, CE) (b) Line partition for $\alpha = \pi/2$ (c) Line partition for $\alpha = 0$

Figure 4.14: The line partition approach incorporating prescribed centers

$B_r = \{6; 8; 2; 9; 4\}$. In this case, the sorted sequence of centers is III, I, II, IV, and, hence $C_{l_{q_l}} = \{III; I\}$ and $C_{rq_l} = \{II; IV\}$. Figure 4.14c shows that center I is located on the "wrong" side of the bisecting line. In this case, I is said to lie outside its sub-problem.

4.5.2.2 Flex-Zone Partition

In order to obtain a flex-zone partition the RPA firstly determines l^* and r^* according to Equations (4.16) and (4.17), and the induced sets B_{ll}, B_{fz} and B_{rr}. Next, it assigns the centers to the sub-problems analogously to the line partition approach, i.e., $C_{l_{q_l}} := \{c_1^k; \ldots; c_{q_l}^k\}$ and $C_{r_{q_l}} := \{c_{q_l+1}^k; \ldots; c_q^k\}$. Finally, the RPA assigns each basic area of the flex-zone to one of the sub-problems. In this case, each assignment is based on the distances to the prescribed centers. More precisely, each basic area $i \in B_{fz}$ is assigned to the sub-problem that contains its closest center, i.e.,

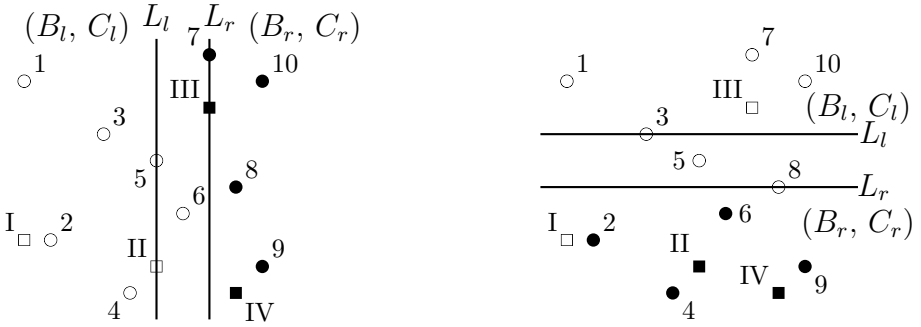
$$B_l := B_{ll} \cup \left\{ i \in B_{fz} \;\middle|\; \arg\min_{h \in C} d_{i,h} \in C_{l_{q_l}} \right\}$$

and

$$B_r := B_{rr} \cup \left\{ i \in B_{fz} \;\middle|\; \arg\min_{h \in C} d_{i,h} \in C_{r_{q_l}} \right\} .$$

Note that Euclidean Distances as well as road distances are usable to determine the closest center. Altogether, this approach results in $FZP_c(k, l^*, r^*, q_l) := (B_l, B_r, C_{l,q_l}, C_{r,q_l})$.

Example 4.5.1 (cont.) Consider the example illustrated in Figure 4.14a and specified in Table 4.14. Moreover, let $\tau = 0.2$. This results in $L_D = 10$, $U_D = 15$, $LL = 20$, and $LU = 30$.



(a) Flex-Zone partition for $\alpha = \pi/2$ (b) Flex-Zone partition for $\alpha = 0$

Figure 4.15: The flex-zone partition approach incorporating prescribed centers

For $\alpha = \pi/2$, the zones are induced by $l^* = 5$, i.e., $b^k_{l^*} = 5$, and $r^* = 7$, i.e., $b^k_{r^*} = 7$ since $w(\{1;\ldots;4\}) = 16 < 20$, $w(\{1;\ldots;5\}) = 20 \geq 20$, $w(\{1;\ldots;7\}) = 29 \leq 30$ and $w(\{1;\ldots;8\}) = 34 > 30$. This leads to the zones $B_{ll} = \{1;2;3;4;5\}$, $B_{fz} = \{6;7\}$ and $B_{rr} = \{8;9;10\}$. Figure 4.15a depicts these corresponding sets. Since $q_l = 2$ holds, the RPA sub-divides the centers into the sets $C_{l_{q_l}} = \{I;II\}$ and $C_{r_{q_l}} = \{III;IV\}$. It holds that $\arg\min_{h\in C} d_{6,h} = II \in C_{l_{q_l}}$ and $\arg\min_{h\in C} d_{7,h} = III \in C_{r_{q_l}}$. Thus, the approach results in $FZP_c(5,7,2) = (\{1;2;3;4;5;6\},\{7;8;9;10\},\{I;II\},\{III;IV\})$.
For $\alpha = 0$, it holds that $B_{ll} = \{7;1;10;3\}$, $B_{fz} = \{5;8\}$, $B_{rr} = \{6;2;9;4\}$, $C_l = \{III;I\}$ and $C_r = \{II;IV\}$. Concerning the flex-zone, it holds that $\arg\min_{h\in C} d_{5,h} = III \in C_{l_{q_l}}$ and $\arg\min_{h\in C} d_{8,h} = III \in C_{l_{q_l}}$. Thus, this approach partitions the basic areas into the subsets $BP_l = \{7;1;10;3;5;8\}$ and $BP_r = \{6;2;9;4\}$. Figure 4.15b illustrates the resulting sub-division.

4.5.3 Choosing a Bisecting Partition

This subsection focuses on the evaluation of bisecting partitions containing prescribed centers and illustrates the differences to the case without centers (cf. Section 4.3.5). In the following, let a partition problem $PP_c = (B, C)$ and a corresponding bisecting partition $BP_c = (B_l, B_r, C_l, C_r)$ be given.

4.5.3.1 Evaluating Balance

Since $|C_l| = q_l$ and $|C_r| = q_r$ holds, the measure described in Section 4.3.5.1 is directly applicable in order to evaluate the balance of a bisecting partition.

4.5.3.2 Evaluating Compactness

The goal of the compactness evaluation of a bisecting partition is the approximation of the compactness of the final solution for generating this bisecting partition. In the case of prescribed centers, the compactness of a district is based on the distances between its center and its allocated basic areas. In contrast to the general case (cf. Section 4.3.5.2), the centers are prescribed and a compactness measure uses them directly. Hence, it makes use of $|C| = q$ centers instead of one center in order to approximate the compactness of (B, C). The RPA provides two approaches to approximate the compactness considering prescribed centers. The first one looks at the problem from the side of the basic areas and allocates each basic area to its closest center. The second approach takes a contrary view on the problem and determines for each center a balanced district around this center.

Weighted Moment of Inertia - Closest Assignment: In order to evaluate a partition problem, the first measure computes for each basic area the distance to its closest center within this partition problem and determines the sum of all basic areas, i.e.,

$$comp_{wmoi-ca}(PP_c) := \sum_{i \in B} w_i \cdot d^2(i, C)\,.$$

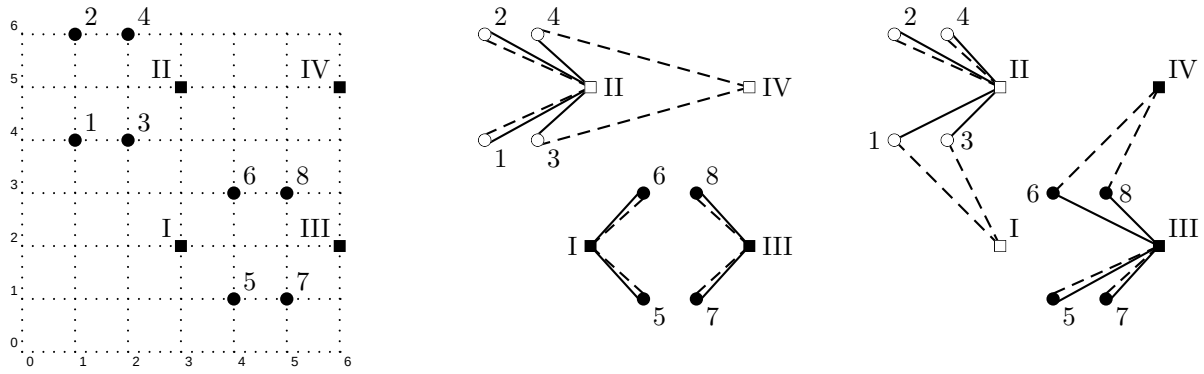
Obviously, $comp_{wmoi-ca}(PP_c)$ is a lower bound for the compactness of a partition problem according to the Weighted Moment of Inertia since no further improvement by reassigning basic areas is possible.
The compactness of a bisecting partition is the sum of the compactness values of its subproblems, i.e.,

$$comp_{wmoi-ca}(BP_c) :=: comp_{wmoi-ca}(B_l, C_l) + comp_{wmoi-ca}(B_r, C_r)\,.$$

The main drawback of this measure is the fact that a closest assignment is usually very unbalanced. Hence, most likely the final solution does not assign many basic areas to the district of their closest center, and, thus, this approximation is very rough. The following example points out this issue.

Example 4.5.2 Figure 4.16a shows a set of basic areas and a set of centers. Assume an activity of one for all basic areas. Figures 4.16b and 4.16c depict two possible bisecting partitions BP_{c1} and BP_{c2}. The solid lines illustrate the allocations considered according to this measure, whereas the dashed lines depict the allocations of the final solution.

In terms of $comp_{wmoi-ca}(\cdot)$ BP_{c1} is evaluated with 22, whereas BP_{c2} with 28. Hence, according to this measure the RPA should prefer BP_{c1}. However, the final solution of the partition



(a) Partition problem (b) Bisecting partition BP_{c1} (c) Bisecting partition BP_{c2}

Figure 4.16: Illustration of $comp_{wmoi-ca}(\cdot)$

problem depicted in Figure 4.16b results in 52 according to the Weighted Moment of Inertia, whereas the final solution of the partition problem depicted in Figure 4.16c results in 40. Hence, BP_{c2} induces a better final solution, although this measure prefers BP_{c1}. For example, for the upper sub-problem of BP_{c1}, this measure allocates all basic areas to center II, whereas the final solution allocates basic areas 3 and 4 to center IV and the corresponding distances are noticeably larger than the distances to center II.

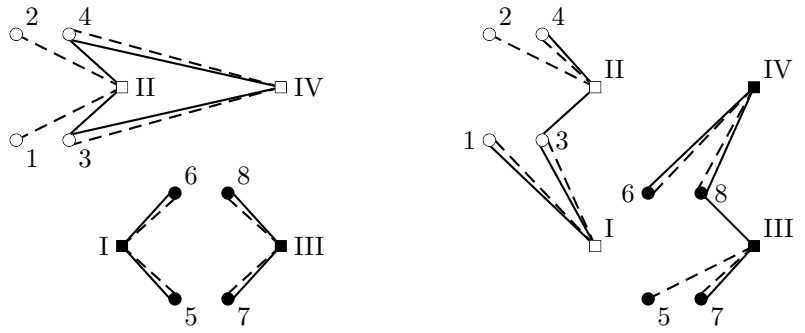
That means, most likely this measure allocates no basic area to a center that is located away from the basic areas. In other words, such a center is not considered in order to approximate the compactness at all. In order to overcome this drawback, the second measure considers all centers anyway.

Weighted Moment of Inertia - Surrounding districts: The main idea of this measure is the approximation of "good" districts in terms of compactness containing for each center $h \in C$ its closest basic areas within the same sub-problem. In order to do so, it includes the closest basic areas as long as the sum of the corresponding activities does not exceed the average activity within this partition problem. Let $b_1^h, b_2^h, \ldots, b_n^h$ be the sequence of basic areas in B sorted in non-decreasing order according to their distances to h. Let a^h be the index, such that

$$w(\{b_1^h; \ldots; b_{a^h}^h\}) \leq \frac{w(B)}{|C|}$$

and

$$w(\{b_1^h; \ldots; b_{a^h+1}^h\}) > \frac{w(B)}{|C|}\,.$$



(a) Bisecting partition BP_{c1} (b) Bisecting partition BP_{c2}

Figure 4.17: Illustration of $comp_{wmoi-st}(\cdot)$

Then,

$$comp_{wmoi-st}(PP_c) := \begin{cases} \sum_{h \in C} \sum_{i=1}^{a^h} w(b_i^h) \cdot d^2(b_i^h, h) & \text{if } |C| > 1 \\ \sum_{i \in B} w_i \cdot d^2(i, C) & \text{if } |C| = 1 \end{cases}$$

defines the compactness of a partition problem. The compactness of a bisecting partition is the sum of the compactness values of its sub-problems, i.e.,

$$comp_{wmoi-st}(BP_c) := comp_{wmoi-st}(B_l, C_l) + comp_{wmoi-st}(B_r, C_r)\,.$$

Example 4.5.2 (cont.) Consider the example illustrated in Figure 4.16a. Figures 4.17a and 4.17b present the bisecting partitions BP_{c1} and BP_{c2} again. A solid line depicts the allocations considered according to this compactness measure, whereas a dashed line depicts the allocations of the final solution.
The compactness values in terms of $comp_{wmoi-st}(\cdot)$ are 46 for BP_{c1} and 34 for BP_{c2}. Hence, with respect to this measure the RPA prefers BP_{c2}. In this case, applying this measure results in the better solution compared to the approach before.

Unfortunately, this measure also has drawbacks. The main drawback is the fact that some basic areas are allocated multiple times to centers, whereas others are not taken into account at all. For example, in order to evaluate BP_{c1} depicted in Figure 4.17a, this measure considers the basic areas 3 and 4 twice, whereas it does not consider the basic areas 1 and 2 at all. However, there are also examples where the first measure performs better than the second.

Remark 4.5.1 The result of $comp_{wmoi-st}(\cdot)$ is not necessarily a lower bound for the compactness of a bisecting partition.

Since both measures have advantages and drawbacks, an obvious approach is to combine them, e.g., by using a weighted sum of their (normalized) evaluation values. Section 4.5.7 will present computational results that confirm the quality of this combination.

4.5.3.3 Evaluating Center Location

At best, the center is a central point of the district. But, since this extension deals with existing centers, it may generate solutions where centers lie at the border or even outside their districts. A center's location within the "wrong zone" of a sub-division, i.e., outside its sub-problem, implies a center's location outside its final district. Hence, this measure penalizes a center in this case. In order to define the penalization, the measure distinguishes between line partitions and flex-zone partitions. Without loss of generality, assume $\alpha = \pi/2$.

Line Partition: Let a^* be defined according to Equation (4.13). Formally, a center h of the left (right) sub-problem lies *outside* the respective sub-problem if its x-value is greater (smaller) than the x-value of $b^k_{a^*}$. In this case, the measure penalizes this center by the distance from c_h to the closest basic area within its sub-problem, i.e.,

$$loc(h) := \begin{cases} 0 & \text{if } (h \in C_l \text{ and } x_h \leq x_{a^*}) \text{ or } (h \in C_r \text{ and } x_h \geq x_{a^*}) \\ d(c_h, B_l) & \text{if } h \in C_l \text{ and } x_h > x_{a^*} \\ d(c_h, B_r) & \text{if } h \in C_r \text{ and } x_h < x_{a^*} \end{cases} .$$

Example 4.5.1 (cont.) Consider the line partition depicted in Figure 4.14b. It locates all centers within their sub-problems, i.e., $loc(I) = loc(II) = loc(III) = loc(IV) = 0$. In contrast to this, consider the line partition illustrated in Figure 4.14c. Here, center I lies outside its sub-problem since it is assigned to the left sub-problem, but located to the right of the bisecting line. Its closest basic area within the left sub-problem is basic area 3, and, thus, $loc(I) = d(I, 3)$ holds. Moreover, $loc(II) = loc(III) = loc(IV) = 0$ holds since this line partition locates centers I, II and III within their sub-problems.

Flex-Zone Partition: Let l^* and r^* be defined according to Equations (4.16) and (4.17). In this case, a center h of the left (right) sub-problem is defined as *outside* its sub-problem if its x-value is greater (smaller) than the x-value of $b^k_{r^*}$ ($b^k_{l^*}$). In other words, a center of the

left (right) sub-problem is outside its sub-problem if it is located in the right (left) zone. Again, its penalty value corresponds to the distance to its closest basic area within the same sub-problem. Usually, a center located in the flex-zone is close to some basic areas also located in the flex-zone. Hence, most likely these basic areas are assigned to the same sub-problem. That is the reason why a center located in the flex-zone is not defined as outside its sub-problem. This implies that the measure does not penalize this center. Altogether,

$$loc(h) := \begin{cases} 0 & (h \in C_l \text{ and } x_h \leq x_{r^*}) \text{ or } (h \in C_r \text{ and } x_h \geq x_{l^*}) \\ d(c_h, B_l) & g \in C_l \text{ and } x_h > x_{r^*} \\ d(c_h, B_r) & g \in C_r \text{ and } x_h < x_{l^*} \end{cases} .$$

defines the measure.

Example 4.5.1 (cont.) Consider the flex-zone partition illustrated in Figure 4.15a. It locates the centers I and II in the left zone, center III in the flex-zone and center IV in the right zone. Hence, this measure penalizes none of these locations.
In contrast to this, consider Figure 4.15b. This flex-zone partition locates center I in the right zone but assigns it to the left sub-problem. This implies $loc(I) = d(I, 3)$. Moreover, evaluating the further location leads to $loc(II) = loc(III) = loc(IV) = 0$.

In order to evaluate a bisecting partition as a whole this measure uses the sum of the evaluations of all centers, i.e.,

$$loc(BP_c) := \sum_{h \in C} loc(h) .$$

4.5.4 Algorithm Overview

The solution approach incorporating prescribed centers is largely identical to the one for the basic model described in Algorithm 4.3.2. Thus, this section restricts itself to the changes. First, C, C_l, and C_r replace q, q_l, and q_r, respectively. Second, allocating each basic area to its closest center generates the best possible solution with respect to compactness according to Equation (4.30) or (4.31), respectively. The algorithm has an optional feature activated by the flag *useCloseAss* that utilizes this fact. If this closest assignment induces a feasible solution for a partition problem, this feature directly uses it. Thus, Algorithm 4.5.1 replaces Step 2 compared to Algorithm 4.3.2.

Algorithm 4.5.1: Extended Recursive Partitioning Algorithms

Input: *Set of basic areas BA, set of centers CE, a set of measures MEA, set of approaches to determine bisecting partitions PA, parameters τ, L_D, U_D, K, $\beta_1, \ldots, \beta_{|MEA|}$, $PPMax$, $RelMax$, $useCloseAss$.*

⋮

if $|C| = 1$ **then**
  Set $S = S \cup \{(B, C)\}$, $UPP = UPP \backslash \{PP_c\}$ and GOTO 5.
if *($useCloseAss = true$) and ($pos(PP) = 0$)* **then**
  forall the $h \in C$ **do**
    Determine $B_h := \left\{ i \in B \;\middle|\; h = \arg\min_{g \in C} d(b_i, c_g) \right\}$.
    if *($w(B_h) < L_D$) or ($w(B_h) > U_D$)* **then** *GOTO 3.*
  end
  forall the $h \in C$ **do**
    Set $S = S \cup \{(B_h, h)\}$.
  end
  Set $UPP = UPP \backslash \{PP_c\}$ and GOTO 5.
end

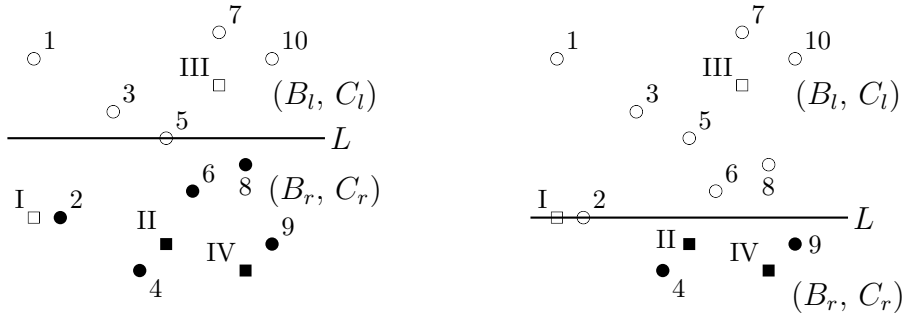
⋮

4.5.5 Complexity

This subsection analyzes the complexity of this extension. In particular, it focuses on the changes compared to the general case (cf. Section 4.3.7).

For each partition problem and each search direction the sorted sequence of centers is computed in $\mathcal{O}(|C| \cdot \log |C|)$ time. Hence, computing a line partition needs $\mathcal{O}(|B| \cdot \log |B| + |C| \cdot \log |C|)$ time. However, since $|C| < |B|$ holds, a line partition is computed in $\mathcal{O}(|B| \cdot \log |B|)$ time. Computing a flex-zone partition requires $\mathcal{O}(|B| \cdot \log |B| + |C| \cdot \log |C| + |B| \cdot |C|)$ time. Since the flex-zone approach considers all distances between basic areas and centers in the worst case and since $|C| < |B|$ holds, the required time is $\mathcal{O}(|B| \cdot (\log |B| + |C|))$.
Evaluating balance still requires $\mathcal{O}(|B|)$ time, whereas evaluating compactness as well as evaluating the centers' locations needs $\mathcal{O}(|B| \cdot |C|)$ time. Hence, generally generating and evaluating one bisecting partition for one partition problem takes $\mathcal{O}(|B| \cdot (\log |B| + |C|))$ time. Determining a closest assignment is in $\mathcal{O}(|B| \cdot |C|)$.
Thus, solving a partition problem requires $\mathcal{O}(K \cdot |B| \cdot (\log |B| + |C|)) + K \cdot \log K)$ time.

The overall complexity of the algorithm is $\mathcal{O}(|C| \cdot K \cdot (|B| \cdot (\log |B + |C|)) + logK)$ since $p = |C|$.



(a) Center located outside its sub-problem (b) Sweeping the bisecting line

Figure 4.18: Making an infeasible line partition feasible

4.5.6 Extensions

This subsection describes two possible extensions. The first one treats the location of the centers within their corresponding districts as a hard criterion and the second one incorporates different capacities of the districts.

4.5.6.1 Center Location as a Hard Criterion

A possible extension requires that each center must lie within its district. Hence, each center must already lie within its sub-problem (cf. Section 4.5.3.3). Consequently, a bisecting partition is now infeasible if $loc(BP_c) > 0$ holds. However, instead of discarding an infeasible bisecting partition right away, this extension tries to make it feasible by reassigning some basic areas. Let a partition problem and the numbers q_l and q_r be given. Without loss of generality, assume $\alpha_k = \pi/2$. In order to reassign some basic areas this extension distinguishes line partitions and flex-zone partitions.

Line Partitions: Let a^* be defined according to Equation (4.13). If $x_{b_{a^*}}$ is smaller (greater) than $x_{c_{q_l}}$ ($x_{c_{q_{l+1}}}$), this extensions shifts all basic areas with $x_{b_i} \leq x_{c_{q_l}}$ ($x_{b_i} \geq x_{c_{q_{l+1}}}$) from B_r (B_l) to B_l (B_r). Figuratively spoken, it sweeps the line unless each center lies within its sub-problem. Unfortunately, the resulting line partition may be infeasible in terms of balance.

Example 4.5.1 (cont.) Consider the line partition depicted in Figure 4.18a. This line partition locates center I outside its sub-problem. Hence, this extension sweeps the line such that center I lies within its sub-problem afterwards. Figure 4.18b illustrates the resulting line partition. In this case, the balance is noticeably worse.

Flex-Zone Partitions: Let l^* and r^* be defined according to Equations (4.16) and (4.17). If $x_{c_{q_l}} \leq x_{b_{r^*}}$ and $x_{c_{q_{l+1}}} \geq x_{b_{l^*}}$, the flex-zone partition is feasible according to this criterion. Otherwise, the extension redetermines l^* and r^* using the bounds LL and LU defined in Equations (4.14) and (4.15), i.e., it allows the flex-zone to exploit the feasible deviation completely. If $x_{c_{q_l}} > x_{b_{r^*}}$ or $x_{c_{q_{l+1}}} > x_{b_{l^*}}$ still holds, the extension discards this search direction.

4.5.6.2 Capacities

In addition, capacities can be associated with the centers, e.g., some service persons work full-time, whereas others work part-time. In this case, a solution is infeasible if it contains at least one district having an activity that exceeds the capacity of the corresponding center. However, a solution where some districts nearly exploit their capacities and some others are almost empty is non-satisfying in terms of balance. Therefore, a solution is balanced if the utilizations of all districts are nearly equal. Let

$$uti(D_g) := \frac{w(B_g)}{cap_{cen_g}}$$

be the utilization of D_g and

$$\mu_{ut} := \frac{\sum\limits_{i \in BA} w_i}{\sum\limits_{h \in C_l} cap_h}$$

the average utilization. Moreover, let τ_{ut} define the feasible deviation from μ_{ut}, i.e., a feasible district in terms of utilization satisfies the inequalities

$$\mu_{ut} - \tau_{ut} \leq uti(D_g) \leq \max\{\mu_{ut} + \tau_{ut}; 1\}\,.$$

Consequently, the balance of a solution is the maximum deviation of one district from μ_{ut}, i.e.,

$$bal_{ut}(S) := \max_{h=1,\ldots,|C|} |uti(D_g) - \mu_{ut}|\,.$$

Accordingly, for line partitions the activity of the left sub-problem should result in

$$W_{l,ut} := w(BA) \cdot \frac{\sum\limits_{h \in C_l} cap_h}{\sum\limits_{h \in C} cap_h}\,.$$

A feasible activity of a partition problem(B, C) satisfies the inequalities

$$(\mu_{ut} - \tau_{ut}) \cdot \sum_{h \in C} cap_h \leq w(B) \leq \max\{\mu_{ut} + \tau_{ut}; 1\} \cdot \sum_{h \in C} cap_h \,.$$

Hence, accordingly, this extension adapts LL and LU.

4.5.7 Computational Results

The following tests are conducted on a dataset containing 44 instances provided by a project partner. The basic areas correspond to locations of customers whereas the prescribed centers correspond to the locations of salespersons. The number of basic areas ranges from 284 to 38667 while the number of prescribed centers varies from 2 to 160. These tests use a combination of line and flex-zone partitions and the parameter settings $\tau = 0.05$, $K = 16$, $PPMax = 10p$, and $RelMax = 3$.

In order to obtain a lower bound for the compactness, for each instance a solution is used where each basic area is assigned to its closest center, regardless of balance. For purposes of comparability, the values of $comp_{wmoi}$ are stated as relative percentage deviations from the values of this closest assignment solution. The balance and contiguity is evaluated analogously to Section 4.4. Moreover, loc_{po} states the percentage of the centers located outside the convex hull of their associated basic areas. Note that a center could be located outside this convex hull, although it lies within the same sub-problem.

The first test addresses the two compactness measures presented in Section 4.5.3.2. In order to compare them, only compactness is used when evaluating a bisecting partition.

comp.	*bal*		*comp*	*ctg*		*loc*
measure	*max*	*ave*	*wmoi*	*max*	*ave*	*po*
$wmoi - ca$	4.83	3.40	68.79	0.509	0.108	10.23
$wmoi - st$	4.84	3.32	89.46	0.797	0.083	11.03
combination	4.84	3.39	57.00	0.494	0.063	10.45

Table 4.15: Comparing different compactness measures

Table 4.15 compares the exclusive use of $comp_{wmoi-ca}$, the exclusive use of $comp_{wmoi-st}$, and an equally weighted combination of them. The results confirm the theoretical thoughts of Section 4.5.3.2. In terms of compactness the combined measure outperforms the single use of one measure having a compactness value of 57.00% compared to 68.79% and 89.46%, respectively. Also in terms of contiguity the combination is slightly better. In terms of

balance and according to the number of centers located outside their districts the results are comparable. Hence, this test points out that combining both approaches in order to measure compactness is advisable.

The second test addresses the evaluation of a bisecting partition. In this case the ranking function (cf. Equation (4.28)) consists of three parts: Balance (cf. Section 4.5.3.1), compactness, and center location (cf. Section 4.5.3.3). For measuring compactness, following the result of the previous test, the combination of both measures is applied.

criteria weights			*bal*		*comp*	*ctg*		*loc*
balance	compact-ness	center location	*max*	*ave*	*wmoi*	*max*	*ave*	*po*
closest assignment			82.35	29.86	0.00	0.00	0.00	0.00
0.00	1.00	0.00	4.84	3.39	57.99	0.494	0.063	10.45
0.00	0.75	0.25	4.85	3.37	57.61	0.345	0.051	8.13
0.00	0.50	0.50	4.84	3.37	59.03	0.345	0.054	7.22
0.00	0.25	0.75	4.84	3.37	59.76	0.345	0.055	6.94
0.25	0.75	0.00	0.37	0.15	74.75	0.582	0.035	12.40
0.25	0.50	0.25	0.34	0.12	75.42	0.183	0.009	10.17
0.25	0.25	0.50	0.28	0.11	78.58	0.129	0.010	9.88
0.50	0.50	0.00	0.27	0.10	78.68	0.588	0.020	12.78
0.50	0.25	0.25	0.22	0.09	79.75	0.288	0.009	9.79
0.75	0.25	0.00	0.19	0.08	81.58	0.609	0.020	13.76

Table 4.16: Varying the weights for evaluating a bisecting partition

Table 4.16 compares the results for varying the weights of these three criteria. In addition, the first row of Table 4.16 states the evaluations of the closest assignment solutions. As expected, these solutions are very unbalanced with an average value of 82.35% for bal_{max}. However, these solutions exhibit no overlap, i.e., $ctg(S) = 0$, and all centers are located within their districts.

Unsurprisingly, in terms of balance the solution improves for increasing the corresponding weight. Even setting this weight to 0.25 results in nearly perfectly balanced districts. Usually, with respect to compactness an increase of the corresponding weight leads to better results. However, ignoring balance but incorporating the center location by setting the corresponding weight to 0.25 slightly improves the solution's compactness compared to exclusively using compactness. Since there is also an improvement in terms of the number of centers located outside, it is recommendable to incorporate the center location criterion into the evaluation of bisecting partitions.

4.6 Incorporating Multiple Activity Measures

Some applications incorporate multiple activities, $a = 1, \ldots, |A|$, for example both working time and sales potential. Therefore, a solution should be balanced with respect to all activity measures. In this case, the ranking function contains one balance measure for each activity. Moreover, in order to treat balance as a hard criterion this extension needs a feasible deviation τ^a for each activity. A solution is feasible if for each activity measure its balance is feasible with respect to this deviation, formally, if

$$(1 - \tau^a) \cdot \mu^a \leq w^a(B_g) \leq (1 + \tau^a) \cdot \mu^a \quad \forall\, g = 1, \ldots, p\,, a = 1, \ldots, A$$

holds. In the following, this subsection presents the necessary modifications of line partitions and flex-zone partitions in order to incorporate multiple activities. All over this section, let β_a be the user-given weight of bal^a in the ranking function.

4.6.1 Line Partition

In a first step, for each activity this extension determines the first line and the last line that is feasible according to the corresponding balance measure. The determination is analogous to the determination of the flex-zone bounds using Equations (4.14) and (4.15). Next, the extension examines only those lines that are feasible according to all dimensions of activity in more detail and evaluates their corresponding line partition by

$$bal_{ma}(LP) := \sum_{a=1}^{A} \beta_a \cdot bal^a(LP)\,.$$

Finally, it chooses the best one according to $bal_{ma}(\cdot)$.

4.6.2 Flex-Zone Partition

In order to obtain a flex-zone partition the extension determines the corresponding flex-zone B^a_{fz} for each activity. Then, it computes the intersection of these flex-zones, i.e.,

$$B^{ma}_{fz} := \bigcup_{a=1,\ldots,|A|} B^a_{fz}\,.$$

All basic areas left (right) to this zone are assigned to the left (right) zone. The assignment of the basic areas of the flex-zone to the sub-problems works as before.

Finally, consider the backtracking mechanism (cf. Section 4.3.6). If a relaxation is necessary, the extension relaxes the bounds in increasing order of their corresponding weights in the ranking function.

4.7 EMS Regions

The topic of this section is an application in medical services. Emergency Medical Services (EMS) are responsible for the treatment of medical emergencies as well as for the transport of patients that need medical assistance during the transport. Usually, the latter are no emergency but scheduled transports. In general, they are transports from a hospital, to a hospital or between hospitals. In Germany these transports are organized or coordinated by EMS regions. Since each coordination center plans the transports within its region, two coordination centers have to coordinate a transport between two different EMS regions. Hence, the handling of so-called cross-border transports is more difficult. Currently, there is a discussion about the reorganization of these regions. Mainly, a reduction of the number of regions in order to reduce costs is desired. For example, in Baden-Württemberg there is a discussion to reduce the number of EMS regions from 34 to just 8.

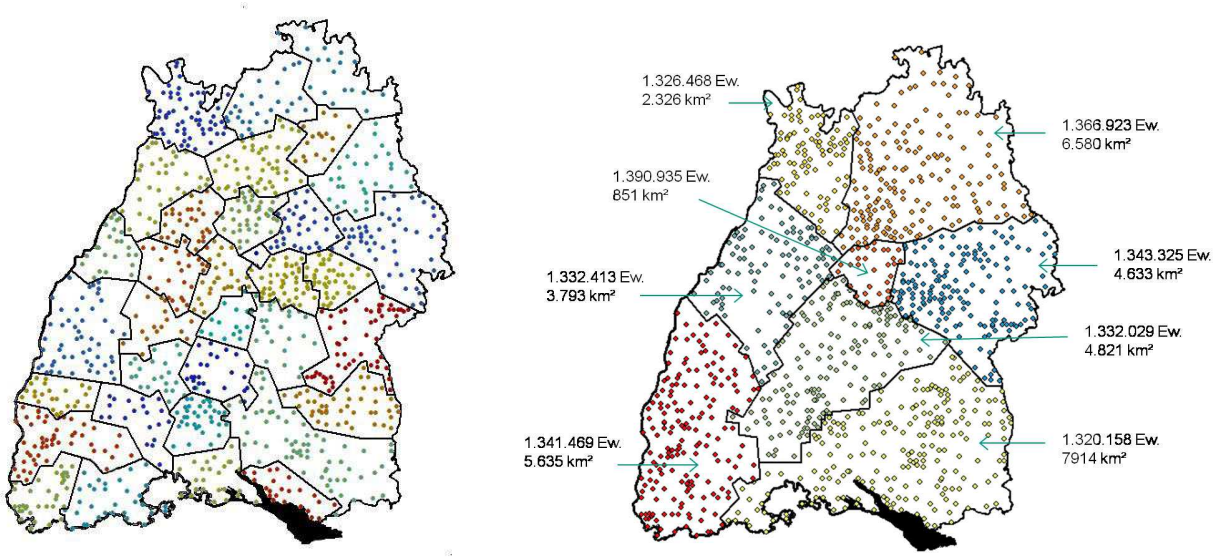
The problem of reorganizing the EMS regions can be interpreted as a districting problem. In this context the basic areas are cities. The EMS regions should be balanced, contiguous and compact. Another goal is the minimization of the number of cross-border transports. Since each hospital or ambulance station is located in a city, the transports are merged to transports between basic areas. Let t_{ij} be the number of expected transports between the basic areas i and j. The number of cross-border transports of a solution S results in

$$cbt(S) := \sum_{g=1}^{p-1} \sum_{h=g+1}^{p} \sum_{i \in D_g} \sum_{j \in D_h} t_{ij} \,.$$

The number of cross-border transports of a bisecting partition results in

$$cbt(BP) := \sum_{i \in B_l} \sum_{j \in B_r} t_{ij} \,.$$

This section outlines the necessary modifications according to the generation of bisecting partitions. A more detailed presentation of this topic is given by Butsch et al. [5]. After determining a bisecting line this extension applies a step that shifts basic areas between the sub-problems in order to reduce the number of transports between them. Let LP^{init} be the bisecting partition induced by the determined line. This step is based on an algorithm of Fiduccia and Mattheyses [7]. The main idea is to shift basic areas from B_l to B_r or vice versa. The goals of this step are the maintenance of the balance and the compactness of the initial line partition and the similarity to the initial line partition. Hence, this step uses a threshold τ_{rcb} for the balance and a maximum deviation of the Weighted Moment of Inertia



(a) Current EMS regions of Baden-Württemberg [5]

(b) Proposed EMS regions for Baden-Württemberg [5]

Figure 4.19: Applying the extended RPA

υ_{rcb}, i.e., a bisecting partition BP^{cur} is feasible if

$$bal(BP^{cur}) \leq \min\{\tau; bal(LP^{init}) + \tau_{rcb}\}$$

and

$$comp_{wmoi}(BP^{cur}) \leq comp_{wmoi}(LP^{init}) \cdot (1 + \upsilon_{rcb}).$$

In order to obtain a bisecting partition similar to the initial line partition this extension fixes the centers of the sub-problems to those of the initial line partition. Moreover, it shifts each basic area at most once. In each iteration and for each basic area unshifted yet, it determines the change in the number of cross-border transports for shifting this basic area. Next, it computes the best candidate for both sub-problems and implements it, even if this shifting increases the number of cross border transports. Moreover, it stores the resulting bisecting partition in a list of feasible bisecting partitions. The approach stops if the candidates of both sub-problems induce an infeasible bisecting partition or if there is no unshifted basic area anymore. Finally, it chooses the bisecting partition of the stored list with the minimal number of cross-border transports.

Since we had no real world transport data, we approximated the number of transports based on the population distribution and the distances between the basic areas, i.e., cities or communities, and the surrounding hospitals. Figure 4.19a presents the current EMS layout

of Baden-Württemberg consisting of 34 EMS regions. Figure 4.19b presents a proposal of a new EMS layout consisting of 8 EMS regions resulting from the extended RPA. With a maximum deviation of 3% from the average number of inhabitants, this layout is much more balanced than the current layout, which has a deviation of up to 216%. Furthermore, based on the assumptions the number of cross-border transports reduces by over 66%.

4.8 Conclusions

This chapter has presented the Recursive Partitioning Algorithm, a geometric heuristic for districting problems considering point represented basic areas. Such problems arise for example in the context of service and sales districting or the design of pickup and delivery districts. The original version of this heuristic proposed by Kalcsics et al. [16] sub-divides the districting problem into smaller and smaller problems recursively by means of lines. However, the original version has some weaknesses in terms of compactness. That is why this chapter has enhanced this approach, for example by presenting a more flexible way of sub-division by introducing a flex-zone. Moreover, it has improved the evaluation of bisecting partitions in terms of compactness. In contrast to many other approaches, the RPA treats both compactness and balance as a soft criterion. Hence, the user defines his preferences by setting weights to these criteria. Tests on real-world data have confirmed the efficiency of this approach and the suitability for an interactive use.

In addition, this chapter has shown how to incorporate network distances into the RPA, although the RPA is a geometric approach. Moreover, it has presented some practical extensions, for example the incorporation of prescribed centers or multiple activity measures. Furthermore, an adaptation of the RPA in order to determine emergency medical service regions has been presented.

A possible extension could be a more exact approximation of the induced routing times for each sub-division. However, several different approaches would be necessary since the routing depends on the visit frequency of the customers, for example.

Some of the enhancements, mainly some compactness approximations for bisecting partitions, lead to an increase of the running times. Hence, some approaches to keep the running times small are possible. Especially the usage of parallelization techniques could be promising because the unsolved sub-problems can be solved independently of each other.

Bibliography

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

[2] J. Beardwood, J. Halton, and J. Hammersley. The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, 55:299–327, 10 1959.

[3] B. Bozkaya, E. Erkut, and G. Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1): 12–26, 2003.

[4] A. Butsch, J. Kalcsics, S. Nickel, and M. Schröder. Geometric Approaches to Districting Problems. Working paper, 2015.

[5] A. Butsch, S. Nickel, and M. Reuter-Oppermann. Applying districting heuristics to determine EMS regions in Germany. Working paper, 2016.

[6] A. Drexl and K. Haase. Fast Approximation Methods for Sales Force Deployment. *Management Science*, 45(10):1307–1323, 1999.

[7] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proceedings of the 19th Design Automation Conference*, DAC '82, pages 175–181, Piscataway, NJ, USA, 1982. IEEE Press.

[8] B. Fleischmann and J. N. Paraschis. Solving a large scale districting problem: a case report. *Computers & Operations Research*, 15(6):521–533, 1988.

[9] E. Forrest. Apportionment by Computer. *American Behavioral Scientist*, 23(7):23–35, 1964.

[10] K. Haase and S. Müller. Upper and lower bounds for the sales force deployment problem with explicit contiguity constraints. *European Journal of Operational Research*, 237(2): 677–689, 2014.

[11] D. Haugland, S. C. Ho, and G. Laporte. Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3):997–1010, 2007.

[12] S. W. Hess and S. A. Samuels. Experiences with a Sales Districting Model: Criteria and Implementation. *Management Science*, 18(4-part-ii):41–54, 1971.

[13] D. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(1):148–162, 1982.

[14] A. I. Jarrah and J. F. Bard. Large-scale pickup and delivery work area design. *Computers & Operations Research*, 39(12):3102–3118, 2012.

[15] J. Kalcsics. *Unified Approaches to Territory Design and Facility Location*. Shaker Verlag, 2006. ISBN 978-3832252595. PhD Thesis, Saarland University, Germany.

[16] J. Kalcsics, S. Nickel, and M. Schröder. Towards a Unified Territorial Design Approach – Applications, Algorithms and GIS Integration. *TOP*, 13(1):1–74, 2005.

[17] R. Klein. *Algorithmische Geometrie*. Addison-Wesely-Longman, Bonn, 1997. ISBN 978-3827311115.

[18] H. Lei, G. Laporte, and B. Guo. Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1–2):67–85, 2012.

[19] H. Lei, G. Laporte, Y. Liu, and T. Zhang. Dynamic design of sales territories. *Computers & Operations Research*, 56:84–92, 2015.

[20] H. Lei, R. Wang, and G. Laporte. Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. *Computers & Operations Research*, 67:12–24, 2016.

[21] R. Z. Ríos-Mercado and E. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776, 2009.

[22] R. Z. Ríos-Mercado and J. F. López-Pérez. Commercial territory design planning with realignment and disjoint assignment requirements. *Omega (United Kingdom)*, 41(3): 525–535, 2012.

[23] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and M. Cabrera-Ríos. New Models for Commercial Territory Design. *Networks and Spatial Economics*, 11(3):487–507, 2011.

[24] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and J. L. González-Velarde. A bi-objective programming model for designing compact and balanced territories in commercial districting. *Transportation Research Part C: Emerging Technologies*, 19(5):885–895, 2011.

[25] H. Zhong, R. W. Hall, and M. Dessouky. Territory planning and vehicle dispatching with driver learning. *Transportation Science*, 41(1):74–89, 2007.

5 Power Diagram Districting Algorithm

Contents

The RPA presented in Chapter 4 is based on a geometric divide and conquer approach. It puts more emphasis on contiguity and balance than on compactness. Therefore, this chapter introduces another geometric approach focusing on compactness. It can either be used as a stand-alone algorithm or as a post-processing step applied to the solutions of the RPA.

The so-called *Power Diagram Districting Algorithm* (PDDA) is based on generalized Voronoi Diagrams, or more precisely on Power Diagrams. For a given set of generator points a Voronoi Diagram partitions an overall area into so-called Voronoi regions such that each region contains all points that are closer to the corresponding generator than to any other generator. A more detailed description of Voronoi Diagrams is given in Aurenhammer [1] or Klein [7]. The usage of Voronoi Diagrams for districting has two main problems. On the one hand Voronoi regions are not necessarily balanced. On the other hand, a Voronoi Diagram needs a set of generators, and, thus, the algorithm has to specify this set. Here, it should be noted that the location of the generators has a significant impact on the solution, that is for example pointed out by Moreno-Regidor et al. [8]. Section 5.5.4 will present some results confirming this point.

In order to overcome the first problem generalized Voronoi Diagrams such as weighted Voronoi Diagrams can be used. Here, an additional weight is associated with each generator. This weight is multiplied with or added to the distances between the corresponding generator and the points of the overall area. More details about generalized Voronoi Diagrams can be found in Aurenhammer [1]. A skillful determination of these weights helps to obtain more balanced regions. Moreover, Power Diagrams are a variation of additively weighted Voronoi Diagrams using squared Euclidean distances. In contrast to multiplicatively weighted Voronoi Diagrams, for additively weighted Voronoi Diagrams or Power Diagrams, respectively, the connectivity of the achieved regions is guaranteed.

The remainder of this chapter is organized as follows. The first section gives some definitions concerning solutions and districts in the context of generalized Voronoi Diagrams. Then, Section 5.2 reviews the literature on districting approaches based on generalized Voronoi Diagrams. Section 5.3 presents our Algorithm in detail followed by a multi-start variant in Section 5.4. After that, Section 5.5 presents the results of extensive computational tests that confirm the suitability of the proposed approach. Finally, Section 5.6 outlines some possible extensions. The chapter concludes with a summary and a short outlook.

5.1 Basic Definitions

A Voronoi Diagram partitions all points of an overall area into regions. Throughout this chapter, let $G := \{g_1; \ldots; g_p\}$ denote a set of points in $\mathbb{R}^2$, the so-called *generator points* or *generators* for short.

Definition 5.1.1 *A* Voronoi Diagram *consist of a set* $R_1^{Vo}(G), \ldots, R_p^{Vo}(G)$ *of* Voronoi regions*, where a Voronoi region* $R_h^{Vo}(G)$ *is formally defined by*

$$R_h^{Vo}(G) := \left\{ x \in \mathbb{R}^2 \mid d(x, g_h) < d(x, g_j) \ \forall j \neq h \right\} .$$

However, in order to solve a districting problem only a predefined set of points has to be partitioned, namely the set of points corresponding to basic areas. In the following, this section gives some definitions of districts and solutions in the context of (generalized) Voronoi Diagrams.

Definition 5.1.2 *A* Voronoi districting plan (VDP) $S^{Vo}(G) := \{D_1^{Vo}(G); \ldots; D_p^{Vo}(G)\}$ *for* BA *with respect to* G *is defined by*

$$D_h^{Vo}(G) := \{ i \in BA \mid d(b_i, g_h) < d(b_i, g_j) \ \forall 1 \leq j, h \leq p, \ j \neq h \} ,$$

where $D_h^{Vo}(G)$ *is called* Voronoi district (VD)*.*

Figuratively spoken, each basic area is assigned to the district of its closest generator. Throughout this section, we assume without loss of generality that no basic area has exactly the same (weighted) distance to two or more generators. Therefore, each basic area is assigned to exactly one district. This assumption can be made, because if there would be more than one generator having the same distance, an assignment rule for equal distances satisfying the criterion of exclusive and total assignment can be defined. Note that for a given set of generators a VDP optimizes the sum of the (weighted) (squared) distances from the basic areas to the corresponding generators. This is similar to the optimization of the (Weighted) Moment of Inertia, a compactness measure defined in Section 3.3.5.1, differing only in the fact that the centers are predefined here. However, a VDP has not to be balanced, even empty VDs are possible.

Example 5.1.1 Let the set of basic areas specified in Table 5.1a and the set of generators specified in Table 5.1b be given.

i	1	2	3	4	5	6	7	8	9	10
x_i	0.5	1	2	2.5	3	3.5	4	4.5	5	5
y_i	5	2	4	1	3.4	2.5	5.5	3	1.5	5
w_i	5	3	4	4	4	6	3	5	7	9

(a) Set of basic areas BA

h	1	2	3
x_h	1	3	4
y_h	3	2	4

(b) Set of generators G

Table 5.1: Specification of the example depicted in Figure 5.1

Figure 5.1a presents the resulting VDP. It illustrates basic areas as circles, whereas it illustrates generators by squares. For purposes of clarity, this figure depicts the convex hulls of the basic areas comprising the districts additionally. The corresponding activities measures are $w(D_1^{Vo}(G)) = 12$, $w(D_2^{Vo}(G)) = 17$ and $w(D_3^{Vo}(G)) = 21$. Here, the average district size is 16.67. Hence, the maximum percentage deviation is 28%, i.e., the solution is not well balanced. Note that often a maximum balance of 5% or 10% is required.

A possible generalization of Voronoi Diagrams facilitating in order to obtain balanced solutions are so-called multiplicatively weighted Voronoi Diagrams, where each generator has an additional weight. Throughout this chapter, let $V := \{v(g_1); \ldots; v(g_p)\}$ denote these weights. In the case of multiplicatively weighted Voronoi Diagrams, these weights are non-negative, i.e., $v(g_h) \in \mathbb{R}_+ \; \forall 1 \leq h \leq p$.

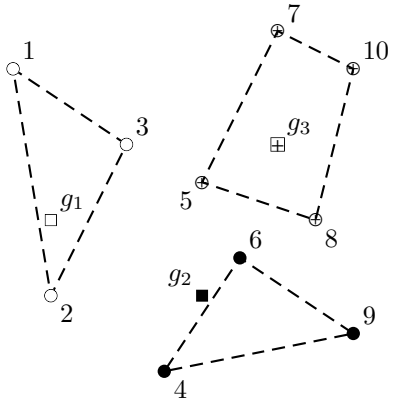
Definition 5.1.3 *For BA, a* multiplicatively weighted Voronoi districting plan (MWVDP) *$S^{MV}(G,V) := \{D_1^{MV}(G,V); \ldots; D_p^{MV}(G,V)\}$ with respect to G and V is defined by*

$$D_h^{MV}(G,V) := \{i \in BA \mid v(g_h) \cdot d(b_i, g_h) < v(g_j) \cdot d(b_i, g_j) \; \forall \; 1 \leq j, h \leq p \; j \neq h\} \; ,$$

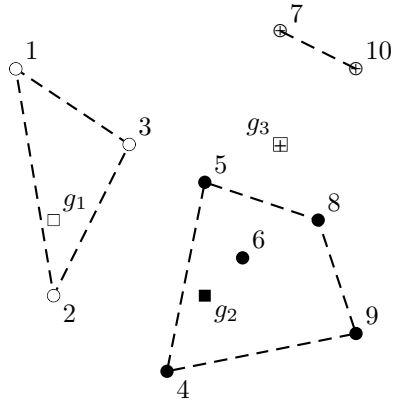
where $D_h^{MV}(G,V)$ is called multiplicatively weighted Voronoi district (MWVD).

This approach differs from the former approach in the usage of weighted distances. The weights control the spatial extension of the corresponding districts. The higher the weight of a generator, the smaller the spatial extension of its district, as the following example shows.

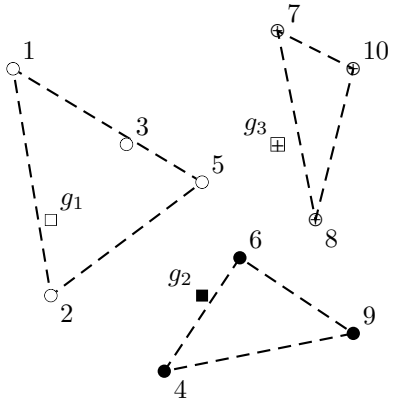
Example 5.1.1 (cont.) Let now the weights $v(g_1) = 1$, $v(g_2) = 1$ and $v(g_3) = 2$ be given. For basic area 5, $d(5, g_1) = 2.06$, $d(5, g_2) = 1.5$ and $d(5, g_3) = 1.12$ holds. Incorporating the generators' weights leads to $v(g_2) \cdot d(5, g_2) = 1.5 < v(g_3) \cdot d(5, g_3) = 2.24$. Hence, basic area 5 is assigned to the district of generator g_2, i.e., $5 \in D_2^{MV}(G,V)$. Furthermore, in this case



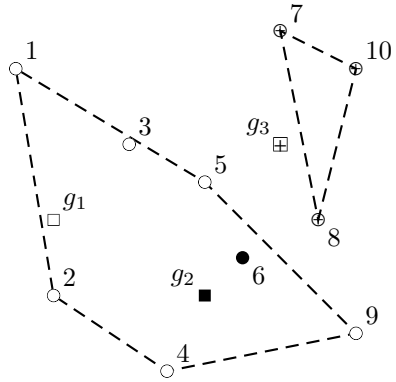
(a) VDP



(b) MWVDP with $V = \{1; 1; 2\}$



(c) MWVDP with $V = \{1; 1.5; 2\}$ and AWVDP with $V = \{0; 1; 1.5\}$



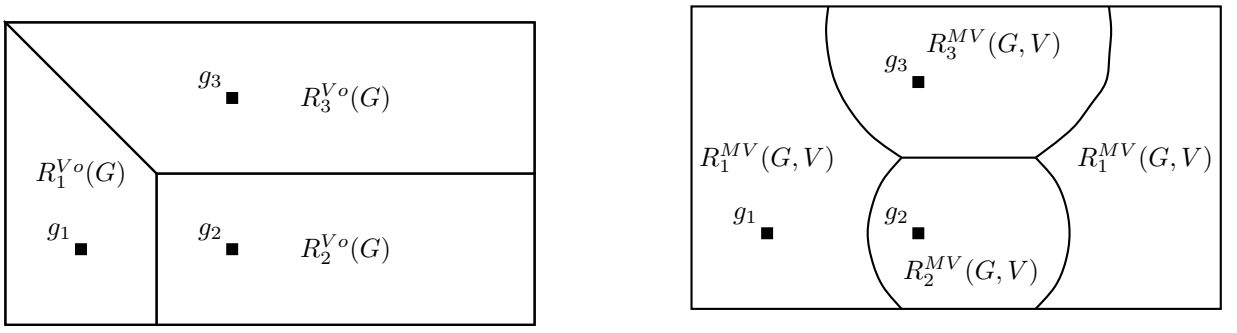
(d) MWVDP with $V = \{0.4; 1; 1\}$

Figure 5.1: Different (generalized) Voronoi districting plans

basic area 8 is also assigned to this district. Figure 5.1b illustrates the resulting MWVDP. Here, the spatial extension of $D_3^{MV}(G, V)$ is noticeably smaller than the spatial extension of $D_3^{Vo}(G)$ depicted in Figure 5.1a.

Therefore, in order to obtain a balanced solution, the weight of a generator has to be increased if the size of its district is too large and it has to be decrease it if this size is too small.

Example 5.1.1 (cont.) Choosing the weights $v(g_1) = 1$, $v(g_2) = 1.5$, and $v(g_3) = 2$ results in the MWVDP presented in Figure 5.1c. Here, the district sizes are $w(D_1^{WV}(G, V)) = 16$, $w(D_2^{WV}(G, V)) = 17$ and $w(D_3^{WV}(G, V)) = 17$. Hence, the maximum percentage deviation from the average district size is only 4% now.



(a) Voronoi Diagram

(b) Multiplicatively weighted Voronoi Diagram with $V = \{1; 2; 2\}$

Figure 5.2: Connectedness of (generalized) Voronoi Diagrams

However, multiplicatively weighted Voronoi Diagrams have one main drawback. The obtained regions are not necessarily connected. Figure 5.2a depcits a Voronoi Diagram for three generators. Assume $v(g_1) = 1$, $v(g_2) = 2$, and $v(g_3) = 2$. Figure 5.2b illustrates the resulting multiplicatively weighted Voronoi Diagram. In this case, the Voronoi region corresponding to generator v_1 is not connected. Thus, this approach may generate a MWVDP that is very poor in terms of contiguity.

Example 5.1.1 (cont.) Let $v(g_1) = 0.4$, $v(g_2) = 1$, and $v(g_3) = 1$. Figure 5.1d depicts the resulting MWVDP. In this case, basic area 9 is assigned to the district of generator g_1. The distances from 9 to the generators are $d(9, g_1) = 4.27$, $d(9, g_2) = 2.06$ and $d(9, g_3) = 2.69$. Thus, $v(g_1) \cdot d(9, g_1) = 1.71$ is smaller than $v(g_2) \cdot d(9, g_2) = 2.06$ and $v(g_3) \cdot d(9, g_3) = 2.69$. Note that $D_2^{MV}(G, V)$ contains only one basic area, namely basic area 6. Obviously, basic area 6 is located inside (the convex hull of) $D_1^{MV}(G, V)$. Hence, according to the criterion of contiguity, this result is not sufficient.

Additively weighted Voronoi Diagrams overcome this drawback. Here, negative weights of the generators are also possible, i.e., $v(g_h) \in \mathbb{R}\ \forall 1 \leq h \leq p$.

Definition 5.1.4 *For a set of basic areas BA, an* additively weighted Voronoi districting plan (AWVDP) $S^{AV}(G, V) := \{D_1^{AV}(G, V); \ldots; D_p^{AV}(G, V)\}$ *with respect to G and V is defined by*

$$D_h^{AV}(G, V) := \{i \in BA \mid d(b_i, g_h) + v(g_h) < d(b_i, g_j) + v(g_j)\ \forall 1 \leq j, h \leq p,\ j \neq h\}\ ,$$

where $D_h^{AV}(G, V)$ is called additively weighted Voronoi district (AWVD).

Unfortunately, also an AWVD can be empty. Assume the condition $v_h > v_j + d(g_h, g_j)$ holds. This implies $v_j + d(b_i, g_j) < v_h + d(b_i, g_h)\ \forall i \in BA$, i.e., no basic area is assigned to the district of generator g_h.

Additively weighted Voronoi approaches in the context of districting have the aim to find an AWVDP that is feasible in terms of balance and minimizes the additively weighted sum of distances from the basic areas to the corresponding generators. In order to find a feasible solution, analogously to the multiplicatively case, additively weighted Voronoi approaches increase (decrease) the weight of a generator if the size of the corresponding district is too large (small).

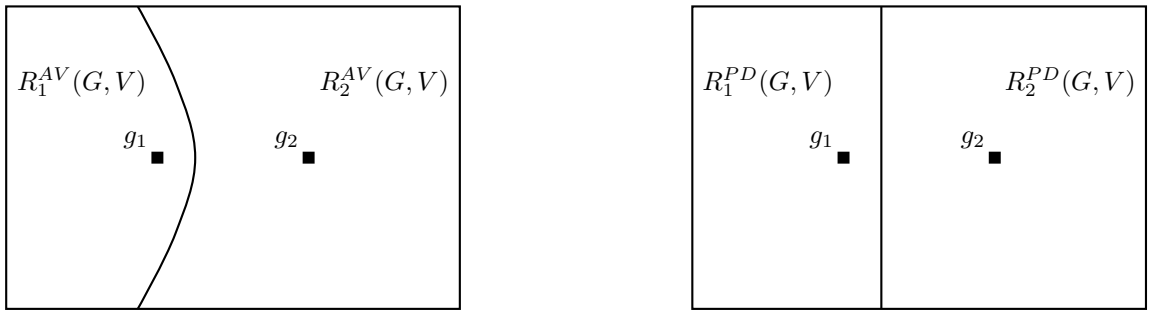
Example 5.1.1 (cont.) For choosing $v(g_1) = 0$, $v(g_2) = 1$, and $v(g_3) = 1.5$, the resulting AWVDP is also the one illustrated in Figure 5.1c.

Fortunately, for additively weighted Voronoi diagrams the obtained regions are guaranteed to be connected as long as Euclidean distances are used. The following lemma and the related proof are based on Sharir [12].

Lemma 5.1.1 *For Euclidean distances and given generators G and weights V, each region defined by $R_h^{AV}(G,V) := \{x \in \mathbb{R}^2 \mid d(x, g_h) + v(g_h) < d(x, g_j) + v(g_j)\ \forall j \neq h\}$ is connected.*

Proof
Let $x_1 \in R_h^{AV}(G,V)$, and, hence, $x_1 \notin R_j^{AV}(G,V)\ \forall j \neq h$. Choose an arbitrary point x_2 located on the segment $\overline{g_h, x_1}$.
Assume that $x_2 \notin R_h^{AV}(G,V)$: Hence, it has to exist another region containing x_2, i.e., $x_2 \in R_j^{AV}(G,V)$, $j \neq h$.
According to the triangle inequality $d(x_1, g_j)+w(g_j) < d(x_1, x_2)+d(x_2, g_j)+v(g_j)$ holds. Since $x_2 \in R_j^{AV}(G,V)$ and $x_2 \notin R_h^{AV}(G,V)$ holds, the inequality $d(x_2, g_j)+v(g_j) < d(x_2, g_h)+v(g_h)$ holds. Furthermore, since x_2 is located on the segment $\overline{g_h x_1}$, $d(x_1, x_2) + d(x_2, g_h)$ equals $d(x_1, g_h)$. Take these considerations together:

$$\begin{aligned} d(x_1, g_j) + v(g_j) &\leq d(x_1, x_2) + d(x_2, g_j) + v(g_j) \\ &< d(x_1, x_2) + d(x_2, g_h) + v(g_h) \\ &= d(x_1, g_h) + v(g_h) \end{aligned}$$



(a) Additively weighted Voronoi Diagram (b) Power Diagram

Figure 5.3: Convexity of generalized Voronoi Diagrams

This means that $x_1 \in V_j^{AV}(G,V)$ contradicting the assumption.
Thus, each point on a segment between the generator and an arbitrary point of its region has to be in its region, too. Thus, the region is connected. □

Unfortunately, $V_h^{AV}(G,V)$ is not necessarily convex. For example, Figure 5.3a depicts two generators having the weights $v(g_1) = 1$ and $v(g_2) = 0$. Obviously, $V_2^{AV}(G,V)$ is not convex.

Concerning the districting problem, this observation implies that an achieved solution is not necessarily contiguous according to the definition introduced in Section 2.2.4. Nevertheless, the computational results in Section 5.5.8 will show that the solutions are quite acceptable in terms of contiguity. In order to ensure a contiguous solution, i.e., $ctg(S) = 0$, a variation of additively weighted Voronoi Diagrams can be used, the so-called *Power Diagrams*.

Definition 5.1.5 *For a set of basic areas BA, a* Power Diagram districting plan (PDDP) *$S^{PD}(G,V) := \{D_1^{PD}(G,V); \ldots; D_p^{PD}(G,V)\}$ with respect to the generators G and V is defined by*

$$D_h^{PD}(G,V) := \left\{ i \in BA \;\middle|\; d^2(b_i,g_h) + v(g_h) < d^2(b_i,g_j) + v(g_j)\ \forall 1 \leq j,h \leq p,\ j \neq h \right\},$$

where $D_h^{PD}(G,V)$ is called Power Diagram district (PDD).

The main difference to the approach before is the usage of squared distances instead of single distances. In the context of districting, a Power Diagram approach has the goal to find the feasible solution minimizing the sum of squared distances from the basic areas to the corresponding generators. This goal can be interpreted as minimizing the (Weighted) Moment of Inertia (cf. Section 3.3.5.1) with predefined centers. There is a pleasant side-effect

for Euclidean distances: If the number of basic areas is equal for every district, minimizing the sum of squared distances from the basic areas to the corresponding generators is equivalent to minimizing the sum of squared pairwise distances between the basic areas of the same district. The following lemmata and the related proofs follow Fryer Jr. and Holden [3].

Lemma 5.1.2 *Consider Euclidean distances: If* $|D_h| = \frac{|BA|}{p}$ $\forall 1 \leq h \leq p$ *holds, minimizing* $ev_{spd}(S)$ *is equivalent to minimizing* $ev_{moi}(S)$*, with*

$$ev_{spd}(S) := \sum_{D_h \in S} \sum_{i \in D_h} \sum_{j \in D_h} d_{i,j}^2 \qquad \textit{and} \qquad ev_{moi}(S) := \sum_{D_h \in S} \sum_{i \in D_h} d^2(b_i, c_h)\,,$$

where $c_h := (c_{hx}, c_{hy}) := \left(\frac{\sum_{i \in D_h} x_i}{|D_h|}, \frac{\sum_{i \in D_h} y_i}{|D_h|} \right)$.

Proof

The proof shows that

$$\sum_{D_h \in S} \sum_{i \in D_h} \sum_{k \in D_h} d_{i,k}^2 = 2 \cdot \frac{|BA|}{p} \cdot \sum_{D_h \in S} \sum_{i \in D_h} d^2(b_i, c_h)$$

holds. Since $2 \cdot \frac{|BA|}{p}$ is a positive constant factor, minimizing $ev_{spd}(S)$ is equivalent to minimizing $ev_{moi}(S)$.

$$\begin{aligned}
&\sum_{D_h \in S} \sum_{i \in D_h} \sum_{j \in D_h} d_{i,j}^2 \\
&= \sum_{D_h \in S} \sum_{i \in D_h} \sum_{j \in D_h} \left(x_i^2 - 2 \cdot x_i \cdot x_j + x_j^2 + y_i^2 - 2 \cdot y_i \cdot y_j + y_j^2 \right) \\
&= \sum_{D_h \in S} \left[\textstyle\sum_{i \in D_h} \sum_{j \in D_h} \left(x_i^2 + y_i^2 \right) + \sum_{i \in D_h} \sum_{j \in D_h} \left(x_j^2 + y_j^2 \right) - \sum_{i \in D_h} \sum_{j \in D_h} (2 \cdot x_i \cdot x_j) - \sum_{i \in D_h} \sum_{j \in D_h} (2 \cdot y_i \cdot y_j) \right] \\
&= \sum_{D_h \in S} \left[|D_h| \cdot \textstyle\sum_{i \in D_h} \left(x_i^2 + y_i^2 \right) + |D_h| \cdot \sum_{j \in D_h} (x_j^2 + y_j^2) - \sum_{i \in D_h} (2 \cdot x_i \cdot \sum_{j \in D_h} x_j) - \sum_{i \in D_h} (2 \cdot y_i \cdot \sum_{j \in D_h} y_j) \right] \\
&= \sum_{D_h \in S} \left[2 \cdot |D_h| \cdot \sum_{i \in D_h} (x_i^2 + y_i^2) + \sum_{i \in D_h} (2 \cdot x_i \cdot c_{hx} \cdot |D_h|) - \sum_{i \in D_h} (2 \cdot y_i \cdot c_{hy} \cdot |D_h|) \right] \\
&= \sum_{D_h \in S} \left[2 \cdot |D_h| \cdot \sum_{i \in D_h} \left(x_i^2 + y_i^2 \right) + 2 \cdot |D_h| \cdot \sum_{i \in D_h} (x_i \cdot c_{hx}) - 2 \cdot |D_h| \cdot \sum_{i \in D_h} (y_i \cdot c_{hy}) \right] \\
&= \sum_{D_h \in S} \left[2 \cdot |D_h| \cdot \sum_{i \in D_h} \left(x_i^2 + y_i^2 - x_i \cdot c_{hx} - y_i \cdot c_{hy} \right) \right]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{D_h \in S} \left[2 \cdot |D_h| \cdot \sum_{i \in D_h} \left(x_i^2 + y_i^2 - 2 \cdot x_i \cdot c_{hx} - 2 \cdot y_i \cdot c_{hy} + x_i \cdot c_{hx} + y_i \cdot c_{hy} \right) \right] \\
&= \sum_{D_h \in S} \left[2 \cdot |D_h| \cdot \sum_{i \in D_h} (x_i^2 + y_i^2 - 2 \cdot x_i \cdot c_{hx} - 2 \cdot y_i \cdot c_{hy}) + 2 \cdot c_{hx} \cdot |D_h| \cdot \sum_{i \in D_h} (x_i) + 2 \cdot c_{hy} \cdot |D_h| \cdot \sum_{i \in D_h} (y_i) \right] \\
&= \sum_{D_h \in S} \left[2 \cdot |D_h| \cdot \sum_{i \in D_h} (x_i^2 + y_i^2 - 2 \cdot x_i \cdot c_{hx} - 2 \cdot y_i \cdot c_{hy}) + 2 \cdot c_{hx}^2 \cdot |D_h|^2 + 2 \cdot c_{hy}^2 \cdot |D_h|^2 \right] \\
&= \sum_{D_h \in S} \left[2 \cdot |D_h| \cdot \sum_{i \in D_h} \left(x_i^2 + y_i^2 - 2 \cdot x_i \cdot c_{hx} - 2 \cdot y_i \cdot c_{hy} \right) + 2 \cdot |D_h| \cdot \sum_{i \in D_h} \left(c_{hx}^2 + c_{hy}^2 \right) \right] \\
&= \sum_{D_h \in S} \left[2 \cdot |D_h| \cdot \sum_{i \in D_h} \left(x_i^2 + y_i^2 - 2 \cdot x_i \cdot c_{hx} - 2 \cdot y_i \cdot c_{hy} + c_{hx}^2 + c_{hy}^2 \right) \right] \\
&= \sum_{D_h \in S} \left[2 \cdot |D_h| \cdot \sum_{i \in D_h} d^2(b_i, c_h) \right] \\
&= 2 \cdot \frac{|BA|}{p} \cdot \sum_{D_h \in S} \sum_{i \in D_h} d^2(b_i, c_h)
\end{aligned}
$$

□

Note that $|D_h| = \frac{|BA|}{p}$ is a necessary precondition, i.e., if this condition is not satisfied, the minimization of the functions $ev_{spd}(S)$ and $ev_{moi}(S)$ are not equivalent.

Example 5.1.2 Assume four basic areas represented by the points $b_1 = (0, 0)$, $b_2 = (0, 1)$, $b_3 = (0, 2)$ and $b_4 = (1, 1)$. For $p = 2$ there are seven possible solutions.

	D_1	D_2	ev_{moi}	ev_{spd}
S_1	$\{1\}$	$\{2;3;4\}$	1.33	8
S_2	$\{2\}$	$\{1;3;4\}$	2.67	16
S_3	$\{3\}$	$\{1;2;4\}$	1.33	8
S_4	$\{4\}$	$\{1;2;3\}$	2.00	12
S_5	$\{1;2\}$	$\{3;4\}$	1.50	6
S_6	$\{1;3\}$	$\{2;4\}$	2.50	10
S_7	$\{1;4\}$	$\{2;3\}$	1.50	6

Table 5.2: Solutions of Example 5.1.2

Table 5.2 states these solutions and their evaluations in terms of $ev_{moi}(S)$ and $ev_{spd}(S)$. Obviously, there are two optimal solutions in terms of $ev_{spd}(S)$, namely $S_5 = \{\{1;2\};\{3;4\}\}$ and $S_7 = \{\{1;4\};\{2;3\}\}$ resulting in an evaluation value of 6. However, the optimal solutions

in terms of $ev_{moi}(S)$ are $S_1 = \{\{1\};\{2;3;4\}\}$ and $S_3 = \{\{3\};\{1;2;4\}\}$. The corresponding evaluation value is 1.33.
If the set of feasible solutions is restricted to those satisfying $|D_1| = |D_2| = 2$, the set of feasible solutions contains only the solutions S_5, S_6, and S_7. In this case, the optimal solutions in terms of $ev_{moi}(S)$ are S_5 and S_7. Hence, for requiring this restriction, the optimal solutions in terms of $ev_{spd}(S)$ and $ev_{moi}(S)$ are equal, as has been proven above.

However, if a solution is balanced the hope is that the number of basic areas per district is approximately equal too. Then, the solution obtained by a Power Diagram approach is also well in terms of the Squared Pairwise Distances, and, hence, very likely also in terms of the Pairwise Distances described in Section 3.3.5.2.
Finally, the following lemmata addresses Power Diagrams in terms of connectivity and convexity. Let $R_h^{PD}(G,V) := \{x \in \mathbb{R}^2 : d^2(x,g_h) + w(g_h) < d^2(x,g_j) + w(g_j)\ \forall j \neq h\}$ be the Power Diagram region of generator g_h. For Euclidean distances the Power Diagram regions of two generators are divided by a line, and, hence, each Power Diagram region is convex. This implies that a PDDP is always contiguous.

Lemma 5.1.3 *Consider Euclidean distances: Two neighbored Power Diagram regions are separated by a line.*

Proof
Let $g_1 := (g_{1x}, g_{1y})$ and $g_2 := (g_{2x}, g_{2y})$ be the generators of two neighbored regions. A point $p := (x,y)$ is located on the border between the regions of g_1 and g_2 if and only if

$$d^2(p,g_1) + w(g_1) = d^2(p,g_2) + w(g_2) \tag{5.1}$$

holds. Thus, in the next step, the proof validates that Equation (5.1) induces a line:

$$\begin{aligned}
& d^2(p,g_1) + v(g_1) = d^2(p,g_2) + v(g_2) \\
\Leftrightarrow\ & (x-g_{1x})^2 + (y-g_{1y})^2 + v(g_1) = (x-g_{2x})^2 + (y-g_{2y})^2 + v(g_2) \qquad (5.2)\\
\Leftrightarrow\ & -2\cdot x\cdot g_{1x} + g_{1x}^2 - 2\cdot y\cdot g_{1y} + g_{1y}^2 + v(g_1) \\
& = -2\cdot x\cdot g_{2x} + g_{2x}^2 - 2\cdot y\cdot g_{2y} + g_{2y}^2 + v(g_2) \\
\Leftrightarrow\ & y\cdot(2\cdot g_{2y} - 2\cdot g_{1y}) = x\cdot(2\cdot g_{1x} - 2\cdot g_{2x}) - g_{1x}^2 - g_{1y}^2 - v(g_1) + g_{2x}^2 + g_{2y}^2 + v(g_2) \\
\Leftrightarrow\ & y = x\cdot\left(\frac{g_{1x}-g_{2x}}{g_{2y}-g_{1y}}\right) + \frac{-g_{1x}^2 - g_{1y}^2 - v(g_1) + g_{2x}^2 + g_{2\cdot y}^2 + v(g_2)}{2\cdot g_{2y} - 2g_{1y}} \qquad (5.3)
\end{aligned}$$

Equation (5.3) defines a line since the two fractions are constants. For the sake of completeness regard the case where $g_{2y} = g_{1y}$ holds, and, hence, $(y - g_{2y})^2 = (y - g_{1y})^2$ holds:

$$\begin{aligned}(5.2) \Leftrightarrow & -2 \cdot x \cdot g_{1x} + 2 \cdot x \cdot g_{2x} = -g_{1x}^2 - v(g_1) + g_{2x}^2 + g_{2y}^2 + v(g_2) \\ \Leftrightarrow & \, x = \frac{-g_{1x}^2 - v(g_1) + g_{2x}^2 + g_{2y}^2 + v(g_2)}{2 \cdot g_{2x} - 2 \cdot g_{1x}}, \end{aligned} \tag{5.4}$$

i.e., the regions are divided by a vertical line, defined as in Equation (5.4). □

Lemma 5.1.4 *If Euclidean distances are used, for given G and V each Power Diagram region is convex.*

Proof
Let $x_1, x_2 \in R_h^{PD}(G,V)$ and $l(h,j)$ be the line separating $R_h^{PD}(G,V)$ from $R_j^{PD}(G,V)$ for an arbitrary $j \neq h$. The assumption $x_1, x_2 \in R_h^{PD}(G,V)$ implies that x_1 and x_2 are not located on $l(h,j)$. Thus, the segment $\overline{x_1, x_2}$ is not a pitch line of $l(h,j)$. Hence, there is at most one point of intersection between $l(h,j)$ and $\overline{x_1, x_2}$. However, if there exists a point of intersection, x_1 and x_2 have to be located on different sides of $l(h,j)$ contradicting that both points are elements of $R_h^{PD}(G,V)$. Hence, there is no point of intersection, and this implies that the whole segment $\overline{x_1, x_2}$ is located on the same side of $l(h,j)$, i.e., for all points x^* of $\overline{x_1, x_2}$ the inequality $d^2(x^*, g_h) + v(g_h) < d^2(x^*, g_j) + v(g_j)$ holds. This argumentation is valid for every $j \neq h$. This implies $x^* \in R_h^{PD}(G,V)$, and, hence, $R_h^{PD}(G,V)$ is convex. □

Example 5.1.3 Recall Figure 5.3b. It depicts two generators g_1 and g_2. The corresponding weights are $v(g_1) = 1$ and $v(g_2) = 0$. Obviously, the Power Diagram regions $R_1^{PD}(G,V)$ and $R_2^{PD}(G,V)$ are divided by a (vertical) line and both regions are convex.

Hess et al. [5] apply the Weighted Moment of Inertia (cf. Section 3.3.5.1) in order to evaluate a districting plan in terms of compactness. Therefore, the following definition introduces an adapted version of Power Diagrams considering the activities of the basic areas:

Definition 5.1.6 *For a set of basic areas BA, a* weighted Power Diagram districting plan (WPDDP) *$S^{WPD}(G,V) := \{D_1^{WPD}(G,V); \ldots; D_p^{WPD}(G,V)\}$ with respect to G and V is defined by*

$$D_h^{WPD}(G,V) := \left\{ i \in BA \;\middle|\; w_i \cdot d^2(b_i, g_h) + w(g_h) < w_i \cdot d^2(b_i, g_j) + w(g_j) \;\forall 1 \leq j, h \leq p,\; j \neq h \right\},$$

where $D_h^{WPD}(G,V)$ is called weighted Power Diagram district (WPDD).

Unfortunately, by including the activities, the described properties in terms of convexity, connectivity and contiguity are not valid anymore.

Example 5.1.4 Assume a small example with two generators and three basic areas, all located on one line. Let the generators $g_1 = (0, 0)$ and $g_2 = (10, 0)$ with $v(g_1) = 0$ and $v(g_2) = 150$ and the basic areas $b_1 = (4, 0)$, $b_2 = (6, 0)$, and $b_3 = (8, 0)$ with $w_1 = 1$, $w_2 = 10$, and $w_3 = 2$ be given.
Since $1 \cdot 4^2 + 0 = 16 < 1 \cdot 6^2 + 150 = 186$ and $2 \cdot 8^2 + 0 = 128 < 2 \cdot 2^2 + 150 = 158$ holds, b_1 and b_3 are assigned to the district of g_1. However, b_2 is assigned to the district of g_2 since $10 \cdot 6^2 + 0 = 360 > 10 \cdot 4^2 + 150 = 310$ holds. Hence, the obtained WPDDP is not contiguous.

Since the various definitions introduced in this section are all based on distances between the generators and the basic areas, the usage of street distances or travel times is straightforward. However, in this case, the described properties in terms of connectivity or contiguity are not necessarily valid any more.

5.2 Literature Review

To the best of our knowledge, there are only a few approaches using Voronoi Diagrams in the context of districting.

Galvão et al. [4] address a parcel delivery problem and propose a multiplicatively weighted Voronoi approach. They assume an existing delivery pattern and define for each district the center of gravity as a generator. The obtained set of generators is fixed during the whole procedure, while the corresponding weights are updated in each iteration. Their update rule includes for each district its total cargo, its total working time including the travel time from the depot to this district, and the approximated travel time within this district.
Novaes et al. [9] continue this work while studying the applicability of (generalized) Voronoi approaches on location-districting problems. They propose a Power Diagram approach and integrate obstacles by re-defining the distance between a point and a generator. In their approach, this distance is the shortest distance not traversing these obstacles.

Ricca et al. [11] apply multiplicatively weighted Voronoi Diagrams in order to solve political districting problems. In a first step, they locate the generators and fix them. In each further step, they update the distances between the generators and the basic areas depending on the population within the corresponding districts. Moreover, by means of a neighborhood graph they ensure connectedness.

Moreno-Regidor et al. [8] solve districting problems by applying an additively weighted Voronoi approach. They also assume prescribed generators. Each iteration updates the weights of the generators considering the current sizes of the districts. However, they conclude that the locations of the generators have a great impact on the resulting solution. In contrast to other approaches, for each generator the required size of the corresponding district can be defined separately, i.e., the districts are actually not balanced.

Finally, Fryer Jr. and Holden [3] focus on the problem of measuring compactness. They propose a relative measure that determines the ratio between the compactness of a solution and the optimal compactness for the same set of basic areas, where compactness is evaluated in terms of Squared Pairwise Distances. Since it is NP-hard to derive the optimal compactness, they propose an approximation approach based on Power Diagrams. This approach determines the initial set of generators depending on a current solution, but in contrast to other approaches it does not fix their locations. Therefore, this approach is a two-stage iterative procedure where one stage relocates the generators and the other stage updates the weights of the generators.

5.3 The Algorithm Framework

After giving some basic definitions and reviewing the literature, this section presents our algorithm framework based on generalized Voronoi Diagrams. The underlying model is already presented in Section 4.2. Like the approach of Fryer Jr. and Holden [3] this approach is a two stage iterative procedure. Since the quality of a districting plan based on (generalized) Voronoi Diagrams highly depends on the generators' locations, this algorithm does not fix these locations, but updates them during each main-iteration. Moreover, each main-iteration executes an iterative sub-process in order to determine a feasible solution. Therefore, each sub-iteration updates the weights of the generators followed by the computation of a new solution. This sub-process stops if the computed solution is feasible, i.e., the balance is smaller than or equal to a given threshold μ. Since there is no guarantee to find a feasible solution at all, a user-given parameter it_{max}^{sub} limits the number of executed sub-iterations. After determining a new feasible solution, the next main-iteration determines new generators based on this solution. The algorithm stops, when one of the following conditions holds:

- There is no improvement in terms of an evaluation function.
- There is no feasible solution after the execution of a main-iteration.
- A maximum number of executed main-iterations it_{max}^{main} is reached.

Algorithm 5.3.1 outlines the general framework.

Algorithm 5.3.1: Algorithm Framework for Voronoi Based Districting Approaches

Input: Set of basic areas BA, number of districts p, parameters it_{max}^{sub}, it_{max}^{main}.
Output: Districting plan S.

1 Initialize G, S^{cur}, S^{best} and set $it_{count}^{main} := 0$.
2 **repeat**
3 | Set $S^{last} := S^{cur}$, $V := \{0; \ldots; 0\}$, $it_{count}^{sub} := 0$.
4 | Determine S^{cur} depending on G and V.
5 | **while** $\left[it_{count}^{sub} < it_{max}^{sub}\right]$ *AND* $[S^{cur}$ *is not feasible*$]$ **do**
6 | | Update V depending on S^{cur} and G.
7 | | Determine S^{cur} depending on G and V.
8 | | Set $it_{count}^{sub} = it_{count}^{sub} + 1$.
| **end**
9 | Update G depending on S^{cur}.
10 | **if** $\left[ev(S^{cur}) < ev(S^{best})\right]$ **then** set $S^{best} := S^{cur}$.
11 | Set $it_{count}^{main} = it_{count}^{main} + 1$.
until $[S^{cur}$ *is not feasible*$]$ *OR* $\left[ev(S^{cur}) \geq ev(S^{last})\right]$ *OR* $\left[it_{count}^{main} = it_{max}^{main}\right]$
12 **return** S^{best}.

This outline leaves some questions open:

- How to compute an initial set of generators?
- How to evaluate a solution?
- How to determine a new set of generators based on a current solution?
- How to update the weights of the generators iteratively?

The subsequent sections address these questions in detail.

5.3.1 Initial Set of Generators

In contrast to Fryer Jr. and Holden [3] our approach assumes that no existing districting plan is available. Therefore, it has to define or determine, respectively, an initial set of generators or an initial solution, respectively.

One option is the Recursive Partitioning Algorithm (RPA) introduced in Section 4. The compactness measure used during the execution of the RPA is the Moment of Inertia with a corresponding weight of 1, i.e., only compactness is used in order to evaluate a bisecting partition. Moreover, the RPA uses exclusively line partitions in order to determine the set of bisecting partitions since the obtained solutions are contiguous in this case. As number of search directions $K = 8$ or $K = 16$ is recommendable. There are some reasons for doing so. At first, this approach is very fast. Furthermore, the achieved solutions are feasible in (almost) every case - during our tests no infeasible solution occurred. Finally, the determined set of initial generators based on this solution has a good spatial distribution. For using line partitions, the RPA requires $\mathcal{O}(|BA| \cdot \log |BA|)$ time (cf. Section 4.3.7.3). Moreover, since $K = 8$ or $K = 16$, assume that $\log K << |B| \cdot \log |B|$. Thus initializing G, S^{cur} and S^{best} requires $\mathcal{O}(p \cdot K \cdot |BA| \cdot \log |BA|)$ time. Algorithm 5.3.2 summarizes the described approach. The tests presented in Section 5.5.4 will compare different kinds of initial solutions and will confirm the suitability of this approach.

Algorithm 5.3.2: Initialize G, S^{cur} and S^{best} by the RPA

Input: Set of basic areas BA, number of districts p.
Output: G, S^{cur}, S^{best}.

1 Determine S^{cur} by applying the RPA.
2 Set $S^{best} := S^{cur}$.
3 Determine G depending on S^{cur}.
4 **return** G, S^{cur}, S^{best}.

Of course, there are also other approaches to generate an initial set of generators. Algorithm 5.3.3 outlines the general process for generating this set.

Algorithm 5.3.3: General Process for Initializing G, S^{cur} and S^{best}

Input: Set of basic areas BA, number of districts p, approach to determine the initial set of generators DG.
Output: G, S^{cur}, S^{best}.

1 Determine G by applying DG and set $V := \{0; \ldots; 0\}$.
2 Determine S^{cur} depending on G and V.
3 **if** [S^{cur} *is feasible*] **then** set $S^{best} := S^{cur}$
else set $S^{best} := NULL$
4 **return** G, S^{cur}, S^{best}.

A straightforward approach chooses these generators randomly. In order to obtain an initial solution each basic area is assigned to its closest generator. Hence, the total process to initialize G, S^{cur} and S^{best} requires $\mathcal{O}(p \cdot |BA|)$ time.
A more sophisticated approach is based on the *k-Means++* clustering algorithm [6]. At first, it chooses one basic area randomly as first generator. Then, it determines the other generators successively as follows: Each iteration computes for each basic area the distance to its closest existing generator. Afterwards, it defines the probability that this basic area is chosen as next generator proportional to the square of this distance. Hence, a good spatial distribution of the generators is expected as well. This approach also needs $\mathcal{O}(p \cdot |BA|)$ time. Section 5.5.4 will show that this approach also delivers good results. Unfortunately, this approach does not necessarily result in a feasible initial solution in terms of balance. In this case, according Algorithm 5.3.3 Line 3 our approach sets $S^{best} := NULL$ and defines $ev(NULL) := \infty$. Hence, in this case also Algorithm 5.3.1 does not necessarily determine a feasible solution at all. This is another reason for preferring the RPA to initialize the generators.

5.3.2 Evaluating a Solution

This subsection addresses the evaluation of a solution. As described above, for the assignment of basic areas to generators classical additively weighted Voronoi Diagrams use (Euclidean) single distances, whereas Power Diagrams use squared (Euclidean) distances. Thus, for additively weighted Voronoi Diagrams the function

$$ev_{AW}(S) := \sum_{h=1}^{p} \sum_{i \in D_h} d(b_i, g_h)\,, \tag{5.5}$$

for Power Diagrams the function

$$ev_{PD}(S) := \sum_{h=1}^{p} \sum_{i \in D_h} d^2(b_i, g_h)\,, \tag{5.6}$$

and for weighted Power Diagrams the function

$$ev_{WPD}(S) := \sum_{h=1}^{p} \sum_{i \in D_h} w_i \cdot d^2(b_i, g_h) \tag{5.7}$$

evaluates a current solution S. Recall that the evaluation function for (weighted) Power Diagrams corresponds to the (Weighted) Moment of Inertia. Thus, strongly spoken, the solutions are only evaluated and compared in terms of compactness. Note that the usage of network distances as distance function $d(\cdot, \cdot)$ is possible. Section 5.5.10 will present some results for using distances and travel times on a road network.

In the following, *additively weighted Voronoi Districting Approach*, or AWVDA for short, denotes Algorithm 5.3.1 if it uses the evaluation function introduced in Equation (5.5). (*Weighted*) *Power Diagram Districting Approach*, or (W)PDDA for short, denotes the algorithm if it uses the functions defined in Equation (5.6) or (5.7), respectively.

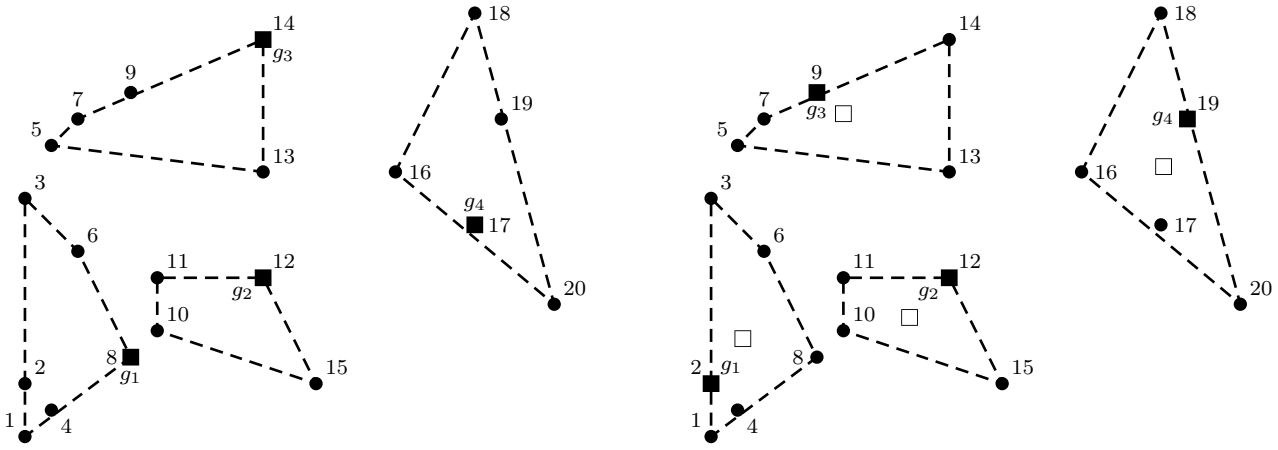
The complexity of evaluating a solution in terms of one of these functions is $\mathcal{O}(p \cdot |BA|)$ since the closest generator has to be found for each basic area.

5.3.3 Updating the Generators' Locations

This subsection explains how to determine a new set of generators based on a current solution S is determined, or more precisely, how a new generator g_h for a district D_h of S is determined. The approach restricts the set of candidates to the set of basic areas of D_h, i.e., it selects the location of an existing basic area as generator. Hence, the usage of network distances is possible as well. Since the applied evaluation function should be minimized by the selection of a new generator, the generator's determination depends on this function.

- **AWVDA**: Equation (5.5) states the considered evaluation function. The new location of g_h is the location b_i of the basic area $i := \arg\min_{j \in D_h} \sum_{k \in D_h} d_{j,k}$. In order to determine this generator, the distances between all pairs of basic areas of D_h are considered, so the computation for one district D_h needs $\mathcal{O}(|D_h|^2)$ time.

- **PPDA**: Equation (5.6) states the considered evaluation function. The new location of generator g_h is the location b_i of the basic area $i := \arg\min_{j \in D_h} \sum_{k \in D_h} d^2_{j,k}$. For



(a) Solution S^{cur}

(b) Updated set of generators G

Figure 5.4: Updating the generators' locations

Euclidean distances the computation for one district D_h needs $\mathcal{O}(|D_h|)$ time since i is the closest basic area to the unweighted center of gravity of D_h. In contrast to this, for network distances the computation needs $\mathcal{O}(|D_h|^2)$ time.

- **WPPDA**: Equation (5.7) states the considered evaluation function. The new location of generator g_h is the location b_i of the basic area $i := \arg\min_{j \in D_h} \sum_{k \in D_h} w_j \cdot d_{j,k}^2$. Again, for Euclidean distances this computation needs $\mathcal{O}(|D_h|)$ time, whereas for network distances it needs $\mathcal{O}(|D_h|^2)$ time for each district.

Example 5.3.1 Let the set of basic areas defined in Table 5.3 be given.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x_i	0	0	0	1	1	2	2	4	4	5	5	9	9	9	11	14	17	17	18	20
y_i	3	5	12	4	14	10	15	6	16	7	9	9	13	18	5	13	11	19	15	8

Table 5.3: Set of basic areas BA

Moreover, let the locations of the generators $g_1, \ldots, g_4$ correspond to basic areas 8, 12, 14, and 17. Setting $v(g_1) = 30$, $v(g_2) = 5$, $v(g_3) = -15$, and $v(g_4) = -20$ leads to the PDDP $S^{PD}(G, V) := \{\{1; 2; 3; 4; 6; 8\}; \{10; 11; 12; 15\}; \{5; 7; 9; 13; 14\}; \{16; 17; 18; 19; 20\}\} := S^{cur}$ depicted in Figure 5.4a.
For $D_1^{PD}(G, V)$ the center of gravity is $(1.2, 6.7)$ and the closest basic area to this point is basic area 2. Hence, the new generator of this district is basic area 2. For the further

districts the centers of gravity are $(7.5, 7.5)$, $(5, 15.2)$, and $(17.2, 13.2)$. Thus, the updated generators are the basic areas 12, 9 and 19. Figure 5.4b illustrates for each district its center of gravity as white square and its new generator as black square.

5.3.4 Updating the Generators' Weights

The last subsection addresses one of the main challenges of districting approaches based on (generalized) Voronoi Diagrams, the update function for the weights of the generators. If the update steps are too small, the changes between the solutions of two consecutive sub-iterations are very small, even the case of no changes can occur. Hence, the number of executed sub-iterations until a feasible solution is found would be quite large. If the update steps are too large, for one generator the size of its district can change from too large to too small in the succeeding solution or the other way around. This effect can result in a deterioration of the balance or in oscillating solutions. Let $S^t := \{D_1^t, \ldots, D_p^t\}$ be the current solution and $v^t(g_1); \ldots; v^t(g_p)$ the current weights of the generators in sub-iteration t. Furthermore, the current absolute (balance) error of a district is defined by

$$aer_h^t := w(D_h^t) - \mu\,,$$

where μ is the average district size. The current relative (balance) error is defined by

$$rer_h^t := \frac{aer_h^t}{\mu}\,.$$

In the following, three update rules are introduced. Rule 1 and 2 have already proposed in the literature, this work introduces rule 3 additionally.

5.3.4.1 Update Rule 1

Moreno-Regidor et al. [8] use additively weighted Voronoi Diagrams for their districting approach. They take the current relative errors, the distances between the generators and a dynamic convergence parameter into account in order to update the generators' weights in each iteration. Moreover, they use a dynamic convergence parameter CP^t, because they conclude that it is impossible to find a universal convergence parameter that is applicable to all instances. They propose the following update rule:

$$v^{t+1}(g_h) := v^t(g_h) + CP^t \cdot \sum_{j \neq h} \frac{rer_h^t - rer_j^t}{d(g_h, g_j)}\,,$$

where

$$CP^{t+1} := \min_{h=1,\dots,p} \left| \frac{\min\limits_{i,k \in D_h, i \neq k} d_{i,k}}{\sum\limits_{j \neq h} \frac{rer_j^t - rer_h^t}{d(g_h, g_j)}} \right| .$$

Since the proposed function is based on additively weighted Voronoi Diagrams, an adaptation in order to make it usable for (weighted) Power Diagrams is necessary:

- Power Diagrams: $v^{t+1}(g_h) := v^t(g_h) + CP^t \cdot \sum\limits_{j \neq h} \frac{rer_h^t - rer_j^t}{d^2(g_h, g_j)}$,
  where $CP^{t+1} := \min\limits_{h=1,\dots,p} \left| \frac{\min\limits_{i,k \in D_h, i \neq k} d_{i,k}^2}{\sum\limits_{j \neq h} \frac{rer_j^t - rer_h^t}{d^2(g_h, g_j)}} \right|$.

- Weighted Power Diagrams: $v^{t+1}(g_h) := v^t(g_h) + CP^t \sum\limits_{j \neq h} \frac{rer_h^t - rer_j^t}{d^2(g_h, g_j)}$,
  where $CP^{t+1} := \min\limits_{h=1,\dots,p} \left| \frac{\min\limits_{i,k \in D_h, i \neq k} w_i \cdot w_k \cdot d_{i,k}^2}{\sum\limits_{j \neq h} \frac{rer_j^t - rer_h^t}{d^2(g_h, g_j)}} \right|$.

Unfortunately, Section 5.5.1 will show that the number of necessary sub-iterations to obtain a feasible solution is comparatively high, while the quality of the solutions is comparatively poor.

5.3.4.2 Update Rule 2

Fryer Jr. and Holden [3] use an update rule based on an algorithm of Aurenhammer et al. [2] for their Power Diagram. This rule incorporates the current sizes of the districts and the current evaluation of the solution. It should be mentioned that the authors do not take activities of basic areas into account. They propose the following update rule:

$$v^{t+1}(g_h) := v^t(g_h) + \frac{\overline{ev_{PD}} - ev_{PD}(S) - \sum\limits_{j=1}^{p} v^t(g_j) \cdot (|D_j^t| - \frac{|BA|}{p})}{\sum\limits_{j=1}^{p} |D_h^t|^2} \cdot (|D_j^t| - \frac{|BA|}{p}) \qquad (5.8)$$

where $\overline{ev_{PD}}$ is an overestimate of the minimum value of $ev_{PD}(\cdot)$. It can be initialized by $ev_{PD}(S')$ for any feasible solution S' and updated according to the current sizes of the districts and the current evaluation of the solution. In order to integrate the activities of

the basic areas, the rule defined in Equation (5.8) can be modified to

$$v^{t+1}(g_h) := v^t(g_h) + \frac{\overline{ev_{PD}} - ev_{PD}(S) - \sum_{j=1}^{p} v^t(g_j) \cdot aer_h^t}{\sum_{j=1}^{p} |w(D_h^t)|^2} \cdot aer_h^t \,.$$

If (weighted) Power Diagrams or additively weighted Voronoi Diagrams are used, only the functions $\overline{ev_{PD}}$ and $ev_{PD}(S)$ have to be replaced by the corresponding evaluation functions. The results presented in Section 5.5.1 are promising, however, the running times are noticeably higher than those of the following approach.

5.3.4.3 Update Rule 3

We propose an update rule based on the current absolute errors and on a dynamic convergence parameter CP^t. That rule is applicable to both additively weighted Voronoi Diagrams and (weighted) Power Diagrams. Our rule says:

$$v^{t+1}(g_h) := v^t(g_h) + CP^t \cdot aer_h^t \tag{5.9}$$

The update of all weights needs $\mathcal{O}(|D_1| + \ldots + |D_h|) = \mathcal{O}(|BA|)$ time. Since a universal convergence parameter for all instances is (nearly) impossible to find or define, this rule uses a dynamic parameter. The dynamic change of this parameter is based on the following ideas:

- If the balance of two consecutive solutions is unchanged, the value of the parameter has to be increased to speed up the convergence.
- If the balance increases, the value of the parameter has to be decreased to avoid further balance increases or oscillations of the solution.
- If the balance decreases, the parameter is reset to an initial value CP^0.

These ideas result in the following update function for this dynamic parameter:

$$CP^t := \begin{cases} 2 \cdot CP^{t-1} & \text{if } bal(S^t) = bal(S^{t-1}) \\ \frac{1}{2} \cdot CP^{t-1} & \text{if } bal(S^t) > bal(S^{t-1}) \\ CP^0 & \text{if } bal(S^t) < bal(S^{t-1}) \end{cases} \tag{5.10}$$

Furthermore, this parameter is set to CP^0 at the beginning of each main-iteration. CP^0 depends on the spatial extension of the overall area, the number of basic areas and the total activity measure. The main idea is to approximate an average distance between two basic

areas and divide the resulting value through the total activity measure of all basic areas: For the AWVDA, this leads to

$$CP^0 := \frac{\sqrt{\max\limits_{i,j \in BA} |x_i - x_j|^2 + \max\limits_{i,j \in BA} |y_i - y_j|^2}}{\sqrt{|BA|} \cdot w(BA)},$$

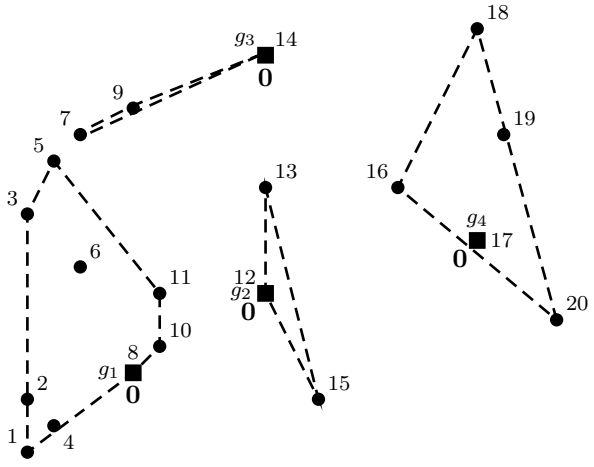
and for both PDDA and the WPDDA, this leads to

$$CP^0 := \frac{\max\limits_{i,j \in BA} |x_i - x_j|^2 + \max\limits_{i,j \in BA} |y_i - y_j|^2}{|BA| \cdot w(BA)}. \tag{5.11}$$

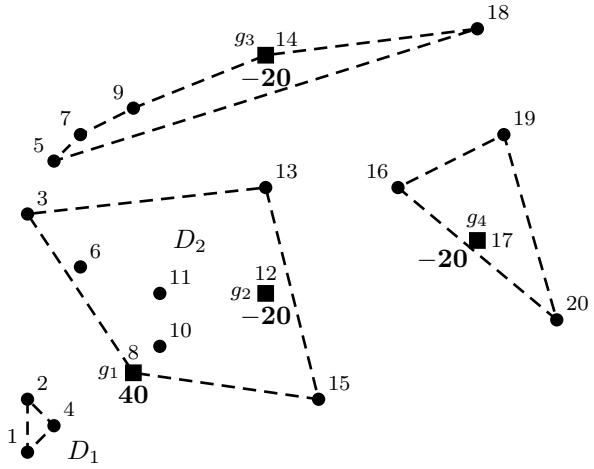
The determination of CP^0 requires also $\mathcal{O}(|BA|)$ time. The results presented in Section 5.5.1 will confirm the efficiency of this approach as well as the quality of the obtained solutions.

Example 5.3.2 Consider the basic areas defined in Table 5.3 and let the locations of the generators $g_1, \ldots, g_4$ correspond to the basic areas 8, 12, 14, and 17. Assume $\tau = 0.2$ and equal activity measures of the basic areas, i.e., $w_i = 1 \ \forall i$. This implies $\mu = \frac{20}{4} = 5$, $L_d = (1 - 0.2) \cdot 5 = 4$ and $U_d = (1 + 0.2) \cdot 5 = 6$, i.e., a solution is feasible if the size of each district is between 4 and 6. Moreover, for purposes of simplification let $CP^0 = 10$.

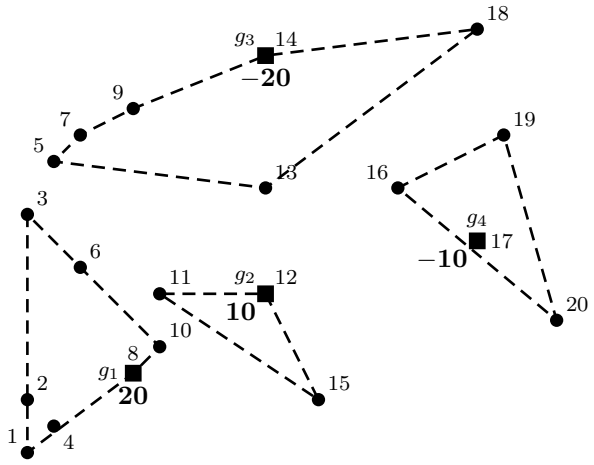
At first, Figure 5.5a depicts the PDDP S^0 for $v^0(g_h) = 0 \ \forall h$. District D_1^0 corresponds to g_1 and consists of nine basic areas. This implies $w(D_1^0) = 9$ and $aer_1^0 = 9 - 5 = 4$, i.e., this district is too large. Hence, the weight of the corresponding generator has to be increased. According to rule 3 this weight results in $v^1(g_1) = v^0(g_1) + CP^0 \cdot aer_1^0 = 0 + 10 \cdot 4 = 40$. In contrast to this, district D_2^0 is too small since $w(D_2^0) = 3$ holds. Therefore, the weight of generator g_2 decreases as follows: $v^1(g_2) = v^0(g_2) + CP^0 \cdot aer_2^0 = 0 + 10 \cdot (-20) = -20$. Moreover, $w(D_3^0) = 3$ and $w(D_4^0) = 5$ holds and leads to $v^1(g_3) = -20$ and $v^1(g_4) = 0$. Furthermore, $bal_{max}(S^0)$ results in 0.8.
Figure 5.5b illustrates the PDDP S^1 for these updated weights. Here, basic area 8 corresponds to the generator of D_1, but it is assigned to D_2. Obviously, the activities of the resulting districts are $w(D_1^1) = 3$, $w(D_2^1) = 8$, $w(D_3^1) = 5$ and $w(D_4^1) = 4$. As required, by updating the weights the activity of D_1 decreases and the activity of D_2 increases. Furthermore, the balance improves since $bal_{max}(S^1) = 0.6$ holds. This implies $CP^1 = CP^0 = 10$. Applying the update rule again results in $v^2(g_1) = 20$, $v^2(g_2) = 10$, $v^2(g_3) = -20$ and $v^2(g_4) = -10$.
Figure 5.5c shows the obtained PDDP S^2 for this set of weights. Now, the activities of the districts are $w(D_1^2) = 7$, $w(D_2^2) = 3$, $w(D_3^2) = 6$ and $w(D_4^2) = 4$. Thus, the balance improves again and results in $bal_{max}(S^2) = 0.4$. This implies $CP^2 = CP^1 = 10$. Updating the weights



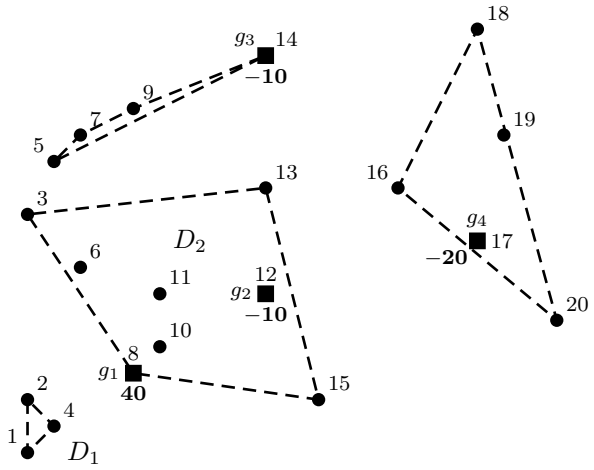
(a) PDDP for $V = \{0; 0; 0; 0\}$



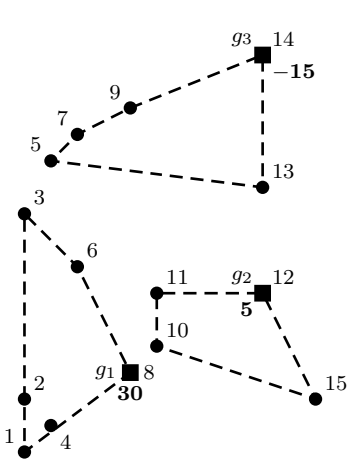
(b) PDDP for $V = \{40; -20; -20; 0\}$



(c) PDDP for $V = \{20; 10; -20; -10\}$



(d) PDDP for $V = \{40; -10; -10; -20\}$



(e) PDDP for $V = \{30; 5; -15; -20\}$

Figure 5.5: Updating the generators' weights

of the generators leads to $v^3(g_1) = 40$, $v^3(g_2) = -10$, $v^3(g_3) = -10$ and $v^3(g_4) = -20$.
Figure 5.5d shows the corresponding PDDP S^3. Here, the activities are $w(D_1^3) = 3$, $w(D_2^3) = 8$, $w(D_3^3) = 4$ and $w(D_4^3) = 5$. Unfortunately, $bal_{max}(S^3)$ equals 0.6, i.e., the balance is worse compared to the iteration before. Hence, this iteration reduces the convergence parameter to $CP^3 = 0.5 \cdot CP^2 = 0.5 \cdot 10 = 5$. Next, it updates the weights to $v^4(g_1) = 30$, $v^4(g_2) = 5$, $v^4(g_3) = -15$ and $v^4(g_4) = -20$.
Figure 5.5e depicts the resulting PDDP S_4. Now, the activities are $w(D_1^4) = 6$, $w(D_2^4) = 4$, $w(D_3^4) = 5$ and $w(D_4^4) = 5$. This implies that the solution is feasible, and, hence, the execution of sub-iterations stops.

5.3.5 Overall Complexity

Finally, this subsection analyzes the complexity of the presented algorithm as a whole. The complexity of the initialization step is $\mathcal{O}(p \cdot K \cdot |BA| \cdot \log |BA|)$ for using the RPA, whereas it is $\mathcal{O}(p \cdot |BA|)$ for using one of the alternative approaches. Each sub-iteration at first updates the generator weights in $\mathcal{O}(|BA|)$ time. After that, it determines a new solution in $\mathcal{O}(p \cdot |BA|)$ time. Finally, it verifies the feasibility of the solution in $\mathcal{O}(|BA|)$ time. Hence, the complexity of each sub-iteration is $\mathcal{O}(p \cdot |BA|)$. Each main-iteration consists of executing at most it_{max}^{sub} sub-iterations, generating a new set of generators, verifying the feasibility and computing the evaluations of the current solution. Thus, for using (weighted) Power Diagrams and Euclidean distances the total complexity of one single main-iteration is $\mathcal{O}(it_{max}^{sub} \cdot p \cdot |BA| + |BA| + |BA| + p \cdot |BA|) = \mathcal{O}(it_{max}^{sub} \cdot p \cdot |BA|)$. However, in general it is $\mathcal{O}(it_{max}^{sub} \cdot p \cdot |BA| + p \cdot |BA|^2 + |BA| + p \cdot |BA|) = \mathcal{O}(p \cdot |BA| \cdot (it_{max}^{sub} + |BA|))$. The algorithm executes at most it_{max}^{main} main-iterations, this implies the overall complexities stated in Table 5.4 for the different versions, where $IT := it_{max}^{main} \cdot it_{max}^{sub}$ Hence, the heuristic is sub-quadratic in p, K, it_{max}^{main}, it_{max}^{sub} and at most quadratic in $|BA|$.

approach	initial solution	complexity
(W)PDDA (Eucl. dist.)	RPA	$\mathcal{O}(p \cdot \|BA\| \cdot (IT + K \cdot \log \|BA\|)$
(W)PDDA (Eucl. dist.)	k-Means++	$\mathcal{O}(IT \cdot p \cdot \|BA\|)$
(W)PDDA (Eucl. dist.)	random	$\mathcal{O}(IT \cdot p \cdot \|BA\|)$
(W)PDDA	RPA	$\mathcal{O}(p \cdot \|BA\| \cdot (IT + it_{max}^{main} \cdot \|BA\| + K \cdot \log \|BA\|))$
(W)PDDA	k-Means++	$\mathcal{O}(p \cdot \|BA\| \cdot (IT + it_{max}^{main} \cdot \|BA\|))$
(W)PDDA	random	$\mathcal{O}(p \cdot \|BA\| \cdot (IT + it_{max}^{main} \cdot \|BA\|))$
AWVDA	RPA	$\mathcal{O}(p \cdot \|BA\| \cdot (IT + it_{max}^{main} \cdot \|BA\| + K \cdot \log \|BA\|))$
AWVDA	k-Means++	$\mathcal{O}(p \cdot \|BA\| \cdot (IT + it_{max}^{main} \cdot \|BA\|))$
AWVDA	random	$\mathcal{O}(p \cdot \|BA\| \cdot (IT + it_{max}^{main} \cdot \|BA\|))$

Table 5.4: Overall complexity

5.4 Multi-Start Algorithm

The solution of an approach presented in the section before highly depends on the initial set of generators, even if the locations of these generators change dynamically during the execution of the algorithm. The computation results in Section 5.5.5 will confirm this. Therefore, this section introduces a multi-start version of this approach in order to use different initial sets of generators and combine their results. This multi-start version uses a parameter it_{max}^{starts} that defines the number of starts. It is also conceivable to define a time limit and generate solutions until this time limit is reached. Section 5.3.1 describes that the RPA (nearly) always computes a feasible solution, whereas the random based approaches sometimes determine no feasible solution. Therefore, the multi-start version uses the RPA for one of the starts and the k-Means++ approach for the $it_{max}^{starts} - 1$ further starts. Note that the k-Means++ approach has advantages in terms of the spatial distribution of the generators over the approach that chooses the generators completely randomly. Of course, the multi-start algorithm chooses the best solution determined during the it_{max}^{starts} runs. In order to compare different solutions an evaluation function ev^{ms} is necessary. Since Voronoi approaches focus on compactness, the usage of a compactness measure as evaluation function is recommendable. Nevertheless, further evaluation functions such as combinations of the measures presented in Section 2.2 are possible. Algorithm 5.4.1 summarizes and outlines this multi-start approach.

Algorithm 5.4.1: Voronoi Based Multi-Start Districting Algorithm

Input: Set of basic areas BA, number of districts p, evaluation function ev^{ms}, parameters it_{max}^{sub}, it_{max}^{main}, it_{max}^{starts}.
Output: Districting plan S.

1 Determine S^{cur} by applying Algorithm 5.3.1 (**Input**: BA, p, it_{max}^{sub}, it_{max}^{main}, and Algorithm 5.3.2).
2 Set $S^{best} := S^{cur}$ and $it_{count}^{starts} := 1$.
3 **while** $[it_{count}^{starts} < it_{max}^{starts}]$ **do**
4 | Determine S^{cur} by applying Algorithm 5.3.1 (**Input**: BA, p, it_{max}^{sub}, it_{max}^{main}, and Algorithm 5.3.3).
5 | **if** $[S^{cur} \textit{ is feasible}]$ AND $\left[ev^{ms}(S^{cur}) < ev^{ms}(S^{best})\right]$ **then** Set $S^{best} := S^{cur}$.
6 | Set $it_{count}^{starts} = it_{count}^{starts} + 1$.
end
7 **return** S^{best}.

5.5 Computational Results

The presented algorithms were coded in C++ and executed on a PC running Windows 7 with a Pentium(R) $E5500$ processor with 2.80 GHz and 2 GB RAM. The tests were mainly conducted on the datasets *PPS* and *ZCA* described in Section 4.4.

Unless specified otherwise, we choose $\tau = 5\%$ as maximum feasible balance deviation, and the initial solution is based on the solution of the RPA using 8 line directions, line partitions as bisecting partitions, and the (unweighted) Moment of Inertia as compactness measure.

This section compares the solutions obtained by applying different algorithm or parameter settings in terms of balance, compactness, contiguity and running times. The evaluation parameters are the same as in Section 4.4. However, Power Diagram regions are always convex, and, hence, the determined solutions are always contiguous (cf. Section 5.1), i.e., $ctg(\cdot) = 0$. For that reason, the following tests state no contiguity results for the PDDA.

5.5.1 Update Rules

The first experiment addresses the different rules for updating the generators' weights presented in Section 5.3.4. Note that rule 3 is the one we have proposed, while rules 1 and 2 have been proposed in the literature. Table 5.5 reports the results for *PPS*, while Table 5.6 reports the results for *ZCA*. The respective first row states the evaluation of the initial solution. The further rows compare the PDDA solutions obtained for setting it_{max}^{main} (maximum number of executed main-iterations) to 50, and it_{max}^{sub} (maximum number of sub-iterations) to 1000 and 5000, while varying the update rule.

Algorithm			*time*	*bal*		*comp*			
	rule	it_{max}^{sub}		*max*	*ave*	*moi*	*wmoi*	*pd*	*wpd*
RPA (MoI)			43	0.80	0.29	−5.63	−5.61	−3.17	−2.94
PDDA	rule 1	1000	1669	0.88	0.35	−5.69	−5.66	−3.20	−2.96
PDDA	rule 2	1000	219	4.37	2.39	−13.61	−13.71	−7.20	−7.14
PDDA	rule 3	1000	90	4.48	2.43	−15.40	−15.52	−8.00	−8.03
PDDA	rule 1	5000	7876	1.02	0.44	−5.75	−5.73	−3.23	−2.99
PDDA	rule 2	5000	563	4.67	2.48	−15.09	−15.30	−7.83	−7.98
PDDA	rule 3	5000	162	4.48	2.42	−15.64	−15.77	−8.06	−8.16

Table 5.5: Dataset *PPS*: Comparing different update rules

First of all, the results show that the running times for using our update rule (rule 3) is noticeably smaller than the running times for using the other approaches. For *PPS* and

Algorithm			*time*	*bal*		*comp*			
	rule	it_{max}^{sub}		*max*	*ave*	*moi*	*wmoi*	*pd*	*wpd*
RPA (MoI)			48	1.97	0.94	−3.03	−2.21	−2.93	−1.30
PDDA	rule 1	1000	3547	2.73	1.43	−5.18	−4.30	−4.02	−2.23
PDDA	rule 2	1000	268	4.05	2.21	−7.88	−7.28	−5.43	−3.67
PDDA	rule 3	1000	153	4.08	2.18	−7.64	−7.07	−5.32	−3.55
PDDA	rule 1	5000	20185	3.74	2.03	−7.43	−6.68	−5.23	−3.32
PDDA	rule 2	5000	1203	4.28	2.29	−8.60	−8.08	−5.84	−4.09
PDDA	rule 3	5000	165	4.21	2.25	−8.08	−7.56	−5.57	−3.81

Table 5.6: Dataset *ZCA*: Comparing different update rules

$it_{max}^{sub} = 5000$ (cf. Table 5.5) our approach determines the solutions of all instances in a total time of 162 seconds, whereas the other approaches need 563 and 7876 seconds, respectively. Nevertheless, the solutions obtained by using rule 1 are still very close to the initial solutions. Thus, they are still balanced but not (very) well in terms of compactness. Table 5.6 depicts that this effect is less pronounced, but still noticeable for *ZCA*. Hence, using rule 1 for the PDDA is not advisable.

For *PPS*, with respect to the compactness, the solutions obtained by using our rule are slightly better than those obtained by using rule 2. For example, for $it_{max}^{sub} = 5000$ in terms of the Weighted Moment of Inertia our rule achieves an average improvement of 15.77% compared to the reference solutions, whereas rule 2 achieves 15.30%. However, for *ZCA* the results are contrary. Here, for $it_{max}^{sub} = 5000$ our rule achieves an improvement of 7.56%, whereas rule 2 leads to an improvement of 8.08%.

Furthermore, the presented results indicate that our rule needs fewer sub-iterations to generate good results. For example, the results for using $it_{max}^{sub} = 1000$ are slightly better in terms of the Weighted Moment of Inertia than those obtained by using rule 2 and $it_{max}^{sub} = 5000$. Finally, in terms of balance the results for our rule and rule 2 are similar.

In summary, our update rule (rule 3) generates good results very fast. Hence, we suggest using this rule for the PDDA. Therefore, the subsequent tests are based on this update rule.

5.5.2 Number of Main-Iterations

This test verifies the advantage of updating the generators' locations dynamically during the execution of the algorithm in contrast to fixing them at the beginning. Therefore, it compares the solutions obtained by allowing only one main-iteration to those obtained by

allowing at most 50 main-iterations, i.e., $it_{max}^{main} = 50$. However, no instance has executed 50 main-iterations, the algorithm always stopped much earlier. This happens if there is no further improvement of a solution during a main-iteration.

it_{max}^{main}	it_{max}^{sub}	*time*	*bal*		*comp*			
			max	*ave*	*moi*	*wmoi*	*pd*	*wpd*
1	100	97	3.26	1.83	−9.51	−9.52	−5.16	−5.01
50	100	100	3.38	1.87	−11.03	−11.11	−5.91	−5.83
1	5000	104	4.34	2.32	−12.29	−12.29	−6.40	−6.25
50	5000	162	4.48	2.43	−15.64	−15.77	−8.06	−8.16

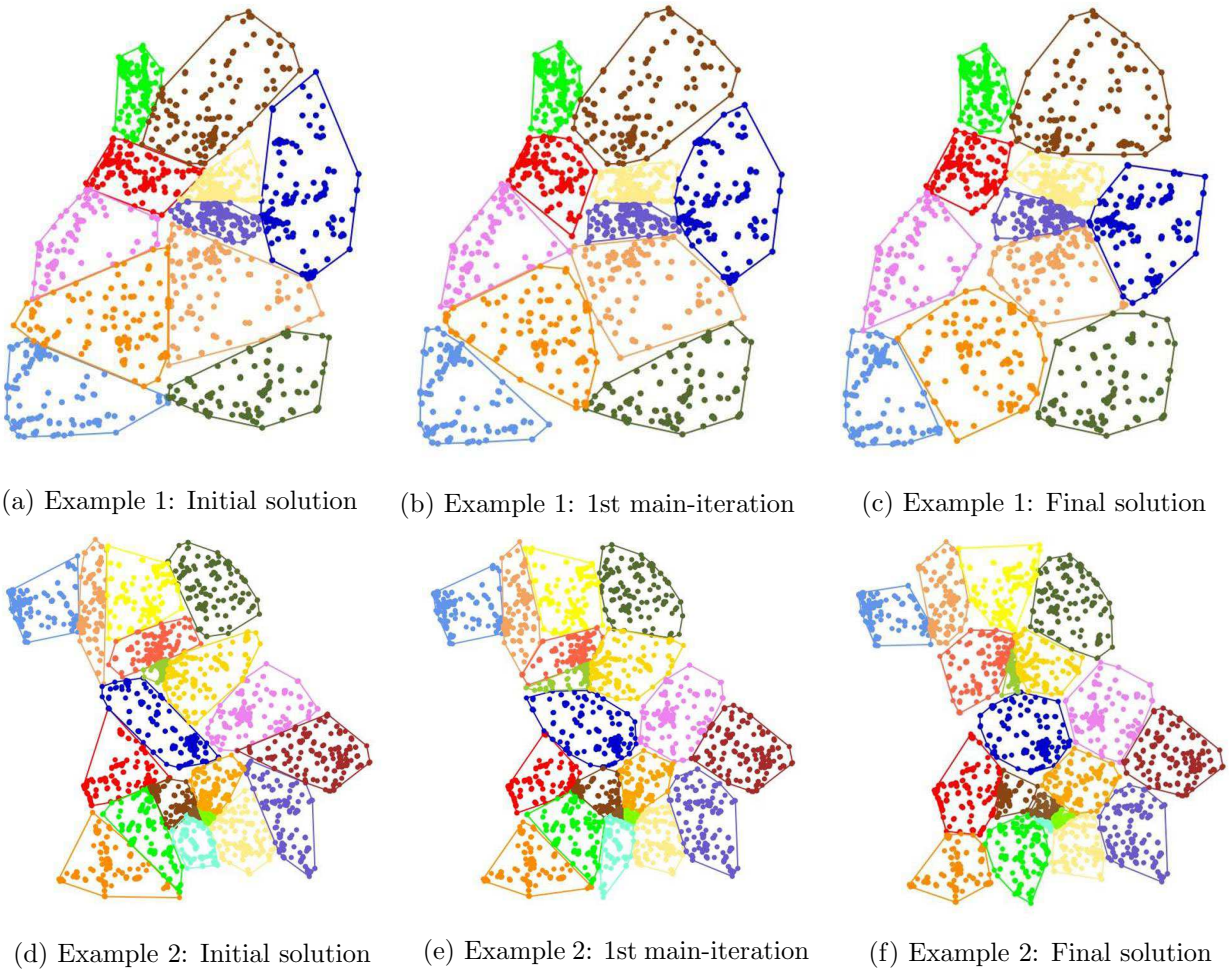
Table 5.7: Dataset *PPS*: Comparing fixed and dynamically changed generators' locations

it_{max}^{main}	it_{max}^{sub}	*time*	*bal*		*comp*			
			max	*ave*	*moi*	*wmoi*	*pd*	*wpd*
1	100	95	2.08	1.01	−3.36	−2.53	−3.12	−1.44
50	100	135	2.09	1.01	−3.38	−2.55	−3.14	−1.45
1	5000	125	4.11	2.14	−6.93	−6.35	−4.96	−3.16
50	5000	165	4.21	2.25	−8.08	−7.56	−5.57	−3.81

Table 5.8: Dataset *ZCA*: Comparing fixed and dynamically changed generators' locations

Table 5.7 states the results for *PPS*. As expected, by executing more main-iterations the running time increases. Moreover, the balance slightly deteriorates. For example, for setting $it_{max}^{sub} = 5000$ the maximum balance increases from 4.34% to 4.48%. However, this test depicts noticeable compactness improvements, e.g., in terms of the Weighted Moment of Inertia there is an improvement from −12.29 to −15.77. Table 5.8 shows similar results for *ZCA*.

Figure 5.6 depicts exemplarily two developments during the execution of the PDDA for one instance of *PPS*. At first, Figure 5.6a illustrates the initial solution. Here, the underlying structure given by line partitions is observable. Figure 5.6b shows the solution after the first main-iteration. Although the initial solution defines the generators, significant changes are noticeable. Here, the improvement in terms of the Weighted Moment of Inertia is 12.58%. After four further main-iterations the algorithm terminates with the solution shown in Figure 5.6c. This solution seems to be visually compact and the improvement compared to the initial solution is 20.26%.
Figures 5.6d to 5.6f present the development of another instance of *PPS*. In this case, with respect to the Weighted Moment of Inertia the final solution is 19.95% better than the



(a) Example 1: Initial solution (b) Example 1: 1st main-iteration (c) Example 1: Final solution

(d) Example 2: Initial solution (e) Example 2: 1st main-iteration (f) Example 2: Final solution

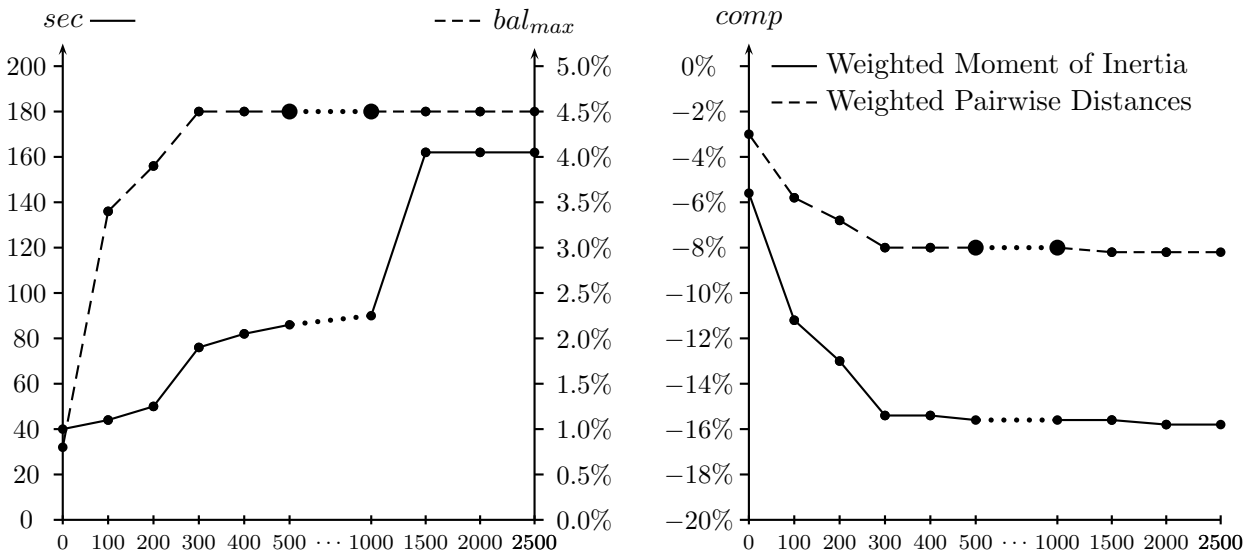
Figure 5.6: Solutions during the execution of the PDDA

initial solution. For example, the orange colored district in the north-west indicates this improvement. It is elongated in the initial solution (cf. Figure 5.6d). Even after executing one main-iteration a small vertical reduction and a small horizontal growth is noticeable (cf. Figure 5.6e). However, in the final solution the district is significantly more compact than in the initial solution (cf. Figure 5.6f).

In summary, this test verifies that updating the locations of the generators dynamically leads to noticeably better results in terms of compactness than working with fixed generators. Therefore, the subsequent tests use $it_{max}^{main} = 50$.

5.5.3 Number of Sub-Iterations

This test addresses the parameter it_{max}^{sub} that defines the maximum number of executed consecutive sub-iterations, i.e., updates of the weights of the generators, until a feasible



(a) Running times and balance

(b) Compactness

Figure 5.7: Dataset *PPS*: Comparing different numbers of sub-iterations

solution is found. For applying the PDDA, this test compares the obtained results for setting $it_{max}^{sub} \in \{100; 200; 300; 400; 500; 1000; 1500; 2000; 2500\}$ while applying the PDDA.

Figure 5.7 presents the results for *PPS*. For purposes of presentability the scale on the number of sub-iterations axis varies. Unsurprisingly, the running time (solid line in Figure 5.7a) increases if the maximum number of sub-iterations increases. Surprisingly, the running times is almost equal for choosing the maximum number of sub-iterations between 1500 and 2500. Hence, in most cases the number of executed sub-iterations until a feasible solution is found is smaller than 1500. Moreover, the balance (dashed line in Figure 5.7a) deteriorates for increasing it_{max}^{sub}. In some instances the balance (nearly) exploits the maximum feasible balance of 5%.

In contrast to this, the compactness becomes better if it_{max}^{sub} increases. As already outlined in Section 5.5.1, the PDDA using rule 3 generates a feasible solution very fast, i.e., it needs a small number of iterations. Figure 5.7b illustrates the corresponding values for the Weighted Moment of Inertia (solid line) and the Weighted Pairwise Distances (dashed line). Here, for allowing more than 300 sub-iterations nearly no further improvement is observable.

Nevertheless, the subsequent tests use $it_{max}^{sub} = 5000$ since the running times are still good and there is most likely no further improvement for increasing it_{max}^{sub}.

5.5.4 Initial Set of Generators

The next test focuses on the initial set of generators. As already pointed out in Section 5.3.1 we prefer using the RPA if only one solution should be generated. Alternatively, the initial generators can be chosen randomly or by an approach based on the k-Means++ algorithm. Since these alternative approaches define the generators randomly, each instance is solved ten times in this case.

Initial solution		*time*	*bal*		*comp*			
			max	*ave*	*moi*	*wmoi*	*pd*	*wpd*
RPA		162	4.48	2.42	−15.64	−15.77	−8.06	−8.16
Random	average	195	4.50	2.38	−12.60	−12.56	−6.71	−6.59
k-Means++	average	151	4.50	2.45	−13.97	−14.11	−7.44	−7.43
Random	moi_{best}	223	4.58	2.47	−15.44	−15.54	−8.08	−8.08
k-Means++	moi_{best}	143	4.60	2.57	−17.27	−17.33	−9.03	−9.06
Random	$wmoi_{best}$	227	4.55	2.46	−15.43	−15.70	−8.08	−8.15
k-Means++	$wmoi_{best}$	143	4.59	2.32	−17.07	−17.49	−8.87	−9.09

Table 5.9: Dataset *PPS*: Comparing different approaches for the initialization

Initial solution		*time*	*bal*		*comp*			
			max	*ave*	*moi*	*wmoi*	*pd*	*wpd*
RPA		165	4.21	2.25	−8.08	−7.56	−5.57	−3.81
Random	average	183	4.40	2.41	−5.12	−4.67	−4.26	−2.55
k-Means++	average	192	4.38	2.40	−5.51	−5.09	−4.47	−2.77
Random	moi_{best}	167	4.52	2.55	−10.89	−10.33	−7.20	−5.03
k-Means++	moi_{best}	179	4.47	2.50	−11.17	−10.75	−7.35	−5.20
Random	$wmoi_{best}$	164	4.44	2.49	−9.90	−11.49	−5.98	−5.68
k-Means++	$wmoi_{best}$	191	4.47	2.49	−10.40	−11.89	−6.45	−5.84

Table 5.10: Dataset *ZCA*: Comparing different approaches for the initialization

Table 5.9 presents the achieved results for *PPS* and Table 5.10 those for *ZCA*. The first row states the results for using the RPA. The second (third) row presents the average results over ten runs for the random (k-Means++) approach, taking only feasible solutions into account. In addition, the further lines present the results for choosing the best feasible solution out of the (at most) ten results according to the (Weighted) Moment of Inertia, where moi_{best} and $wmoi_{best}$ denote these solutions.

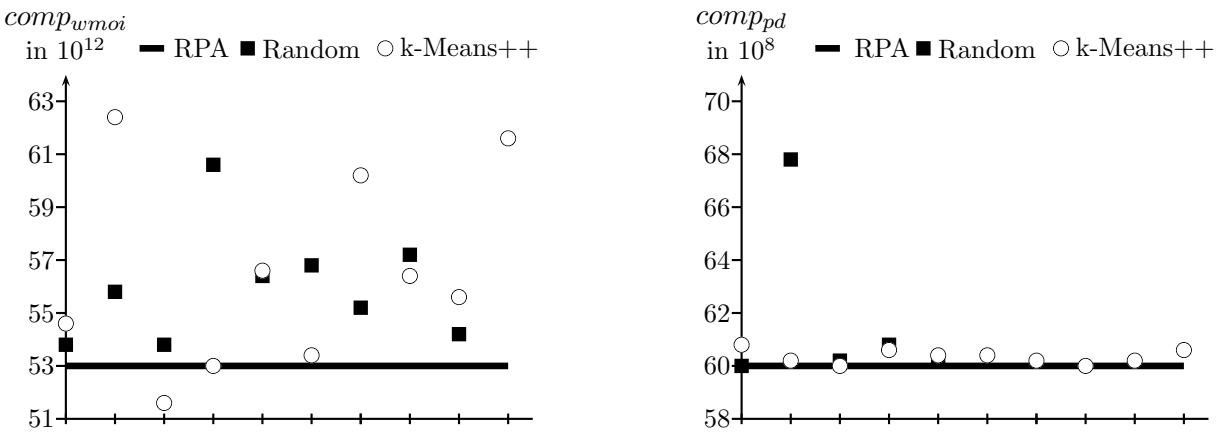
Comparing the results, there are only small differences in terms of balance. The maximum balance ranges from 4.48% (4.21%) to 4.60% (4.52%) while the average balance ranges from 2.32% (2.25%) to 2.57% (2.55%) for *PPS* (*ZCA*).

However, in terms of compactness the solutions using the RPA are noticeably better than the average of the solutions using the random approach or the k-Means++ approach, respectively. For example, for *PPS* in terms of the Weighted Moment of Inertia for *PPS* the RPA solutions are on average 15.77% better than the reference solutions, whereas the k-Means++ solutions are only 13.97% better. Considering the best solutions using the random approach, they are closer to the solutions using the RPA, but according to the (Weighted) Moment of Inertia as well as to the Weighted Pairwise Distances still slightly worse. In terms of Pairwise Distances they are actually slightly better.

Finally, the best solutions using k-Means++ are noticeably better than those using the RPA. For example, in terms of the Weighted Moment of Inertia the best solutions are on average 17.49% better than the reference solutions, whereas the solutions using the RPA are only 15.77% better. For *ZCA* the best solutions of both the random approach and the k-Means++ approach are more compact than those of the PPA approach, where the results of the k-Means++ approach are better than the results of the random approach.

But keep in mind that for each instance the solution using the RPA is compared to the best solution out of ten runs for the other approaches. Furthermore, the main disadvantage of both the random and the k-Means++ approach is that they may not generate a feasible solution at all. During our tests, for *PPS* this occurred on average in 2.52 runs for the random approach and at most in nine of ten runs for a single instance. For the k-Means++ approach the average is only 0.58 infeasible runs per instance, however there is one single instance with ten infeasible solutions after ten runs. For *ZCA* 24 instances are still unsolved after ten runs.

For one instance of *PPS*, Figure 5.8a shows exemplarily the evaluations in terms of the Weighted Moment of Inertia. The solutions using k-Means++ (illustrated by white circles) show a high variance of their compactness values, containing one solution that is better than the compactness value of the solution using the RPA (illustrated by the solid line). The distribution for the random approach (illustrated by black squares) is smaller, but still high. For this exemplary instance only nine of ten runs for the random approach generate a feasible solution. The presented results confirm the assumption that the quality of the obtained solution highly depends on the initial set of generators. Figure 5.8b presents for another instance the evaluations in terms of Pairwise Distances. Here, the results using k-Means++ are comparable to those using the RPA. Using the random approach only five



(a) Weighted Moment of Inertia

(b) Pairwise Distances

Figure 5.8: Examples of compactness distribution for different initial sets of generators

runs have generated a feasible solution, where one solution is significantly worse than the other solutions.

This comparison of different approaches for initializing the generators confirms that the initial set of generators highly influences the quality of the solution. Using the RPA in order to initialize the generators results (nearly) always in feasible solutions, usually doing well in terms of compactness. Therefore, we suggest using the RPA if only one solution should be determined. Moreover, for the multi-start approach, described in Section 5.4, we suggest to determine one solution using the RPA and the further solutions using the k-Means++ approach.

5.5.5 Multi-Start Algorithm

This section examines the results for applying the multi-start Algorithm introduced in Section 5.4. We choose $it_{max}^{main} = 10$, i.e., the algorithm starts ten times. The first time the initial solution is generated using the RPA and subsequently using the k-Means++ approach. The evaluation function corresponds to a compactness measure, i.e., for each instance the algorithm chooses the best solution with respect to this measure.

Tables 5.11 and 5.12 present the results for using the (Weighted) Moment of Inertia as well as (Weighted) Pairwise Distances as evaluation function. The respective first row states the results of the single-start version, i.e., when only using the RPA. The further rows present the results of the multi-start version for different evaluation functions. As additional information, these tables state the attribute *ss-sol* describing the percentage of instances for

version	ev^{ms}	*time*	*bal*		*comp*				*ss-sol*
			max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	
single-start		162	4.48	2.42	−15.64	−15.77	−8.06	−8.16	100.00
multi-start	MoI	1564	4.62	2.59	−17.34	−17.30	−9.04	−8.98	15.15
multi-start	WMoI	1531	4.68	2.50	−17.10	−17.49	−8.83	−9.07	27.27
multi-start	PD	1490	4.62	2.58	−16.92	−16.85	−9.14	−8.04	15.15
multi-start	WPD	1518	4.61	2.51	−16.77	−17.01	−9.06	−9.23	27.27

Table 5.11: Dataset *PPS*: Comparing single-start and multi-start approaches

version	ev^{ms}	*time*	*bal*		*comp*				*ss-sol*
			max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	
single-start		165	4.21	2.25	−8.08	−7.56	−5.57	−3.81	100.00
multi-start	MoI	1602	4.35	2.40	−11.12	−10.40	−7.34	−5.08	30.67
multi-start	WMoI	1534	4.45	2.44	−10.74	−11.93	−6.52	−5.91	26.00
multi-start	PD	1642	4.35	2.41	−10.15	−18.58	−8.17	−4.24	25.00
multi-start	WPD	1644	4.37	2.41	−10.01	−11.30	−6.05	−5.85	28.33

Table 5.12: Dataset *ZCA*: Comparing single-start and multi-start approach

which the solution using the RPA is the best overall solution. In about one quarter of the instances the chosen solution is the single-start solution.

As expected, the running time of the multi-start version for ten runs is approximately ten times higher as for the single-start version. In terms of the applied evaluation function the multi-start solution is at least as good as the single-start solution since for each instance one of the at most ten solutions is the single-start solution. In terms of balance the multi-start solutions are similar but less bad than the single-start solutions.

In terms of compactness, Tables 5.11 and 5.12 point out noticeable improvements of the multi-start solutions compared to the single-start solutions. For example, using the Weighted Moment of Inertia as evaluation function the improvement in terms of the Weighted Moment of Inertia compared to the reference solutions is 17.49%, whereas for the single-start solution this improvement is only 15.64%

In summary, we recommend using the multi-start PDDA if the running time is not the main criterion or the most critical point for the user. In the following, we always choose $it_{max}^{main} = 10$.

5.5.6 Running Times

Table 5.13 shows the running times for a selection of instances from *PPS*. Moreover, it contains running times of some additional large instances provided by our project partner.

nb of basic areas	1092	2019	3531	4049	4971	9847	21315	38667
nb of districts	4	18	13	18	46	41	46	160
single-start	1	4	4	8	15	70	421	4038
multi-start	2	23	35	41	341	1005	7156	22232

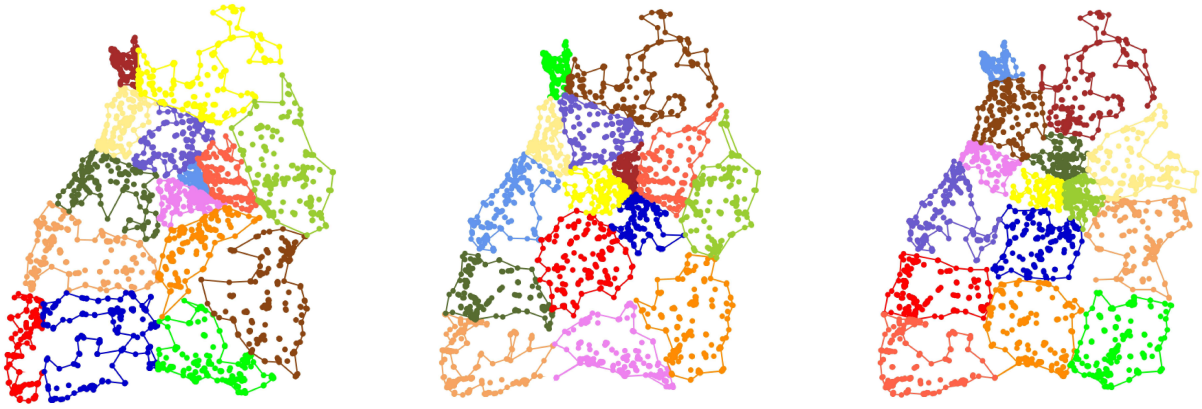
Table 5.13: Running times for some instances (sec)

The running time for the instance having 4049 basic areas and 18 districts is eight seconds in the single-start case, while it is 41 seconds in the multi start case. Even the instance having 9847 basic areas and 41 districts is still solved in about one minute in the single-start case and in about 17 minutes in the multi-start case. For a really large instance having 38667 basic areas more than one hour is necessary in the single-start case and approximately 6.25 hours are necessary in the multi-start case. Usually, the running time for the multi-start case is not ten times the running time of the single start case since the running time depends on the initial set of generators and the induced number of executed iterations.

Having in mind that a tactical or strategical problem is solved these running times are still acceptable. Moreover, on a multi-core processor further improvements of the running time are expected if the starts are parallelized.

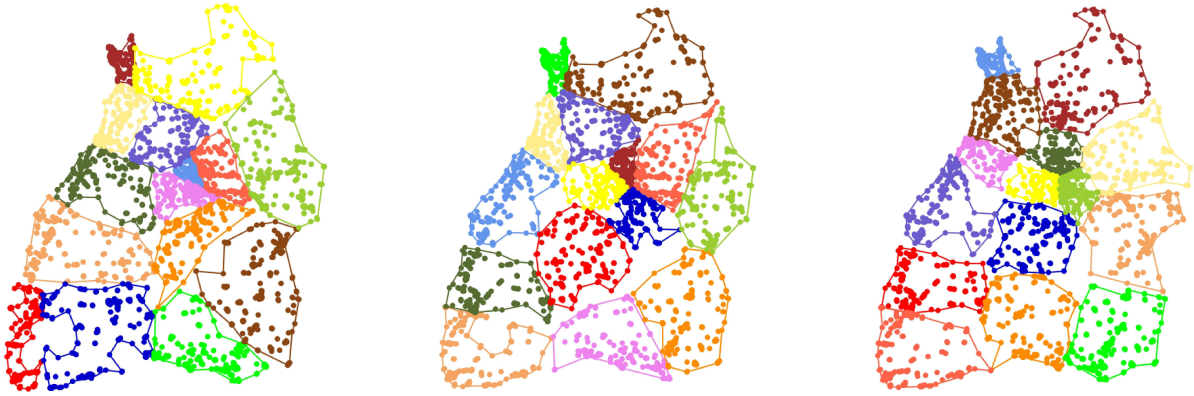
5.5.7 Compactness

The previous sections have shown that the PDDA improves the compactness noticeably during its execution. Furthermore, regarding the results presented in Figures 5.6 the final solutions seem to be visually more compact than the initial solutions. This subsection wants to verify this impression by applying compactness measures, namely the Reock-Test (cf. Section 3.3.1.1), the Gibbs-Test (cf. Section 3.3.1.2), and the reciprocal value of the Schwartzberg-Test (cf. Section 3.3.2.1). These tests are originally defined for polygons (cf. Chapter 3). However, here the basic areas are points. Hence, for each district its area has to be approximated. Section 3.5 provides an overview how this can be done. The following comparison uses convex hulls as well as χ-shapes setting l_χ to 0.5 and 0.75 on solutions obtained by applying the RPA, the PDDA and the multi-start PDDA. Note that an RPA solution corresponds to the initial solution of the PDDA.



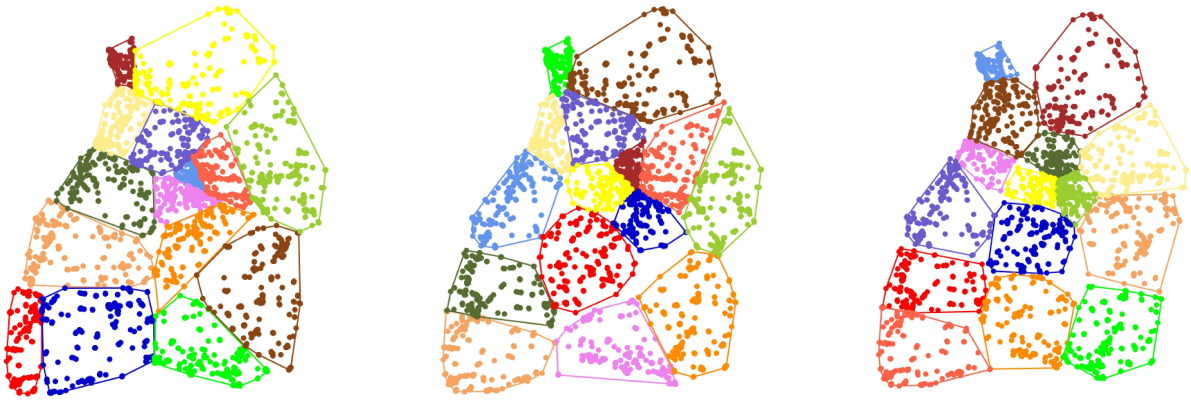
(a) RPA (b) PDDA (c) Multi-start PDDA

Figure 5.9: χ-shapes for $l_\chi = 0.5$



(a) RPA (b) PDDA (c) Multi-start PDDA

Figure 5.10: χ-shapes for $l_\chi = 0.75$



(a) RPA (b) PDDA (c) Multi-start PDDA

Figure 5.11: Convex hulls

Figure 5.9a illustrates the χ-shapes for $l_\chi = 0.5$ for the RPA solution of one instance of *PPS*. Figure 5.9b (5.9c) illustrates the corresponding solution for the PDDA (multi-start PDDA). First of all, this definition of l_χ leads to non-intuitive shapes such as the shape of the brown-colored district in the north of Figure 5.9b. Figure 5.10 shows the χ-shapes $l_\chi = 0.75$ of the same solution. These shapes are more intuitive. Finally, Figure 5.11 depicts the convex hulls for the same solutions. Recall that the evaluation of a district depends on the shape approximation. For example, consider the yellow-colored district in the north of the RPA solution and apply the Reock-Test. For using convex hulls (Figure 5.11a), it is the best evaluated district of this solution; its evaluation is 0.666. For using $l_\chi = 0.5$ (Figure 5.9a), its evaluation is only 0.318, and, hence, it is worse than the average of 0.377 of this solution. A similar effect occurs for the blue-colored area in the south. However, in each case (Figures 5.9 to 5.11), the multi-start PDDA solution is visually more compact than the PDDA solution and noticeable more compact than the solution of the RPA.

Since the applied compactness measures are defined on single districts, for each instance the minimum, maximum and average values of its districts have to be considered in order to compare the different approaches. The results reported in the following tables are again average values over all instances. For *PPS*, Table 5.14 presents the results for applying the Reock-test to χ-shapes and convex hulls. Recall that the optimal evaluation by the Reock-test is 1 if and only if the shape of the district is a circle.

algorithm	χ-shape $l_\chi = 0.5$			χ-shape $l_\chi = 0.75$			convex hull		
	min	*max*	*ave*	*min*	*max*	*ave*	*min*	*max*	*ave*
RPA	0.233	0.526	0.385	0.279	0.590	0.442	0.333	0.648	0.507
PDDA	0.273	0.573	0.427	0.328	0.652	0.498	0.399	0.714	0.567
multi-start PDDA	0.280	0.581	0.438	0.345	0.655	0.511	0.411	0.720	0.581

Table 5.14: Dataset *PPS*: Reock-Test

These results confirm the previous observations. For every kind of shape, the evaluations of the multi-start PDDA solutions are slightly better than the evaluations of the PDDA solutions. Moreover, both the multi-start PDDA solutions and the PDDA solutions are noticeably better evaluated than the RPA solutions. These observations are valid for the minimum, maximum and average values. For example, using convex hulls as the districts' shapes and the average districts' compactness to evaluate a solution, for *PPS* the result is 0.507 applying the RPA, 0.567 applying the PDDA, and 0.581 applying the multi-start PDDA.

As mentioned before, using convex hulls the yellow-colored district in the north is the best

evaluated district of the solution illustrated in Figure 5.11a. Here, the average compactness is 0.534 and the worst compactness is 0.325. Figure 5.11b depicts the PDDA solution where the best evaluated district has an evaluation of 0.789, the worst an evaluation of 0.395. The average is 0.556. Finally, in Figure 5.11c the best evaluated district of the multi-start PDDA is the brown-colored in the north-east having an evaluation of 0.800. The worst evaluated district is the red-colored in the south-east having an evaluation of 0.510. The average evaluation of 0.661 is very good and corresponds to the visual impression.

algorithm	χ-shape $l_\chi = 0.5$			χ-shape $l_\chi = 0.75$			convex hull		
	min	*max*	*ave*	*min*	*max*	*ave*	*min*	*max*	*ave*
RPA	0.236	0.548	0.394	0.282	0.610	0.453	0.334	0.676	0.520
PDDA	0.275	0.592	0.437	0.334	0.669	0.510	0.403	0.736	0.581
multi-start PDDA	0.283	0.603	0.450	0.349	0.681	0.525	0.414	0.748	0.596

Table 5.15: Dataset *PPS*: Gibbs-Test

algorithm	χ-shape $l_\chi = 0.5$			χ-shape $l_\chi = 0.75$			convex hull		
	min	*max*	*ave*	*min*	*max*	*ave*	*min*	*max*	*ave*
RPA	0.499	0.798	0.664	0.633	0.874	0.775	0.768	0.925	0.866
PDDA	0.480	0.807	0.664	0.664	0.891	0.801	0.819	0.944	0.897
multi-start PDDA	0.486	0.814	0.668	0.660	0.897	0.807	0.822	0.945	0.901

Table 5.16: Dataset *PPS*: Schwartzberg-Test

Table 5.15 shows similar results for applying the Gibbs-Test. Mostly, the results for applying the Schwartzberg-Test are similar, too. However, for example, for χ-shapes with $l_\chi = 0.5$ Table 5.16 states a minimal evaluation of 0.499 for the RPA, whereas the evaluation for the PDDA is only 0.480 and for the multi-start PDDA 0.486. Nevertheless, for the average values the multi-start version having an evaluation of 0.668 is slightly better than the PDDA and the RPA, both having an evaluation of 0.664.

The presented results confirm the visual impression that the solutions obtained by using the PDDA are more compact than the solutions achieved by using the RPA.

5.5.8 AWVDA and WPDDA

Until now, the tests have focused on the PDDA. However, Algorithm 5.3.1 presents a general framework that is applicable to other kinds of generalized Voronoi Diagrams as well. This subsection addresses the AWVDA based on additively weighted Voronoi Diagrams and the

WPDDA based on weighted Power Diagrams, and compares their results to those of the PDDA.

Algorithm	*time*	*bal*		*comp*				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
PDDA	162	4.48	2.42	−15.64	−15.77	−8.06	−8.16	0.00	0.00
AWVDA	155	4.36	2.28	−14.21	−14.43	−7.78	−7.99	1.513	0.195
WPDDA	198	4.37	2.37	−10.57	−14.15	−4.08	−7.17	10.221	2.364

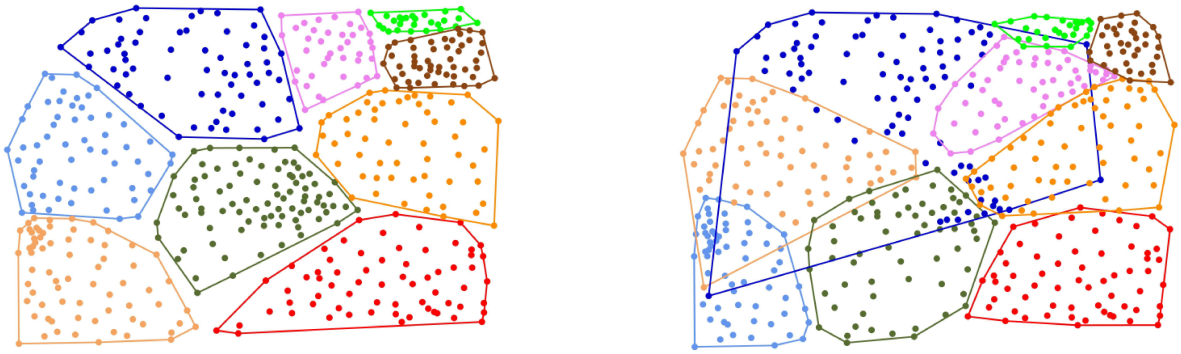
Table 5.17: Dataset *PPS*: Comparing the AWVDA and the (W)PDDA

Algorithm	*time*	*bal*		*comp*				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
PDDA	165	4.21	2.25	−8.08	−7.56	−5.57	−3.81	0.000	0.000
AWVDA	169	4.29	2.35	−8.53	−8.31	−5.63	−4.69	0.099	1.244
WPDDA	120	4.13	2.20	2.30	−8.87	3.92	−4.58	9.200	42.419

Table 5.18: Dataset *ZCA*: Comparing the AWVDA and the (W)PDDA

Table 5.17 states the results for *PPS* and Table 5.18 for *ZCA*. In terms of balance the results are comparable.

In terms of compactness the results differ. For *PPS* the PDDA outperforms both the AWVDA and the WPPDA. For example, in terms of the Weighted Moment of Inertia its solutions are 15.77% better than the reference solutions, whereas the solutions of the AWVDA (WPDDA) are only 14.43% (14.15%) better. In contrast to this, for *ZCA* for each compactness measure the AWVDA solutions are slightly better than the PDDA solutions. However, the AWVDA solutions are not necessarily contiguous, for example there is one instance having an overlap of more than 1.244%. Hence, there is a trade-off between contiguity and compactness. The WPDDA solutions are noticeably better in terms of the Weighted Moment of Inertia. The obtained solutions are 8.87% better than the reference solutions, whereas the AWVDA (PDDA) solutions are only 8.31% (7.56%) better. This result is not surprising since the definition of the WPDDA (cf. Definition 5.1.6) is based on the Weighted Moment of Inertia. Also in terms of Weighted Pairwise Distances applying the WPDDA results in good results. Unfortunately, in terms of the unweighted versions, the solution quality is poor, even worse compared to the reference solutions. The instances of *ZCA* have a noticeable higher variation of the basic areas' activities than the instances of *PPS*. Hence, optimizing the weighted version of a compactness measure does not necessarily lead to the optimization of the unweighted version. Unfortunately, in this case the WPDDA solutions are also very poor with respect to the contiguity. A basic area having a very small activity



(a) PDDA (b) WPDDA

Figure 5.12: Comparison of different Voronoi based districting approaches

can be assigned to a generator far away, but having a small weight. Figure 5.12b depicts an example. The weight of the generator corresponding to the blue district is very small compared to the other weights. Moreover, the basic area in the south-west assigned to the blue district has an activity of only 2.09, while the average activity is about 83. Obviously, this assignment leads to a very large area of intersection between the districts, and, hence, the solution is visually non-satisfying. In contrast to this, Figure 5.12a depicts the corresponding PDDA solution having no intersections.

In summary, the usage of the WPPDA is not advisable since the achieved solutions are very poor in terms of contiguity. If small intersections are acceptable and if there is a high variation in the basic area's activities, the usage of the AWVDP results in satisfactory solutions. However, in general the usage of the PDDA is recommendable since it leads to the best overall solutions.

5.5.9 Further Approaches

After comparing different parameter settings and different types of Voronoi based districting approaches, this test compares the solutions of the PDDA to the solution of further districting approaches.

The first row of Table 5.19 presents the results of the PDDA. The second row depicts the results of the basic version of the RPA that is used as reference solution in terms of compactness. The solutions are contiguous and well balanced, but rather non-compact. The third row reports the evaluations for the solutions of an improved version of the RPA combining flex-zone partitions and line partitions (cf. Section 4.3.3), and applying Weighted Pairwise Distances for evaluating bisecting partitions in terms of compactness (cf. Section 4.3.5.2). In

terms of balance, its solutions are similar to the PDDA solutions. However, they are worse in terms of compactness and slightly worse in terms of contiguity.
The fourth row corresponds to a location-allocation approach (cf. Section 2.3). Here, the balance is worse compared to the PDDA, especially the average balance is 4.08% compared to 2.42%. Usually, the districts obtained by the location-allocation approach nearly exploit the feasible balance tolerance. In terms of the (Weighted) Moment of Inertia the PDDA solutions are slightly better, whereas in terms of the (Weighted) Pairwise Distances the location-allocation solutions are slightly better. In contrast to the PDDA solutions, there are small intersections of at most 0.303% between the districts of the location-allocation solutions. In summary, the solutions of the PDDA seem to be slightly better than the location-allocation solutions.

Algorithm	*bal*		*comp*				*ctg*	
	max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
PDDA	4.48	2.42	−15.64	−15.77	−8.06	−8.16	0.000	0.000
RPA (basic)	0.78	0.31	0.00	0.00	0.00	0.00	0.000	0.000
RPA (improved)	4.54	2.82	−11.20	−12.13	−6.66	−7.09	0.052	0.006
loc-alloc	4.86	4.08	−14.95	−15.68	−8.16	−8.48	0.303	0.036
multi-start PDDA	4.68	2.50	−17.10	−17.49	−8.83	−9.07	0.000	0.000
multi-start loc-alloc	4.97	4.26	−17.79	−18.43	−9.62	−9.90	0.440	0.052

Table 5.19: Dataset *PPS*: Comparing different districting approaches

The last two rows compare the multi-start versions of the PDDA and the location-allocation approach. In terms of compactness, both approaches show noticeable improvements compared to the single-start versions. However, for the location-allocation approach these improvements come along with a further deterioration in terms of balance and contiguity. In particular, the maximum balance comes close to the bound of 5%. Hence, the results of the PDDA are again better in terms of balance and contiguity. However, in this case they are worse in terms of compactness. Thus, there is a trade-off between different criteria and it depends on the preferences of the user or on the application which solution is more suitable.

In summary, this test confirms the quality of the solutions obtained by the PDDA. The obtained solutions are balanced, very good in terms of compactness, and contiguous in any case.

5.5.10 Network Distances

This test evaluates the integration of network distances into the PDDA. On the one hand, the distances measure has an effect on the evaluation of a solution since the evaluation is based on the distances between the basic areas and the generators (cf. Section 5.3.2). On the other hand, the distances measure influences the update of the generators' locations. Since the location of a generator corresponds to the location of a basic area the usage of network distances is straightforward (cf. Section 5.3.3).

The first part of this test includes 23 instances of *PPS* where our practical partner provides distances on a road network. The applied PDDA initializes the generators by the means of RPA and executes at most 5000 sub-iterations and 50 main-iterations. The multi-start approach uses ten starts. The distance-based compactness values are based on road distances. The basic version of the RPA provides the reference solutions, again.

Algorithm	distances	*bal*		*comp* (road distances)				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
PDDA	road	4.41	2.42	−20.31	−20.01	−8.78	−8.71	2.549	1.206
PDDA	Eucl.	4.51	2.47	−18.21	−18.28	−8.66	−8.74	0.000	0.000
multi-start PDDA	road	4.68	2.63	−21.93	−21.60	−9.73	−9.60	2.398	1.146
multi-start PDDA	Eucl.	4.65	2.58	−20.65	−20.92	−9.86	−10.14	0.000	0.000
RPA (basic)	Eucl.	0.22	0.11	0.00	0.00	0.00	0.00	0.000	0.000
RPA (improved)	road	4.42	2.74	−13.39	−13.59	−8.00	−8.14	0.236	0.042
loc-alloc	road	4.95	4.20	−17.93	−18.23	−7.99	−8.13	4.809	1.505
multi-start loc-alloc	road	4.98	4.20	−22.17	−22.57	−10.49	−10.51	3.126	1.216

Table 5.20: Different districting approaches incorporating road distances

Table 5.20 presents the results for applying different districting approaches on these instances. The first row presents the results for the PDDA using network distances during its execution, whereas the second row for using Euclidean distances. In this case, the former results are better in terms of the (Weighted) Moment of Inertia. Nevertheless, in this case, Euclidean distances approximate network distances well. Unfortunately, in contrast to Euclidean distances, networks distances results in overlaps between the districts. Here, the contiguity evaluates at most 2.549%. For example, Figure 5.13 depicts solutions having a contiguity of more than 2%.

Rows three and four compare the corresponding multi-start PDDA versions and show similar results. Comparing the PDDA with the RPA and the location-allocation approach leads to the same observations as in Section 5.5.9. The PDDA performs better than the location-allocation approach in terms of balance and contiguity and comparable in terms

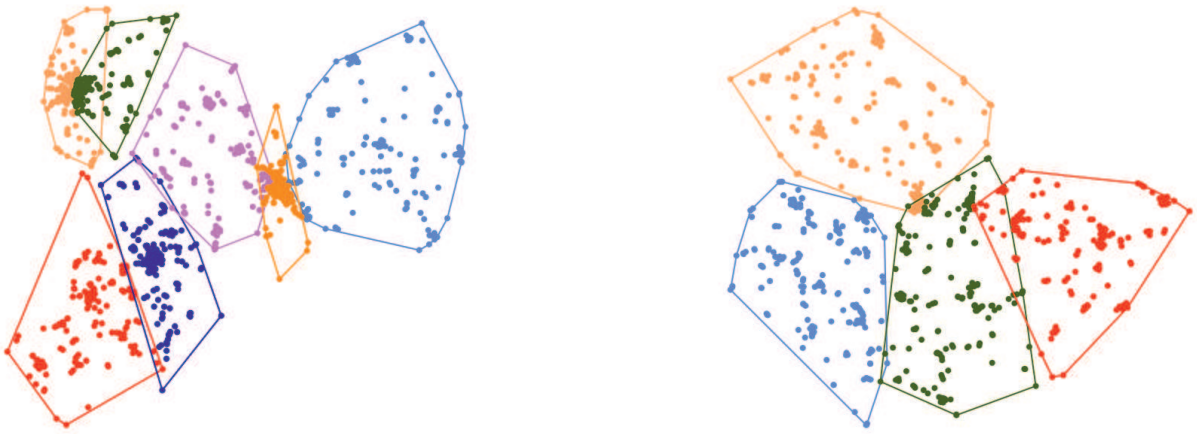


Figure 5.13: PDDA incorporating road distances: Overlapping districts

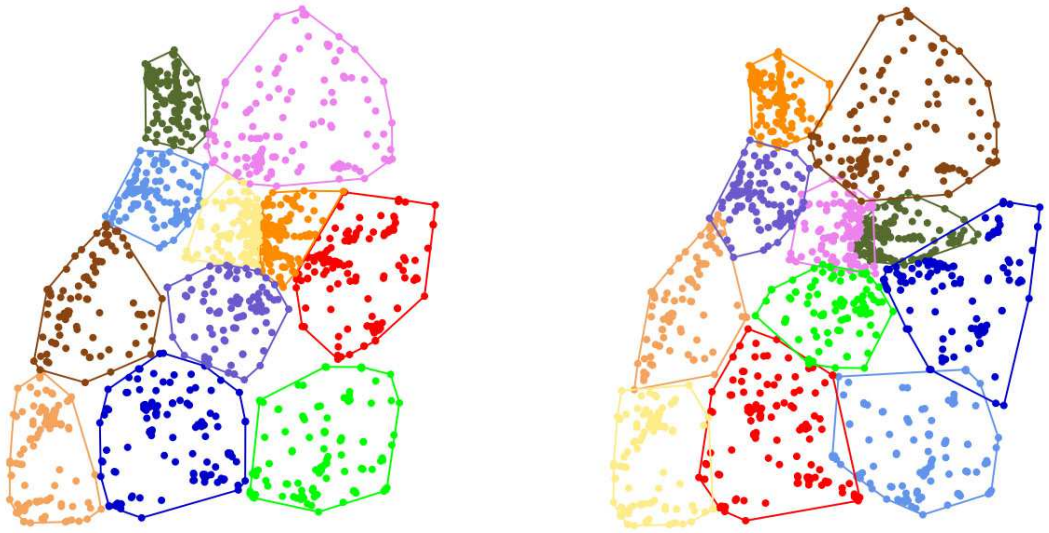
of compactness. Here, the single-start version of the PDDA generates more compact solutions than the single-start version of the location-allocation approach and vice versa for the multi-start versions.

The second part of this test uses 12 instances of *PPS* where our practical partner provides travel times on a road network.

Algorithm	distances	*bal*		*comp*				*ctg*	
		max	*ave*	*moi*	*wmoi*	*pd*	*wpd*	*max*	*ave*
PDDA	travel	4.26	2.26	−18.44	−19.16	−6.91	−7.45	12.033	3.525
PDDA	Eucl.	3.99	2.08	−11.32	−11.74	−6.02	−6.57	0.000	0.000
multi-start PDDA	travel	4.33	2.31	−21.71	−22.47	−7.73	−8.17	6.439	2.732
multi-start PDDA	Eucl.	4.56	2.47	−15.70	−16.81	−7.15	−8.07	0.000	0.000
RPA (basic)	Eucl.	0.34	0.17	0.00	0.00	0.00	0.00	0.000	0.000
RPA (improved)	travel	4.16	2.51	−11.11	−12.78	−6.18	−6.84	0.106	0.015
loc-alloc	travel	4.95	3.87	−18.86	−20.16	−6.86	−7.35	12.094	4.861
multi-start loc-alloc	travel	4.95	4.05	−22.49	−24.05	−8.11	−9.03	8.835	2.860

Table 5.21: Districting approaches according to travel times

Table 5.21 presents the corresponding results. Here, the differences between using network distances and Euclidean distances are more significant. Hence, Euclidean distances do not approximate network distances well. The fact, that the distance travelled in ten minutes differs noticeably if the driver uses a highway or an inner-city road explains this observation. For example, in terms of the Weighted Moment of Inertia using Euclidean distances leads to solutions 11.74% better than the reference solution, whereas using travel times is 19.16% better. However, the overlaps are significantly larger for travel times than for using road



(a) Euclidean distances

(b) Travel times

Figure 5.14: Illustration of the difference between using Euclidean distances and travel times

distances, i.e., the contiguity values are larger. This holds for both the PDDA and the location-allocation approach. The corresponding single-start solutions have a contiguity of at most about 12%. Altogether, Comparing the different approaches results in the same observations as before.

Figure 5.14 illustrates the difference between applying the PDDA based on Euclidean distances and travel times. Figure 5.14a depicts a non-overlapping solution for Euclidean distances, whereas Figure 5.14b shows the corresponding solution for network distances. The underlying dataset is taken from Baden-Württemberg. The orange district in the west of Figure 5.14b is located along the highway A5, and, hence, rather long-shaped than square. Moreover, the lake Constance leads to the overlap between the red and the light blue district in the south-east since the red district contains the region south to it, whereas the blue one contains the region north to it.

This test shows that the PDDA can handle network distances, although it is a geometrically motivated approach. The usage of Euclidean distances as proxy for road distances is sufficient, but the usage as proxy for travel times is unsatisfactory. The comparison to the further districting approaches leads to the same results as for Euclidean distances, reported in Section 5.5.9.

5.6 Extensions

Both the RPA and the PDDA are based on the model presented in Section 4.2. However, the basic model presented in Chapter 2 contains some additional components. This section outlines extensions of the PDDA including some of these components.

5.6.1 Incorporating Prescribed Centers

The first extension addresses prescribed centers, e.g., existing residences of salespersons (cf. Section 4.5). The integration of this extension is straightforward: It defines the set of centers as initial set of generators and fixes them by setting $it_{max}^{main} = 1$, i.e., the generators are unchanged during the execution of the algorithm.

In addition, capacities can be associated with these centers (cf. Section 4.5.6.2). In this case, a solution is infeasible if the activity of at least one district is greater than the capacity of its center and a solution is balanced if the utilizations of all districts are equal. A solution is feasible if the maximum deviation of a district's utilization is smaller than or equal to a feasible deviation τ_{ut}. Hence, the extension defines the current absolute (utilization) error of a district as follows:

$$aer_g^t := w(B_g) - \mu_{ut} \cdot cap_{cen_g} .$$

The remainder of Algorithm 5.3.1 stays as before.

5.6.2 Incorporating Multiple Activities

Some applications take multiple activities into account (cf. Section 4.6). A solution should be balanced with respect to all activity measures.

This extension uses one weight for each generator, but it uses one dynamic convergence parameter $CP^{t,a}$ for each activity measure a, defined analogously to Equations (5.10) and (5.11). Each sub-iteration of the extended algorithm updates the weights of all generators and the convergence parameters of all activity measures. According to Equation (5.9) the update of a generator's weight depends on the convergence parameter, but there is one parameter for each activity. Therefore, each iteration uses for each generator g_h the convergence parameter corresponding to the currently worst balanced activity of district D_h. Let $a^* = \underset{a=1,\dots,A}{\arg\max}\, bal^a(D_h)$, then it is

$$v_{g_h}^{t+1} := v_{g_h}^{t+1} + CP^{t,a^*} \cdot aer_h^{t,a} .$$

Unfortunately, it can occur that one district is too small with respect to one activity measure, but too large with respect to another activity measure. In this case, both decreasing and increasing this generator's weight leads to a further deterioration in terms of balance for at least one activity measure. Therefore, if for one district one activity exceeds the corresponding upper bound and another activity falls below the lower bound, it is most likely impossible to obtain a feasible solution using the corresponding generator. In this case, the extended algorithm removes the respective generator(s) from the set of generators. Then, it relocates the remainder generators according to the procedure described in Section 5.3.3. Afterwards, it relocates the missing generators by the means of the k-Means++ algorithm (cf. Section 5.3.1), resets all generator weights to zero, and starts a new main-iteration.

5.7 Conclusions

This chapter has proposed an algorithm framework for districting problems based on generalized Voronoi Diagrams. The algorithm is a two-stage iterative approach, where one stage relocates the generators and the other stage recalculates the corresponding weights. By dynamically changing the locations of the generators it outperforms approaches that fix the generators at the beginning. Even though the generator locations are dynamic, result quality is still highly dependent on the initial set of generators. Therefore, this chapter has introduced a multi-start variant in addition. In contrast to the RPA presented in the previous chapter, this algorithm puts more emphasis on compactness than on balance. Its optimization goal is mainly compactness, while it uses balance in order to decide whether a solution is feasible or not. Moreover, this chapter has compared different generalized Voronoi Diagrams and concluded that in the context of districting the usage of Power Diagrams is recommendable. Power Diagrams are contiguous if Euclidean distances are used and the allocation of the basic areas to generators is related to the Weighted Moment of Inertia.

A possible extension could address the evaluation functions comparing solutions of different main-iterations or of different starts, respectively. They could combine the evaluations of different planning criteria such as compactness and balance to reflect the user preferences more precisely. In this case, the Pareto front of feasible solutions with respect to these criteria could be approximated additionally, as it was done for example by Paquette et al. [10].

Concerning to the execution of a main-iteration: After achieving a feasible solution the update of the weights could be continued until there is no further improvement or until the obtained solution is infeasible again.

Nevertheless, tests on two datasets comprising 56 instances in total and comparisons to other approaches have confirmed the efficiency of the proposed algorithm framework and the quality of the obtained solution.

Bibliography

[1] F. Aurenhammer. Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys*, 23(3):345–405, 09 1991.

[2] F. Aurenhammer, F. Hoffmann, and B. Aronov. Minkowski-Type Theorems and Least-Squares Clustering. *Algorithmica*, 20, 1998.

[3] R. G. Fryer Jr. and R. Holden. Measuring the Compactness of Political Districting Plans. *Journal of Law and Economics*, 54(3):493–535, 2011.

[4] L. C. Galvão, A. G. N. Novaes, J. E. Souza de Cursi, and J. C. Souza. A multiplicatively-weighted Voronoi diagram approach to logistics districting. *Computers & Operations Research*, 33:93–114, 2006.

[5] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. Nonpartisan Political Redistricting by Computer. *Operations Research*, 13(6):998–1006, 1965.

[6] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry: Theory and Applications*, 28(2-3 SPEC. ISS.):89–112, 2004.

[7] R. Klein. *Algorithmische Geometrie*. Addison-Wesely-Longman, Bonn, 1997. ISBN 978-3827311115.

[8] P. Moreno-Regidor, J. García López de Lacalle, and M.-A. Manso-Callejo. Zone design of specific sizes using adaptive additively weighted Voronoi diagrams. *International Journal of Geographical Information Science*, 26(10):1811–1829, 2012.

[9] A. G. N. Novaes, J. E. Souza de Cursi, A. C. L. da Silva, and J. C. Souza. Solving continuous location-districting problems with Voronoi diagrams. *Computers & Operations Research*, 36:40–59, 2009.

[10] J. Paquette, J.-F. Cordeau, G. Laporte, and M. M. Pascoal. Combining multicriteria analysis and tabu search for dial-a-ride problems. *Transportation Research Part B: Methodological*, 52:1–16, 2013.

[11] F. Ricca, A. Scozzari, and B. Simeone. Weighted Voronoi region algorithms for political districting. *Mathematical and Computer Modelling*, 48:1468–1477, 2008.

[12] M. Sharir. Intersection and Closest-Pair Problems for a Set of Planar Discs. *SIAM Journal on Computing*, 14(2):448–468, 1985.

Part III

Districting on Road Networks

6 Districting for Arc Routing Applications

Contents

This chapter focuses on districting problems where the basic areas correspond to the edges of a graph, for example, streets on a road network. Classical applications are the design of districts for mail or leaflet delivery, waste collection, salt spreading, snow removal, or meter reading. Typically, each street must be serviced exactly once and the required service time as well as the deadheading time along each street is given. The deadheading time is the time necessary to traverse the street without providing service. The aim is to partition the set of edges into a given number of districts such that each district is balanced, compact, connected, and has a small unproductive deadheading time. The balance of a district is based on the total working time required to service all of its edges, including service times and travel times.

6.1 Literature Review

In contrast to the literature concerning polygonal representations of basic areas, the literature on algorithms based on edge representations of basic areas is rather limited.

Bodin and Levy [2] discuss the Arc Partitioning Problem. They have in mind applications such as postal delivery or meter reading. In the given street network, each side of a street requiring service is modeled as a separate arc having the working time that is necessary to service it as its weight. If only one side of a street has to be serviced, the graph is augmented by a parallel opposite arc having a weight of zero. Therefore, the resulting undirected multigraph is Eulerian. In their heuristic the authors firstly select a set of nodes which serve as seeds for the districts. In each step then they assign a parallel pair of street sides to a district, considering balance in terms of service times and connectedness. By adding parallel pairs of streets sides the obtained sub-networks are also Eulerian. After all arcs have been assigned to a district, they apply three exchange steps in order to improve the balance of the districts. Connectedness is not affected by the exchange steps. If the balance is still unsatisfactorily, the center of each district is chosen as seed and the heuristic restarts. The aim of this approach is to find a feasible solution in terms of balance, not an optimal solution, i.e., the service time of each district has to be between given lower and upper bounds. Moreover, compactness is no planning criterion at all. Furthermore, finding a minimum deadheading time Euler cycle is not part of this procedure.
In a second work, the authors refer to the Arc Oriented Location Routing Problem [1]. In this case, an added parallel opposite arc has the deadheading time of the street as arc weight and a depot is located within each district. Moreover, minimizing the number of depots and minimizing the total deadheading time are further planning criteria.

Hanafi et al. [14] consider a districting problem for municipal solid waste collection and present local search methods in order to improve an already given districting plan. The underlying road network is represented by a mixed multi-graph. The regarded problem contains two special features: First, the districts are explicitly allowed to be disconnected. Second, since the number of times a basic area is serviced per week differs, it contains daily varying collection and districting plans. Thus, each basic area is exclusively assigned to a district within each day, but it can be assigned to different districts on different days. The authors consider the total working time of each district, i.e., the service times plus the routing times. Their objectives are the minimizations of the working time imbalance, of the number of connected components, and of the maximum workload of a single district. Furthermore, they treat balance as a constraint by defining minimal and maximal working times for each district. Their model does not take compactness into account. As they allow disconnected districts, the resulting routing problem is a rural postman problem that is NP-hard. Thus, they approximate the routing times based on travel times between the centers of gravity of the connected components, the depot, and the dump site.

Muyldermans et al. [19] address the problem of designing districts for salt spreading or road maintenance. They want to create connected, balanced, and compact sub-networks with centralized depots that support good routing within these sub-networks. If the underlying graph is not Eulerian, they match the odd degree vertices at minimal costs in a pre-processing step. Their approach is similar to that of Bodin and Levy [2], but instead of considering parallel edge pairs, they aggregate the edges into small cycles of edges and treat these cycles as basic areas. Hence, the generated sub-networks by aggregating these cycles are Eulerian in order to enable tours from the depots with no deadheading. For each district its seed is its prescribed depot. If a cycle is close to one of these depots, an initial step assigns it to this depot directly. Then, this approach assigns the further cycles to the districts, first considering balance and closeness, later considering compactness, balance, and estimated number of trucks. The applied assignment rules ensure connectedness.
In Muyldermans et al. [20] the authors present three further heuristics. In order to obtain compact districts, two of these heuristics use a closest assignment rule to assign basic areas to districts while ensuring that districts are connected and balanced. They differ in the definition whether single edges or edge cycles are treated as basic areas. Their conducted experiments show that the larger the vehicle capacity, the better the cycle approach. Most likely, this is because they only consider the radial distances from edges to depots during their assignment procedure, but not the routing costs within the districts. Thus, assigning cycles tends to yield shorter tours than using single edges. For small capacities, the situation is the reverse. The third heuristic reduces the augmented graph in size by directly allocating

basic areas close to the depots and by merging basic areas using structural properties of the graph. Then a mixed integer program is solved, focusing on minimizing the number of required trucks and on compactness. However, this approach requires noticeable more computation time than the former. They apply their approaches to real-world data in Antwerp, Belgium.
An overview and some discussions about the approaches of Muyldermans et al. and some former approaches are given by Perrier et al. [22, 23].

Perrier et al. [24] also focus on problems such as salt spreading and snow disposal on a road network that is modeled as a mixed multi-graph. They deal with the partitioning of a road network into sectors as well as the assignment of these sectors to depots. They assume that each sector is serviced by a single tour. The main idea of their approach is to treat both problems successively by solving mathematical programs. The first approach, called *assign first – partition second*, first assigns street segments to depots, considering capacities, contiguity, and transportation costs. Their formulation of considering transportation costs can be interpreted as a compactness measure. After that, for each depot its streets are partitioned into contiguous sectors while minimizing the number of trucks and considering capacity constraints. The second approach, called *partition first – assign second* is the other way around, i.e., it determines sectors in a first step and assigns these sectors to depots in a second step. The proposed mathematical programs do not model balance explicitly, however, they model different capacity constraints and try to minimize the number of trucks. The authors apply both approaches to real-world street data of Montreal, Canada. They conclude that the first approach outperforms the second one, mainly, the partition problem is not suitable solvable in reasonable time for larger instances.

Mourão et al. [18] focus on a waste collection problem, where a road network should be partitioned into sectors and a route in each sector should be determined. However, their approach is applicable to other problems as well. The road network is modeled as a mixed multi-graph, where each side of a street requiring service is modeled as a separate arc. If both sides can be served simultaneously, the corresponding street is modeled as two opposite arcs, but only one of them has to be visited within a route. The authors present different heuristics that solve the stated problem. Partly, they augment the graph to be Eulerian in a pre-processing step by adding arcs on the shortest path between odd degree vertices. In an initial step they select a seed for each district. After that, at each iteration they either add one single required arc or a small cycle to one district. In order to obtain balanced and compact districts, they chose the smallest district in terms of workload and use a closest assignment rule to add an arc or cycle to it, ensuring that capacity constraints are satisfied. However, no compactness measure is used to evaluate an assignment. The corresponding

routing within each district is either determined simultaneously with the generation of the districts, or in a second phase after finishing the districting phase. In contrast to most other approaches, the authors consider forbidden turns explicitly. However, they do not consider contiguity explicitly, and, hence, the set of arcs defining a district may not be connected.

In the context of meter reading in power distribution networks de Assis et al. [6] address a redistricting problem. In this context, there is a large variation on the set of customers over time. Hence, from time to time a redistricting is necessary. The customers are aggregated on edges of an undirected graph and each edge has two activities: A reading time and a number of customers. The authors propose a bi-criteria mathematical programming formulation that tries to maximize balance and compactness while ensuring connectedness and limiting the number of reassignments. In order to solve the problem, they construct in a first step the dual graph where demand occurs at nodes. Thus, the districting problem on edges is transformed into a districting problem on nodes. In order to solve the latter, or more precisely to approximate its Pareto Frontier, they apply a GRASP heuristic that uses multi-criteria scalarization techniques.

García-Ayala et al. [10] discuss the problem of designing districts for implementing various arc routing operations on them. They model the underlying road network as an undirected graph, where each edge corresponds to a street. Moreover, they consider a set of prescribed depots and there is a one-to-one relation between these depots and the generated districts. The aim of their work is to present an integer linear programming model that includes contiguity (connectedness), compactness, deadheading times, and balance. As in the mathematical districting model of Hess et al. [15], this model uses optimizing compactness as objective function, while it treats restricting balance within a given tolerance as a constraint. However, it considers balance only in terms of service times, but not in terms of total working times including travel times. It also models contiguity as a constraint, analogously to Ríos-Mercado and Fernández [25] for example. However, the innovation of their approach is the introduction of node parity constraints to facilitate Eulerian districts. They define that a node having even (odd) degree in the overall graph *loses parity* if there is (are) at least one (two) district (districts) where the degree of this node is odd in the corresponding sub-graph (sub-graphs). They model the node parity constraints by limiting the ratio of nodes loosing parity. In order to solve the presented model, they propose an exact solution algorithm based on branch and bound with a cut generation strategy.

Some authors present school districting problems. However, most of the proposed models are not based on streets as basic areas. Some authors aggregate streets to clusters and deal with them as basic areas. These clusters are called residence tracts [27], planning polygons [7], or

Reference	Balance of service times	Balance of total working times	Compactness	Connectedness	Minimizing deadheading times	Additional criteria
Bodin and Levy (1989) [1]	+	-	-	+	+	minimizing number of depots
Bodin and Levy (1991) [2]	+	-	-	+	-	
Hanafi et al. (1999) [14]	-	+	-	-	+	minimizing number of connected components
Muyldermans et al. (2002) [19]	+	-	+	+	0	minimizing number of trucks
Muyldermans et al. (2003) [20]	-	-	0	+	0	minimizing number of trucks
Perrier et al. (2008) [24]	0	-	0	+	-	minimizing number of trucks
Mourão et al. (2009) [18]	0	0	0	-	0	
de Assis et al. (2014) [6]	+	-	+	+	-	minimizing number of reassignments
García-Ayala et al. (2016) [10]	+	-	+	+	+	

Table 6.1: Districting based on edge representations of basic areas: Included criteria

blocks [4]. Chapleau et al. [5] define stopping points of the school bus and assign students to the closest stop. To the best of our knowledge, only Ferland and Guénette [9] work directly with streets as basic areas. They assign them to schools using an allocation procedure based on closest assignments under consideration of capacity constraints.

Table 6.1 summarizes the presented districting approaches according to their included planning criteria. An entry of '+' ('-') indicates the (non-)consideration of the corresponding criterion. An entry of '0' indicates that the corresponding criterion is considered only implicitly. For example, a closest assignment rule is often used in order to achieve compact districts, or the underlying graph is made Eulerian in a pre-processing step in order to obtain districts inducing small deadheading times.
There is no approach that explicitly considers the total workload including the routing distances within the districts, as well as compactness. For example, Bodin and Levy [1, 2] do not consider compactness, whereas other approaches consider compactness only implicitly by using the closest assignment rule in order to obtain compact districts [14, 18, 20]. Some former works do not include routing distances at all [1, 6, 24], whereas others make the underlying graph in a pre-processing step Eulerian in order to obtain always Eulerian districts [1, 18, 20]. García-Ayala et al. [10] consider compactness as well as routing distances,

however, they try to achieve balanced districts in terms of service times, but not in terms of total working times. Moreover, in fact, balance and deadheading times are not minimized, they are treated as constraints, i.e., they are bounded to be in prescribed ranges.

The goal of this chapter is to present an algorithm that considers compactness as well as routing distances explicitly. This algorithm tries to optimize compactness, total routing distances, and balance in terms of total working times. It has already been published in Butsch et al. [3] and the following description is based on this work.

The remainder of the chapter is organized as follows: The next section will introduce the model for the arc districting problem. Section 6.3 describes the different components of the proposed algorithm. Section 6.4 presents the results of extensive computational tests, in order to assess the algorithm in terms of solution quality and running times. Moreover, Section 6.5 outlines some possible extensions. This chapter concludes with a summary and a short outlook.

6.2 The Model

The underlying road network is modeled as a connected undirected multi-graph $G = (V, BA)$ called the *street graph*. Each edge of this graph corresponds to a street segment of the underlying road network and the street is stored by means of a connected collection of line segments, where each line segment is specified by the geographic coordinates of its end-points. The node set V corresponds to street crossings or dead ends. A node having an even degree is denoted as *even node*, whereas a node having an odd degree is denoted as *odd node*. The focus is on applications where districts are serviced on foot or by bike. Classical examples for such applications are mail and leaflet delivery, the reading of gas and electricity meters, or door-to-door campaigning. Based on these applications, the following assumptions can be made:

- Without loss of generality, the street graph is connected. If the graph is not connected, each connected component can be considered separately.
- The street graph is undirected. The districts are serviced on foot or by bike since in these cases one-way streets are not prohibitive.
- Each edge is fully serviced in one single traversal. If one street can be serviced in a zig-zag pattern, this street can be modeled as one edge, otherwise, this street can be modeled as two parallel edges since a multi-graph is used.
- Each edge is a required edge. Strongly spoken, an edge is required if there is at least one household or mail box on the corresponding street segment. Streets without households are most likely highways, where the traversing by foot or bike is prohibited. Hence, these streets need not to be modeled. Within cities there are usually no further non-required arcs.
- Each district corresponds to a single round tour. Hence, no specified start- or end-points are defined.
- Depots are not considered since stem distances from a depot to the district are either not an issue, such as in meter reading, or negligible compared to the working time in the district, as, e.g., in mail and leaflet delivery. For example, in leaflet delivery, the deliverer might pick up the leaflets at the supermarket before he distributes them in the neighborhood.
- All kind of turns are feasible since the districts are serviced on foot or by bike.
- While servicing a district, the traversing of other districts is allowed. If necessary, a service person will simply walk the shortest path between two edges he has to service, even if some traversed streets are serviced by another service person.

6.2.1 Components

Chapter 2 has presented a general model for the districting problem. This subsection provides an overview how its components are adapted for the usage in the context of edge representations.

6.2.1.1 Basic Areas

Here, the *basic areas* correspond to edges of the street graph. Each basic area is fully serviced in a single traversal and has two quantifiable activity measures: A *service time* and a *deadheading time*. The service time s_i is the time needed to serve the demand of all customers of the edge, whereas the deadheading time d_i corresponds to the time needed to traverse the edge without providing service, obviously $d_i \leq s_i$ holds.
The underlying street is stored as a sequence of points $(x_i^1, y_i^1), \ldots, (x_i^{m(i)}, y_i^{m(i)})$ representing the end-points of a connected collection of line segments, where $m(i)$ is the number of end-points.

6.2.1.2 Distances

The distance $d(i,j) := d_{i,j}$ between two edges i and j is defined as the minimum distance between the end-nodes of i and j. The distance between two nodes is the distance of a shortest path between them with respect to the deadheading times of the edges.

6.2.1.3 Districts

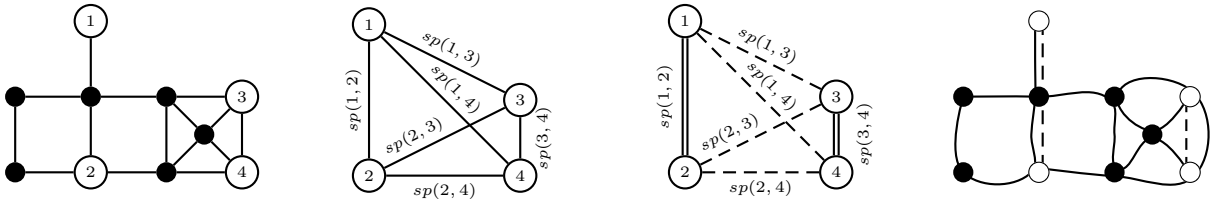
A *district* D_g consists of the set of basic areas $B_g \subseteq BA$ serviced in a single tour. Furthermore, the sub-graph $H(D_g)$ of G induced by a district D_g is defined as $H(D_g) := (V_g, B_g)$ with $V_g = \{q \in V \mid \exists r : (q,r) \in B_g \text{ or } (r,q) \in B_g\}$.
A basic area i is adjacent to district D_g if i is adjacent to at least one basic area $j \in B_g$.

The *total working time* $w(D_g)$ of a district is defined as the total time required to serve the demand of all of its basic areas including travel times. The first time an edge i is visited it is serviced, i.e., its traversing time is s_i. If this edge is visited again, no service will be provided, i.e., its traversing time is d_i.

In order to determine $w(D_g)$ a Chinese postman problem (CPP) is solved. The CPP is to find the shortest cycle that traverses every edge of a connected undirected graph at least once. This cycle is called *Chinese postman tour* (CPT). The problem was introduced by Guan [13] and a solution approach was proposed by Edmonds and Johnson [8].
If the graph is Eulerian, i.e., each node is an even node, the shortest cycle traverses every edge exactly once. Thus, the total working time is the sum of the service times of all edges. For

(a) Graph G_1 (b) CPT on G_1

Figure 6.1: CPT on a Eulerian graph



(a) Graph G_2 (b) Matching graph $M(G_2)$ (c) MCM on $M(G_2)$ (d) CPT on G_2

Figure 6.2: CPT on a non-Eulerian graph

example, Figure 6.1a depicts a Eulerian graph and Figure 6.1b illustrates one corresponding CPT.

If the original graph is not Eulerian, the CPT has to traverse some edges at least twice. Obviously, these edges are on paths between odd nodes. Hence, the problem of finding a CPT can be transformed into the problem of finding a set of pairs of odd nodes, such that each odd node belongs to exactly one pair and the total length of the shortest paths of these pairs is minimized. To this end, a complete graph called *matching graph* consisting of the odd nodes of the original graph is defined, where the length of an edge is that of a shortest path between its end-nodes in the original graph.

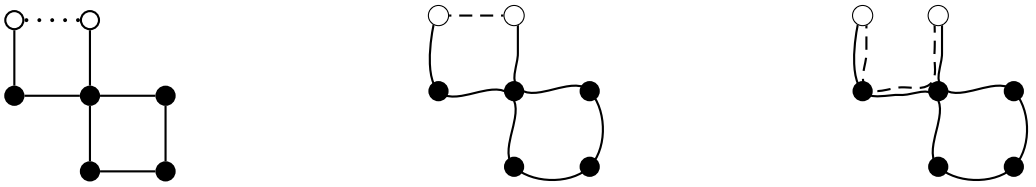
Considering the sub-graph $H(D_g)$ the corresponding matching graph $M(D_g)$ is formally defined as follows:

$$M(D_g) := (V_g^o, B_g^o) \quad \text{with} \quad \begin{cases} V_g^o = \{r \mid r \text{ has an odd degree in } H(D_g)\} \\ B_g^o = \left\{(r,q) \,\middle|\, r, q \in V_g^o\right\} \end{cases} \tag{6.1}$$

The length of an edge $(r, q) \in B_g^o$ of $M(D_g)$ is that of a shortest path $sp(r, q)$ between the nodes r and q in G with respect to the deadheading times.

The problem mentioned above is the problem of finding a *Minimum Cost Matching* (MCM) on the matching graph [12].

Example 6.2.1 Figure 6.2a depicts a non-Eulerian graph, i.e., this graph has odd nodes (white) as well as even nodes (black). Figure 6.2b illustrates the corresponding matching



(a) District D_g

(b) CPT of D_g allowing traversing edges of other districts

(c) CPT of D_g without traversing edges of other districts

Figure 6.3: CPT within a district

graph and Figure 6.2c shows the resulting MCM as double-lined edges. Finally, Figure 6.2d depicts the corresponding CPT, where an edge that is serviced is illustrated as a solid line, while an edge that is only traversed is illustrated as a dashed line, i.e., this edge is a deadheaded edge.

In order to determine $sp(r, q)$ not only the sub-graph $H(D_g)$, but the entire graph G is taken into account. Figure 6.3 illustrates this definition exemplarily: Figure 6.3a depicts the edges of D_g true to scale as solid lines. Moreover, the edge illustrated as dashed lines is assigned to another district. There are only two odd nodes in $H(D_g)$, hence, a CPT traverses the shortest path between them. In real-world applications the service person would most probably choose the tour illustrated in Figure 6.3b since its total length is shorter than the length of the tour illustrated in Figure 6.3c, although this tour traverses a street of another district.

Finally, the total working time of D_g is the sum of the service times of its basic areas, plus the deadheading time induced by the CPT. The corresponding set of deadheaded edges is denoted by TS_{D_g}. Therefore, the deadheading time $DH(D_g)$ of D_g results in

$$DH(D_g) := \sum_{i \in TS_{D_g}} d_i$$

and the total working time $w(D_g)$ results in

$$w(D_g) := \sum_{i \in B_g} s_i + DH(D_g)\,.$$

6.2.1.4 Districting Plan

A *districting plan* or *solution* S is a set of districts $S := \{D_1; \ldots; D_p\}$, where p is the given number of districts. The *total deadheading time* $DH(S)$ of a districting plan is defined as

the sum of the deadheading times of its districts, i.e.,

$$DH(S) := \sum_{g=1}^{p} DH(D_g)\,. \tag{6.2}$$

Analogously, the *total working time* $w(S)$ of a districting plan is defined as the sum of the total working times of its districts, i.e.,

$$w(S) := \sum_{g=1}^{p} w(D_g) = \sum_{i \in BA} s_i + DH(S)\,.$$

In general, the working times of two different districting plans are not equal since the total deadheading times induced by the CPTs can be different for each plan.

6.2.2 Planning Criteria

The aim of the problem can be described as follows: Partition all basic areas (edges) BA into p districts that are connected, balanced, locally and globally compact, and have a small total deadheading time. Next, this section will explain the meanings of these criteria in detail. These criteria can be classified as hard and soft criteria. When a hard criterion is not satisfied, the solution is infeasible; the soft criteria are part of the objective function.

Hard Criteria

A feasible solution must satisfy the following two hard criteria.

6.2.2.1 Complete and Exclusive Assignment

Each basic area must be assigned to exactly one district. An edge is assigned to a district D_g if it is serviced in the corresponding CPT, i.e., if it is an edge of the corresponding sub-graph $H(D_g)$. Note that an edge does not belong to a district if is only traversed in the corresponding CPT without providing service.
Furthermore, a node of G can be assigned to more than one district or sub-graph, respectively.

6.2.2.2 Connectedness

The sub-graph $H(D_g)$ of G induced by a district D_g must be connected. Although disconnected districts are not explicitly forbidden in the applications mentioned above, such districts are nevertheless perceived as highly undesirable by planners. For example, de Assis et al. [6] enforce connected districts for meter reading.

Soft Criteria

The following subsections describe the four considered soft planning criteria. Moreover, they explain their formulations as relative measures in order to make them comparable and applicable to an additively weighted objective function.

6.2.2.3 Balance

Recall (cf. Section 2.2.2) that a common way to measure the balance of a district is to compute the relative percentage deviation of its working time from the average working time $\mu := w(S)/p$, that is

$$bal(D_g) := \frac{|w(D_g) - \mu|}{\mu}\,.$$

The larger this deviation, the worse the balance. In order to measure the balance of an entire districting plan, a classical approach is to use either the maximum relative percentage deviation

$$bal_{max}(S) := \max_{g=1,\dots,p} bal(D_g) \tag{6.3}$$

or the sum (or the average) of all relative percentage deviations. However, both approaches have drawbacks. Hence, our approach combines both ideas using a convex combination to define the balance of a districting plan:

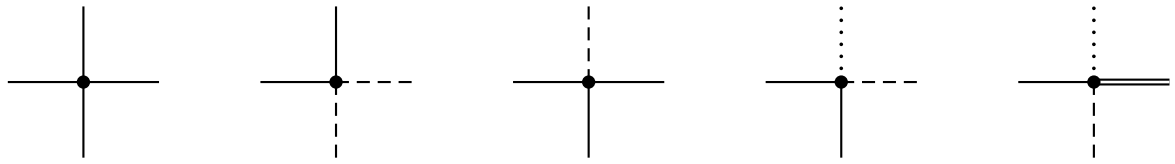
$$bal(S) = \alpha \cdot \frac{1}{p} \cdot \sum_{g=1}^{p} bal(D_g) \;+\; (1-\alpha) \cdot \max_{g=1,\dots,p} bal(D_g)\,, \tag{6.4}$$

where $\alpha \in [0,1]$.

Note that the average working time μ depends on the solution S since the total working time $w(S)$ depends on the deadheading times induced by the CPTs. That means, that by applying an exchange operation the balance of all districts can change even if only a few districts are involved directly.

6.2.2.4 Deadheading Time

A well balanced district does not necessarily have a small deadheading time. For example, a solution with a service time of 30 minutes and a deadheading time of 30 minutes for each district is perfectly balanced, but probably unsatisfactory from an economic point of view.



(a) (b) (c) (d) (e)

Figure 6.4: Assignments of a node having a degree of 4 in G

(a) (b) (c)

Figure 6.5: Assignments of a node having a degree of 3 in G

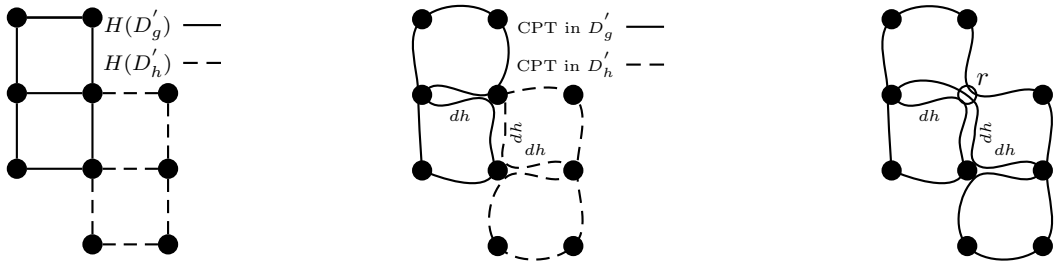
Therefore, the minimization of the total deadheading time $DH(S)$ is regarded as a separate planning criterion.

In order to obtain a relative measure, the relative deviation to a lower bound on $DH(S)$ can be used. For purposes of simplification, in the following the term "district" denotes a district D_g as well as its induced sub-graph $H(D_g)$. At first, Figure 6.4 (6.5) shows exemplarily how the edges incident to an even (odd) node of G can be assigned to districts. Figure 6.4a as well as Figure 6.4b illustrates that an even node of G can be also an even node of all districts it is assigned to. However, it can also be an odd node in some districts as depicted in Figures 6.4c, 6.4d and 6.4e. In contrast to this, an odd node of G will be an odd node of at least one district in any case. For example, the node depicted in Figure 6.5 is either an odd node of one district (Figures 6.5a and 6.5b) or of three districts (Figure 6.5c). Hence, each odd node of G belongs to at least one matching graph in every solution. Thus, a MCM on the matching graph of G, and, hence, a CPT through all basic areas induces a lower bound on $DH(S)$.

Lemma 6.2.1 *Let S be a solution and let TS_{BA} be the set of deadheaded edges in a CPT through all basic areas BA. Then*

$$DH(BA) := \sum_{i \in TS_{BA}} d_i \,,$$

defines a lower bound of $DH(S)$, i.e., the deadheading costs of a CPT through all basic areas BA define a lower bound.



(a) Districts $H(D_g')$ and $H(D_h')$ (b) Corresponding CPTs (c) Merged tour

Figure 6.6: Merging the CPTs of two districts

Proof

Assume that $DH(BA)$ is not a lower bound on $DH(S)$, i.e., a solution $S' := \{D_1'; \ldots; D_p'\}$ exists, such that $DH(S')$ is smaller than $DH(BA)$. Let D_g' and D_h' be two arbitrary neighboring districts, i.e., there is at least one node r belonging to $H(D_g')$ as well as to $H(D_h')$. That means, r is visited at least once in a CPT in district D_g' as well as in a CPT in district D_h'. Therefore, a merged tour visiting all edges of D_g' and D_h' can be constructed as follows: Visit the edges of D_g' according to the corresponding CPT until node r is reached for the first time. Then interrupt this CPT and visit all edges of D_h' according to the corresponding CPT starting and ending in node r. Recall that a CPT corresponds to a cycle. After finishing the CPT in D_h', continue the CPT in D_g' until all remaining edges are visited. For example, Figure 6.6c shows the resulting tour for merging the CPTs of D_g' and D_h' illustrated in Figure 6.6b. Here, *dh* denotes a deadheaded edge. Obviously, the total working time of the merged tour results in $w(D_g') + w(D_h')$ and its deadheading time results in $DH(D_g') + DH(D_h')$. Note that this merged tour is not necessarily the shortest tour visiting all edges of D_g' and D_h' at least once.

In the next steps, this tour can be merged with a further CPT of a neighboring district, and so on. Since G is connected, the CPTs of all districts can be merged to a tour T visiting every edge of G at least once. Then, the total working time of T is the sum of the working times of all districts and its deadheading time is

$$\sum_{g=1,\ldots,p} DH(D_g') := DH(T)\,.$$

According to Equation (6.2) $DH(T)$ equals $DH(S')$, and, hence, $DH(T)$ is smaller than $DH(BA)$. That means, a tour visiting every edge of G at least once and having a smaller deadheading time than a CPT on G exists. This contradicts the definition of a CPT.

Hence, no solution S' exists such that the inequality $DH(S') < DH(BA)$ holds, i.e., $DH(BA)$ is a lower bound on $DH(S)$. □

Therefore, a relative measure to evaluate the deadheading time of a districting plan is the relative deviation of the total deadheading time from this lower bound, i.e.,

$$dh(S) := \frac{DH(S) - DH(BA)}{DH(BA)} . \tag{6.5}$$

Hence, a solution S with $dh(S) = 0$ could have deadheaded edges, but the deadheading is caused by the underlying street graph and not by the districting.

6.2.2.5 Local and Global Compactness

Finally, a district is said to be *compact* if it is nearly round-shaped or square, undistorted, without holes, and has a smooth boundary. See Chapter 3 for a comprehensive overview of proposed measures in the literature. In general, one can distinguish between relative and absolute measures as well as between local and global measures. A relative measure compares the compactness of a district to an ideal value and usually results in a score in the interval $[0, 1]$. For absolute measures this is not the case. A local measure assesses the compactness of a single district, whereas a global measure computes the compactness of the entire districting plan. Based on the recommendations proposed in Section 3.2, the algorithm presented in this chapter uses one local and one global measure for the compactness of a districting plan.

Local Compactness

Most of the available local and global measures work with polygonal basic areas and do not transfer to edges. One of the few exceptions are distance-based measures, which can obviously be adapted to basic areas represented by points or lines (cf. Section 3.5.1.2). The heuristic presented in this chapter uses a measure related to the Moment of Inertia introduced in Section 3.3.5.1. It computes the compactness of a district D_g as the sum of the distances from the assigned basic areas to the district center cen_g:

$$comp(D_g) := \sum_{i \in B_g} d(i, cen_g) ,$$

where

$$cen_g := \arg\min_{i \in B_g} \sum_{j \in B_g} d_{i,j}$$

is the basic area minimizing the sum of distances to all other basic areas.
Distance-based measures are absolute measures. Hence, they are not directly applicable to an additively weighted objective function. Moreover, they are not independent of scale. In

order to overcome these drawbacks, this approach normalizes the result by a factor based on the set of basic areas. Therefore, it computes a global center

$$cen_{BA} := \arg\min_{i \in BA} \sum_{j \in BA} d_{i,j}$$

and uses the aggregated distance of all basic areas BA to this center as normalization factor. Let

$$comp(BA) := \sum_{i \in BA} d(i, cen_{BA}) \,,$$

then, the local compactness of a district results in

$$lc(D_g) := \frac{comp(D_g)}{comp(BA)} \,.$$

As for balance, the local compactness of the districting plan is defined as a convex combination of the average compactness of all districts and the maximum compactness of a single district:

$$lc(S) := \beta \cdot \frac{1}{p} \cdot \sum_{g=1}^{p} lc(D_g) \; + \; (1-\beta) \cdot \max_{g=1,\ldots,p} lc(D_g) \,, \tag{6.6}$$

where $\beta \in [0,1]$.

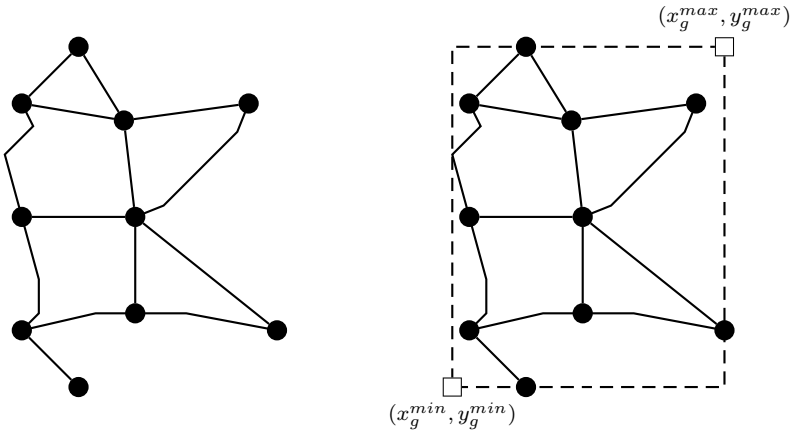
Global Compactness

There is no straightforward definition of a district's shape consisting of basic areas represented by line segments. Following Jarrah and Bard [16], our model uses the smallest enclosing axis-parallel rectangle to approximate the shape of a district. Let $\underline{x}_i$ ($\overline{x}_i$) define the smallest (largest) x-value of the underlying street for basic area i, that means

$$\underline{x}_i := \min_{l=1,\ldots,m(i)} x_i^l \qquad \text{and} \qquad \overline{x}_i := \max_{l=1,\ldots,m(i)} x_i^l \,.$$

Moreover, let $\underline{y}_i$ ($\overline{y}_i$) be defined analogously. Then, the smallest enclosing axis-parallel rectangle $ER(D_g)$ of a district D_g is described by the opposite vertices (x_g^{min}, y_g^{min}) and (x_g^{max}, y_g^{max}) with

$$x_g^{min} := \min_{i \in B_g} \underline{x}_i \,, \qquad y_g^{min} := \min_{i \in B_g} \underline{y}_i \,, \qquad x_g^{max} := \max_{i \in B_g} \overline{x}_i \qquad \text{and} \qquad y_g^{max} := \max_{i \in B_g} \overline{y}_i \,.$$



(a) Road Network (b) Enclosing Rectangle

Figure 6.7: Defining an enclosing rectangle

Example 6.2.2 Figure 6.7a shows the underlying road network of a district D_g. Figure 6.7b illustrates its corresponding axis-parallel enclosing rectangle and depicts its vertices.

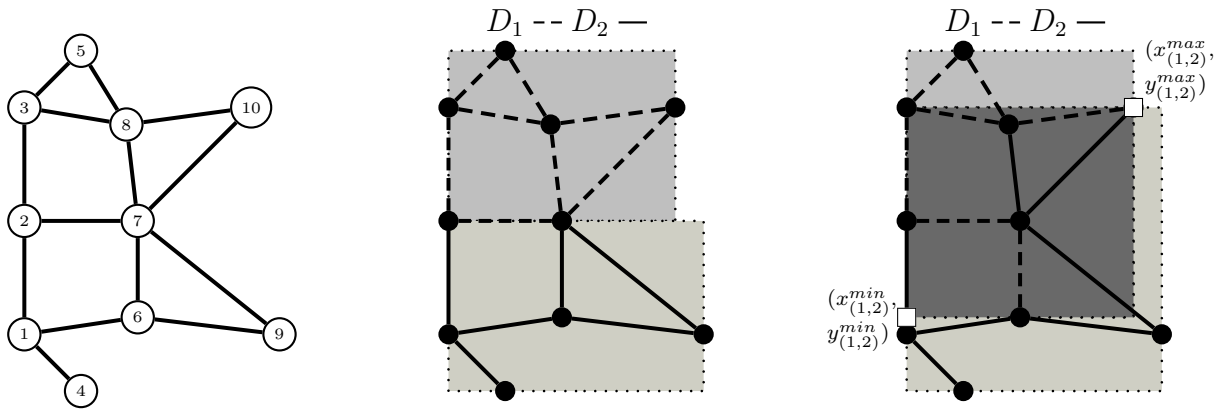
Without loss of generality and for purposes of simplification, in the following illustrations and examples each basic area consists of only one line segment.

The shape of each district is defined independently of the shape of any other district, hence, there are no common borders in general. Thus, taking the total length of common boundaries into account as it is done by Perimeter-Tests (cf. Section 3.3.3.1) is not applicable in this case. Moreover, there can be intersections between the enclosing rectangles of the districts or open spaces within the overall area. Unfortunately, intersections are visually not appealing. Furthermore, in the case of intersections the areas of responsibility for the different service persons are not clearly defined. Therefore, this model defines a districting plan to be globally compact if these enclosing rectangles are non-overlapping. Since this is usually impossible to achieve, the global compactness measure of a districting plan determines the sum of the areas of intersection between pairs of these enclosing rectangles in relation to the area of the enclosing axis-parallel rectangle of all basic areas, i.e.,

$$gc(S) := \frac{1}{area(ER(BA))} \cdot \sum_{g=1}^{p-1} \sum_{h=g+1}^{p} area(ir(g,h)) \,, \tag{6.7}$$

where $ER(BA)$ denotes the smallest enclosing axis-parallel rectangle of BA having the area

$$area(ER(BA)) := (\max_{i \in BA} \overline{x}_i - \min_{i \in BA} \underline{x}_i) \cdot (\max_{i \in BA} \overline{y}_i - \min_{i \in BA} \underline{y}_i) \,.$$



(a) Graph (b) No intersection (c) Large area of intersection

Figure 6.8: Measuring global compactness

Moreover, $ir(g,h)$ denotes the intersection between $ER(D_g)$ and $ER(D_h)$. This intersection is either empty or also an axis-parallel rectangle. Hence, the area of this intersection is given by

$$area(ir(g,h)) = \max\{(x^{max}_{(g,h)} - x^{min}_{(g,h)}); 0\} \cdot \max\{(y^{max}_{(g,h)} - y^{min}_{(g,h)}); 0\}$$

with

$$x^{min}_{(g,h)} := \max\{x^{min}_g; x^{min}_h\}, \qquad y^{min}_{(g,h)} := \max\{y^{min}_g; y^{min}_h\},$$
$$x^{max}_{(g,h)} := \min\{x^{max}_g; x^{max}_h\}, \qquad y^{max}_{(g,h)} := \min\{y^{max}_g; y^{max}_h\}.$$

Note that $gc(S) > 1$ may occur if more than two areas overlap in a sufficiently large region.

Example 6.2.3 Table 6.2 defines the nodes of the graph depicted in Figure 6.8a.

node	1	2	3	4	5	6	7	8	9	10
x	0	0	0	1	1	2	2	1.8	4.5	4
y	1	3	5	0	6	1.3	3	4.7	1	5

Table 6.2: Nodes of the graph depicted in Figure 6.8a

Figure 6.8b shows a partition into two globally compact districts D_1 and D_2. The enclosing rectangle of D_1 is described by the vertices $(0,3)$ and $(4,6)$, while the enclosing rectangle of D_2 is described by the vertices $(0,0)$ and $(4.5,3)$. These enclosing rectangles do not overlap, and as expected this implies $area(ir(1,2)) = \max\{(4-0); 0\} \cdot \max\{3-3); 0\} = 4 \cdot 0 = 0$.

In contrast to this, Figure 6.8c depicts a partition into two districts having a large area of intersection. In this case, the enclosing rectangle of D_1 is described by $(0, 1.3)$ and $(4, 6)$, and of D_2 by $(0, 0)$ and $(4.5, 5)$. This implies $x^{min}_{(1,2)} = 0$, $x^{max}_{(1,2)} = 4$, $y^{min}_{(1,2)} = 1.3$, and $y^{max}_{(1,2)} = 5$, and, hence, $area(ir(1, 2)) = 14.8$. The corresponding area is colored dark gray. The enclosing rectangle of BA is described by the vertices $(0, 0)$ and $(4.5, 6)$. Consequently, it results in $area(ER(BA)) = 27$ and $cg(S) = 0.55$.

6.3 The Algorithm

This section presents our heuristic for the arc districting problem on an undirected graph. In the first phase it constructs an initial solution and then it improves this solution by means of a two-stage iterative approach combining tabu search and adaptive randomized neighborhood search.

The neighborhood of the current solution consists of all solutions resulting from an *operation* applied to the current solution. The heuristic applies three different operations which reassign one or two basic areas to other districts. Since this neighborhood can be quite large, we have developed four different *strategies* that restrict the search to specific subsets of the neighborhood of the current solution. Each strategy focuses on neighboring solutions likely to yield an improvement with respect to one of the four soft criteria. In order to explore the different neighborhoods of the current solution determined by the strategies, we have developed a set of *sub-routines*. There is one sub-routine per strategy, plus one that searches the complete neighborhood of the current solution. Each sub-routine applies tabu search to the respective neighborhood, and stops if no improvement has occurred for a certain number of consecutive iterations. The best solution encountered during this search is the initial solution for the next sub-routine. The sub-routines are randomly selected according to a roulette wheel mechanism, as in adaptive large neighborhood search [26]. The probability of selecting a sub-routine depends on its past performance and on user-defined weights. The algorithm stops after a certain number of consecutive sub-routine executions without improvement.

As explained above the algorithm evaluates a solution in terms of four soft criteria. Hence, essentially, it solves a multi-criteria problem. However, in order to obtain a single objective function, it merges the evaluations of these four criteria, defined in Equations (6.4), (6.5), (6.6) and (6.7) into a weighted objective function:

$$F(S) := w_1 \cdot bal(S) + w_2 \cdot dh(S) + w_3 \cdot lc(S) + w_4 \cdot gc(S)\,, \tag{6.8}$$

with $w_i \geq 0$ for all i and $\sum_{i=1}^{4} w_i = 1$. The weights $w_1, \ldots, w_4$ are specified by the user and reflect the relative priorities of the corresponding criteria. The aim of the algorithm is the generation of a solution satisfying both hard criteria and minimizing F. However, the soft criteria are conflicting, a fact that Section 6.4 will examine. Thus, the algorithm also determines and stores a set of alternative solutions, in addition to the best solution with respect to F using the concept of *Pareto-optimality*. Defining the multi-criteria function

$$MF(S) := (MF^1(S), \ldots, MF^4(S)) := (bal(S), dh(S), lc(S), gc(S))\,,$$

S dominates S' if $MF^c(S) \leq MF^c(S')$ for $c = 1, \ldots, 4$ holds, and there exists one c such that $MF^c(S) < MF^c(S')$. A districting plan S is called *locally Pareto* with respect to a given set of solutions if there is no solution in this set that dominates S. In order to obtain a set of alternative solutions PS, the algorithm stores all locally Pareto solutions with respect to the solutions encountered during its execution.

In the following, this section will explain the construction heuristic. After that, it describes the operations applied in order to create new solutions and the strategies for a systematic search in the neighborhood. Finally, it explains the sub-routines and provides a full description of the algorithm.

6.3.1 Construction Heuristic

The algorithm determines a feasible initial solution by applying a basic version of the Recursive Partitioning Algorithm (RPA), described in detail in Chapter 4. Since the RPA is based on basic areas represented as points and having one activity measure, for each edge a proxy point is defined. Its location is the middle point of the edge and its activity measure is its service time. This middle point is defined as the point located on the collection of line segments and having the same distance to (x_i, y_i) and $(x_i^{m(i)}, y_i^{m(i)})$ measured along these segments. The RPA quickly computes a well balanced and globally compact districting plan.

Unfortunately, the resulting districts are not necessarily connected, because the neighborhood information induced by the street graph is not taken into account. Therefore, in order to obtain connected districts, the algorithm carries out the following post-processing-step: For each disconnected district D_g it determines all of its connected components and the corresponding service times. Let $C_g^1, \ldots, C_g^c$ denote the sets of edges of these components. Moreover, let C_g^{max} denote the largest component in terms of service times. After that, the algorithm reduces each disconnected district to its largest component, i.e., it removes from B_g all edges except those of C_g^{max}. Let B_{un} denote the set of all unassigned basic areas. As a result, all districts are now connected, but the criterion of complete assignment is no longer satisfied.

In order to restore the complete assignment, the algorithm iteratively assigns the basic areas of B_{un} to districts as follows: One iteration firstly determines the set of assignment candidates $AC \subseteq B_{un} \times \{1; \ldots; p\}$, where $(i, g) \in AC$ if $i \in B_{un}$ is adjacent to D_g. After that, it ranks these candidates according to their objective value $F(D_1; \ldots, D_g \cup \{i\}; \ldots, D_p)$, and realizes the best ranked assignment. The algorithm repeats this procedure until all basic areas are assigned to districts. Although connectedness and full assignment are guaranteed by this

procedure, the solution needs no longer to be balanced.
Algorithm 6.3.1 summarizes the steps of the construction heuristic.

Algorithm 6.3.1: Construction Heuristic

Input: *Set of basic areas BA, number of districts p.*

Output: *A feasible districting plan S.*

1 *Compute $S = RPA(BA, p)$ and set $B_{un} = \emptyset$, $AC = \emptyset$.*

2 **for** $g = 1, \ldots, p$ **do**
  Calculate $C_g^1, \ldots, C_g^c$ and C_g^{max}.
  Set $B_g = C_g^{max}$.
  foreach $C_g^j \neq C_g^{max}$ **do** $B_{un} = B_{un} \cup C_g^j$.
end

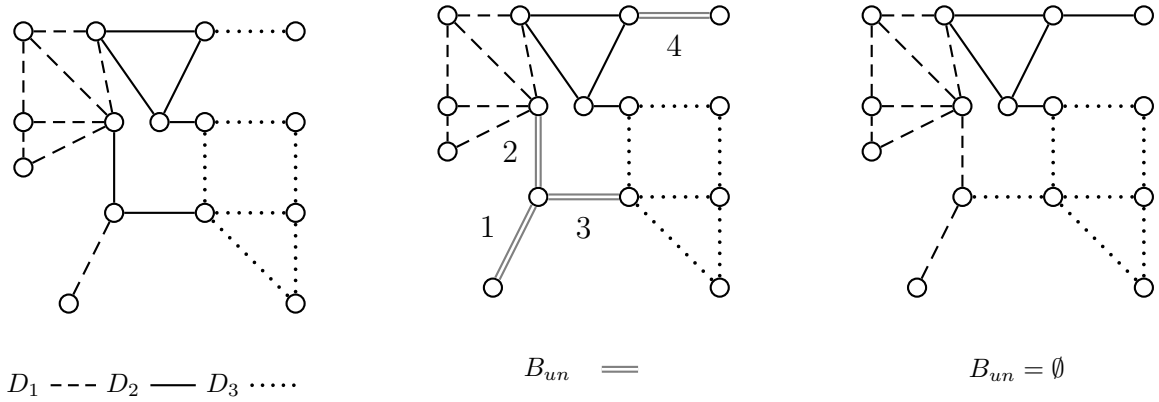
3 **while** $B_{un} \neq \emptyset$ **do**
  $AC = \emptyset$
  foreach $i \in B_{un}$ **do**
    for $g = 1, \ldots, p$ **do**
      if *i is adjacent to g* **then** *Set* $AC = AC \cup \{(i, g)\}$.
    end
  end
  Calculate $(a, t) = \underset{(i,g) \in AC}{\arg\min} F(D_1; \ldots; D_g \cup \{i\}, \ldots, D_p)$.
  Set $D_t = D_t \cup \{a\}$ *and* $B_{un} = B_{un} \backslash \{a\}$.
end

4 **return** S.

Example 6.3.1 Figure 6.9 illustrates this procedure. The solution of the RPA contains three disconnected districts, see Figure 6.9a. Assuming equal service times for each edge, the districts are reduced to those illustrated in Figure 6.9b. Thus, basic areas 1, 2, 3, and 4 are unassigned, i.e., $B_{un} = \{1; 2; 3; 4\}$. Basic area 2 is adjacent to D_1, 3 is adjacent to D_3 and 4 is adjacent to D_2. Since basic area 1 is only adjacent to unassigned basic areas, the set of assignment candidates is $AC = \{(2, 1); (3, 3); (4, 2)\}$. First, the algorithm assigns 4 to D_2 since D_2 is highly unbalanced. In the next two iterations, it assigns basic area 2 to D_1 and basic area 3 to D_3. The final outcome of the algorithm is depicted in Figure 6.9c.

6.3.2 Operations and Neighboring Solutions

The heuristic applies three operations in order to create alternative solutions. Two of them are *shift-operations*, the last one is a *swap-operation*. All operations maintain the complete and exclusive assignment of the basic areas to the districts.



(a) Solution generated by the RPA (b) Solution after restoring connectedness (c) Final solution of the construction phase

Figure 6.9: Illustration of the construction heuristic

6.3.2.1 Shift-Operation

The operation $shift(i, g)$ changes the assignment of basic area i from its current district $D_h = D_{(i)}$ to another district D_g $(g \neq h)$, i.e., $B_h = B_h \backslash \{i\}$ and $B_g = B_g \cup \{i\}$.

6.3.2.2 Double-Shift-Operation

The operation $double\text{-}shift(i, j, g)$ changes the assignment of basic areas i and j from their current district $D_h = D_{(i)} = D_{(j)}$ to another district D_g $(g \neq h)$, i.e., $B_h = B_h \backslash \{i; j\}$ and $B_g = B_g \cup \{i; j\}$.

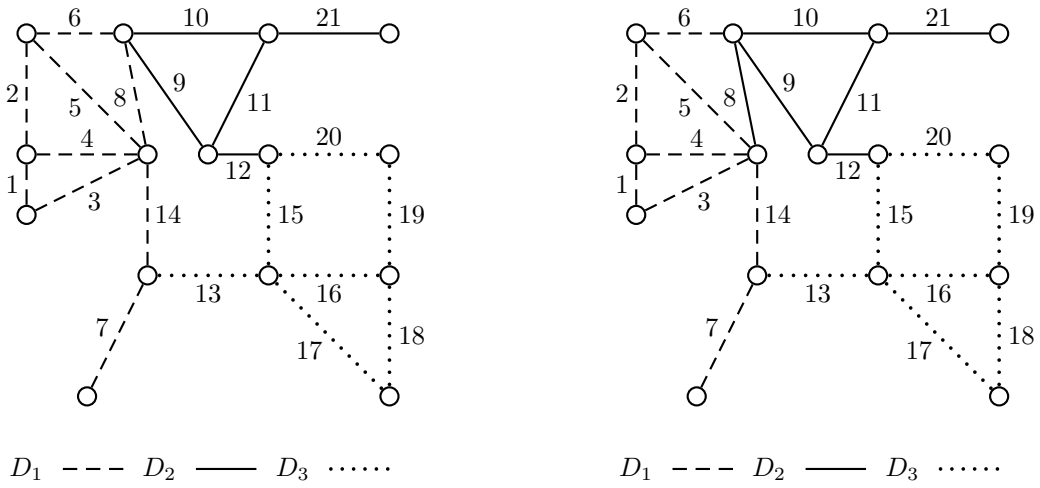
6.3.2.3 Swap-Operation

The operation $swap(i, j)$ changes the assignment of basic area i from its current district $D_h = D_{(i)}$ to district $D_g = D_{(j)}$ $(g \neq h)$ and the assignment of basic area j from its district D_g to district D_h. This leads to $B_h = (B_h \cup \{j\}) \backslash \{i\}$ and $B_g = (B_g \cup \{i\}) \backslash \{j\}$.

An operation is feasible if

- the involved basic areas are not in the current tabu list (see Section 6.3.4);
- the involved districts are still connected after the execution of the operation.

The heuristic executes only feasible operations. The districting plan resulting from the execution of an operation on S is called a *neighboring solution* of S. Assuming $i < j$ for swap-operations, there is a one-to-one correspondence between neighboring solutions and feasible operations. Finally, the set $NH(S)$ of all neighboring solutions of S resulting from feasible operations is called the neighborhood of S. The majority of the strategies abstains from using double-shift-operations. On the one hand a double-shift-operation can



(a) Current solution

(b) Solution after applying $shift(8, 2)$

Figure 6.10: Illustration of an operation

be interpreted as consecutive execution of two shift-operations and on the other hand the number of feasible double-shift-operations is most likely quite large. Therefore, $NH_s(S)$ is the set of all neighboring solutions resulting from feasible shift- and swap-operations. If the meaning is not ambiguous, for short (i, g), (i, j, g), and (i, j) denote $shift(i, g)$, $double\text{-}shift(i, j, g)$, and $swap(i, j)$, respectively.

Example 6.3.2 Figure 6.10a shows a districting plan. Assuming an empty tabu list there are the following nine feasible shift-operations: $(9, 1)$, $(10, 1)$, $(13, 1)$, $(6, 2)$, $(8, 2)$, $(15, 2)$, $(20, 2)$, $(7, 3)$, and $(12, 3)$. For example, $(14, 3)$ is not feasible. The execution of this operation splits D_1 into two connected components, one containing the basic areas 1, 2, 3, 4, 5, 6, 8 and the other containing basic areas 7.
Moreover, in this case the following four swap-operations are feasible: $(6, 9)$, $(6, 10)$, $(8, 9)$, and $(8, 10)$.
Furthermore, the set of feasible double-shift-operations contains altogether 22 operations, namely $(9, 10, 1)$, $(9, 12, 1)$, $(10, 21, 1)$, $(13, 15, 1)$, $(13, 16, 1)$, $(13, 17, 1)$, $(2, 6, 2)$, $(5, 6, 2)$, $(3, 8, 2)$, $(4, 8, 2)$, $(5, 8, 2)$, $(6, 8, 2)$, $(13, 15, 2)$, $(15, 16, 2)$, $(15, 17, 2)$, $(15, 20, 2)$, $(19, 20, 2)$, $(7, 14, 3)$, $(9, 12, 3)$, and $(11, 12, 3)$.
Finally, Figure 6.10b illustrates the districting plan after applying $shift(8, 2)$.

6.3.3 Strategies

Depending on the number of required districts, the structure of the graph, and the length of the tabu list, the cardinality of $NH(S)$ can be very large. Therefore, the algorithm

uses four specialized strategies in order to restrict the size of the neighborhood in the local search phase of a sub-routine. These strategies reduce the running time of the algorithm and can also be used to choose neighboring solutions which specifically improve on a particular criterion. Next, these strategies are presented. Each strategy defines a subset $CL \subseteq NH(S)$ of neighboring solutions, called the *candidate list*.

6.3.3.1 Improve Balance

The *improve balance strategy* (BL) chooses neighboring solutions with the aim of improving balance (cf. Section 6.2.2.3). To this end, it only includes operations that involve highly unbalanced districts. A district is deemed to be highly unbalanced if its balance exceeds a given threshold value bal^{max}. If there are fewer than nb^{bal} districts exceeding this threshold, the nb^{bal} worst balanced districts are considered. An unbalanced district is characterized by a total working time which is either too small or too large. If the total working time of an unbalanced district is too small (large), the heuristic restricts itself to those operations that add (remove) a basic area to (from) this district. In the following, a district having a total working time which is too small (large) is denoted as a small (large) district. In order to allow some flexibility such an operation is not forbidden even if the second district involved in the operation is highly unbalanced as well. Therefore, CL consists of all solutions resulting from feasible shift-operations (i, g) which fulfill one of the following two conditions:

- $D_{(i)}$ is a large district and basic area i is adjacent to district D_g;
- D_g is a small district and basic area i is adjacent to district D_g.

This strategy does not use swap operations. Since a swap operation simultaneously adds and removes a basic area to a district, its impact on balance is only marginal in general. Double-shift operations are also not included.

6.3.3.2 Improve Deadheading Time

The choice of neighboring solutions for the *improve deadheading time strategy* (DH) is motivated by the goal of reducing the total deadheading time (cf. Section 6.2.2.4). As above, this strategy considers the set LDT containing the nb^{dh} districts of the current solution with the largest deadheading times. In order to determine good candidate solutions, a look at the minimal cost matching that determines the CPTs of these districts is necessary. The matching graph $M(D_g)$ is defined in Equation (6.1). The deadheading time of D_g corresponds to the costs $MC(D_g)$ of a MCM on $M(D_g)$. A shift- or swap-operation that modifies $H(D_g)$ changes $M(D_g)$ and, therefore, also the matching costs of D_g.

Let $H^{'}(D_g)$ be the graph obtained from $H(D_g)$ after adding an edge $i = (r, q)$, maintaining connectedness. Concerning the corresponding matching graph $M^{'}(D_g)$ the following effects can occur (see Figure 6.11):

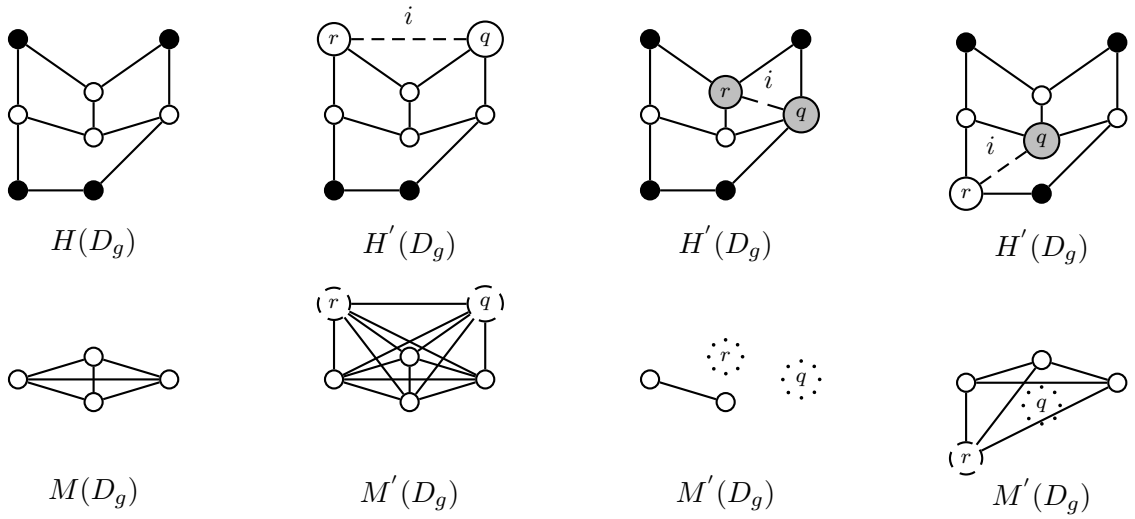
- **Addition of two nodes compared to $\mathbf{M(D_g)}$**: If r and q are even nodes in $H(D_g)$, they are odd nodes in $H^{'}(D_g)$. Hence, in contrast to $M(D_g)$, $M^{'}(D_g)$ contains nodes corresponding to r and q (see Figure 6.11b).
- **Removal of two nodes compared to $\mathbf{M(D_g)}$**: If r and q are odd nodes in $H(D_g)$, they are even nodes in $H^{'}(D_g)$. Hence, in contrast to $M(D_g)$, $M^{'}(D_g)$ contains no nodes corresponding to r or q, respectively (see Figure 6.11c).
- **Replacement of one node compared to $\mathbf{M(D_g)}$**: If r is an even node and q an odd in $H(D_g)$, then r is an odd node and q and even node in $H^{'}(D_g)$. Hence, in contrast to $M(D_g)$, $M^{'}(D_g)$ contains a node corresponding to r, but no node corresponding to q. That means that r replaces q (see Figure 6.11d).

Removing an edge $i = (r, q)$ from $H(D_g)$ produces similar effects on $H^{'}(D_g)$ and $M^{'}(D_g)$, respectively:

- **Addition of two nodes compared to $\mathbf{M(D_g)}$**: If r and q are even nodes in $H(D_g)$, they are odd nodes in $H^{'}(D_g)$. Hence, in contrast to $M(D_g)$, $M^{'}(D_g)$ contains nodes corresponding to r and q (see Figure 6.12b).
- **Removal of two nodes compared to $\mathbf{M(D_g)}$**: If r and q are odd nodes in $H(D_g)$, they are even nodes in $H^{'}(D_g)$. Hence, in contrast to $M(D_g)$, $M^{'}(D_g)$ contains no nodes corresponding to r or q, respectively (see Figure 6.12c).
- **Replacement of one node compared to $\mathbf{M(D_g)}$**: If r is an even node and q an odd in $H(D_g)$, then r is an odd node and q and even node in $H^{'}(D_g)$. Hence, in contrast to $M(D_g)$, $M^{'}(D_g)$ contains a node corresponding to r, but no node corresponding to q. That means that r replaces q (see Figure 6.12d).

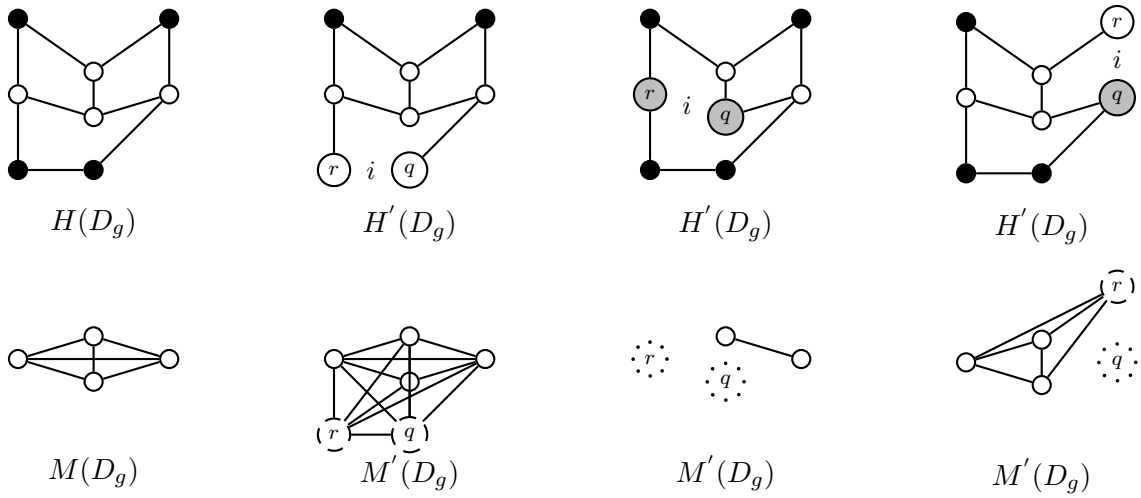
Unfortunately, there exists no general way of knowing which effect results in a reduction of the matching costs.

Example 6.3.3 Figure 6.13a depicts a matching of cost 3 on a matching graph having two nodes. In Figure 6.13c the addition of two nodes r and q to $M(D_g)$ causes an increasing of $MC(D_g)$ to 6. Here, r and q are matched with each other in the resulting MCM. In this case, the resulting MCM of $MC(D_g)$ increases in any case. In contrast to this, in Figure 6.13b the addition of two nodes r and q causes a decrease of $MC(D_g)$ to 2. In this case, r and q are matched with already existing nodes. However, such matchings do not necessarily result in a decrease of $MC(D_g)$, as Figure 6.13d shows, where $MC(D_g)$ results in 4.



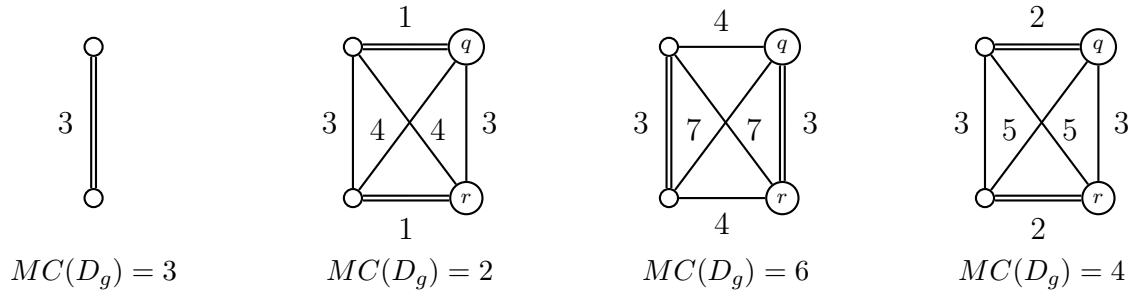
(a) Initial graphs (b) Addition of two nodes (c) Removal of two nodes (d) Replacement of one node

Figure 6.11: Effects on $M(D_g)$ after adding an edge i to $H(D_g)$



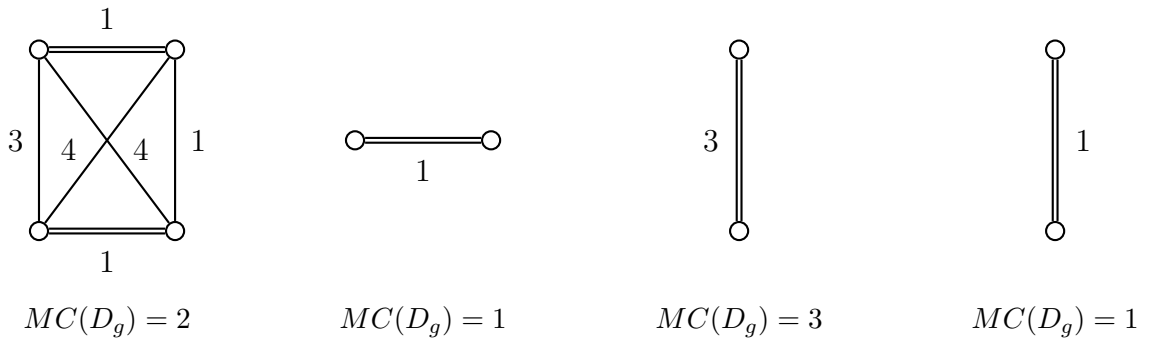
(a) Initial graphs (b) Addition of two nodes (c) Removal of two nodes (d) Replacement of one node

Figure 6.12: Effects on $M(D_g)$ after removing an edge i from $H(D_g)$



(a) Current $M(D_g)$ (b) Decrease in $MC(D_g)$ (c) Increase in $MC(D_g)$ (d) Increase in $MC(D_g)$

Figure 6.13: Changes in $MC(D_g)$ when adding two nodes to $M(D_g)$



(a) Current $M(D_g)$ (b) Decrease in $MC(D_g)$ (c) Increase in $MC(D_g)$ (d) Decrease in $MC(D_g)$

Figure 6.14: Changes in $MC(D_g)$ when removing two nodes from $M(D_g)$

The same effects result from the removal of two nodes.

Example 6.3.4 Removing two nodes from the matching graph depicted in Figure 6.14a leads to the matching graphs illustrated in Figures 6.14b, 6.14c, and 6.14d. The corresponding matching costs are 1, 3, and 1, compared to 2 of the original matching.

If two nodes matched with each other are removed, the matching costs decrease in any case since the further matching remains unchanged. In the other case, an increase as well as a decrease of the matching costs could occur.

Finally, replacing one node in the graph illustrated in Figure 6.13a causes an increase or decrease of the matching costs depending on the distance of the added node to the remaining node.

However, if there is no odd node in $H(D_g)$, i.e., $M(D_g)$ is empty, there are no deadheading costs. Thus, reducing the number of nodes in $M(D_g)$ seems to be advisable. Moreover, if two nodes r and q are odd nodes in $H(D_g)$ and close to each other, their corresponding nodes are most likely matched in a MCM on $MC(D_g)$. Due to the kind of operations applied, the nodes in $H(D_g)$, corresponding to two nodes added to (removed from) $M(D_g)$, are most likely close to each other. Thus, in most cases two nodes added to $M(D_g)$ are matched in the updated MCM, and, hence, $MC(D_g)$ increases. Moreover, in most cases two nodes removed from $M(D_g)$ are matched in the previous MCM, and, hence, $MC(D_g)$ decreases. Therefore, it is more likely that removing nodes from $M(D_g)$ induces a reduction in $MC(D_g)$ than exchanging nodes. Likewise, exchanging nodes more likely yields a decrease in $MC(D_g)$ than adding nodes to $M(D_g)$. Therefore, for every $D_g \in LDT$ the feasible operations that change $H(D_g)$ are partitioned into three disjoint sets $REM(D_g)$, $ADD(D_g)$, and $REP(D_g)$.

The set $REM(D_g)$ is defined as follows:

- $shift(i,g) \in REM(D_g)$ holds if the operation causes a removal of two nodes from $M(D_g)$;
- $shift(i,h) \in REM(D_g)$ with $D_{(i)} = D_g$ holds if the operation causes a removal of two nodes from $M(D_g)$;
- $swap(i,j) \in REM(D_g)$ with $D_{(i)} = D_g$ or $D_{(j)} = D_g$ holds if the operation causes a decrease of the number of nodes in $M(D_g)$.

The set $REP(D_g)$ is defined as follows:

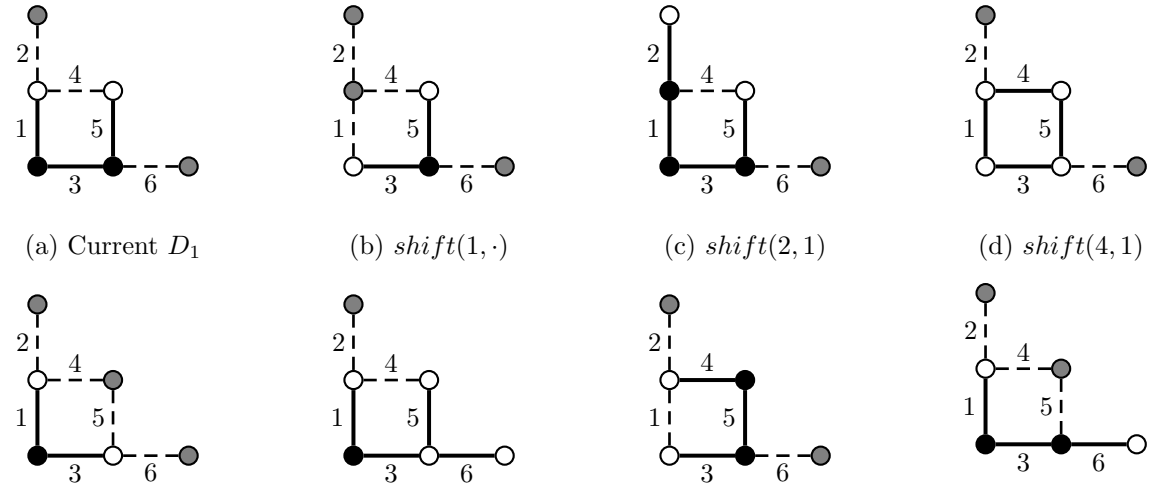
- $shift(i,g) \in REP(D_g)$ holds if the operation causes a replacement of one node of $M(D_g)$;
- $shift(i,h) \in REP(D_g)$ with $D_{(i)} = D_g$ holds if the operation causes a replacement of one node of $M(D_g)$;
- $swap(i,j) \in REP(D_g)$ with $D_{(i)} = D_g$ or $D_{(j)} = D_g$ holds if the number of nodes in $M(D_g)$ stays unchanged after executing the operation.

The set $ADD(D_g)$ is defined as follows:

- $shift(i,g) \in ADD(D_g)$ holds if the operation causes an addition of two nodes to $M(D_g)$;
- $shift(i,h) \in ADD(D_g)$ with $D_{(i)} = D_g$ holds if the operation causes an addition of two nodes to $M(D_g)$;
- $swap(i,j) \in ADD(D_g)$ with $D_{(i)} = D_g$ or $D_{(j)} = D_g$ holds if the operation causes an increase of the number of nodes in $M(D_g)$.

The strategy firstly computes the change of $MC(D_g)$ for every element of $REM(D_g)$. Only for those operations that cause a decrease of $MC(D_g)$ the corresponding solutions are added to CL. If there exists no such operation, the strategy examines the elements of $REP(D_g)$ and computes the respective variations of $MC(D_g)$. Again, for the operations that cause a decrease of $MC(D_g)$, the solutions are added to CL. If there are still no such operations, it finally considers the elements of $ADD(D_g)$ and adds the respective solutions to CL if they improve $MC(D_g)$.
Again, this strategy does not include double-shift-operations since they can be interpreted as a consecutive execution of two shift-operations.

Example 6.3.5 Figure 6.15a depicts the district D_1 represented by solid lines. Odd nodes of $H(D_1)$ are colored white, whereas even nodes are colored black. Hence, $M(D_1)$ consists of two nodes. The remaining illustrations in Figure 6.15 show the feasible operations.



(a) Current D_1 (b) $shift(1,\cdot)$ (c) $shift(2,1)$ (d) $shift(4,1)$

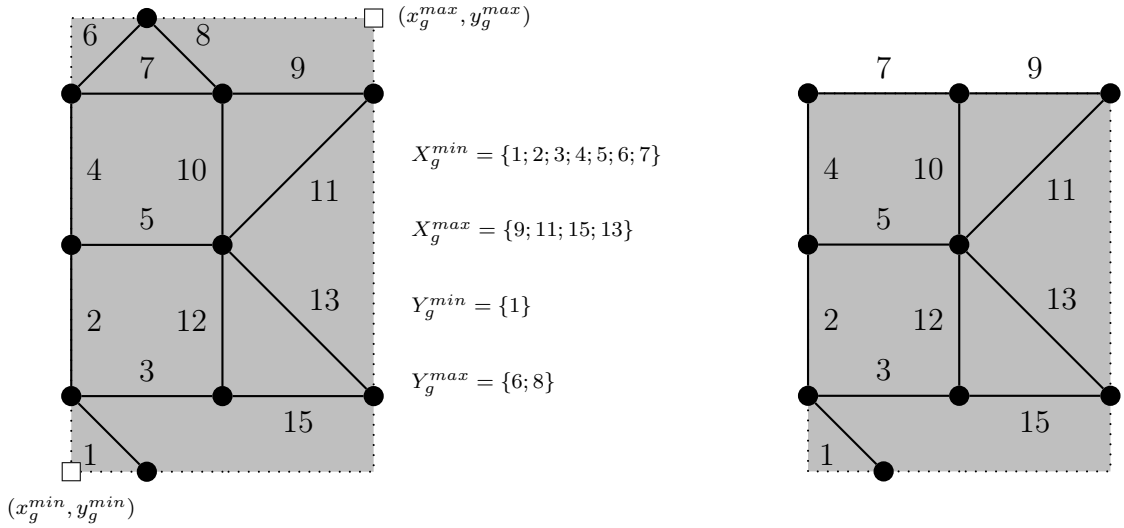
(e) $shift(5,\cdot)$ (f) $shift(6,1)$ (g) $swap(1,4)$ (h) $swap(5,6)$

Figure 6.15: Removing two nodes from $M(D_g)$

The operation $shift(4,1)$ reduces the number of odd nodes. In this case, the resulting subgraph is actually Eulerian. There is no further operations that reduces the number of odd nodes, this implies $REM(D_1) = \{shift(4,1)\}$.
After applying most of the possible operations the number of odd nodes stays unchanged, i.e., $REP(D_1) = \{shift(1,\cdot); shift(2,1); shift(5,\cdot); swap(1,4); swap(5,6)\}$.
Finally, the operation $shift(6,1)$ results in a matching graph consisting of four nodes. Hence, $shift(6,1) \in ADD(D_1)$.

6.3.3.3 Improve Local Compactness

The neighboring solutions for the *improve local compactness strategy* (LC) are chosen with the aim of reducing the sum of distances from all basic areas to the district center cen_g (cf. Section 6.2.2.5). The strategy defines the nb^{lct} districts with the largest local compactness as highly non-compact, and it only considers these. Note that a change of district D_g may imply a change of cen_g.
In general, the larger the distance of a basic area i to cen_g, the larger the improvement in local compactness resulting from the removal of i from that district. Let LD_g be the set of the lc^{nr} basic areas with the largest distances to cen_g. The elements of LD_g are candidates to be moved from a district D_g with a poor compactness value to another district. Furthermore, the operation $swap(i,j)$ improves the local compactness of $D_g = D_{(i)}$ only if $d(j, cen_g) < d(i, cen_g)$ holds. Hence, this strategy considers such an operation only if $i \in LD_g$. Usually, adding basic area i to district D_g causes a deterioration of the local



(a) $ER(D_g)$ (b) $ER(D_g \backslash Y_g^{max})$

Figure 6.16: Change of the smallest enclosing axis-parallel rectangle

compactness of D_g since the corresponding measure sums up all distances to cen_g. Thus, the strategy omits operations that move a basic area to a district with a poor compactness. CL contains all neighboring solutions resulting from the following feasible operations:

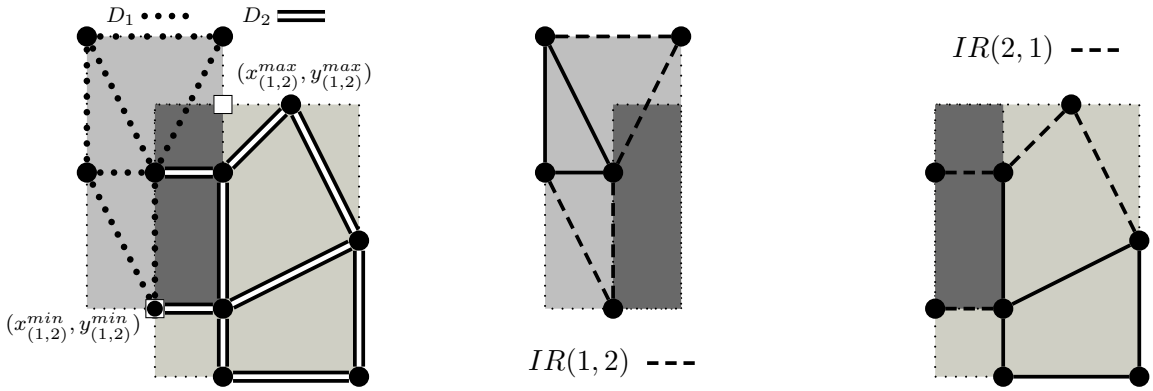
- $shift(i, h)$ if $D_g = D_{(i)}$ is highly non-compact and $i \in LD_g$;
- $swap(i, j)$ if $D_g = D_{(i)}$ is highly non-compact, $i \in LD_g$ and $d(j, cen_g) < d(i, cen_g)$;
- $swap(i, j)$ if $D_g = D_{(j)}$ is highly non-compact, $j \in LD_g$ and $d(i, cen_g) < d(j, cen_g)$.

6.3.3.4 Improve Global Compactness

Finally, the heuristic also uses a strategy to focus on solutions such that an improvement of the global compactness is expected, the *improve global compactness strategy* (GC) (cf. Section 6.2.2.5). Recall that $ER(D_g)$ is described by its two opposite vertices (x_g^{min}, y_g^{min}) and (x_g^{max}, y_g^{max}). For each bounding edge of $ER(D_g)$ there exists at least one edge of D_g having a point of contact with this edge. Therefore, the following sets are defined:

$$
\begin{aligned}
X_g^{min} &:= \left\{ i \;\middle|\; i \in B_g \text{ and } i \text{ has a point of contact with } x = x_g^{min} \right\}, \\
X_g^{max} &:= \left\{ i \;\middle|\; i \in B_g \text{ and } i \text{ has a point of contact with } x = x_g^{max} \right\}, \\
Y_g^{min} &:= \left\{ i \;\middle|\; i \in B_g \text{ and } i \text{ has a point of contact with } y = y_g^{min} \right\}, \\
Y_g^{max} &:= \left\{ i \;\middle|\; i \in B_g \text{ and } i \text{ has a point of contact with } y = y_g^{max} \right\}.
\end{aligned}
$$

Obviously, the corresponding bounding edge moves if and only if all edges of such a set are removed from D_g, and, hence, $ER(D_g)$ shrinks.



(a) Enclosing rectangles of D_1 and D_2 (b) Candidate edges of D_1 to reduce $ir(1,2)$ (c) Candidate edges of D_2 to reduce $ir(1,2)$

Figure 6.17: Area of intersection between two districts' enclosing rectangles

Example 6.3.6 Figure 6.16a illustrates the sets X_g^{min}, X_g^{max}, Y_g^{min}, and Y_g^{max}. Moreover, Figure 6.16b illustrates the result for removing all basic areas $i \in Y_g^{max}$ from D_g.

The intersection $ir(g,h)$ between $ER(D_g)$ and $ER(D_h)$ is again a rectangle. In order to reduce the area of intersection, one has to delete one of the sets that induce a bounding edge of the intersection rectangle from D_g or D_h. For example, if $x^{min}_{(g,h)}$ equals x^{min}_g and all edges of X^{min}_g are removed from D_g, then $area(ir(g,h))$ shrinks. $IR(g,h)$ denotes the set of all edges of D_g whose removal from D_g may result in a reduction of $area(ir(g,h))$. Note that $ir(g,h) = ir(g,h)$, but $IR(g,h) \neq IR(g,h)$.

Example 6.3.7 Figure 6.17a shows two overlapping enclosing rectangles. The corresponding sets $IR(1,2)$ and $IR(2,1)$ are illustrated in Figure 6.17b and 6.17c, respectively. Both sets contain four edges. Unfortunately, in both cases only removing one edge does not decrease the area of intersection.

Since the sets X_g^{min}, X_g^{max}, Y_g^{min}, and Y_g^{max} generally contain more than one edge, the execution of the operation *double-shift* increases the probability of shrinking $ER(D_g)$, and, hence, of $IR(g,h)$, $h \neq g$. As an operation shifting more than two basic areas at once may cause a considerable deterioration of the balance, this strategy does not include such operations. Moreover, a swap-operation deletes one basic area from a district, but adds another one, thus, the change of the enclosing rectangle is in general rather small. Finally, in order to compute solutions that improve global compactness, this strategy only incorporates the pairs (g,h) with the largest area of intersection. Let LA denote the corresponding set of

pairs. Then, CL contains all neighboring solutions resulting from the following operations:

- $shift(i, g)$ with $D_{(i)} = D_h$ if $i \in IR(h, g)$ and $(h, g) \in LA$;
- $double\text{-}shift(i, j, g)$ with $D_{(i)} = D_{(j)} = D_h$ if $i \in IR(h, g)$ and $(h, g) \in LA$.

6.3.4 Sub-Routines

All sub-routines are based on tabu search. Tabu search is a neighborhood based local search method proposed by Glover [11]. In order to escape from local minima and prevent cycling during the neighborhood search, a move is declared *tabu* for a number of iterations. The set of moves which are declared tabu define a *tabu list* (TL). Tabu search proceeds as follows: Starting from an initial solution, during each iteration the heuristic chooses the best non-tabu solution from the neighborhood of the current solution, even if this causes a deterioration of the current solution. As a result, the heuristic is able to escape from local minima. The heuristic ends if a given stopping criterion is fulfilled. In the following, the iterations of a sub-routine are called *sub-iterations*, whereas the top-level iterations of the whole algorithm are called *main-iterations*.

There is one sub-routine for each of the four strategies presented above. The notation of these sub-routines uses the same abbreviations as for the strategies, i.e., BL, DH, LC, and GC. In addition, there is one sub-routine that does not focus on a particular criterion, but contains all neighbors of the current solution based on shift- and swap-operations, that means $CL = NH_s(S)$. This sub-routine is called the *brute force sub-routine* (BF). Its usage is sometimes useful for escaping from extremal solutions generated by the algorithm after applying a particular sub-routine too often.

Starting from the current solution S and the corresponding tabu list TL, a sub-routine tries to find a better solution $S^L \in CL$. In order to evaluate the balance of a solution in CL correctly, the algorithm has to recompute the average working time μ (cf. Section 6.2.2.3). Moreover, for each involved district D_g its center cen_g has to be recomputed (cf. Section 6.2.2.5) in order to achieve the correct compactness evaluation. Since these computations are rather time consuming, the algorithm does not update these two values for each solution. Therefore, the obtained evaluations $bal_a(S)$ and $lc_a(S)$ are only approximations of the correct values $bal(S)$ and $lc(S)$. However, this approximation is in general sufficiently close to the true value (on average, the relative deviation is below 0.1%). Only at the end of an iteration of the sub-routine the algorithm updates μ and cen_g.

If a solution performs poorly for a specific criterion, the algorithm sometimes wants to put more emphasis on it. To this end, a special feature of this algorithm is the usage of local

weights w_1^L, w_2^L, w_3^L, w_4^L for each criterion in addition to the global user weights. The objective function with respect to these local weights is

$$F^L(S) := w_1^L \cdot bal(S) + w_2^L \cdot dh(S) + w_3^L \cdot lc(S) + w_4^L \cdot gc(S)\,,$$

and the objective function using the approximations is

$$F_a^L(S) := w_1^L \cdot bal_a(S) + w_2^L \cdot dh(S) + w_3^L \cdot lc_a(S) + w_4^L \cdot gc(S)\,.$$

More details on the local weights and their motivation will be provided in Section 6.3.5.

During each sub-iteration the algorithm chooses the neighboring solution S with the best value according to F_a^L. For this solution, it then determines $F^L(S)$ and tests whether the solution improves upon the currently best solution S^L with respect to F^L. It also checks whether the solution improves the objective value F of the currently best solution S^* with respect to the global weights. Furthermore, it updates the set of locally Pareto solutions PS as follows. If a solution $S' \in PS$ dominates S, S can be discarded. Otherwise, the algorithm adds S to PS and deletes all solutions from PS that are dominated by S. A sub-routine stops if there are $maxIT$ successive sub-iterations without an improvement of F^L. The best solution S^L with respect to F^L and the corresponding tabu list are the result of the sub-routine and the initial starting point for the next sub-routine.
Algorithm 6.3.2 provides a formal description of this procedure.

Algorithm 6.3.2: Outline of a Sub-Routine

Input: *PS, S^*, S, TL, local weights $w_1^L, \ldots, w_4^L$.*

Output: *A feasible districting plan S.*

Parameter: *An iteration limit maxIt.*

1 *Set $S^L = S$, $TL^L = TL$, and $it = 0$.*

2 **while** $it < maxIt$ **do**

Determine CL and compute $S = \arg\min_{S' \in CL} F_a^L(S')$.

Update TL.

if $F^L(S) < F^L(S^L)$ `// update` μ `and` cen_g
then *set $S^L = S$, $TL^L = TL$, and $it = 0$*
else *Set $it = it + 1$.*

Update PS with S.

if $F(S) < F(S^*)$ **then** *Set $S^* = S$.*

end

3 **return** *PS, S^*, S, and TL.*

6.3.5 Local Weights

The main goal of this heuristic is the determination of a solution minimizing F, where F is computed using the global user-defined weights $w_1, \ldots, w_4$; see Equation (6.8). In order to escape from local minima of F or to put temporarily a higher or lower emphasis on a certain criterion during the execution of the heuristic, it can be useful to change the values of these weights. To this end, the *local weights* $w_1^L, \ldots, w_4^L$ are introduced in Section 6.3.4. Generally, a large local weight w_r^L means that an improvement with respect to the r^{th} soft criterion is desired. In addition, in order to use the local weights in the evaluation of a solution during a sub-iteration, these weights are used to calculate the probabilities of applying the different sub-routines; see Section 6.3.6 for details.

Since the local weights are used to focus on certain criteria, they are updated at the end of each main-iteration. The update of a single local weight depends on its corresponding global weight as well as on the variation of the evaluation of its corresponding criterion between the current solution S and the solution S' of the previous main-iteration. If the evaluation of the criterion for S is worse than for S', the local weight increases, and vice versa. More precisely, the algorithm applies the following update rule:

$$w_i^L := w_i^L \cdot \begin{cases} 1 - w_i \cdot \min\left\{1,\ 10 \cdot \frac{MF^i(S') - MF^i(S)}{MF^i(S')}\right\} & \text{if } MF^i(S) < MF^i(S') \\ 1 & \text{if } MF^i(S) = MF^i(S') \\ 1 + w_i \cdot \min\left\{1,\ 10 \cdot \frac{MF^i(S) - MF^i(S')}{MF^i(S')}\right\} & \text{if } MF^i(S) > MF^i(S') \end{cases} \tag{6.9}$$

Hence, for a relative improvement (deterioration) of up to 10% in the value of the criterion between S and S', the change is proportional to the global weight and to the relative improvement (deterioration). For a larger variation the change is only proportional to the global weight.

6.3.6 Sub-Routine Selection

In each main-iteration the algorithm randomly selects one of the five sub-routines. The probability $p(BF)$ of selecting the brute force sub-routine is fixed here by the parameter p_{BF}. The probability of selecting another sub-routine is proportional to the corresponding local weight. For example, the probability for the *improve balance* sub-routine is determined as

$$p(BL) := (1 - p_{BF}) \cdot \frac{w_1^L}{w_1^L + w_2^L + w_3^L + w_4^L}\,. \tag{6.10}$$

The probabilities $p(DH)$, $p(LC)$, and $p(GC)$ are defined analogously. In general, the higher the value of w_r^L is, the higher the probability of selecting the corresponding sub-routine. If a sub-routine was executed without finding a better solution, the sub-routine is declared *inactive* until another sub-routine can improve the solution. An inactive sub-routine may not be selected, except for the brute force sub-routine which is never inactive. Let IS denote the set of inactive sub-routines. Strongly spoken, declaring a sub-routine inactive changes the probabilities according to Equation (6.10). Technically, the heuristic repeats the selection step until a sub-routine is chosen that is not inactive.

Although balance is only a soft criterion, it is often desired to avoid highly unbalanced solutions. Thus, if the balance of a district exceeds a user-given threshold bal^{inf}, the Improve Balance sub-routine is always chosen, unless it is inactive. Note that for $bal^{inf} = \infty$ this rule is invalid. Exceeding bal^{inf} especially occurs after the construction phase since the applied post-processing usually causes a deterioration of the balance.
Algorithm 6.3.3 summarizes the selection step.

Algorithm 6.3.3: Selection of Sub-Routines

Input: *A feasible districting plan S, local weights $w_1^L, \ldots, w_4^L$, a set of inactive sub-routines IS.*
Output: *A sub-routine R.*

1 **if** $(\max_{g=1,\ldots,p} bal(D_g) > bal^{inf})$ *and* $BL \notin IS$ **then** *Select* $R = BL$.
else
  repeat
    foreach $R \in \{BL; DH; LC; GC\}$ **do** *Calculate* $p(R)$.
    Draw a random variable uniformly distributed over $[0, 1]$ *and select the corresponding sub-routine* $R \in \{BL; DH; LC; GC; BF\}$.
  until $R \notin IS$
end
2 **return** R.

6.3.7 Overview: Improvement Heuristic

This subsection provides an overview over the complete improvement heuristic; see also Algorithm 6.3.4. The improvement phase starts with an initial solution S. During each main-iteration, it randomly selects a sub-routine (Section 6.3.6) depending on the current local weights (Section 6.3.5), and it applies this sub-routine to the current solution S using the current tabu list TL. After the execution of a sub-routine, the heuristic updates the local weights depending on the variations of the evaluations of the four soft criteria. After

that, the heuristic checks whether the executed sub-routine has found a better solution with respect to F or whether the set of locally Pareto solutions has changed. If neither of these two events occurs for $maxMIt$ consecutive main-iterations, the algorithm stops.

Algorithm 6.3.4: Outline of the Improvement Phase

Input: *A feasible districting plan S, global weights $w_1, \ldots, w_4$.*

Output: *A feasible districting plan S^*, the set of locally Pareto solutions PS.*

Parameter: *An iteration limit $maxMIt$.*

1 *Set $TL = \emptyset$, $S^* = S$, $PS = \{S^*\}$, $IS = \emptyset$, $it = 0$, and $w_i^L = w_i$ for $i = 1, \ldots, 4$.*

2 **while** *$it < maxMIt$* **do**

Randomly select a sub-routine R. `// Algorithm 6.3.3`

Execute R using PS, S^*, S, TL, and $w_1^L, \ldots, w_4^L$. `// Algorithm 6.3.2`

Update the local weights $w_1^L, \ldots, w_4^L$. `// Equation` (6.9)

if *(S has improved or PS has changed)* **then**
Set $it = 0$ and $IS = \emptyset$
else
Set $it = it + 1$ and $IS = IS \cup \{R\}$ if $R \neq BF$.
end

3 **return** *S^*, PS.*

6.4 Computational Results

This Section presents the results of our computational tests. The algorithm was coded in C++ and executed on a PC running Windows 7 with a Pentium(R) Dual-Core E5500 processor with 2.80 GHz and 2 GB RAM. The calculation of a minimal cost perfect matching is based on the blossom algorithm. Our implementation uses the blossom implementation of Kolmogorov [17].

The tests are conducted on 24 different problem instances based on the German road network generated as follows. Using the ArcView GIS[1], all streets within a rectangular area were selected and converted into a street graph, storing for each street the list of connected line segments representing it. The instances differ in the number of streets, the aspect ratio of the rectangle, and in whether they are situated in an urban (U) or in a rural (R) area. The instances are labelled $U*$ and $R*$, respectively, where $*$ is the number of streets of the graph. The deadheading times are taken to be proportional to the length of the streets, and the service time of each street is a random multiple of the deadheading time.

The number p of districts ranges from four to eight for instances with fewer than 400 basic areas. For 400 up to 600 basic areas five, seven, eight and ten districts are considered. Finally, for more than 600 basic areas six, eight, ten and twelve districts are considered. With respect to the other parameters, $bal^{max} = 0.1$, $bal^{inf} = 0.1$, $nb^{bal} = nb^{dh} = nb^{lct} = \lfloor p/2 \rfloor$, $nb^{gc} = p$, $lc^{nr} = 15$, $p_{BF} = 0.2$, $maxIT = 25$, $maxMIt = 5$ is used. Since the algorithm contains a random component, each instance is solved 10 times. The results present only the average over these 10 runs. The percentage of the standard deviation from the mean objective function value is just 6.26% on average (with a maximum of 20.4% for one instance).

6.4.1 Soft Criteria

The first test assesses the relevance of each criterion for obtaining a good overall solution. To this end, it computes for each instance a solution for which the objective function contains only one of the criteria, i.e., the user-given weight for the chosen criterion is set to one and all other weights are zero. In the following, let $S(BL)$, $S(DH)$, $S(LC)$, and $S(GC)$ denote the respective single criterion solutions for balance, deadheading, local compactness, and global compactness. Afterwards, the test evaluates each single criterion solution with respect to the other three criteria as well. Concerning the balance, it uses $bal_{max}(S)$ (cf. Equation (6.3)) instead of $bal(S)$ (cf. Equation (6.4)) since it is easier to interpret this result. After that, this test computes for each solution $S(\cdot)$ and each criterion the absolute difference in the

[1] ESRI®, www.esri.com

value and the relative percentage deviation with respect to the value of the corresponding single criterion solution. For example, for each instance it computes for $S(BL)$ the difference and relative deviation of $lc(S(BL))$ to $lc(S(LC))$ (cf. Equation (6.6)) and of $gc(S(BL))$ to $gc(S(LC))$ (cf. Equation (6.7)). There is one exception: Concerning the deadheading times, it computes the difference of $dh(S(BL))$ to $dh(S(DH))$ (cf. Equation (6.5)), but the relative percentage deviation of $DH(S(BL))$ to $DH(S(DH))$ (cf. Equation (6.2)) since there are some instances where $dh(S(DH)) = 0$. In other words, the absolute deadheading times are used in order to determine the relative percentage deviation.

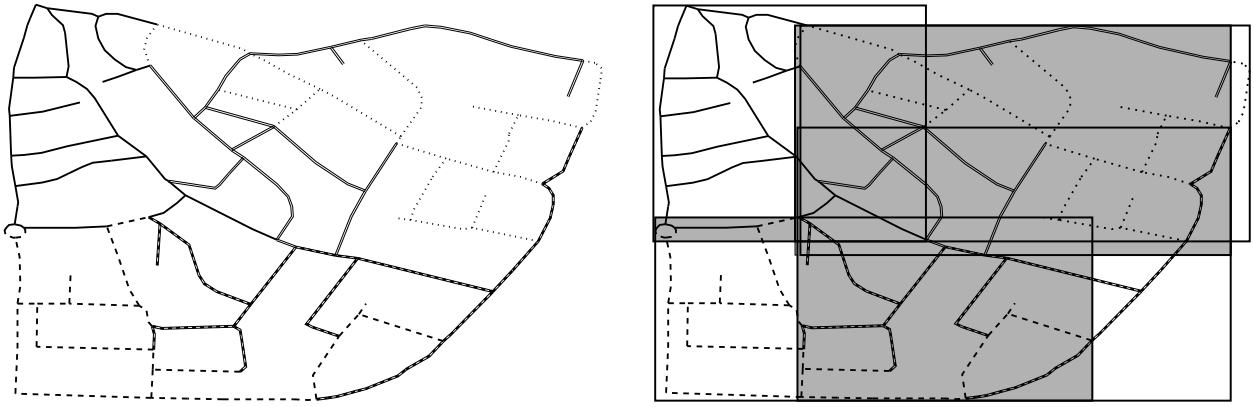
	balance	deadheading	local comp.	global comp.
$S(BL)$	–	0.477 (46%)	0.022 (45%)	0.631 (682%)
$S(DH)$	0.198 (41180%)	–	0.022 (50%)	0.340 (369%)
$S(LC)$	0.153 (39416%)	0.413 (39%)	–	0.375 (414%)
$S(GC)$	0.463 (115790%)	0.283 (27%)	0.041 (82%)	–

Table 6.3: Average absolute values and relative deviations (in brackets) between the single criterion solutions

Table 6.3 shows the results, which are the averages over the 24 problem instances and the different values of p. Taking row one for example, the deadheading costs of the solution $S(BL)$ are 0.477 larger than the deadheading costs of $S(DH)$, i.e., $dh(S(BL)) - dh(S(DH)) = 0.477$. This corresponds to an increase in the costs of 46%.

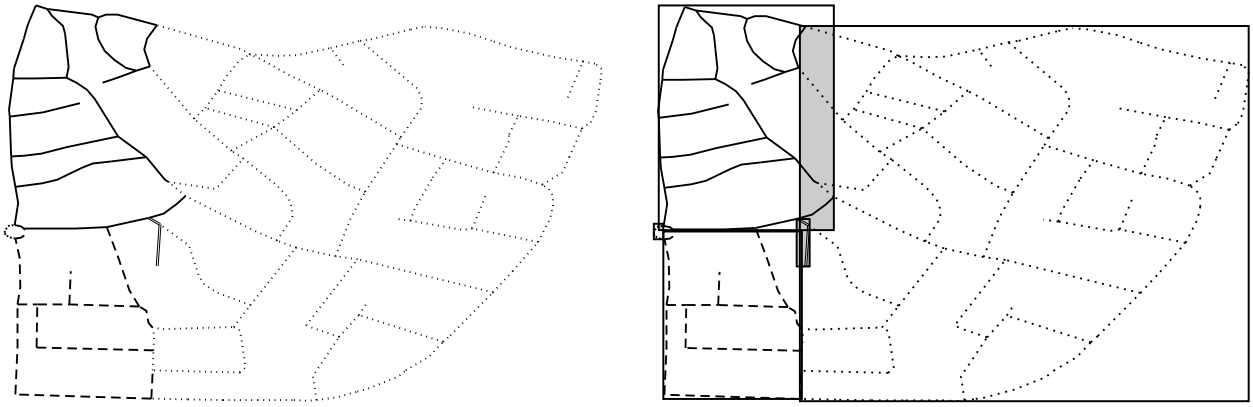
The results exhibit huge balance deviations. The reason is that the solutions $S(BL)$ are nearly perfectly balanced, with an average balance of only 0.4%. Note that a balance of 0.4% corresponds to $bal(\cdot) = 0.004$, consequently a balance of 5%, i.e., $bal(\cdot) = 0.05$, has a percentage deviation of 1150%. Unfortunately, perfectly balanced solutions usually have very high deadheading costs and induce larger overlaps of districts. Figuratively spoken, all service persons work approximately the same, but very much. Hence, the achieved solutions are not sufficient from an economic point of view. Moreover, the areas of responsibility are not separated well. Namely, the average value of the global compactness criterion for these solutions is 0.751, i.e., the total area of intersection between the districts is about three quarter of the whole area.

Figure 6.18a depicts an example of an almost perfectly balanced solution with five districts for the instance $U132$. The evaluations are $bal_{max} = 0.0008$, $dh = 0.88$, $lc = 0.103$ and $gc = 1.491$, while the corresponding values of the single criteria solutions are $dh^* = 0.03$, $lc^* = 0.100$ $gc^* = 0.05$. Especially in the north-east, the district illustrated by dotted lines



(a) Solution (b) Enclosing rectangles

Figure 6.18: Optimizing balance (U132 with $p = 5$)



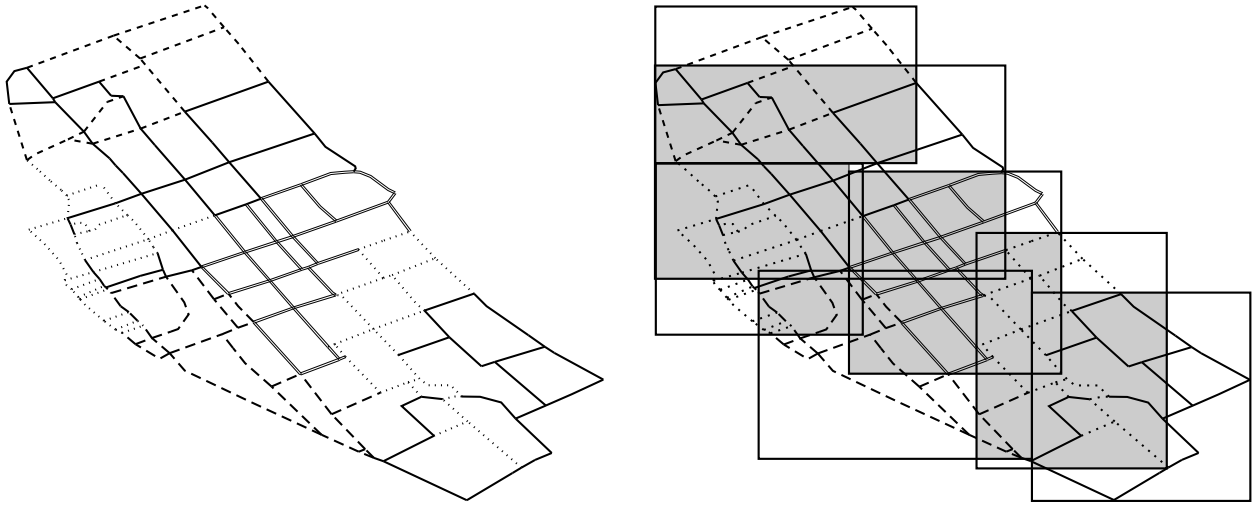
(a) Solution (b) Enclosing rectangles

Figure 6.19: Optimizing global compactness (U132 with $p = 5$)

and the district illustrated by double-lines are more or less totally overlapped. Figure 6.18b illustrates the corresponding enclosing rectangles.

Considering global compactness, the districts of the solutions $S(GC)$ are well separated with an average value of 0.12 for this criterion. Unfortunately, the balance is often way off. The average balance of the solutions considered in this test is 48%. Figure 6.19a depicts an example for a globally compact solution (instance $U132$ with $p = 5$) with $gc = 0.05$. Figure 6.19b illustrates the corresponding enclosing rectangles. The overlap is almost zero, but the districts are highly unbalanced ($bal_{max} = 1.598$). Especially the very small district located in the west consists of only one street illustrated as dotted line. In contrast to this, the large district located in the east contains more than half of the total demand. The further evaluations are $dh = 0.07$ and $lc = 0.323$.

Another conflict exists between the goals of minimizing deadheading times and maximizing global compactness. A solution with small deadheading times usually consists of districts



(a) Solution (b) Enclosing rectangle

Figure 6.20: Optimizing deadheading (U212 with $p = 7$)

whose sub-graphs are nearly Eulerian. Hence, the algorithm tries to generate districts consisting of cycles. These cycles are often interwoven with cycles of other districts, and, thus, these solutions are unsatisfactory with respect to global compactness. Figure 6.20a shows this effect on the instance U212 with $p = 7$ resulting in $gc = 0.60$. Figure 6.20b depicts the corresponding enclosing rectangles.

Finally, the test points out similar observations for local compactness. Although the conflicts are less pronounced, a local compact solution has some weaknesses in terms of the other criteria.

Summarizing the results of this test, the various criteria pursue conflicting objectives and none is fulfilled implicitly by another one. Therefore, the next tests focus on solutions that try to achieve a good compromise between these conflicting criteria.

6.4.2 Equally Weighted Solutions

This set of tests starts with a test where the four soft criteria have the same weight, i.e., $w_1 = w_2 = w_3 = w_4 = 0.25$. Table 6.4 shows an extract of the results. A complete list of the results can be found in the online appendix of Butsch et al. [3]. For each criterion, Table 6.4 firstly presents the absolute values and secondly it compares these values to those of the corresponding single criterion solutions. For example, for the instance $U132$ with $p = 5$ the balance of the equally weighted solution is 0.110 larger than $bal_{max}(S(BL))$, which corresponds to a relative increase of 14093%. In contrast to this, the deadheading costs of the equally weighted solution are identical to those of $S(DH)$.

		Absolute values				Deviations to the single criterion solutions			
	p	BL	DH	LC	GC	BL	DH	LC	GC
$U132$	5	0.111	0.030	0.093	0.194	0.110 (14093%)	0.000 (0%)	0.020 (28%)	0.145 (290%)
$U132$	6	0.136	0.142	0.079	0.299	0.133 (4966%)	0.142 (14%)	0.022 (39%)	0.217 (262%)
$U132$	7	0.130	0.112	0.057	0.112	0.127 (4048%)	0.077 (7%)	0.015 (37%)	0.037 (50%)
$U212$	5	0.067	0.163	0.094	0.260	0.067 (21512%)	0.163 (16%)	0.021 (28%)	0.175 (206%)
$U212$	6	0.048	0.111	0.068	0.256	0.047 (6974%)	0.059 (6%)	0.019 (38%)	0.142 (125%)
$U212$	7	0.036	0.181	0.060	0.303	0.035 (3547%)	0.144 (14%)	0.022 (56%)	0.120 (66%)
$U448$	5	0.045	0.082	0.091	0.325	0.045 (28008%)	0.082 (8%)	0.021 (30%)	0.245 (307%)
$U448$	7	0.052	0.167	0.053	0.425	0.051 (22548%)	0.167 (17%)	0.013 (32%)	0.293 (223%)
$U448$	8	0.093	0.211	0.050	0.415	0.093 (16191%)	0.179 (17%)	0.014 (38%)	0.203 (95%)
$U627$	6	0.108	0.125	0.078	0.407	0.108 (63106%)	0.104 (10%)	0.009 (13%)	0.298 (272%)
$U627$	8	0.118	0.243	0.054	0.346	0.118 (25434%)	0.144 (13%)	0.010 (22%)	0.124 (56%)
$U627$	10	0.182	0.182	0.034	0.426	0.182 (22412%)	0.180 (18%)	0.002 (5%)	0.197 (86%)
$R412$	5	0.091	0.223	0.037	0.114	0.091 (37353%)	0.199 (19%)	0.001 (3%)	0.069 (155%)
$R412$	7	0.076	0.133	0.024	0.087	0.076 (21195%)	0.117 (11%)	0.002 (10%)	0.037 (72%)
$R412$	8	0.266	0.346	0.023	0.072	0.266 (61445%)	0.346 (35%)	0.008 (52%)	0.041 (134%)

Table 6.4: Results for equally weighted criteria

The results show that equally weighted solutions constitute a good compromise between deadheading times and local compactness. In terms of global compactness the relative deviations are much larger, but still acceptable. Unfortunately, in terms of balance, already the absolute values are unsatisfactory. Balance is usually a very important criterion and deviations of up to 27% from the mean district size are not acceptable in many applications. Often, it is desired that the maximum deviation, i.e., the balance, is at most than 10%.

6.4.3 Increasing Balance Weight

The next test discusses the effect on balance when the user weight w_1 increases, while keeping the other criteria equally weighted. Table 6.5 shows some results, which are again only an extract of the conducted experiments.

As expected, balance improves with increasing weights w_1. Already for $w_1 = 0.4$ the solutions are balanced, i.e., $bal_{max} \leq 0.10$, expect for $U627$ with $p = 10$ and for $R412$ with $p = 8$. Fortunately, the deterioration with respect to the other three criteria is rather moderate. Table 6.6 presents the corresponding relative deviations.

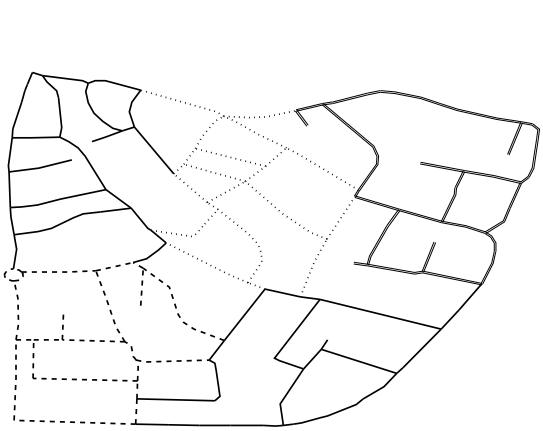
Figure 6.21 illustrates two respective solutions exemplarily, one for an urban area and one for a rural area.

	p	BL	DH	LC	GC	BL	DH	LC	GC	BL	DH	LC	GC
weight		0.4	0.2	0.2	0.2	0.5	0.17	0.17	0.17	0.57	0.14	0.14	0.14
$U132$	5	**0.040**	0.061	0.095	0.249	**0.035**	0.048	0.092	0.288	**0.029**	0.067	0.093	0.274
$U132$	6	**0.049**	0.255	0.069	0.343	**0.018**	0.379	0.071	0.429	**0.022**	0.373	0.068	0.421
$U132$	7	**0.048**	0.195	0.053	0.176	**0.049**	0.195	0.053	0.176	**0.026**	0.238	0.051	0.271
$U212$	5	**0.027**	0.136	0.097	0.301	**0.015**	0.187	0.098	0.298	**0.015**	0.216	0.098	0.281
$U212$	6	**0.023**	0.170	0.073	0.246	**0.021**	0.174	0.074	0.260	**0.010**	0.251	0.072	0.270
$U212$	7	**0.033**	0.169	0.054	0.300	**0.021**	0.179	0.054	0.328	**0.013**	0.285	0.055	0.334
$U448$	5	**0.016**	0.121	0.093	0.344	**0.008**	0.137	0.093	0.336	**0.004**	0.199	0.091	0.374
$U448$	7	**0.031**	0.220	0.054	0.337	**0.020**	0.230	0.056	0.373	**0.008**	0.264	0.056	0.370
$U448$	8	**0.035**	0.263	0.045	0.318	**0.031**	0.302	0.048	0.363	**0.015**	0.301	0.045	0.330
$U627$	6	**0.073**	0.147	0.075	0.407	**0.031**	0.115	0.073	0.480	**0.071**	0.144	0.078	0.588
$U627$	8	**0.015**	0.253	0.049	0.430	**0.016**	0.249	0.049	0.425	**0.005**	0.304	0.050	0.446
$U627$	10	**0.147**	0.198	0.049	0.419	**0.101**	0.211	0.048	0.467	**0.010**	0.309	0.042	0.535
$R412$	5	**0.021**	0.246	0.056	0.113	**0.079**	0.170	0.038	0.149	**0.018**	0.264	0.053	0.114
$R412$	7	**0.023**	0.167	0.028	0.108	**0.009**	0.202	0.028	0.103	**0.018**	0.202	0.027	0.106
$R412$	8	**0.184**	0.331	0.023	0.094	**0.184**	0.325	0.023	0.093	**0.179**	0.315	0.024	0.124

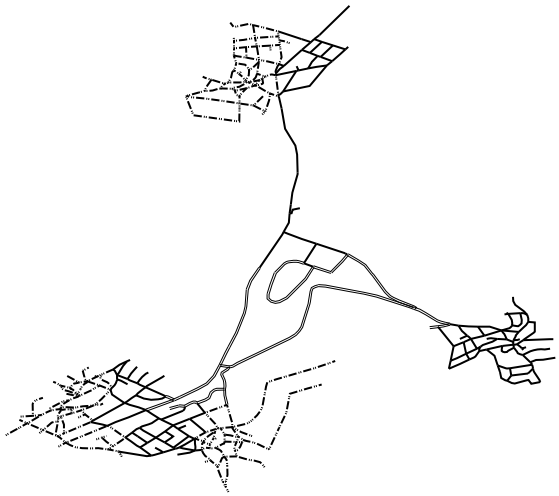
Table 6.5: Results for increasing the user weight for balance

	p	BL	DH	LC	GC	BL	DH	LC	GC	BL	DH	LC	GC
weight		0.4	0.2	0.2	0.2	0.5	0.17	0.17	0.17	0.57	0.14	0.14	0.14
$U132$	5	**64**%	−3%	−1%	−28%	**69**%	−2%	1%	−48%	**74**%	−4%	0%	−41%
$U132$	6	**64**%	−10%	12%	−15%	**87**%	−21%	10%	−43%	**84**%	−20%	14%	−41%
$U132$	7	**63**%	−7%	8%	−56%	**62**%	−7%	8%	−56%	**80**%	−11%	10%	−141%
$U212$	5	**60**%	2%	−3%	−16%	**78**%	−2%	−4%	−14%	**77**%	−5%	−4%	−8%
$U212$	6	**51**%	−5%	−7%	4%	**55**%	−6%	−9%	−2%	**78**%	−13%	−6%	−5%
$U212$	7	**7**%	1%	10%	1%	**42**%	0%	10%	−8%	**63**%	−9%	8%	−10%
$U448$	5	**63**%	−4%	−2%	−6%	**82**%	−5%	−1%	−3%	**91**%	−11%	1%	−15%
$U448$	7	**41**%	−5%	−1%	21%	**61**%	−5%	−6%	12%	**84**%	−8%	−5%	13%
$U448$	8	**63**%	−4%	9%	23%	**66**%	−7%	4%	13%	**83**%	−7%	9%	21%
$U627$	6	**33**%	−2%	3%	0%	**71**%	1%	6%	−18%	**34**%	−2%	0%	−44%
$U627$	8	**88**%	−1%	10%	−24%	**86**%	0%	9%	−23%	**95**%	−5%	**7**%	−29%
$U627$	10	**19**%	−1%	−46%	2%	**45**%	−3%	−43%	−10%	**94**%	−11%	−25%	−26%
$R412$	5	**77**%	−2%	−51%	1%	**13**%	4%	−1%	−31%	**80**%	−3%	−43%	−1%
$R412$	7	**70**%	−3%	−18%	−24%	**88**%	−6%	−18%	−19%	**77**%	−6%	−13%	−22%
$R412$	8	**6**%	−1%	1%	3%	**6**%	0%	0%	4%	**8**%	1%	−6%	−28%

Table 6.6: Results for increasing the user weight for balance



(a) U132 with five districts



(b) R412 with seven districts

Figure 6.21: Two solutions for $w_1 = 0.4$

6.4.4 Varying Weights

Analogously to balance, the next tests addresses the effects of increasing the weights for the other three criteria. Table 6.7 presents a summary of these results. It states average results over all 24 instances.

weights				deviations			
BL	DH	LC	GC	BL	DH	LC	GC
0.40	0.20	0.20	0.20	−50%	2%	−1%	11%
0.50	0.17	0.17	0.17	−66%	5%	0%	19%
0.57	0.14	0.14	0.14	−76%	6%	1%	27%
0.20	0.40	0.20	0.20	72%	−4%	1%	9%
0.17	0.50	0.17	0.17	81%	−6%	2%	18%
0.14	0.57	0.14	0.14	91%	−7%	3%	21%
0.20	0.20	0.40	0.20	4%	−3%	−7%	−1%
0.17	0.17	0.50	0.17	5%	−3%	−9%	−1%
0.14	0.14	0.57	0.14	9%	−2%	−10%	0%
0.20	0.20	0.20	0.40	104%	3%	0%	−19%
0.17	0.17	0.17	0.50	176%	4%	1%	−27%
0.14	0.14	0.14	0.57	228%	5%	4%	−32%

Table 6.7: Average deviations to the equally weighted solutions

For example, the first row ($w_1 = 0.4, w_2 = w_3 = w_4 = 0.2$) describes that the balance (local compactness) of these solutions is on average 50% (1%) smaller than the balance (local

compactness) of the corresponding equally weighted solution. With respect to deadheading and global compactness, these solutions are on average 2% and 11% worse than the equally weighted ones. As already described before, if the weight of balance increases, the solution becomes better with respect to the balance. The same effect occurs for the other criteria, although the percentage improvements are smaller. Increasing the weight for local compactness shows surprising effects. On average the deadheading times and the global compactness slightly improves as well. Unfortunately, the equally weighted solutions are already unsatisfactory in terms of balance and the balance deteriorates again. However, altogether, the user-given weights work as expected.

In addition, Table 6.8 shows extracts of the results for deadheading, Table 6.9 for local compactness, and finally Table 6.10 for global compactness in more detail. The online appendix of Butsch et al. [3] presents the complete results.

	p	BL	DH	LC	GC	BL	DH	LC	GC	BL	DH	LC	GC
weight		0.4	0.2	0.2	0.2	0.5	0.17	0.17	0.17	0.57	0.14	0.14	0.14
*U*132	5	0.053	**0.090**	0.096	0.241	0.096	**0.073**	0.097	0.239	0.074	**0.030**	0.093	0.239
*U*132	6	0.106	**0.078**	0.071	0.325	0.149	**0.075**	0.079	0.317	0.184	**0.046**	0.083	0.315
*U*132	7	0.160	**0.050**	0.071	0.127	0.197	**0.042**	0.070	0.138	0.161	**0.048**	0.071	0.130
*U*212	5	0.072	**0.080**	0.097	0.280	0.085	**0.050**	0.098	0.331	0.095	**0.039**	0.095	0.379
*U*212	6	0.081	**0.077**	0.070	0.282	0.059	**0.073**	0.075	0.270	0.076	**0.056**	0.062	0.249
*U*212	7	0.056	**0.083**	0.055	0.308	0.067	**0.078**	0.058	0.329	0.085	**0.077**	0.059	0.338
*U*448	5	0.025	**0.088**	0.094	0.358	0.027	**0.076**	0.091	0.377	0.050	**0.072**	0.095	0.353
*U*448	7	0.071	**0.109**	0.059	0.344	0.051	**0.067**	0.056	0.337	0.030	**0.012**	0.056	0.352
*U*448	8	0.074	**0.134**	0.047	0.366	0.066	**0.106**	0.046	0.356	0.078	**0.087**	0.047	0.376
*U*627	6	0.025	**0.085**	0.071	0.410	0.043	**0.036**	0.072	0.485	0.071	**0.028**	0.077	0.528
*U*627	8	0.139	**0.192**	0.053	0.363	0.134	**0.161**	0.054	0.403	0.183	**0.160**	0.056	0.394
*U*627	10	0.135	**0.158**	0.049	0.467	0.131	**0.157**	0.048	0.540	0.064	**0.171**	0.044	0.577
*R*412	5	0.160	**0.026**	0.037	0.083	0.162	**0.026**	0.037	0.083	0.148	**0.025**	0.037	0.085
*R*412	7	0.085	**0.061**	0.025	0.082	0.083	**0.059**	0.024	0.083	0.083	**0.060**	0.025	0.087
*R*412	8	0.155	**0.190**	0.022	0.142	0.197	**0.162**	0.021	0.150	0.205	**0.166**	0.021	0.147

Table 6.8: Results for increasing the user weight for deadheading

	p	BL	DH	LC	GC	BL	DH	LC	GC	BL	DH	LC	GC
weight		0.4	0.2	0.2	0.2	0.5	0.17	0.17	0.17	0.57	0.14	0.14	0.14
$U132$	5	0.079	0.049	**0.092**	0.204	0.090	0.068	**0.089**	0.197	0.084	0.071	**0.087**	0.236
$U132$	6	0.082	0.239	**0.065**	0.280	0.067	0.229	**0.064**	0.315	0.066	0.246	**0.062**	0.185
$U132$	7	0.088	0.247	**0.049**	0.179	0.095	0.233	**0.050**	0.184	0.089	0.235	**0.049**	0.192
$U212$	5	0.059	0.061	**0.087**	0.204	0.029	0.062	**0.088**	0.243	0.038	0.079	**0.087**	0.235
$U212$	6	0.020	0.074	**0.061**	0.239	0.027	0.088	**0.061**	0.228	0.037	0.088	**0.058**	0.251
$U212$	7	0.034	0.064	**0.050**	0.205	0.045	0.056	**0.048**	0.204	0.047	0.059	**0.048**	0.201
$U448$	5	0.017	0.050	**0.083**	0.297	0.022	0.046	**0.080**	0.298	0.023	0.052	**0.079**	0.288
$U448$	7	0.033	0.065	**0.051**	0.256	0.032	0.062	**0.052**	0.270	0.039	0.055	**0.051**	0.266
$U448$	8	0.031	0.106	**0.043**	0.319	0.028	0.074	**0.044**	0.326	0.038	0.074	**0.043**	0.328
$U627$	6	0.011	0.062	**0.070**	0.237	0.011	0.064	**0.070**	0.240	0.014	0.064	**0.070**	0.242
$U627$	8	0.030	0.163	**0.050**	0.378	0.032	0.157	**0.049**	0.379	0.028	0.188	**0.046**	0.365
$U627$	10	0.181	0.192	**0.034**	0.394	0.182	0.198	**0.035**	0.406	0.100	0.201	**0.034**	0.428
$R412$	5	0.161	0.038	**0.037**	0.071	0.139	0.071	**0.037**	0.095	0.126	0.089	**0.037**	0.125
$R412$	7	0.107	0.082	**0.024**	0.080	0.081	0.122	**0.024**	0.083	0.073	0.135	**0.024**	0.096
$R412$	8	0.086	0.031	**0.018**	0.131	0.093	0.049	**0.018**	0.157	0.084	0.063	**0.018**	0.171

Table 6.9: Results for increasing the user weight for local compactness

	p	BL	DH	LC	GC	BL	DH	LC	GC	BL	DH	LC	GC
weight		0.4	0.2	0.2	0.2	0.5	0.17	0.17	0.17	0.57	0.14	0.14	0.14
$U132$	5	0.092	0.129	0.096	**0.210**	0.090	0.053	0.090	**0.179**	0.123	0.177	0.098	**0.170**
$U132$	6	0.147	0.157	0.073	**0.144**	0.293	0.213	0.076	**0.108**	0.312	0.294	0.082	**0.106**
$U132$	7	0.197	0.112	0.063	**0.114**	0.228	0.118	0.065	**0.113**	0.303	0.269	0.065	**0.112**
$U212$	5	0.078	0.161	0.099	**0.250**	0.158	0.226	0.112	**0.215**	0.124	0.261	0.099	**0.201**
$U212$	6	0.117	0.196	0.068	**0.222**	0.175	0.225	0.072	**0.227**	0.226	0.339	0.070	**0.192**
$U212$	7	0.083	0.265	0.064	**0.265**	0.158	0.273	0.065	**0.222**	0.197	0.263	0.066	**0.208**
$U448$	5	0.042	0.182	0.096	**0.317**	0.066	0.190	0.095	**0.296**	0.066	0.190	0.101	**0.283**
$U448$	7	0.042	0.088	0.053	**0.192**	0.081	0.107	0.053	**0.161**	0.086	0.156	0.053	**0.145**
$U448$	8	0.089	0.329	0.046	**0.297**	0.176	0.312	0.045	**0.255**	0.208	0.333	0.046	**0.235**
$U627$	6	0.121	0.127	0.075	**0.410**	0.120	0.181	0.073	**0.377**	0.106	0.164	0.073	**0.356**
$U627$	8	0.193	0.262	0.058	**0.282**	0.270	0.268	0.059	**0.251**	0.275	0.268	0.062	**0.241**
$U627$	10	0.180	0.213	0.034	**0.395**	0.156	0.216	0.038	**0.363**	0.154	0.208	0.039	**0.349**
$R412$	5	0.071	0.231	0.041	**0.091**	0.084	0.232	0.040	**0.083**	0.052	0.251	0.053	**0.102**
$R412$	7	0.088	0.120	0.024	**0.073**	0.079	0.139	0.024	**0.072**	0.079	0.139	0.024	**0.072**
$R412$	8	0.246	0.289	0.025	**0.104**	0.252	0.360	0.023	**0.065**	0.259	0.347	0.027	**0.072**

Table 6.10: Results for increasing the user weight for global compactness

6.4.5 Running Times

Finally, this subsection takes a look at the running times of the algorithm. Table 6.11 contains the average running times in seconds for solving the 24 equally weighted problem instances. Since there is no clear trend in running times with respect to p, the results are also averaged over p. Moreover, the tests do not show any significant differences in running times with respect to different user weights.

Instance	$U132$	$U137$	$U147$	$U212$	$U264$	$U268$	$U269$	$U274$
seconds	13	18	25	54	76	94	102	145

Instance	$R287$	$U325$	$U367$	$R412$	$U429$	$U448$	$U479$	$U485$
seconds	26	91	145	93	173	346	347	255

Instance	$U509$	$R544$	$U584$	$U627$	$R629$	$U741$	$U771$	$U857$
seconds	464	377	534	515	364	1096	1396	1237

Table 6.11: Average running times for equally weighted criteria

Taking into account that the heuristic solves a tactical problem, the running times are acceptable.

6.5 Extensions

This section outlines some extensions considering further requirements or variations of the general model.

6.5.1 Incorporating Non-Required Edges

The model presented in Section 6.2 assumes that each edge is a required edge since most likely there are no non-required edges within cities. This extension addresses non-required edges, i.e., streets segments having no demand ($s_i = 0$). Let BA_n denote the set of non-required edges or streets segments, respectively. This extension distinguishes whether each non-required street has to be assigned to exactly one district or not.

In the first case, the heuristic treats a non-required street just like a required street and ends up with a solution, where each non-required street is visited at least once on a CPT.

However, there can be a better tour in terms of the total working time if this street has not to be visited. Hence, in the second case, non-required streets are excluded from the complete and exclusive assignment. That means, a non-required street needs not to be assigned to a district. Therefore, a solution contains a set of unassigned streets $B_{un} \subseteq BA_n$ in addition. Concerning the heuristic, there are additional shift-operations in order to assign unassigned basic areas or the other way around:

Shift-Assignment: The operation $shift_a(i, g)$ assigns an unassigned basic area $i \in B_{un}$ to a district D_g, i.e., $B_{un} = B_{un} \backslash \{i\}$ and $B_g = B_g \cup \{i\}$.

Shift-Unassignment: The operation $shift_u(i)$ unassigns a basic area $i \in BA_n$ from a district $D_g = D_{(i)}$, i.e., $B_g = B_g \backslash \{i\}$ and $B_{un} = B_{un} \cup \{i\}$.

Both operations maintain connectedness and the complete and exclusive assignment of the required streets.

6.5.2 Incorporating Depots

The next extension includes depots. Depending on the application either the location of depots is part of the planning process, e.g., locating boxes where the deliverer picks up the mail of one tour, or the depots are already existing, e.g., supermarkets where the deliverer picks up the leaflets.

Since each tour is a round-trip, the depot can be located everywhere on this tour. Hence, the first case is easy to handle.

If the depots are prescribed, each tour is enlarged by the shortest path from the depot to a point on the tour and back. Let C denote the set of depots. Moreover, let each district D_g contain exactly one depot $c_g \in C$. In this case, the total working time of D_g results in

$$w(D_g) := \sum_{i \in B_g} s_i + DH(D_g) + 2 \cdot d(c_g, B_g)\,,$$

where $d(c_g, B_g) := \min_{i \in B_g} d(c_g, i)$. Hence, in this case minimizing the total distances between the depots and the tours is a further optimization goal. Therefore, the heuristic optimizes an objective function consisting of five criteria and applies a further strategy in order to improve these distances. For each district D_g having the depot not on the corresponding tour, it defines neighbored basic areas, located closer to c_g than the closest basic area $i \in B_g$, as candidates that can be shifted to D_g. Note that this approach also works if more than one tour is associated with each depot.

6.5.3 Incorporating One-Way-Streets

If districts are serviced by car, one-way streets have to be considered. Unfortunately, in this case a tour within a district is not a CPT any longer. Hence, its computation is more complex, and, hence, the running time increases. However, from a theoretical point of view the presented approach is still applicable. Depending on the problem size an approximation of the tour length for evaluating a neighbored solution is necessary.

6.6 Conclusions

This chapter has proposed a heuristic for a districting problem arising in many arc routing applications, but did not yet attract the attention of many researchers. The presented model contains two hard criteria and four soft criteria which were weighted in a linear multi-criteria objective function. The proposed heuristic solves the problem by combining features of tabu search and adaptive randomized neighborhood search. Tests on graphs derived from real-world street data confirm the quality of the solutions.

A possible extension of this work could be to approximate the Pareto front of all feasible solutions with respect to these criteria, for example, as in Paquette et al. [21]. Moreover, the variation of the local compactness measure could be an additional feature. Some further measures are presented in Chapter 3.

In addition, some improvements according to the running times are possible. For example, for updating the deadheading times some observations described in Section 6.3.3.2 could be used in order to determine the MCM incrementally. Nevertheless, the current implementation already has confirmed the efficiency of the proposed methodology.

Bibliography

[1] L. D. Bodin and L. Levy. The arc oriented location routing problem. *INFOR*, 27(1): 74–94, 1989.

[2] L. D. Bodin and L. Levy. The arc partitioning problem. *European Journal of Operational Research*, 53(3):393–401, 1991.

[3] A. Butsch, J. Kalcsics, and G. Laporte. Districting for arc routing. *INFORMS Journal on Computing*, 26(4):809–824, 2014.

[4] F. Caro, T. Shirabe, M. Guignard, and A. Weintraub. School redistricting: embedding GIS tools with integer programming. *The Journal of the Operational Research Society*, 55(8):836–849, 2004.

[5] L. Chapleau, J.-A. Ferland, and J.-M. Rousseau. Clustering for routing in densely populated areas. *European Journal of Operational Research*, 20(1):48–57, 1985.

[6] L. S. de Assis, P. M. Franca, and F. L. Usberti. A redistricting problem applied to meter reading in power distribution networks. *Computers & Operations Research*, 41 (1):65–75, 2014.

[7] M. desJardins, B. Bulka, R. Carr, A. Hunt, P. Rathod, and P. Rheingans. Heuristic Search and Information Visualization Methods for School Redistricting. In *Proceedings of the 18th conference on Innovative applications of artificial intelligence - Volume 2*, pages 1774–1781. AAAI Press, 2006. ISBN 978-1577352815.

[8] J. Edmonds and E. L. Johnson. Matching, Euler tours and the Chinese postman. *Mathematical Programming*, 5(1):88–124, 1973.

[9] J. A. Ferland and G. Guénette. Decision support system for the school districting problem. *Operations Research*, 38:15–21, 1990.

[10] G. García-Ayala, J. González-Velarde, R. Ríos-Mercado, and E. Fernández. A novel model for arc territory design: promoting Eulerian districts. *International Transactions in Operational Research*, 23(3):433–458, 2016.

[11] F. Glover. Heuristic for integer programming using surrogate constraints. *Decision Sciences*, 8(1):156–166, 1977.

[12] J. L. Gross and J. Yellen. *Handbook of Graph Theory*. CRC Press, Boca Raton, Florida, 2003. ISBN 978-1584880905.

[13] M. K. Guan. Graphic programming using odd or even points. *Chinese Mathematics*, 1: 273–277, 1962.

[14] S. Hanafi, A. Fréville, and P. Vaca. Municipal solid waste collection: An effective data structure for solving the sectorization problem with local search methods. *INFOR*, 37: 236–254, 1999.

[15] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. Nonpartisan Political Redistricting by Computer. *Operations Research*, 13(6):998–1006, 1965.

[16] A. I. Jarrah and J. F. Bard. Large-scale pickup and delivery work area design. *Computers & Operations Research*, 39(12):3102–3118, 2012.

[17] V. Kolmogorov. Blossom V: A new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation*, 1:43–67, 2009.

[18] M. Mourão, A. Nunes, and C. Prins. Heuristic methods for the sectoring arc routing problem. *European Journal of Operational Research*, 196:856–868, 2009.

[19] L. Muyldermans, D. Cattrysse, D. Van Oudheusden, and T. Lotan. Districting for salt spreading operations. *European Journal of Operational Research*, 139(3):521–532, 2002.

[20] L. Muyldermans, D. Cattrysse, and D. Van Oudheusden. District design for arc-routing applications. *Journal of the Operational Research Society*, 54(11):1209–1221, 2003.

[21] J. Paquette, J.-F. Cordeau, G. Laporte, and M. M. Pascoal. Combining multicriteria analysis and tabu search for dial-a-ride problems. *Transportation Research Part B: Methodological*, 52:1–16, 2013.

[22] N. Perrier, A. Langevin, and J. F. Campbell. A survey of models and algorithms for winter road maintainance. Part I: System design for spreading and plowing. *Computers & Operations Research*, 33:209–238, 2006.

[23] N. Perrier, A. Langevin, and J. F. Campbell. A survey of models and algorithms for winter road maintainance.Part II: System design for snow disposal. *Computers & Operations Research*, 33:239–262, 2006.

[24] N. Perrier, A. Langevin, and J. F. Campbell. The sector design and assignment problem for snow disposal operations. *European Journal of Operational Research*, 189(2):508–525, 2008.

[25] R. Ríos-Mercado and E. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36 (3):755–776, 2009.

[26] S. Ropke and D. Pisinger. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4):455–472, 2006.

[27] O. B. Schoepfle and R. L. Church. A New Network representation of a "classic" School districting problem. *Socio-Economic Planning Sciences*, 25(3):189–197, 1991.

Part IV

Implementation of Districting Problems

7 Lizard

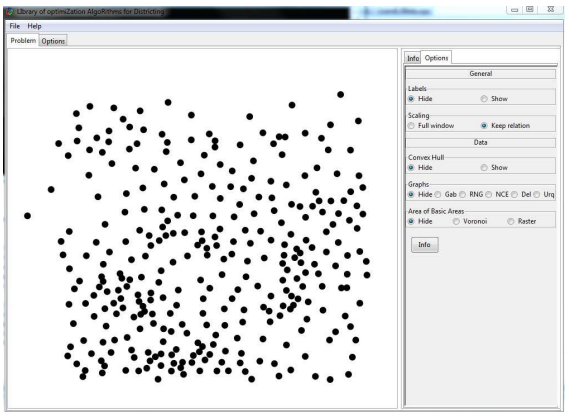
Within the scope of this thesis we have developed an open source C++ library of algorithms to solve districting problems called *"Lizard"* (LIbrary of optimiZation AlgoRithms for Districting). *Lizard* is freely available under `www.lizard.ior.kit.edu` as a Windows executable. Moreover, this homepage makes the source code and some exemplary problem instances available. In addition, we have developed an interface to the geographic information system OpenStreetMap[1]. This chapter provides an overview over the contents and options of *Lizard*.

First of all, *Lizard* provides a module to load and visualize an existing instance. Figure 7.1a depicts an exemplary instance after loading it. The graphical front-end is based on the library GTKMM[2]. Moreover, *Lizard* contains some tools to visualize surrogates in the context of compactness and contiguity evaluation. For example, *Lizard* can determine and display the different kind of neighborhood graphs presented in Section 2.2.4 or of the basic areas' shapes stated in Section 3.5.4. For the instance illustrated in Figure 7.1a, Figure 7.1b depicts the corresponding Relative Neighborhood Graph and Figure 7.1c shows the corresponding Voronoi Regions exemplarily.

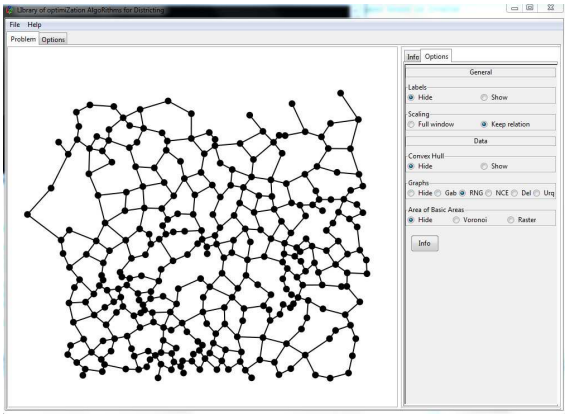
After loading an instance, the user specifies the problem instance in more detail, chooses a solution approach and defines the corresponding parameter settings. Figure 7.1d depicts the interface for choosing the Recursive Partitioning Algorithm. Here, among other parameters, the user can specify the compactness measure (cf. Section 4.2.2.3) and the kind of bisecting partition (cf. Section 4.3.3).

After specifying and solving the districting problem, *Lizard* displays the solution. In order to distinguish them, *Lizard* displays the different districts in different colours. Again, *Lizard* provides some tools to analyze the solutions, both graphically and textually. For example, the different approaches of determining shapes of districts are implemented. Figure 7.2a depicts the convex hulls of the districts (cf. Section 3.5.3.3), whereas Figure 7.2b shows their

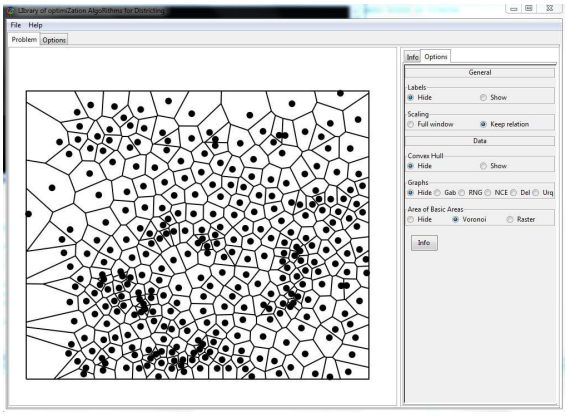
[1] http://www.openstreetmap.org
[2] http://www.gtkmm.org



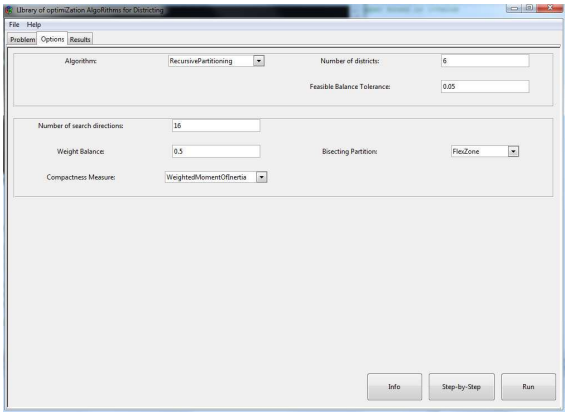
(a) Districting instance



(b) Relative Neighborhood Graph



(c) Voronoi regions

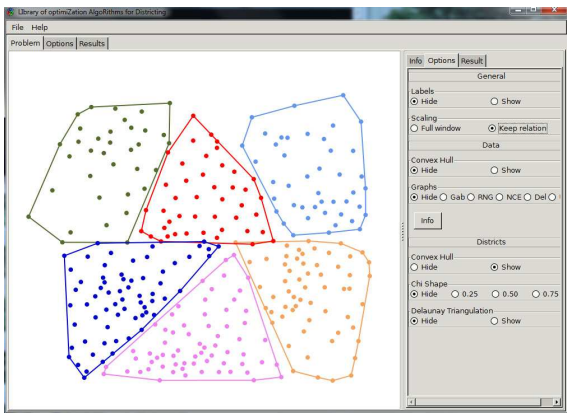


(d) Choosing an algorithm

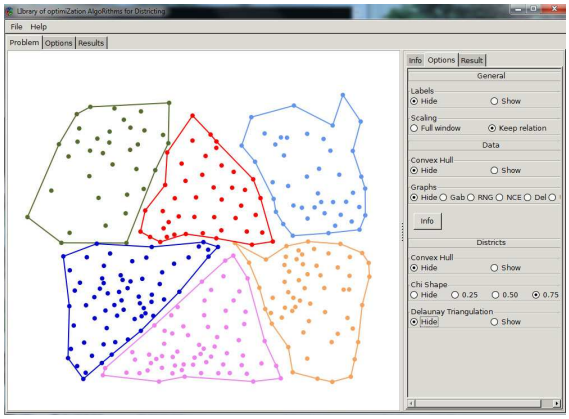
Figure 7.1: Specifying a problem instance and a solution approach

χ-shapes (cf. Section 3.5.3.4). Moreover, the user can evaluate a solution or the corresponding districts, respectively, in terms of a number of measures. *Lizard* provides different variations of balance (cf. Section 2.2.2), compactness (cf. Chapter 3) and contiguity (cf. Section 2.2.4) measures. Figure 7.2c shows the drop menu for choosing a measure and Figure 7.2d shows some exemplary results.

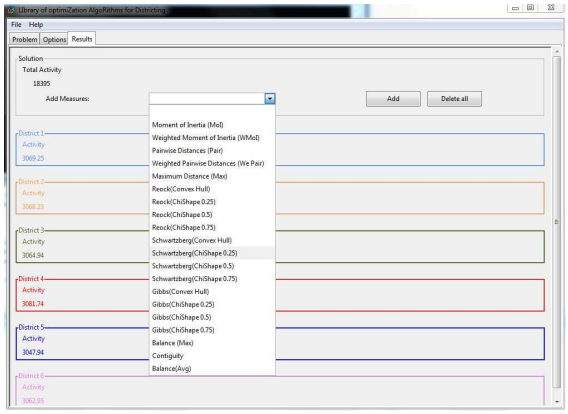
In addition, *Lizard* contains a step-by-step version of the RPA in order to make it applicable for teaching. For a better understanding, *Lizard* visualizes each sub-division and reports the corresponding evaluations of the generated bisecting partitions. For example, Figure 7.3a shows the first sub-division of a districting problem, where the two lines illustrate the borders of the corresponding flex-zone. The window on the right provides information about the generated bisecting partitions for all search directions. It states basic information such as the angle, or further information details such as evaluations in terms of balance and compactness. Figure 7.3b illustrates the situation some sub-divisions later. The window on the right allows the user to navigate through the sub-division history.



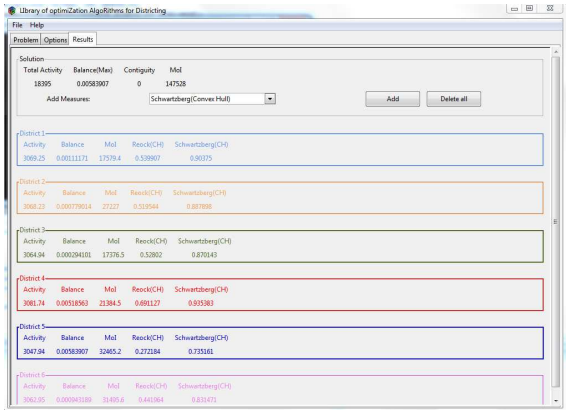
(a) Illustrated as convex hull



(b) Illustrated as χ-shapes

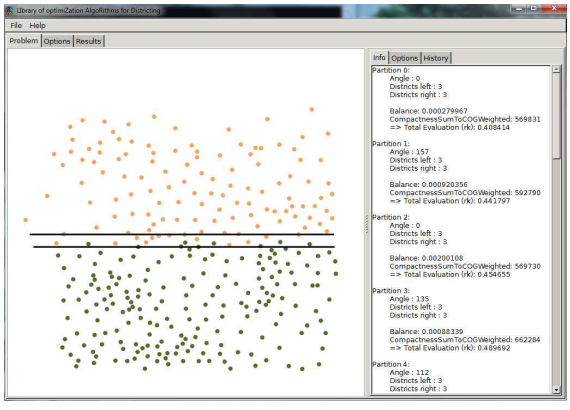


(c) Choosing a measure

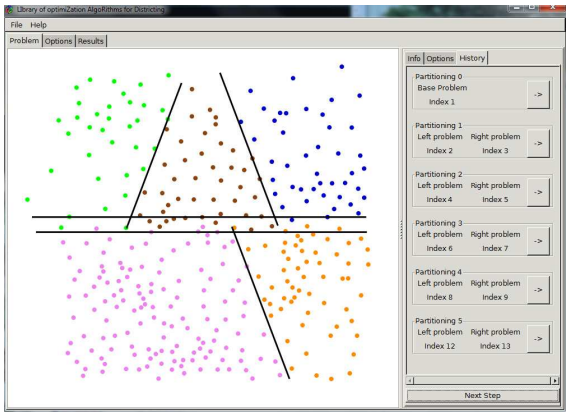


(d) Resulting evaluations

Figure 7.2: Illustrations of a solution

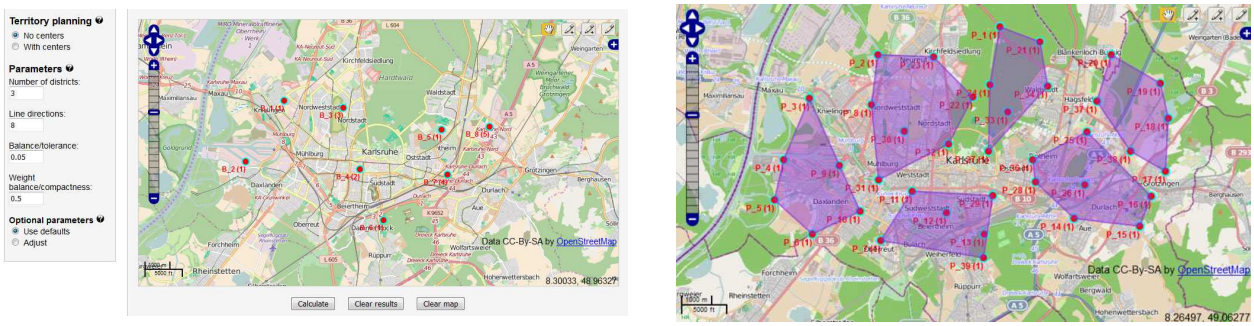


(a) Illustration of the first sub-division



(b) Illustration of a later sub-division

Figure 7.3: Step-by-step version



(a) Web front-end

(b) Illustration of a solution

Figure 7.4: GIS intergration of *lizard* (Map: ©OpenStreetMap)

Beside the offline version, we have developed an online version that can be called without any local installation of software. The user interface is shown in the web browser. Here, the user places basic areas on a map provided by OpenStreetMap and specifies the parameters using an input mask. Figure 7.4a shows the described web front-end. After solving the problem, the web browser shows the calculated result as well, as Figure 7.4b illustrates exemplarily. Altogether, *Lizard* is an open source C++ library including a graphical front-end and a GIS integration that allows solving districting problems and visualizing and analyzing the obtained results.

8 Conclusions and Outlook

This thesis has addressed different aspects of districting problems. In the first part, the most common components and planning criteria have been reviewed. In particular, compactness has been investigated in very much detail. Even if it is theoretically nearly impossible to define a comprehensive compactness measure, this thesis has pointed out that some measures proposed in literature perform well in practice. In addition, this thesis has introduced a couple of practical approaches for measuring compactness when basic areas are represented by lines or points. If a service person has to travel within a district, the requirement for compact districts is based on the assumption that a compact district induces small expected travel times. Future research could address this correlation in more detail and enhance existing measures or develop new ones that incorporate an approximation of the travel times. This approximation is challenging since travel times depend on requirements of the customers such as visit frequencies or time windows, but also on stochastic factors such as the day-to-day demand and traffic jams. Moreover, the developed measures should not relax the requirement for visually compact districts.

This thesis has focused on solution approaches for applications where basic areas are represented by lines and points since these cases have not yet attracted the attention of several researchers. The latter case is the content of the second part of this thesis, where the presented approaches are based on ideas from computational geometry. In particular, there is a special feature of these presented approaches: Even though the presented approaches are geometrically motivated, they are able to incorporate distances on a road network.

The Recursive Partitioning Algorithm (RPA) is an existing geometrically motivated heuristic that yields in nearly perfectly balanced districts. This thesis has overcome the RPA's weaknesses in terms of compactness; the solutions of the improved RPA are considerably more compact, while the quality in terms of balance and contiguity is still good. Hence, the RPA delivers good overall solutions that are a compromise between the different planning criteria. Furthermore, fast running times allow an interactive use.

Concerning applications that focus on compactness during the planning process, the Power Diagram Districting Algorithm (PDDA) introduced in this thesis achieves further improvements. The compactness improvements come at a cost of deteriorations of balance, however, the algorithm keeps balance within predefined limits. The algorithm has been evaluated for many practical examples with a problem size of up to some thousand basic areas. Running times for these problems are still within a few seconds.

The different versions and extensions of the RPA and the PDDA that have been developed in this thesis have been made available as the library *Lizard*. When using this library, users can determine districting plans depending on their specific requirements.

Beside the presented or outlined extensions, further planning scenarios could be addressed in future research. For example, similarity to an existing districting plan is often sought. In this case, the evaluation function for a solution should at least incorporate a similarity measure. Moreover, a way to integrate this requirement into the RPA could be the restriction to currently unsatisfactory parts of the solution. For example, the RPA could be applied to a sub-problem consisting of a district evaluated as poor and (a subset of) its neighboring districts. In order to integrate this requirement into the PDDA, the locations of the generators could be restricted to be in defined regions around their current locations.

For the case of line representations, the third part of this thesis has developed a heuristic for districting problems where the edges of a road network have to be serviced. The proposed heuristic focuses on problems where the service within each district is provided by bike or foot. However, it can be adapted to further applications where the service is provided by car or truck. This heuristic combines ideas from geometrical approaches, tabu search, and adaptive randomized neighborhood search. Its innovation is the fact that it takes into account both compactness and routing distances explicitly. Tests on real-world street data confirm the efficiency of this approach. Hence, in future works the proposed methodology can be integrated in further practical districting algorithms in the context of arc routing, for example for snow removal, waste collection, and similar applications.

In conclusion, this thesis has developed the modelling and the solution of districting problems, mainly in the context of basic areas represented by points or lines. The proposed approaches are applicable to many practical problems, for example in the context of the design of districts for field staff members or mail deliverers. Moreover, these algorithms can be a basis of solution approaches where districting problems occur as part of another problem, such as routing or facility location.

Erklärung

gemäß §4, Abs. 4 der Promotionsordnung vom 15.August 2006:

Ich versichere wahrheitsgemäß, die Dissertation bis auf die in der Abhandlung angegebene Hilfe selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und genau kenntlich gemacht zu haben, was aus Arbeiten anderer und aus eigenen Veröffentlichungen unverändert oder mit Abänderungen entnommen wurde.

Hiermit erkläre ich, dass ich bisher an keiner anderen Hochschule ein Promotionsgesuch eingereicht habe.

Karlsruhe, 25.04.2016

Alexander Butsch