Beyond Zooming: Models and Algorithms for Mobile Maps with Extended Interaction Techniques

Dissertation

zur Erlangung des Grades

Doktor der Ingenieurwissenschaft (Dr.-Ing.)

der Agrar-, Ernährungs- und Ingenieurwissenschaftlichen Fakultät

der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Sven Gedicke

geboren in

Münster

Bonn, 2024

Referent:	Prof. Dr.-Ing. Jan-Henrik Haunert
Korreferenten:	Prof. Dr. Sara Irina Fabrikant
	Prof. Dr. Stephan Winter

Tag der mündlichen Prüfung:	24. Juli 2024

Angefertigt mit Genehmigung der Agrar-, Ernährungs- und Ingenieurwissenschaftlichen Fakultät der Universität Bonn.

Acknowledgments

First and foremost, I wish to acknowledge my doctoral supervisor Jan-Henrik Haunert for his continuous support, encouragement, and patience throughout the course of my doctorate. The mutual exchange of ideas and your insightful advice have contributed significantly to the development of this thesis. Also, I would like to express my gratitude for the opportunities Jan provided me, including attending international conferences and undertaking research visits to Switzerland and Australia during my doctoral studies. Thank you Jan!

In relation to my international stays, I extend my gratitude to Sara Irina Fabrikant and Stephan Winter for their friendly support and guidance during my time abroad. Both stays offered unforgettable experiences that I am grateful for.

I would like to give special thanks to Benjamin Niedermann and Youness Dehbi for their exceptional support, expertise, and advice. Since the beginning of my doctoral studies, you have always supported me in my research endeavors and guided me through the academic challenges.

Many thanks to all my colleagues who have accompanied me throughout my doctoral journey. The time we shared, whether during retreats, staff meetings, or casual interactions at the coffee machine or in the canteen, has greatly enriched my experience. Special thanks go to Annika, whose invaluable research-related insights and long-standing friendship I deeply cherish. You always lent a listening ear, and I sincerely appreciate it. I also want to acknowledge Axel, my longtime office colleague, for his support with technical challenges and guidance on both professional and personal matters. I will fondly remember our time as *geotastic buddies*. A heartfelt thanks to Alina, with whom I not only enjoy spending time privately, but also as colleagues in the office. Thanks to all of you!

Furthermore, I wish to acknowledge all my co-authors who have collaborated with me on various research projects throughout my academic journey. Your collective expertise and teamwork have been instrumental in not only shaping but also implementing many of the ideas and findings presented in this thesis.

Finally, I would like to thank my family and friends. I am very fortunate to have such a network of loved ones who have supported and encouraged me along the way. Cheers to the *Affenbande* and a special thank you to Corinna!

Thanks everyone!

Abstract

This thesis focuses on the development and evaluation of novel interfaces for the presentation of large and detailed geographic data sets on mobile devices. Especially when the density of information is high, a major challenge emerges due to the constrained screen space, making it difficult to maintain uncluttered map representations. Established interfaces rely on not showing all information simultaneously and providing basic interaction capabilities such as zooming and panning for further diving into a map's details. However, to thoroughly explore the information in an area of interest, such zoom-and-pan strategies often require a user to zoom in to a very large scale. To retain the spatial context, the user has to repeatedly zoom and pan the map section. Not only can this process be tedious, but the changes in scale and level of detail also impede the cognitive linking of successive visualizations.

Addressing this issue, the thesis follows the premise that advancing beyond common zoom-and-pan methods is possible by shifting attention towards specialized interaction techniques. In this context, *zoomless maps* will be introduced. Such maps implement advanced interaction capabilities, which enable users to explore all available information without the need for zooming or panning, thus maintaining the map scale and spatial context unchanged. These novel interactions can also complement traditional zoom-and-pan techniques, broadening the capabilities of established map interfaces.

In particular, different variants of zoomless maps will be developed, each implementing a distinct interaction technique. The implementations rely on mathematical models that consider multiple criteria tailored to a corresponding interaction. A criterion is either optimized or enforced as a constraint within a mathematical optimization problem. Taking into account the computational complexity of a problem, suitable optimization methods are applied, with a primary focus on *combinatorial optimization*.

Exact methods, such as integer linear programming, will be introduced, which provide mathematically optimal solutions. These solutions enable the verification of the mathematical models using different quality metrics. Through the comparison with optimal solutions of specialized models – each optimizing only one criterion at a time – it will be shown that the multi-criteria models allow for a satisfactory trade-off across all considered criteria. However, since exact methods may prove to be too slow for interactive scenarios, heuristics will be presented that rather focus on efficiency than on optimality. Still, using the optimal results of exact methods as benchmarks, it will be shown that the heuristics yield high-quality solutions with regard to the optimized objectives.

Beyond quantitatively evaluating the developed models and algorithms, one key objective of this thesis is to verify the usability and utility of zoomless maps and to prove their merits over established zoom-and-pan strategies. To that end, a comprehensive empirical user study will be presented that compares three variants of zoomless maps against a standard zoom-and-pan interface. For the common task of browsing through a large collection of features to find an object that suits a user's preferences best, it will be shown that the use of zoomless maps has a positive impact on accuracy in terms of the given search criteria. Additionally, it reduces the number of performed zooming and panning operations and leads to a smaller scale with more map context being available when selecting an object.

The exploration of ways to validate a broader generalizability of the benefits of zoomless maps as well as possible development potentials are outlined as part of future research.

Kurzfassung

Diese Arbeit konzentriert sich auf die Entwicklung neuartiger Kartenanwendungen zur Darstellung großer geografischer Datensätze auf mobilen Geräten. Insbesondere bei einer hohen Informationsdichte stellt der begrenzte Platz auf dem Bildschirm eine Herausforderung dar, die eine übersichtliche Kartendarstellung erschwert. Etablierte Anwendungen zeigen nicht alle Informationen gleichzeitig an und ermöglichen es durch Interaktionen wie Zoomen und Verschieben, tiefer in die Details einer Karte einzutauchen. Um die Informationen in einem bestimmten Bereich umfänglich erkunden zu können, erfordern solche Strategien allerdings oft ein Heranzoomen auf einen sehr großen Maßstab. Um dabei den räumlichen Kontext nicht zu verlieren, muss wiederholt gezoomt und der Kartenausschnitt verschoben werden. Dieser Prozess ist mühsam und erschwert die Orientierung innerhalb der Karte.

In dieser Arbeit wird diese Problematik aufgegriffen und mit *Zoomless Maps* ein alternativer Ansatz vorgestellt. Durch neuartige Interaktionsmöglichkeiten erlauben diese Karten eine umfängliche Informationserkundung, ohne den Maßstab ändern zu müssen. Die neuen Interaktionen können herkömmliche Zoom-und-Verschiebe-Techniken ergänzen und somit die Funktionen etablierter Kartenanwendungen erweitern.

Es werden verschiedene Varianten von *Zoomless Maps* entwickelt, wobei jede eine unterschiedliche Interaktionstechnik umsetzt. Die Implementierungen beruhen auf mathematischen Modellen, die mehrere auf die jeweilige Interaktion zugeschnittene Kriterien berücksichtigen. Ein Kriterium wird entweder optimiert oder als Bedingung innerhalb eines mathematischen Optimierungsproblems gefordert. Unter Berücksichtigung der Komplexität eines Problems werden geeignete Optimierungsmethoden angewendet, wobei der algorithmische Fokus auf *kombinatorischer Optimierung* liegt.

Es werden exakte Verfahren vorgestellt, die mathematisch optimale Lösungen liefern. Diese Lösungen ermöglichen die Überprüfung der mathematischen Modelle unter Verwendung verschiedener Qualitätsmetriken. Durch den Vergleich mit optimalen Lösungen spezieller Modelle, die jeweils nur ein Kriterium gleichzeitig optimieren, kann gezeigt werden, dass die multikriteriellen Modelle einen zufriedenstellenden Kompromiss zwischen allen betrachteten Kriterien ermöglichen. Da exakte Methoden für interaktive Szenarien jedoch oft zu langsam sind, werden zudem effiziente Heuristiken vorgestellt. Durch den Vergleich mit optimalen Ergebnissen der exakten Methoden wird gezeigt, dass die entwickelten Heuristiken neben einer schnellen Laufzeit qualitativ hochwertige Lösungen hinsichtlich der optimierten Ziele erreichen.

Über die quantitative Bewertung der entwickelten Modelle und Algorithmen hinaus ist ein wichtiges Ziel dieser Arbeit, die Nützlichkeit und Benutzerfreundlichkeit von *Zoomless Maps* zu überprüfen und ihre Vorzüge gegenüber etablierten Anwendungen nachzuweisen. Hierzu wird eine empirische Nutzerstudie präsentiert, die drei Varianten von *Zoomless Maps* gegen eine Standard-Zoom-und-Verschiebe-Karte vergleicht. Für die Suche nach einem Objekt, das festgelegten Kriterien am besten entspricht, wird gezeigt, dass sich die Verwendung von *Zoomless Maps* positiv auf die Genauigkeit in Bezug auf die gegebenen Suchkriterien auswirkt. Darüber hinaus reduziert sie die Anzahl der durchgeführten Zoom- und Verschiebevorgänge und führt dazu, dass bei der Auswahl eines Objektes mehr Kartenkontext sichtbar ist.

Als Ausblick für zukünftige Forschung werden Potenziale zur Weiterentwicklung der vorgestellten Interaktionstechniken skizziert, sowie weiterführende Formen von Nutzerstudien diskutiert.

Note on Attributions

In this thesis, several figures contain maps as background. The map tiles used in these figures are either provided by *Stamen*[1] or *Carto*[2], both under the Creative Commons Attribution 3.0 License (CC BY 3.0). For both, the underlying map data is sourced from ©*OpenStreetMap*[3] under the Open Database License (ODbL). To save space and ensure clarity, we will use the following abbreviations in the figure captions:

Stamen:	Map tiles by Stamen, under CC BY 3.0. Data by ©OpenStreetMap, under ODbL.
Carto:	Map tiles by Carto, under CC BY 3.0. Data by ©OpenStreetMap, under ODbL.

These abbreviations will be used throughout the thesis whenever a figure contains a map background.

[1] https://maps.stamen.com

[2] https://carto.com/basemaps

[3] https://www.openstreetmap.org

CONTENTS

1 INTRODUCTION

Automation in cartography has a long history, dating back to the advent of digital maps in the second half of the 20th century. Over the past several years, we have become accustomed to computers playing an essential role in the processing, provision, and presentation of spatial data. With the transition from analog to digital map products, the focus of cartographic research has changed, and cartography has become a highly advanced and versatile domain. Manual processes that sufficed in analog cartography become impractical when dealing with the efficiency demands and vast amounts of data that come with digital maps. Thus, automating cartographic processes is an integral issue in modern cartography. While the focus of corresponding research has long been on static maps (e.g., classical topographic maps), the emphasis today has shifted more and more towards highly interactive maps. Especially mobile maps on small devices – such as smartphones and smartwatches – have become an integral part of our daily lives, making access to spatial data ubiquitous. Location-based services have become indispensable tools, e.g., for wayfinding or exploring an unknown environment. However, despite the ever-advancing evolution of mobile devices, their small displays and restricted processing power are limitations that require special consideration when developing mobile map interfaces (Muehlenhaus, 2013). Especially when the density of information is high, it is challenging to create small-screen maps that combine both information fidelity and legibility, i.e., maps that convey all the information contained in a certain area of interest without producing clutter. This thesis contributes to tackling this challenge by introducing novel mathematical models and algorithms for presenting large collections of information in mobile maps.

1.1 Motivation

Rather than displaying all information contained in an area of interest simultaneously, established map interfaces reduce clutter by showing a non-overlapping subset of features. By providing basic map interactions, users can access hidden information (Chittaro, 2006). Notably, most of such interfaces are primarily based on zooming and panning (see Figure 1.1). To reveal information that is not shown in the current visualization, a user can enlarge the map by further zooming in. When zooming in far enough, any map feature can be displayed without conflicting (i.e., overlapping and obscuring) with other features. With regard to such *zoom-and-pan interfaces*,

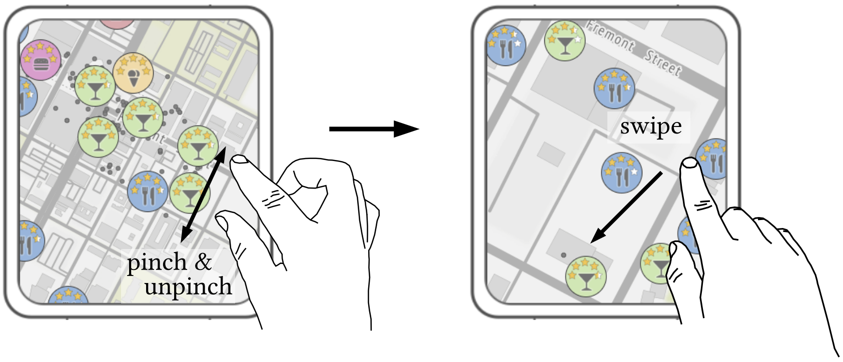


Fig. 1.1: A touch-based zoom-and-pan interface for exploring food services. A user can zoom the map by pinching and pan the map by swiping. *Map attribution: Stamen.*

previous research has focused on developing advanced zooming capabilities that facilitate the users' ability to track visual changes as they change the map scale. In particular, approaches have been developed in the context of both map generalization (Chimani et al., 2014; Sester and Brenner, 2004; Li, 2009; Suba et al., 2016; van Oosterom and Meijers, 2014) and the placement of map labels (Been et al., 2006, 2010; Liao et al., 2016; Zhang et al., 2020) that ensure certain criteria of consistency across scales. Further, researchers introduced algorithms that continuously (Nöllenburg et al., 2008; Gao et al., 2020; Peng et al., 2023; Li and Mao, 2023) or gradually (Peng et al., 2017, 2020) animate the transition between different scales.

Despite the progress that has been made in recent years towards the development of specialized interaction and visualization strategies for multi-scale environments, the majority of map interfaces continue to rely on conventional zoom-and-pan techniques (Gruget et al., 2023). However, we believe that alternative forms of interaction need to be given greater focus. We argue that improvements beyond current standards can be attained by exploring innovative forms of interaction, as we see the following issues with interfaces that rely on zooming and panning only.

While the user digs into the details of the map by zooming in to a large scale, information about the surrounding area gets lost. The larger the map scale, the less map area and, thus, less context is available for the user to spatially orient themselves. In order not to lose the overall context and to situate certain information in the large-scale context, the user has to repeatedly zoom in and out and pan the map section (Schmid et al., 2010; Burigat and Chittaro, 2011). Such a procedure is particularly necessary when a task requires the user to solve several sub-tasks simultaneously (e.g., searching for a hotel while inspecting the geographic context) or sequentially (e.g., searching for the train station before searching for a nearby restaurant). Not only can repeatedly zooming in and out become tedious and cumbersome, but the changes in scale and level of detail also impede the ability to cognitively link successive visualizations (Ulugtekin and Dogru, 2007). Distinctive visual cues can change in their rendering or even disappear, making it difficult to use them as points of reference while interacting with the map (Touya et al., 2021, 2023a). Consequently, the user needs some time to get oriented again after each interaction. This effect is commonly referred to as *desert fog* (Jul and Furnas, 1998). Furthermore, the user does not know in advance where information that is not shown in the current visualization is located in the map. It can easily happen that they do not zoom in far enough for all available information to become visible. Therefore, despite an intensive exploration of the map, information may remain hidden and the user may not be able to find the most suitable objects matching their task.

1.2 Objective and Main Contribution

Addressing the disadvantages of established zoom-and-pan interfaces, we argue that accessing large and detailed geographic data sets should be possible without requiring the user to zoom to an extremely large scale. In particular, we state the following central hypothesis.

Central Hypothesis. *Organizing the content of a dense map on a small screen into manageable and easily accessible pieces improves the interaction with the map in such a way that users find the information they search for at a smaller scale and, thus, zooming to a larger scale can be avoided.*

We consider displayed information to be *manageable* if it is legible at all times. Further, we regard information as *accessible* if a user is able to retrieve it with few interactions.

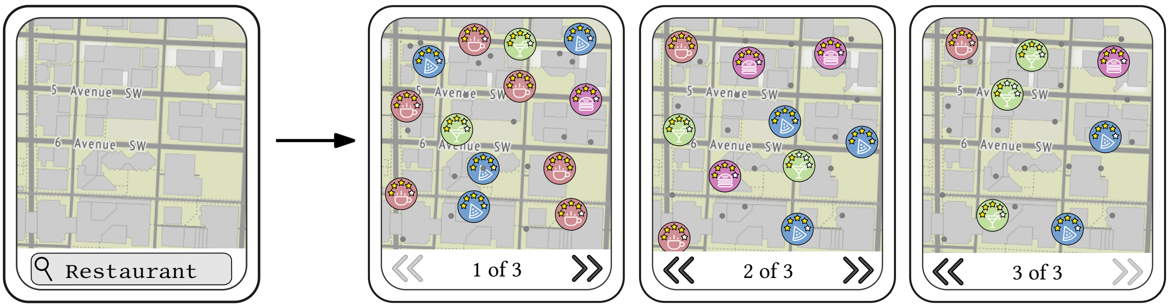


Fig. 1.2: Example of one of our novel interfaces. After querying for restaurants, the results are distributed on three pages, each showing the same background map and a different subset of the queried features. The user can browse through the pages by clicking on the arrow-shaped buttons. On each page, the gray dots indicate further restaurants. For more details on this approach see Chapter 3 and Chapter 4. *Map attribution: Stamen.*

The primary objective of this thesis is to verify the central hypothesis by developing and comparing different strategies for presenting large collections of spatial information in mobile maps. We introduce novel strategies that go beyond conventional zoom-and-pan interfaces, incorporating specialized interaction techniques. In general, these techniques are intended to allow users to explore all information without resorting to zooming and panning. Not all information is displayed at the same time, but the user can interactively browse through the features while both the map scale and the shown map section remain fixed. The idea is not to exclude standard interactions, but to reduce the necessity of relying on them. One illustrative example of a map interface implementing such a strategy is given in Figure 1.2. In this example, the information is distributed on multiple pages which a user can navigate through by clicking navigation buttons. Since such advanced map interfaces aim to enable a thorough information exploration without having to repeatedly zooming in and out, we subsume them under the term *zoomless maps* in the remainder of this thesis.

Generally, we consider zoomless maps to be suitable for a broad range of tasks and not specifically tailored to a certain area of application. Throughout the thesis, however, we mainly focus on a specific user task that is employed as a recurring example. In particular, we concentrate on the common scenario of a user browsing through a dense set of point features to find an object that best suits their needs, e.g., a restaurant that matches certain preferences. In the context of vision science, such a task can be understood as a ranking task within a visual search (Wolfe, 2020). The user engages in a visual search to identify relevant objects, and subsequently ranks them based on how well they match specific attributes. With regard to the ranking, it is assumed that each feature is assigned a weight that expresses its importance, e.g., a user-given rating as it is common on many search and recommendation portals. We consider it a realistic procedure for a user to first get a rough overview of, for instance, an entire city, and then to select a specific area of interest by zooming and panning the map (see Figure 1.3). The user then queries for specific information entering a suitable search term, e.g., *restaurant*. As a result of the query, matching map features are presented and can be further explored by the user. Linking to the last step of this procedure, zoomless maps are intended to enable a thorough information exploration without the need of zooming.

Regarding the visualization of information, we assume that each point feature is annotated with a descriptive *label*. Although in classical cartography the term label was used exclusively for text annotations, we use this term in a more flexible context that also allows descriptive symbols and pictograms. This understanding of a label is consistent with contemporary cartographic research (see Section 2.4). By using symbols as labels, additional information (e.g., ratings or feature types) can be presented in an easily understandable way.

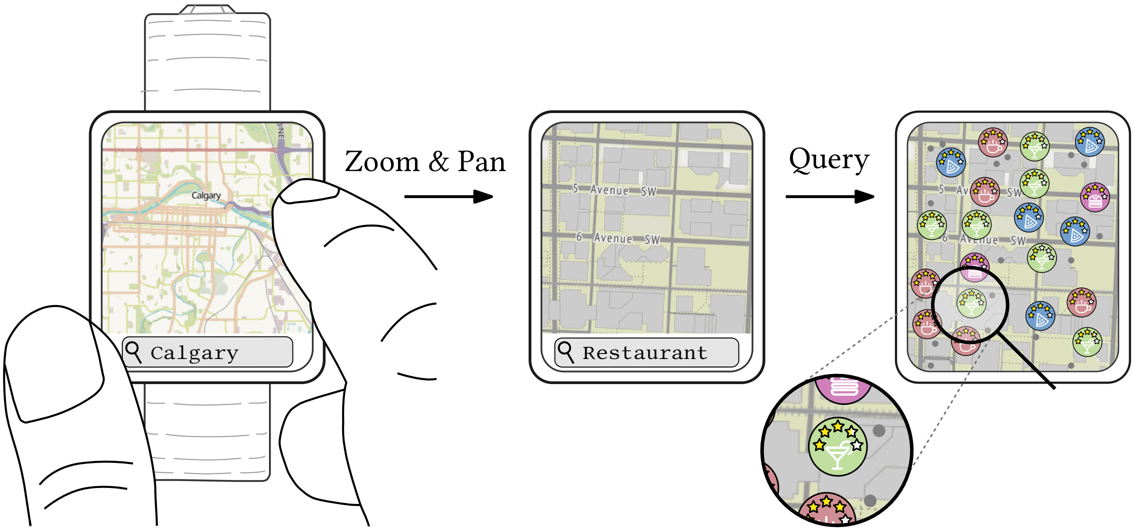


Fig. 1.3: A common user's procedure when searching for specific information. First, a broad overview of a city is obtained. Then, a specific area of interest is selected, restaurants are queried, and the results are displayed on the map. *Map attribution: Stamen.*

1.3 Research Cycle

We consider the development process of zoomless maps as an iterative cycle with different phases (see Figure 1.4). Starting with a broad idea, it is important to refine and concretize this idea, i.e., to *conceptualize* the map interface. For each interface, we define its underlying *concept* to be the strategy that specifies how the set of all features is partitioned, when and how each feature is displayed, and what specific type of interaction is used to provide access to all features.

After a concept has been established, the *modeling phase* is primarily concerned with translating the concept into a mathematical framework. We formalize optimization problems which mathematically express strict requirements as well as quantitative criteria that specify the quality of a solution. To this end, we take into account both general principles and observations from literature on cartography and information visualization as well as the requirements and preferences of users and experts. Alongside a review of existing literature, we conduct user and expert interviews to gain direct insights and opinions.

Once a mathematical model has been formalized, we develop algorithms for solving the established optimization problems in the phase of *algorithmic design*. Thereby, we focus on both the efficiency and the exactness of our algorithms. We analyze a problem in the context of its computational complexity and apply corresponding optimization methods. We tackle NP-hard problems by means of exact approaches – such as integer linear programming – which yield optimal solutions. Even though such approaches require an exponential running time in the worst case, in practice, optimal solutions can often be found within a reasonable time. Addressing the application of our models in interactive interfaces, we further focus on efficient algorithms. More specifically, we develop fast heuristics that do not guarantee optimality but still yield solutions of high quality with respect to the defined objective.

Directly related to the algorithmic design is the actual *implementation* of the models and algorithms as interactive interfaces. The realization of our strategies into usable applications allows us to get a feel of the usage and thereby to uncover potential weaknesses early on. Considering the subsequent *evaluation phase*, it further enables assessing our interfaces in terms of usability and utility with test users. Generally, we evaluate our novel strategies with regard to two key aspects:

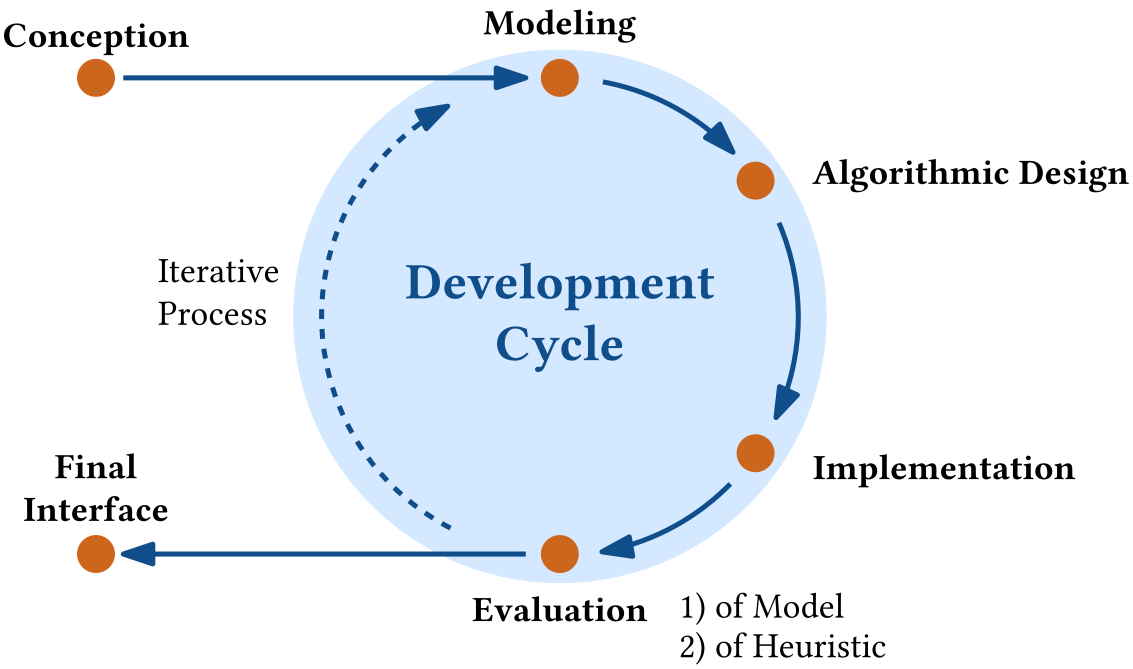


Fig. 1.4: Iterative development cycle of zoomless maps.

(1) the verification of our mathematical models

(2) the quality assessment of heuristic solutions with respect to the model

To verify our models (1), we quantitatively compare optimal solutions with respect to different quality metrics. In most of our models, multiple criteria are considered simultaneously. To assess how well such a multi-criteria model satisfies each criterion, we compare its optimal solutions with those of specialized models that optimize only one of the criteria at a time. Another crucial aspect of the model evaluation is the verification of the actual usability and utility of our interfaces. In this context, we apply the concept of *user-centered design* (UCD), which generally describes the active involvement of the user in the development of an application or interface (Abras et al., 2004; Robinson et al., 2005). We conduct both user and expert studies to assess our interfaces from a practical perspective and to obtain input for improvements. To evaluate the quality of solutions that we obtain by applying heuristic approaches (2), we analyze how a heuristic and an exact method perform in comparison to each other. More specifically, we compare the quality of their solutions according to the defined objective and their running times in practice. Since zoomless maps are indented to be highly interactive, we aim for algorithms that perform well in terms of both running time and quality.

It is particularly important to emphasize the iterative nature of the applied development process. Especially with regard to certain model assumptions made, it is likely that the evaluation step of the first iteration will reveal further potentials for improvement. Going through successive cycles of the development process allows such potentials to be picked up and the existing models and algorithms to be adapted accordingly.

1.4 Outline

This thesis is structured as follows. First, Chapter 2 addresses the opportunities and challenges of automation in cartography. We discuss progress and the current state of the art and go into more detail on subdomains that we consider to be closely related to zoomless maps. In particular, we address cartography on mobile devices, map generalization, and label placement. Linking to the methodological background of this thesis, we further explain theoretical foundations of combinatorial optimization: a prominent strategy employed for mathematically solving cartographic tasks. Concluding the chapter, we present own research as a concrete example in which we apply combinatorial optimization to a generalization task. In Chapters 3 to 5, we focus on the presentation of different concepts of zoomless maps that we have developed in the last years. In addition to a detailed description of our methodologies, we evaluate each mathematical model quantitatively and assess the quality of solutions of our algorithms. Towards validating the central hypothesis of this thesis, Chapter 6 presents a comprehensive empirical user study assessing and comparing the usability and utility of the previously presented concepts. While in general zoomless maps have broad applicability across various domains, Chapter 7 concentrates on one practical use case. In particular, we address map interfaces tailored for emergency services within the field of disaster and emergency response, presenting a novel approach for automatically placing tactical symbols on situation maps. In the concluding Chapter 8, we summarize our main results obtained in this thesis, discuss the findings in the context of the central hypothesis, and provide possible directions for future research.

The contents of Chapters 3 to 7 are mainly drawn from own research published in peer-reviewed journals or international conference proceedings. In detail, the chapters are based on the following publications.

Chapter 3 Sven Gedicke, Benjamin Niedermann, and Jan-Henrik Haunert. **Multi-page labeling of small-screen maps with a graph-coloring approach.** In *Advances in Cartography and GIScience of the ICA: Proc. of the Int. Conf. on Location Based Services (LBS'19)*, 2:1–8. Springer, 2019. doi: 10.5194/ica-adv-2-4-2019

Chapter 4 Sven Gedicke, Adalat Jabrayilov, Benjamin Niedermann, Petra Mutzel, and Jan-Henrik Haunert. **Point feature label placement for multi-page maps on small-screen devices.** *Computers & Graphics*, 100:66–80, 2021. doi: 10.1016/j.cag.2021.07.019

Chapter 5 Sven Gedicke, Annika Bonerath, Benjamin Niedermann, and Jan-Henrik Haunert. **Zoomless maps: External labeling methods for the interactive exploration of dense point sets at a fixed map scale.** *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1247–1256, 2021. doi: 10.1109/TVCG.2020.3030399. (© 2021 IEEE. Reprinted, with permission)

Chapter 6 Sven Gedicke and Jan-Henrik Haunert. **An Empirical Study on Interfaces for Presenting Large Sets of Point Features in Mobile Maps.** *The Cartographic Journal*, 60(1):25–42, 2023. doi: 10.1080/00087041.2023.2182354

Chapter 7 Sven Gedicke, Lukas Arzoumanidis, and Jan-Henrik Haunert. **Automating the External Placement of Symbols for Point Features in Situation Maps for Emergency Response.** *Cartography and Geographic Information Science*, 50(4):385–402, 2023. doi: 10.1080/15230406.2023.2213446

2 AUTOMATION IN CARTOGRAPHY

Digital representations of geospatial data – and especially interactive maps – have become an indispensable part of our everyday lives. With the transition from analog to digital map products, the focus of cartographic research has changed, and cartography has become a highly advanced and versatile domain (MacEachren and Kraak, 1997; Roth, 2013). Manual processes that sufficed in analog cartography become impractical when dealing with the efficiency demands and vast amounts of data that come with digital maps. Thus, automating cartographic processes is an integral issue in modern cartography.

2.1 Challenges, Methods, and Trends

The shift from analog to digital cartography and the associated paradigm shift from manual to automatic processes brings with it a lot of opportunities, but also novel challenges. Since the emergence of the first digital maps, cartographic research has been intensively engaged in exploiting the potential of the technology with regard to diverse cartographic tasks and coping with the associated challenges.

General Challenges One fundamental challenge is to ensure the quality and reliability of automated processes. While automation has the potential to speed up cartographic tasks and reduce manual labor, maintaining high standards of cartographic quality remains essential. To that end, it becomes crucial to establish metrics that reasonably quantify the quality of outputs (Tomlin, 2016). Modeling cartographic problems requires the mathematical formalization of such quality criteria and requirements. Often, established conventions and principles from traditional cartography are adapted and converted into algorithmic form. To that end, geospatial data needs to be interpreted correctly, which requires an understanding of the semantics of geographic features (Bill et al., 2022). These semantics include not only the physical attributes of the features but also their contextual meaning, which can vary due to environmental, cultural, and historical factors. To formally represent the semantics and relationships among entities, onthologies are increasingly used in the field of cartography (Touya et al., 2023b). However, establishing practices becomes challenging due to the diversity of disciplines collaborating in the field. Cartographers, geographers, engineers, mathematicians, computer scientists, and other domain experts all contribute their expertise, but this diversity can lead to difficulties in communicating and agreeing on terminologies and standards (Southard, 1987; Rieger and Coulson, 1993). Ensuring effective communication and alignment of objectives is crucial to fully exploiting the potential of automation.

Initial Approaches Towards Automation During the early phases of automation in cartography, *rule-based expert systems* (Doerschler and Freeman, 1989; Buttenfield and Mark, 1991; Mackaness et al., 1986) were the prevalent methodology to facilitate decision-making processes. Based on the knowledge of domain experts, these systems encode cartographic rules that trigger actions, i.e., specific algorithms. However, such rule-based approaches require the collection of extensive cartographic knowledge that applies to a wide variety of special cases. This limitation is also known as *knowledge acquisition bottleneck* (Weibel, 1995). *Agent systems* emerged as pioneering implementations of artificial intelligence concepts, allowing for the simulation of individual entities' behaviors within spatial environments (Nickerson, 1988; Duchêne et al., 2018). Geographical features are modeled as 'intelligent' *agents* that are able to interact with each other and their environment (Harrie and Weibel, 2007). The agents' behavior is controlled by objectives, which they iteratively strive to satisfy. In each iteration, an agent evaluates its associated objectives and, if these are not yet sufficiently met, follows a *plan* that triggers suitable algorithms.

Automation by Optimization One of the predominant strategies in contemporary cartography is the modeling of tasks as *optimization problems*. This approach allows using mathematical optimization techniques that aim to find the best solution among a set of feasible solutions, i.e., to maximize or minimize an objective function subject to a set of constraints. The objective function expresses the cartographic quality, whereas the constraints reflect strict rules that must be adhered to. While *continuous optimization* models a corresponding problem with continuous variables, *combinatorial optimization* involves discrete variables (Papadimitriou and Steiglitz, 1998), making it particularly suitable for addressing cartographic tasks with discrete characteristics. Since the models and algorithms presented in this thesis are mainly based on combinatorial optimization, we will discuss this branch in more detail in Section 2.5. Before developing a particular optimization approach, theoretical research often focuses on the computational complexity of specific cartographic problems (Marks and Shieber, 1991). This is particularly relevant to decide what kind of method is appropriate. If, for example, an optimization problem turns out to be NP-hard, it is very unlikely to find a polynomial-time algorithm (Knuth, 1974). In such a case, exact approaches, approximation algorithms, or heuristics can be reasonable strategies to tackle the problem. The latter two methods do not guarantee optimality of a solution, but they often provide near-optimal solutions within a short running time.

Map Interactivity The first thoughts on and contributions to automation in cartography related to the creation of static maps (Keates, 1989). Nowadays, the cartographic focus mainly shifted to interactive maps that allow the user to adapt the map dynamically, e.g., by zooming and panning (Roth, 2013). While static maps were designed to communicate geographical information by presenting it to a map user, the user is now able to manipulate the map and thus explore information in a self-determined way. In this context, *geovisualization* describes the relevant research field focused on techniques for exploring geospatial data via interactive visualization methods (Kraak, 2003). In a comprehensive review, Roth (2013) discusses the fundamental questions regarding the application of cartographic interaction. Generally, the ability and flexibility of a user to explore information increases with the number of provided interaction funcionalities (Chittaro, 2006; Roth and Harrower, 2008). However, a high degree of freedom in interaction is also accompanied by a high level of *interface complexity*, which can hinder the completion of specific spatial tasks (Keehner et al., 2008; Jones et al., 2009; Roth and MacEachren, 2016; Wang et al., 2018). In an empirical study on spatial decision making, Vincent et al. (2019) show that limiting the provided interaction functionalities leads to more correct decisions, lower perceived difficulty, and greater user confidence. The interactivity of contemporary maps poses further challenges for the underlying algorithms, especially concerning their running times. It is essential to generate and visualize results promptly in response to user interactions, ensuring that users do not experience substantial delays. In this context, it is known that about a tenth of a second is the limit up to which the user has the feeling that the system reacts immediately (Card et al., 1991; Nielsen, 1993). Response times exceeding one second result in users perceiving the system as sluggish, disrupting the smoothness of interaction. Furthermore, mathematical models should take into account that the visualized information changes consistently when the user interacts with the map. Particularly when switching between different zoom levels, maintaining consistency across different scales is crucial to ensure that a user can track visual changes. Users rely on consistent visual cues and spatial relationships to interpret the map correctly, regardless of scale. Inconsistencies, such as abrupt changes in the visualization, can lead to confusion and disorientation (Jul and Furnas, 1998; Touya et al., 2023a). Thus, modeling consistency is essential in many cartographic domains, including map generalization (see Section 2.3) and map labeling (see Section 2.4). Particular challenges are posed by interactive maps on mobile devices with small screens, which have become ubiquitous for location-based services such as navigation applications. Section 2.2 discusses this area of research in more detail.

Contemporary Research Trend The spectrum of applications in modern cartography, along with methods for automating corresponding tasks, is highly diverse and continuously evolving. Due to the increasing use and provision of *Volunteered Geographic Information* (VGI) (Goodchild, 2007) and geo-referenced data from social media (Stefanidis et al., 2013), new cartographic application areas are constantly emerging. Prominent examples are the mapping of human activities and their dynamics (Huang and Wong, 2016; Zhu et al., 2018; Xin and MacEachren, 2020), as well as the mapping of disaster events (Feng et al., 2022; Tavra et al., 2021). Progressing technological advancements are enabling the utilization of new media platforms for cartographic applications. In this context, *augmented reality* (AR) is becoming increasingly relevant for cartographic applications (Cheliotis et al., 2023). Modern AR-based technology allows information to be overlaid onto the user's real-world view through the camera of a mobile device. This can be used to display customized navigation guidance (Verma et al., 2020; Huang et al., 2020; Nurminen and Sirvio, 2021) or to selectively query information about real-world objects (Burigat and Chittaro, 2005; Mohammed-Amin et al., 2012; Cranmer et al., 2020). Beyond the use of AR on smartphones, cartographic research is increasingly focusing on applications for digital glasses (Zhang et al., 2021; Liu et al., 2021; Guedira et al., 2022; Lakehal et al., 2023). An emerging methodological trend in this era of cartography is the use of *artificial intelligence*, particularly through machine learning strategies such as *deep learning*. In many other domains, deep learning has already conquered the current state of the art (Shinde and Shah, 2018). In the context of cartography, Harrie et al. (2024) provide a comprehensive discussion on the opportunities and limitations of using machine learning techniques. They highlight that such techniques have the potential to cope with the challenge of formalizing cartographic knowledge, which is widely recognized as one major obstacle in the automation of cartographic tasks (Weibel, 1995). Furthermore, they stress the complexity associated with rule-based and constraint-based approaches, which require considerable effort in developing algorithmic models (Touya et al., 2010) and tuning model parameters (Zhou and Li, 2016). Machine learning offers the potential to dispense with such tasks. Especially to facilitate the adaptation to specific contexts or scales, pattern recognition is a valuable area of application (Knura, 2023; Li et al., 2023a; Xi et al., 2023). Particularly in the context of map generalization (Touya et al., 2019) and map labeling (Oucheikh and Harrie, 2024), deep learning is currently being addressed as a possible paradigm shift (see Section 2.3 and Section 2.4). Regardless of the field of application, however, key limitations of machine learning are the substantial computing resources it requires and the fact that the relationships between the input data and the output are difficult to trace.

2.2 Mobile Cartography

With the advent of the first mobile devices, cartographic research has increasingly focused on so-called *small-display cartography* or *mobile cartography*. Due to their portability, devices such as tablets, smartphones, smartwatches, and meanwhile even smart glasses offer new possibilities for cartography in mobile operating environments. *Location-based services* have become ubiquitous, e.g., for navigation assistance, route tracking, social networking, and gaming (Huang et al., 2018). The areas of application are becoming increasingly diverse, and technical progress is enabling ever more sophisticated functionalities.

2.2.1 Mobile Map Design

Despite the opportunities mobile devices bring for cartographic applications, new challenges arise compared to visualizations and methodologies for desktop environments (Muehlenhaus, 2013). Technical limitations and the highly variable physical environment when using mobile devices mean that cartographic applications designed for desktop systems cannot be transferred directly to mobile devices (Chittaro, 2006). New paradigms for information presentation and modes of interaction are needed that adapt to the small devices.

Technical Challenges One key limitation is the restricted screen size of mobile devices, constraining available space and flexibility for presenting maps and associated information. It is particularly difficult to display detailed maps without compromising clarity, i.e., without producing clutter (see Section 2.2.2). In addition, the lack of map space hinders the inclusion of auxiliary elements such as a legend, which complicates the map's interpretability (Reichenbacher, 2004). Another difficulty arises from the typically lower computing power and memory available on mobile devices compared to desktop computers. This can be particularly problematic as mobile maps are usually highly interactive and long running times of the underlying algorithms can lead to noticeable delays for the user. One solution might be client-server communication, in which algorithms are executed on external servers. However, this approach requires stable network availability, which can not be guaranteed in all physical environments (Meng, 2005). The diverse range of screen sizes, formats, and hardware specifications across devices, as well as the shift in user interaction from click to touch input, pose further challenges for mobile cartography. Beyond the established zoom-and-pan interactions, research increasingly focuses on enhancing these existing standards. Particularly, interaction strategies have been proposed that are specialized for one-handed mobile use (Lai et al., 2017; Boring et al., 2012; Hasan et al., 2016) and very small smartwatch displays (Singh et al., 2018; Neshati et al., 2021; Stanke et al., 2022). In recent developments, even gesture-based (Jiang et al., 2021) and gaze-based (Lei et al., 2023) approaches have received a lot of attention.

Adaptive Design Beyond the rather technical challenges, the consideration of contextual factors is crucial in the design of cartographic applications in the mobile context (Huang et al., 2018). A key objective is to reduce interaction complexity through service adaptation and to offer the user context-aware and personalized information (Raubal and Panov, 2009). Generally, this concept is referred to as *egocentric design* (Meng, 2005) or *adaptive design* (Pombinho et al., 2015). Considerable research effort has been dedicated to defining and classifying diverse contextual factors, as well as exploring how mobile map applications can adapt to these aspects (Reichenbacher, 2003; Meng, 2005; Pombinho et al., 2015; Bartling et al., 2021). However, despite the plethora of contributions, there remains a lack of consensus regarding the term context and its appropriate modeling (Griffin et al., 2017). Griffin et al. (2017) categorize context into the four interconnected components *user*, *environment*, *activity*, and *map*. The user component includes personal aspects such as the social situation, mental state, abilities, skill, and experience. The considerable variability of the user's physical environment – caused by the nature of mobile use – is another component that includes factors such as changing lighting conditions, noise levels, and external distractions. Covered by the activity component are the purpose of map use, incidental actions performed while using the map, and the mode of movement (i.e., on foot, by bike, or by car). Aspects such as the device used, the type of data, and the interaction capabilities are included in the map component. Especially with regard to the variety of mobile devices, Roth et al. (2018) argue for the necessity of a *responsive design* that automatically adapts the visualization of a mobile map application to the device used. The survey of Akpinar et al. (2020) provides a comprehensive review of empirical literature examining the impact of various contextual factors on user performance.

User-Centered Design There is a unanimous opinion that the needs and perceptions of a user should be a central concern when developing mobile map interfaces (Roth et al., 2015; Tsou, 2011; Nivala et al., 2005). In this context, *user-centered design* (UCD) describes an early focus and active integration of the user in the design process (Abras et al., 2004). Typically, UCD includes *utility* and *usability* measures (Bartling et al., 2021; Roth et al., 2015). The term utility encompasses characteristics such as suitability, usefulness, and fitness for purpose. However, utility says nothing about how easy an interface is to use. In this context, usability describes the extent to which a user can easily get to know an interface, use it efficiently, and achieve their intended goals satisfactorily (Grinstein et al., 2003). Directly related to usability is the *user experience* (UX), which describes a user's preferences and emotional perception of an interface (Hassenzahl, 2005). Design considerations are crucial for UX and therefore an own complex and evolving area of research across multiple disciplines (Roth, 2017; White, 2017). One of the most basic strategies in UCD is conducting user tests in which participants perform specific tasks with the interface under test (Nivala et al., 2005). By recording quantitative metrics during the execution of tasks, the performance of the participants and thus the utility of the interface can be assessed. Further, interviews and questionnaires are established strategies to gain additional insight into a user's perceptions and hence to capture user experience (Laugwitz et al., 2008). A fairly modern strategy for assessing UX is the use of eye tracking (Bojko, 2013). Monitoring a user's gaze makes it possible to recognize frequently fixated elements and draw direct conclusions about mental attention (Webb and Renshaw, 2008; Duchowski, 2017). With respect to all these evaluation strategies, UCD is a highly iterative process in which the insights gained about utility and usability are used to refine the tested interface (Nielsen, 1993; Robinson et al., 2005). Specifically concerning research on interactive maps and visualizations, Roth et al. (2017) emphasize the importance of empirical studies and provide methodological guidance.

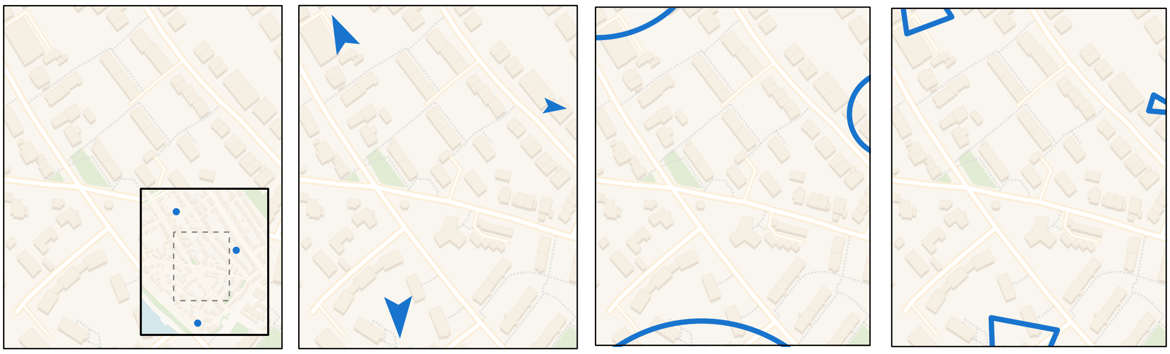
2.2.2 Legibility in Mobile Maps

Especially when dealing with high information densities, it is challenging to create small-screen maps that ensure legibility without omitting information (Muehlenhaus, 2013). A central concern of mobile cartography is to find a balance between the provision of comprehensive geographical information and clear visualization. The high interactivity of mobile maps enables a variety of techniques and strategies that adapt the presentation of information to small screens.

Clutter Reduction Due to the limited space for visualization, maps on mobile devices easily become cluttered. Presenting too much information at the same time causes an information overload (Eppler and Mengis, 2004). Users are overwhelmed and cannot effectively process the information presented on the map. Especially when displaying annotated maps, a high information density leads to a high map complexity (Liao et al., 2019) and a reduced map readability (Harrie et al., 2015). Thus, clutter reduction is an essential task for providing high-quality mobile maps. A systematic analysis of existing clutter reduction strategies is carried out by Ellis and Dix (2007). They provide a taxonomy that highlights and discusses the strengths and weaknesses of each technique. Korpi and Ahonen-Rainio (2013) adapt this taxonomy with the goal of identifying methods that are suitable for reducing clutter in interactive map mashups. Focusing on annotated point data, they establish a link between cartographic generalization operators and clutter reduction strategies. Especially when adapting visualizations to devices with small screens, generalization operators can be used to simplify the presentation of complex information while retaining its essential aspects. In this regard, techniques for data abstraction such as displacement (Tauscher and Neumann, 2016; Opach et al., 2019) and aggregation (Huang and Gartner, 2012; Elmqvist and Fekete, 2009; Opach et al., 2019) have been suggested to reduce clutter. Another straightforward strategy is reducing information complexity by selection. Usually, selection is based on filtering the data, i.e., on querying a set of map features (Korpi and Ahonen-Rainio, 2013). Unlike sampling, where a random selection of spatial features is made, filtering is based on specific criteria, e.g., the features' relevance.

Effect of Desert Fog While reducing or abstracting spatial information improves the map readability (Phillips and Noyes, 1982), it can lead to the omission of information relevant to the user. Therefore, strategies for clutter reduction are often accompanied by interaction techniques such as zooming and panning (Heidmann et al., 2003; Korpi and Ahonen-Rainio, 2013). When a user zooms into an area of interest, the density of information is reduced. This enables the display of objects that were previously hidden without any overlaps. However, by zooming in, visual access to information in the large-scale context outside the viewing area gets lost. Hence, to situate specific information in the large-scale context, the user must repeatedly zoom in and out and pan the map section (Schmid et al., 2010). This process is not only tedious and cumbersome, thus reducing the user's satisfaction (Burigat and Chittaro, 2011), but the accompanying changes in scale and level of detail also impede the ability to cognitively link successive visualizations. Visual cues may alter in their appearance or even vanish entirely, posing challenges to their function as points of orientation. Consequently, the user needs some time to get oriented again after each interaction; an effect known as *desert fog* (Jul and Furnas, 1998). Dumont et al. (2020) investigate which aspects of the map design affect the cognitive load during zooming and how these can be used to mitigate the effect of desert fog. Touya et al. (2023a) argue that disorientation arises from a discrepancy between the current map view and the user's mental map, emphasizing the crucial role of landmarks as orientation points. Based on this argumentation, Touya et al. (2021) introduce a research project which aims at developing new strategies that reduce or even avoid desert fog. Salient landmarks are considered as *anchors* that are visible across consecutive scales and are highlighted when zooming.

Indicating Off-Screen Objects Early approaches tackle the issue of losing contextual information when zooming by using *overview+detail* visualizations (see Figure 2.1a). With this approach, one or multiple small-scale overviews are displayed as small thumbnails superimposed on the large-scale map section (Plaisant et al., 1995). This makes it possible to access information that is not located in the current map section. Although studies have proven the general user acceptance (Hornbæk et al., 2002) and effectiveness of this strategy in completing spatial search tasks (Pietriga et al., 2007) for desktop environments, it is less suitable for mobile devices. The limited space on the screen poses a challenge both for displaying the views simultaneously and for establishing an association between them (Chittaro, 2006). Alternative techniques were proposed which indicate the location of important off-screen objects at the border regions of the map, i.e., by providing so-called *contextual cues* (Burigat and Chittaro, 2011). Approaches have been developed that use scaled icons (e.g., arrows) that point to the position of off-screen objects and that convey distance information through size, length, or color coding (Burigat et al., 2006; Gustafson and Irani, 2007). Figure 2.1b shows an example for this strategy. Aiming to improve distance indication, Baudisch and Rosenholtz (2003) introduced an approach called *Halo*, in which the positions of objects beyond the screen are indicated by encircling them with circles



(a) overview+detail **(b)** scaled arrows **(c)** Halo **(d)** Wedge

Fig. 2.1: Different strategies for indicating off-screen objects. *Map attribution: Carto.*

sized to reach into the boundary region of the shown map section (see Figure 2.1c). However, Halo tends to produce cluttered configurations with overlapping arcs. To tackle this problem, Irani et al. (2006) suggest using oval halos, and Gustafson et al. (2008) suggest a technique called *Wedge*, which uses pointed triangles instead of circles (see Figure 2.1d).

While all of these approaches share the common idea of using visualizations to indicate the distance and direction of locations beyond the screen, they do not account for semantic information. Addressing this issue, Li et al. (2014) proposed an approach that uses cartographic symbols to represent the identities of off-screen locations. Li (2020) further refined this method by manipulating visual variables (i.e., icon size, fuzziness, and transparency) to effectively convey distance information. Both of these approaches have been shown to enhance spatial learning and improve spatial orientation, particularly for people with a low sense of direction.

With the development of mobile technology, strategies for displaying off-screen objects have expanded for augmented reality (AR) and virtual reality (VR). Corresponding AR and VR techniques have been developed that adopt the concepts of overview+detail (Carmo et al., 2020), directional arrows (Schinke et al., 2010; Biswas et al., 2023), Halo (Perea et al., 2017, 2019; Gruenefeld et al., 2018), Wedge (Gruenefeld et al., 2018), and icon-based visualizations (Li, 2023). Empirical research further focused on comparing these different strategies in varying settings (Afonso et al., 2023; Wieland et al., 2022). While most approaches focus on indicating off-screen point data such as points of interest, visual cues have also been employed in other cartographic domains, e.g., for visualizing the off-screen evolution of trajectories (Forsch et al., 2022).

Focus+Context Maps Instead of indicating off-screen objects when the map is zoomed in to a large scale, alternative approaches focus on visualizations that minimize the need for zooming; a strategy also adopted in this thesis in the development of zoomless maps. Existing methods are based on the idea of displaying a user's area of interest at a larger scale than the surrounding map that shows the broader context (Boutoura, 1994; Fairbairn and Taylor, 1995; Harrie et al., 2002; Li, 2009). Similar to excentric labeling (see Section 2.4.3), this area of interest is denoted as *focus region* and can be interpreted as a circular (or rectangular) lens that can be shifted over the context map (Guerra and Boutoura, 2001). Using a fish-eye projection, a smooth transition between the focus and context region can be achieved (Furnas, 1986). Such *focus+context maps* provide more space for visualizing information in the focus region while the spatial relations to the context remain visible. However, focus+context maps tend to be highly distorted. Several strategies have already been proposed to reduce these distortions. Yamamoto et al. (2009) mitigate the negative effects of distortion by representing the transition between the focus and context region as an additional intermediate *glue region*. It absorbs the distortions and ensures that both the focus and context region retain a fixed scale. Haunert and Sering (2011) argue that the distortions should preferably occur in parts of the map with a low density of map features. They present a graph-based optimization approach, extended by van Dijk and Haunert (2014) with regard to its applicability for interactive scenarios. Enabling even more interactivity, Zhao et al. (2020) present a method that allows users to specify the size, shape, and number of focus areas. Göbel et al. (2020) suggest an approach that combines focus+context maps with gaze-adaptive interactions. Based on the visual focus of a user, additional information is displayed. In a pilot study, they found that gaze-adaptive interactions can lead to an increase in usability compared to common mouse-based interactions.

2.3 Map Generalization

When deriving a smaller-scale map from a given map, *map generalization* or *cartographic generalization* describes the process of abstracting and reducing spatial information to maintain legibility. This becomes particularly crucial for maps on mobile devices, where the restricted screen space necessitates a careful consideration of the information presentation. To achieve a true-to-life representation, the content of a map would have to be reduced in size in proportion to the decreasing scale. However, due to space and resolution limitations, this procedure would inevitably lead to illegibility at a certain scale. Hence, when moving to a smaller scale, it becomes necessary to generate an abstracted and less intricate representation of the spatial data. One straight-forward strategy is to omit map objects of minor importance. Although this means that some information is neglected, it creates more space for visualizing objects that reflect the key characteristics of the map. However, the process of generalization goes beyond the simple omission of information and particularly involves the modification of the representation of spatial features. Such modifications allow important geographic information to be retained and conveyed in a simplified and more accessible way without losing the overall meaning or context of the data. By adjusting the way spatial features are represented rather than removing them entirely, map generalization strives to maintain a balance between reducing detail and retaining the information necessary for effective map communication. To achieve this objective, the general task is to identify the regions and entities to be generalized and to apply suitable *generalization operators* (Regnauld and McMaster, 2007). An operator describes the type of a transformation, which can be implemented by applying a suitable *generalization algorithm* (Weibel and Dutton, 1999).

Generalization Operators Since the beginning of research in the field of automatic map generalization, many authors have proposed different types and definitions of generalization operators as well as frameworks and workflows for their deployment. In this context, Roth et al. (2011) provide a comprehensive literature review and a survey of existing taxonomies of generalization operators. In one of the first formal models of the generalization process, Robinson et al. (1978) propose an approach that is divided into two parts. In a first pre-processing step, the operator of *selection* is used to identify important map features that should retain in the generalized map. In the subsequent generalization, the operators *simplification*, *classification*, *symbolization* and *induction* are applied to these features. By simplification, Robinson et al. (1978) understand the removal of unnecessary details while retaining the character of a feature. They describe classification as the division of spatial entities into categories, symbolization as the graphical abstraction of features, and induction as the derivation of further informative graphics from the data. With the work of Robinson et al. (1978) as an impetus, a variety of taxonomies were developed in the following years that adopt, supplement, and refine the established operators (DeLucia and Black, 1987; McMaster and Monmonier, 1989; Slocum, 1998; Yaolin et al., 2001; Regnauld and McMaster, 2007; Roth et al., 2011). In one of the most popular taxonomies, McMaster and Shea (1992) identify ten spatial operators and two additional operators at the attribute level. While

Tab. 2.1: Generalization operators according to McMaster and Shea (1992).

spatial	
simplification	reduction of data points of a feature while retaining its characteristics
smoothing	relocation of data points to remove perturbations along a line
aggregation	combination of point features to form a continuous area
amalgamation	combination of area features without modifying the dimensionality
merging	combination of line features without modifying the dimensionality
collapse	down-conversion of dimensionality (e.g., area to symbolized point feature)
refinement	reduction of the number of features by displaying representative subsets
exaggregation	amplification of shape and size of features to emphasize them
enhancement	alteration of a feature's symbology
displacement	offset feature positions to avoid conflation with other features
attributive	
classification	grouping of features sharing similar attribution
symbolization	graphical abstraction of features based on their characteristics

the spatial operators refer to transformations that change the representation of a feature from a geographical or topological perspective, the attributive operators refer to manipulations of a feature's statistical properties. A list with definitions of the operators is given in Table 2.1. One special characteristic is that the operators are differentiated according to the dimensionality of the features they address. In particular, *aggregation*, *amalgamation*, and *merging* describe a similar type of process (i.e., combining features), but relate to point, line, and area features, respectively (McMaster and Shea, 1992). In many textbooks, this distinction is not made and the term aggregation is used collectively among all spatial dimensions (e.g., see Hake et al. (1994)).

Difficulties of Standardization and Orchestration The scientific cartographic discourse is characterized by diverse interpretations such as exemplified above. Not only do authors differentiate or summarize operators differently but they also employ diverse definitions for identical terms or use different terms for the same definition (Rieger and Coulson, 1993; Gould and Mackaness, 2016). Despite the extensive body of research, a unanimous consensus on the definition of generalization operators remains elusive. This lack of standardization can be an impediment to the successful automation of the generalization process. Without a universally agreed-upon set of principles, automated systems may struggle to interpret and apply operators consistently. However, even if we presuppose that there is a set of operations that cartographers are willing to agree upon, orchestrating these generalization processes is a particularly challenging task (Touya et al., 2010). The individual spatial transformations caused by the generalization requirements often influence each other. This interdependence between the operators demands specific instructions as to which algorithms are to be used for which procedures and in which order. One challenge in establishing such a workflow is that generalization is a highly subjective process in which more than one 'ideal' result is possible (McMaster, 1987). Automated approaches struggle to capture the nuanced preferences and contextual considerations that human cartographers contribute. In this regard, Regnauld and McMaster (2007) state that operators and algorithms should provide metadata that enables an automated system to operate efficiently and adapt dynamically to the context of use. Gould and Mackaness (2016) argue that – to facilitate the automatic selection of operators and algorithms – it is essential to encapsulate cartographic knowledge through the use of ontologies. Unambiguous rules in machine-readable form are necessary to give precise instructions for specific input that produce a desired output, i.e., the generalized map.

Modeling the Generalization Process The first attempts to automate and, thus, model the entire generalization process were made in the 1980s. The conceptual approach pursued at the time is commonly known as *condition-action modeling* (Harrie and Weibel, 2007). The approach consists of two phases: a structural knowledge acquisition and a subsequent execution based on a set of rules. Within the knowledge acquisition, object conditions, i.e., structures and relationships between map features, are recognized. Based on this knowledge, applicable rules are identified which trigger specific generalization actions (i.e., algorithms). One key limitation of condition-action modeling is the diversity of constellations of map features, which requires an immense number of specific rules (Müller et al., 1995). Such rules are usually based on professional judgment, which – as mentioned earlier – is often subjective and intuitive and therefore difficult to formalize. Given these limitations, a prevalent opinion in the early 1990s was to step back from the full automation of the generalization process. In this context, Weibel (1991) suggested the concept of *amplified intelligence*, which describes the use of computational tools as a supplement to the decision-making capabilities of human cartographers. Corresponding generalization systems relied on *human interaction modeling*, in which the human component provides the expertise required to make informed decisions about which features to process and how, while the computer provides the processing power and accuracy required for the respective generalization operation. While such semi-automated systems enable the qualitative evaluation of generalization results, cartographers still have to control the operations and, thus, invest a lot of time in the overall process. The limitations of both rule-based and semi-automated workflows led to the development of a novel conceptual strategy known as *constraint-based modeling*.

2.3.1 Constraint-Based Modeling

The essential philosophy behind constraint-based modeling is that a readable cartographic output requires an accurate description of what one wants to achieve through the generalization process (Burghardt et al., 2007). To express such a description, Beard (1991) was one of the first authors that argued for the use of *constraints* instead of rules. While rules specify the actions to be performed, constraints specify required conditions of the desired output. Instead of adhering to a particular sequence of condition-action rules, constraint-based modeling asks for a state that satisfies a given set of constraints, which allows a high degree of flexibility with regard to the generalization process.

Generally, a distinction is made between two types of constraints: *readability constraints* and *representation constraints*. This distinction reflects the key concern of generalization, which is to find a balance between the readability of the map and the preservation of information. Thus, readability constraints refer to the preservation of the visual recognizability and legibility of a map, while representation constraints limit deviations of the displayed information from reality. Beyond these two main categories, different typologies have been suggested that further differentiate between constraints. Based on the work of Ruas and Plazanet (1996), the work of Harrie and Weibel (2007) identifies six categories, which are listed and described in Table 2.2.

A key requirement for constraint-based modeling is that the constraints are tied to metrics that

Tab. 2.2: Categories of generalization constraints according to Harrie and Weibel (2007).

constraint type	
position	maintaining the absolute and relative positioning of map features
topology	maintaining topological relationships
shape	maintaining key characteristics of map features
structural	maintaining patterns of map features
functional	maintaining functionality with regard to the map purpose
legibility	adherence to graphical limits and prevention of visual conflicts, overlaps, overly small representations, etc.

provide quantifiable indicators of how well these constraints are satisfied. For example, area sizes can specify the minimum dimensions of polygonal objects in the target scale or distances can specify the maximum distance by which an object may be displaced. By incorporating such metrics, constraint-based modeling integrates an assessment mechanism for controlling the generalization process. This is particularly relevant as quality assessment has been recognized as an inseparable part of generalization (Ehrliholzer, 1995; Weibel, 1995). However, constraints often compete with and exclude each other, which can make it impossible to find a solution that satisfies all constraints. The strategy for overcoming this issue is to relax constraints so that compromises between conflicting requirements can be found. The task is then to find a state that minimizes the degree of violation of the relaxed constraints while satisfying all non-relaxed constraints.

Thus, constraint-based modeling inherently enables the application of optimization strategies to address the overall generalization task. In particular, *mathematical optimization* is an established strategy for solving generalization problems (Sester, 2005). Furthermore, *multi-agent systems* have been applied to several generalization tasks (Duchêne et al., 2018), providing the impetus for continuing research towards artificial intelligence.

2.3.2 State of the Art

The previous sections have discussed the fundamentals of automatic map generalization that have emerged since the beginning of the digital age. In this section, we will now look at more recent scientific advances and highlight the current state of research.

The majority of attempts towards holistic generalization approaches have been initiated by National Mapping Agencies (NMAs) to improve their map production lines (Duchêne et al., 2014). In collaboration with research institutes, workflows have been developed that are mainly concerned with the generalization of topographic maps (Revell et al., 2011; Stoter et al., 2014; Blana et al., 2023). Typically, such workflows orchestrate existing generalization tools in commercial software. Revell et al. (2011) focus on producing a background map for Great Britain that is specifically tailored for use on the web and establish a process chain for the generalization of the map. Similarly, Stoter et al. (2014) provide a fully automated workflow to generalize a 1:50k map from 1:10k data of the Dutch cadastre. Most recently, Blana et al. (2023) provide a constraint-based approach for generalizing line and area features in topographic maps that includes a quality control mechanism.

Addressing such holistic approaches, Stoter et al. (2010) identify two main issues: the difficulty of tuning algorithm parameters and the lack of automatic recognition tools that are able to distinguish between specific contexts. Further, as highlighted by Gould and Mackaness (2016), most of the developments have been scale-specific and are therefore only generalizable to a limited extent. By now, the demand for automatic generalization approaches has expanded beyond the restriction to topographic maps to include a broad spectrum of thematic map types (Raposo et al., 2020). Current research includes the generalization of tactile maps (Wabiński and Mościcka, 2019; Štampach and Mulíčková, 2016), sketch maps (Manivannan et al., 2022; Tversky, 2014), land-cover data (Thiemann and Sester, 2018; Haunert and Wolff, 2010), metro maps (Dong et al., 2010), as well as geological maps (Smirnoff et al., 2012; Sayidov et al., 2020).

The increasing diversity of applications and the issues mentioned above are triggering considerations for a potential paradigm shift in map generalization. In this context, Touya et al. (2019) discuss whether deep learning is the innovative strategy with which research will close the gap to fully automated map generalization. One compelling aspect that supports the use of deep learning in map generalization is the availability of extensive training data for neural networks in the form of existing maps. Through its training process, a deep learning model is able to acquire implicit knowledge by extracting relevant features and representations from the input data. However, as a model learns to generalize based on examples from a specific place and scale, it might be limited in its ability to adapt to different geographical scenarios. Another challenge lies in the fact that the majority of current deep learning approaches are based on image processing. To adopt such approaches, geographical vector data would have to be rasterized, which is accompanied by a reduction in information and the occurrence of fuzziness (Liao et al., 2012; Touya et al., 2019). Addressing this issue, an ongoing project of Knura (2021) aims to develop a deep learning model for map generalization that is based on the processing of vector data.

Although deep learning has not yet been successfully applied for the holistic modeling of the generalization process, it has established its role as an approach for specific tasks within the orchestrated generalization model. Focusing on road generalization, research addressed the automatic selection of important roads and road sections (Karsznia et al., 2023; Zhou and Li, 2017; Touya and Lokhat, 2020), their simplification (Courtial et al., 2020, 2023), and the evaluation of corresponding results (Courtial et al., 2022). In addition, deep learning was applied for building classification (Knura, 2023; Yan et al., 2021) and generalization (Sester et al., 2018; Feng et al., 2019; Fu et al., 2023; Yan and Yang, 2023), as well as for the generalization of coastlines (Du et al., 2022; Jiang et al., 2023).

Addressing interactive maps, many scientific contributions are concerned with maintaining consistency in map generalization. To ensure that a user can seamlessly track visual changes when zooming the map, consistency criteria must be met across different scales. It is particularly important to prevent objects from constantly appearing and disappearing during zooming, an effect known as *flickering*. In this context, *continuous map generalization* (CMG) aims to ensure that visualizations change smoothly when switching between scales. Over the past years, approaches have been introduced to continuously or gradually generalize buildings (Sester and Brenner, 2004; Fisher et al., 2005; Li et al., 2017b; Peng and Touya, 2017; Touya and Dumont, 2017; Rottmann et al., 2021; Peng et al., 2023), road networks (Suba et al., 2016; Chimani et al., 2014), river systems (Peng et al., 2012; Huang et al., 2017), and polygonal regions (Danciger et al., 2009; van Oosterom and Meijers, 2014; Peng et al., 2016, 2017, 2020). Furthermore, authors developed strategies for morphing polylines (Nöllenburg et al., 2008; Peng et al., 2013; Deng and Peng, 2015; Gao et al., 2020; Qian et al., 2022; Li et al., 2017a; Li and Mao, 2023).

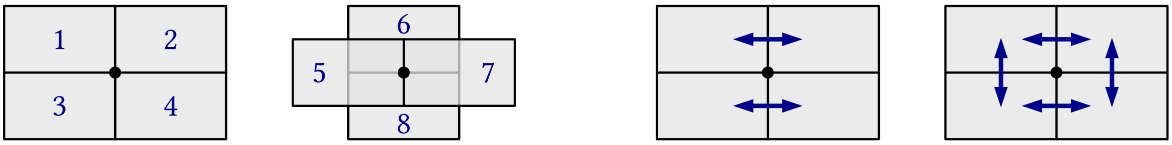
While there is a vast amount of research on particular generalization operations tailored to specific feature types (e.g., clustering point features or aggregating building footprints), approaches of *collaborative generalization* account for the synchronization of processes and, in particular, for geographic correlations among different map features (Touya et al., 2010). A frequently addressed task in this context is the generalization of buildings while avoiding spatial conflicts with the road network (Wei et al., 2018; Pilehforooshha et al., 2021; Li et al., 2023b; Ma et al., 2023). Typically, operators such as displacement, aggregation, elimination, and simplification are coordinated taking into account the spatial relationships between buildings and their relative position in the road network. Another use case is the generalization of the road network while maintaining road connections to and between important point features (e.g., health facilities) and residential areas (Touya, 2010; Samsonov and Krivosheina, 2012; Karsznia et al., 2022; Lyu et al., 2022). In Section 2.6, we present own research that also contributes to the domain of collaborative generalization, introducing an approach for aggregating land-use polygons constraint by line features such as roads.

2.4 Map Labeling and Symbol Placement

Related to map generalization is the process of *map labeling*, in which spatial features are annotated with relevant information. This important step ensures that geographic elements are clearly identifiable and comprehensible, which improves the overall usability and clarity of the map. Map labeling can be traced back to the concept of *map lettering*, wherein textual information is added to a map to identify and describe geographic features (Yoeli, 1972; Imhof, 1975). Text annotations are strategically placed on the map to communicate essential details about locations, including the names of cities, rivers, mountains, and other landmarks. Such annotations are not only an instrument for carrying information, but also have a substantial influence on the aesthetic appearance and thus on the overall quality of a map (Ahn and Freeman, 1983). While the term map lettering highlights the historical association with the manual rendering of letters and characters on maps, contemporary map labeling has become established as the discipline that, in addition to the placement of text, also addresses the placement of descriptive symbols, icons, logos (Zhang and Harrie, 2006; Niedermann and Haunert, 2019; Harrie et al., 2022), and even small images (Wu et al., 2011).

Throughout the history of cartography, map labeling has always been a demanding task that used to be accomplished manually by cartographers. This process is known to be highly time-consuming and can account for more than half of the total production time of analog maps (Morrison, 1980). However, technological progress and the associated digitalization have unlocked the potential to considerably reduce this workload. With the advent of the first digital maps, cartography has become more and more concerned with the automation of map labeling.

Labeling Models Generally, map labeling relies on cartographic principles that express the quality of the label placement. Foundational guidelines have been introduced by cartographers like Alinhac (1963) and Imhof (1962, 1975) covering key aspects such as *legibility*, *unambiguity*, *representation of hierarchies*, and *label distribution*. One of the key challenges in automating the labeling process is transferring the existing cartographic knowledge into a machine-readable form. To evaluate the quality of a label position, measures must be established that quantify the different quality criteria. Yoeli (1972) was the first to tackle this challenge, proposing an algorithm based on the guidelines of Imhof (1962). In his approach, a finite set of potential positions is established around each map feature, defining the eligible options for placing the label. Over time, this concept evolved into what is now known as the *fixed-position model* (see Figure 2.2a). Well-known variants are the 2- and 4-position models introduced by Formann and Wagner (1991), as well as the 8-position model (Christensen et al., 1995). Allowing additional freedom in placement, Hirsch (1982) introduced the *slider model* in which the labels can be



(a) 4- and 8-position model

(b) 2- and 4-slider model

Fig. 2.2: Variants of (a) the fixed-position model and (b) the sliding model.

moved continuously in one or more directions (see Figure 2.2b). This relaxation of the restriction to fixed positions facilitates the resolution of cartographic conflicts (e.g., label overlaps). Common variants are the 1-, 2-, and 4-slider models introduced by van Kreveld et al. (1999). In the course of further research, extended concepts of both the fixed-position and sliding model have been developed. Such variants allow an adjustment of the label sizes (Strijk and Van Kreveld, 2002; Iturriaga and Lubiw, 2003), a label rotation (Doddi et al., 1997), or a specific distance between the feature and the label (Wu et al., 2017; Meng et al., 2015; Li et al., 2016).

2.4.1 Point Feature Label Placement

Most scientific contributions and developed algorithms are specifically concerned with the labeling of point features. In general, the *point feature label placement* (PFLP) refers to the task of placing labels adjacent to their point features such that the cartographic quality of the map is maximized (Christensen et al., 1995). It is often required that the labels must not overlap or that their overlap is minimized. A common way to model such problems is to build a *conflict graph* $G = (V, E)$ based on the set of label candidates, i.e., each label candidate is a vertex in G and there is an edge between two vertices if the corresponding labels overlap or they belong to the same feature. Hence, the problem of finding an overlap-free label placement maximizing the number of labeled features, also known as LabelNumberMaximization, reduces to identifying a *maximum independent set* (MIS) in G (Wagner and Wolff, 1998). To additionally take other cartographic criteria into account, the label candidates can be assigned weights that quantify cartographic qualities. The aim is then to find an overlap-free subset of label candidates that maximizes the sum of the weights. In the literature, this approach is known as *maximum weighted independent set* (MWIS) (Pardalos and Xue, 1994). Even for the simple case that each feature has one label candidate and the labels are unit squares, computing such an independent set is NP-hard (Fowler et al., 1981). PFLP also includes problems which require each feature to be labeled. Well-known variants are FreeLabelMaximization, which admits overlapping labels and maximizes the number of non-overlapping ones, and LabelSizeMaximization, which prohibits overlaps and maximizes the label size. Both of these variants are proven to be NP-hard (Formann and Wagner, 1991; de Berg and Gerrits, 2012). Due to its inherent complexity, PFLP is often approached through optimization strategies and, in particular, by means of combinatorial optimization.

Combinatorial Optimization for PFLP Given the complexity results of PFLP, theoretical research focused on the development of approximation algorithms to tackle these challenging optimization tasks. Initial constant-factor approximations for LabelNumberMaximization were developed under the assumption of unit squares, considering both the fixed-position model (Agarwal et al., 1998; Chan, 2004) and the slider model (van Kreveld et al., 1999). Ongoing research is continuing to improve the quality of the approximation algorithms considering arbitrarily sized rectangles, not only for the unweighted case (MIS) (Chalermsook and Chuzhoy, 2009; Chuzhoy and Ene, 2016; Adamaszek and Wiese, 2013; Gálvez et al., 2022; Mitchell, 2021) but also for the weighted case (MWIS) (Poon et al., 2004; Erlebach et al., 2010; Chan and Har-Peled, 2009; Chalermsook and Walczak, 2021). For LabelSizeMaximization, Formann and Wagner (1991) proposed a 2-approximation algorithm assuming the 4-position model, which was further improved by Wagner and Wolff (1997). Theoretical exploration has extended to various special cases of LabelSizeMaximization including sliding labels (Qin and Zhu, 2002), circular labels (Jiang et al., 2005; Jiang, 2006), rotating labels (Doddi et al., 1997), and a variant where a pair of labels must be placed for each feature (Jiang et al., 2005; Jiang, 2006; Zhu and Poon, 1999; Qin et al., 2000). With regard to FreeLabelMaximization, approximation algorithms have been devised for different variants of both fixed and sliding labels (de Berg and Gerrits, 2012; Higashikawa et al., 2021). Besides these rather theoretical contributions, many authors approached PFLP by means of either heuristics or exact approaches. For a more detailed overview of the related literature, please refer to the work of Rylov and Reimer (2014) and Marín and Pelegrín (2018). The most established heuristic strategies include simulated annealing (Christensen et al., 1995; Zoraster, 1997; Zhang and Harrie, 2006; Rylov and Reimer, 2014, 2015; Li et al., 2016) and genetic algorithms (Verner et al., 1997; Raidl, 1998; Yamamoto and Lorena, 2005; Bae et al., 2011; Gomes et al., 2016). With the aim of finding a mathematically optimal solution for the placement of labels, Zoraster (1986) pioneered the formalization of an integer linear program (ILP) . Different candidate positions are assigned costs according to their cartographic priority, and the objective is to find a cost-minimal set of labels without any overlaps. In follow-up work, Zoraster (1990) suggests a Lagrangian relaxation to obtain close-to-optimal results. Successive studies adopted the strategy of formulating label placement as ILP and leveraging Lagrange heuristics to solve large instances (Ribeiro and Lorena, 2008; Ribeiro et al., 2011; Mauri et al., 2010). While the majority of mathematical programming formulations focus on variants of the fixed-position model, Klau and Mutzel (2003) present an ILP specifically tailored to express LabelNumberMaximization for sliding labels. Their approach is based on constraint graphs that encode the positioning relations in vertical and horizontal direction.

Multi-Criteria Modeling Beyond the general PFLP variants, many algorithmic contributions extend the basic problems by introducing multi-criteria models. These aim to improve the cartographic quality and versatility of the label placement. To holistically quantify the quality of label positions, Edmondson et al. (1996) pioneered in computing one single score that summarizes different cartographic metrics. In an even more comprehensive objective function, van Dijk et al. (2002) quantify most of the established cartographic aspects such as aesthetics, feature and label visibility, and unambiguity. In the same year, Huffman and Cromley (2002) proposed a multi-criteria ILP that extends LabelNumberMaximization with criteria of position preference, feature importance, and label size. Zhang and Harrie (2006) introduce a combinatorial methodology for annotating a map with text and icons simultaneously considering label overlaps, label-feature association, and the occlusion of cartographic content. The latter criterion is also taken up by Luboschik et al. (2008), who present a particle-based strategy that prevents the labels from obscuring other visual elements of the map. Bae et al. (2011) treat both discrete and continuous label candidates and introduce a cost function that accounts for label-label conflicts, label-feature conflicts, and positional preferences. With the aim of annotating not only a subset but all map features, Gomes et al. (2013) maximize the minimum distance between the placed labels to reduce clutter. A more comprehensive model is proposed by Rylov and Reimer (2014), who identify nine cartographic principles for designing the metrics in their objective function. Haunert and Wolff (2017) extend the model of Rylov and Reimer (2014) and introduce an ILP that additionally considers the influence of labels on other labels. More precisely, they avoid ambiguous label-feature associations and labels that are packed too densely. With a closely related objective of integrating ambiguity as an indicator of solution quality, Marín and Pelegrín (2018) introduce different ILPs that model possible ambiguities of feature and label pairs. Further accounting for maps with high graphical density, Rylov and Reimer (2015) propose a raster-based model that takes into account visual contrast, layer hierarchy, spatial distribution, and background homogeneity. Similarly, Cao et al. (2023) address the label placement of dense and complex point data sets and propose an algorithm that considers label correlations and the spatial distribution of the features.

2.4.2 Dynamic Map Labeling

With the use of mobile devices, the demand for dynamic representations in digital maps has become increasingly relevant. When dynamically changing the map through interactions such as zooming and panning, the label placement has to be adjusted to the current map scale and section of view. The high level of interactivity necessitates dynamic labeling methodologies capable of generating solutions on demand; ideally without causing noticeable latency for the user. Consequently, research in map labeling promptly shifted focus towards strategies that prevent the labeling process from having to be completely redone every time the zoom level changes. Petzold et al. (2003) propose a data structure called *reactive conflict graph*. By storing potential conflicts and essential labeling information depending on the extent of occurrence, their structure aims to reduce dynamic query time after a pre-processing step. For the adaptive retrieval of solutions at different scales, Poon and Shin (2005) established a hierarchical data structure in which each level represents a specific scale. Mote (2007) introduces a scalable method with the goal of labeling features in dynamic maps in real time, eliminating the need for a pre-processing stage. Recently, Pavlovec and Cmolik (2021) proposed a fast greedy labeling method designed to operate on the graphics processing unit (GPU), enabling the parallel positioning of labels. Beyond the requirement of real-time solution generation, the study of Dumont et al. (2020) demonstrates the crucial role of consistency in dynamic map design.

Consistency When interacting with the map, it is important that a user is able to track the visual changes. To avoid labels from constantly appearing and disappearing (flickering) or shifting in unexpected ways, it is crucial to ensure temporal consistency. Rather than optimizing the selection and placement of labels for each individual view of the map, Been et al. (2006) propose an algorithmic framework that assigns an activity interval to each label. With regard to four different consistency criteria, their model guarantees temporally consistent map labelings. Later, Been et al. (2010) refined this model, proved NP-hardness for different problem settings, and introduced a suite of exact approaches and approximation algorithms. Subsequent theoretical contributions yielded further enhancements in approximation algorithms (Zhang et al., 2015; Liao et al., 2016). Extensions to the activity-interval framework were considered that provide greater freedom in the placement of the labels, including models that allow for continuous shifting of the labels (Gemsa et al., 2011b; Schwartges et al., 2014) and models that allow for a distance between a feature and its label (Wu et al., 2017). Other approaches integrate the priority of labels by determining the length of the label activity intervals as a function of importance values (Schwartges et al., 2014; Zhang et al., 2020). When zooming in on a map, more important labels are displayed across a larger range of scales. Under the assumption of uniform weights, Zhang et al. (2020) strive for a *fair* (i.e., balanced) distribution of activity interval lengths. While these contributions primarily consider zooming and panning functionalities in dynamic maps, ongoing research is dedicated to achieving a consistent label placement when rotating the map (Gemsa et al., 2016a,b; Yokosuka and Imai, 2017; Rudi, 2021). Cano et al. (2019) address both the scaling and the rotation of the map and prove a number of properties of optimal solutions that allow a significant reduction of the solution space of corresponding ILPs. Aiming towards a more holistic framework, Gemsa et al. (2020) propose a model that considers zooming, panning, and rotating in a unified way. Ensuring that the visualization of labels changes smoothly when switching between scales is part of continuous map generalization (CMG), which we address in Section 2.3.2.

2.4.3 External Map Labeling

Unlike traditional labeling methods that focus on directly labeling features within the map, *external map labeling* strategies offer a distinct perspective by shifting labels outside the map. This approach is particularly useful when dealing with densely packed maps or complex spatial arrangements where internal labeling may result in clutter or overlap. A detailed survey on external labeling techniques is given by Bekos et al. (2019). Usually, to maintain an unambiguous label-feature assignment, labels are visually connected to their map features by thin lines called *leaders*. Employing such leaders necessitates the establishment of additional quality criteria alongside the existing cartographic standards (Ali et al., 2005; Niedermann et al., 2017). Most commonly, crossing leaders should be avoided, the length of the leaders should be small, and there should be sufficient space between leaders. Particularly for sustaining the legibility of the map, the shape of the leaders is also a crucial factor. Bekos et al. (2019) present a systematic classification of different leader types, as depicted in Figure 2.3. These leaders differ in the shape of their segments which run either parallel (p), orthogonal (o), or diagonal (d) to the edge of the map. Barth et al. (2019) conducted a comprehensive user study on the readability of these different leader types, focusing on the assignment performance – the speed at which viewers can associate features with their labels and vice versa.

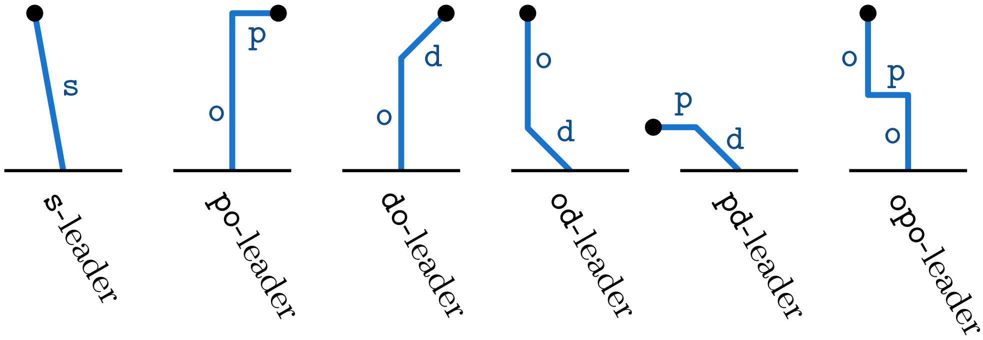


Fig. 2.3: Different leader types according to Bekos et al. (2019).

Boundary Labeling From an algorithmic point of view, Bekos et al. (2004) introduced the first formal model for *boundary labeling*, a variant of external labeling which assumes that the map is rectangular and the labels are placed alongside of the boundary of the map. While their fundamental framework builds upon leaders containing parallel and orthogonal segments, the introduction of octilinear leaders (i.e., leaders with diagonal parts) was investigated in later contributions (Benkert et al., 2009; Bekos et al., 2010). Expanding the scenario beyond labels being limited to a single side of the map, research focused on the development of algorithms for *multi-sided boundary labeling* (Kindermann et al., 2016; Bose et al., 2018, 2021). While these approaches assign each label to precisely one feature, Lin et al. (2008) were among the first to explore scenarios where a label can be associated to multiple features. Additionally, in tackling what is known as *many-to-one boundary labeling*, both Lin (2010) and Bekos et al. (2015) proposed methodologies that merge all leaders attached to the same label. Most contributions on boundary labeling present highly specialized algorithms that are targeted at static figures and maps without providing any user interaction. As an exception, Nöllenburg et al. (2010) consider boundary labeling for dynamic maps taking zooming into account. Beyond the annotation of information in maps, boundary labeling is established in diverse fields of application, including panorama images (Gemsa et al., 2011a, 2015), dynamic scenes (Bobák et al., 2019, 2020), and logic puzzles (Klute et al., 2021).

Excentric Labeling Another variant of external labeling is so-called *excentric labeling* which is primarily aimed at reducing visual clutter. Only map features located within a limited *focus region* are annotated with labels. Usually, this focus region is described by a circle that can be interpreted as a lens that can be shifted over the map (Bekos et al., 2019). To allow a large number of labels to be displayed without producing clutter, Fekete and Plaisant (1999) first suggested an approach in which the labels are arranged in stacks that are placed next to the focus region. Later, Bertini et al. (2009) consider the stack arrangement in a dynamic environment that allows the labels to be explored interactively by scrolling, sorting, and filtering. Moreover, alternative models have been introduced that place the labels directly at the boundary of the focus region (Fink et al., 2012; Haunert and Hermes, 2014) or in its immediate vicinity (Heinsohn et al., 2014). Niedermann and Haunert (2019) propose a hybrid variant in which labels can additionally be placed inside the focus region and are allowed to slightly overlap each other. Similarly, Čmolík et al. (2022) propose a mixed labeling strategy that combines both internal and external labeling techniques.

2.5 Fundamentals of Combinatorial Optimization

Combinatorial optimization has become widely established as a methodological approach for addressing a diverse range of cartographic challenges. Serving as a fundamental methodological basis for this thesis, we delineate its principal characteristics and essential methodologies below. Combinatorial optimization relies on discrete decision variables to model and solve optimization problems. The objective of such optimization problems is to find the best object from a finite set that minimizes (or maximizes) an optimization function. Typically, the sought object is an integer, a combination of elements, or a graph. Notable examples of optimization problems include the *travelling salesperson problem* (TSP), which aims to find the shortest round trip through a set of cities, and the *knapsack problem*, which seeks to maximize the value of the selected items subject to a constraint on their total weight (Cormen et al., 2022). For a comprehensive discussion of the fundamentals and methodologies of combinatorial optimization, please refer to the works of Papadimitriou and Steiglitz (1998) and Korte et al. (2011).

Solving combinatorial problems is often computationally challenging as the search space can grow exponentially with increasing problem size, i.e., with increasing number of variables. Generally, we can distinguish problems according to whether or not they can be solved efficiently. In fact, many combinatorial problems can be proven to be NP-hard, meaning that there is a high improbability of finding a polynomial-time algorithm for solving them. However, to tackle such problems, exact methods, approximation algorithms, as well as heuristics can be employed. Exact methods rigorously seek the optimal solution but can be computationally impractical for large problem instances. As an alternative, approximation algorithms can run in polynomial time and provide solutions that guarantee a specific quality in relation to the optimal solution. Heuristics prioritize speed, but do not guarantee a level of quality. However, they often lead to near-optimal solutions which makes them particularly useful for complex problems and applications where solutions need to be computed on demand (e.g., interactive interfaces).

In the following, *linear programming* as an exact method and *local search* as a heuristic method are discussed in more detail, since the algorithmic contributions of this thesis are mainly based on these strategies. Additionally, the fundamental optimization problem of *graph coloring* is outlined, as it constitutes the basis for many mathematical models presented in this thesis.

2.5.1 Linear Programming

Linear programming is a modeling technique within mathematical optimization. More formally, linear programming aims to optimize a linear objective function while adhering to linear constraints. It can be considered a hybrid optimization problem that spans both continuous and combinatorial optimization domains. In fact, linear programming is inherently a continuous optimization problem, since it is based on real values. However, it also serves as a fundamental concept in combinatorial optimization. This dual role arises from the fact that many combinatorial problems can be solved by adapting linear programming techniques (Yannakakis, 1991).

An instance of linear programming is called a *linear program* (LP). Depending on whether the objective function is to be minimized or maximized, an LP expresses a minimization or a maximization problem, respectively. Any minimization problem can be converted into a maximization problem by reversing the sign of the objective function. The constraints of an LP can be specified either by inequalities or equations. An LP that uses inequalities is considered to be in its *canonical form*, while the use of equations implies the *standard form*. We note that any equality constraint can also be represented by two opposing inequality constraints. In its canonical form, an LP can be represented as follows.

Given a matrix $A \in \mathbb{Z}^{m \times n}$ and two vectors $b \in \mathbb{Z}^m$ and $c \in \mathbb{Z}^n$

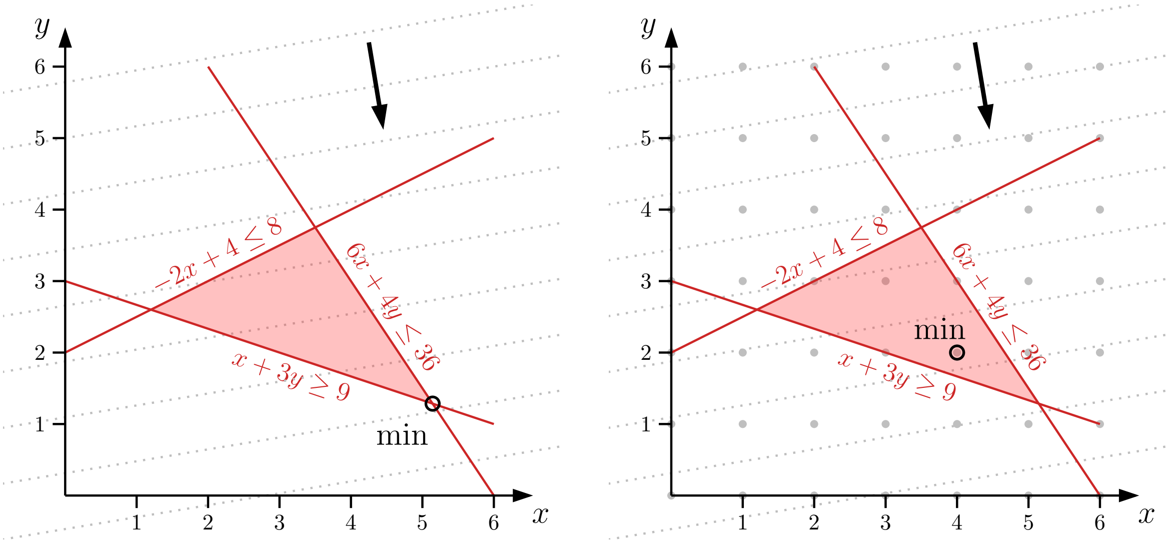
$$\begin{array}{ll} \text{minimize} & f(x) = c^\intercal \cdot x \\ \text{subject to} & A \cdot x \geq b \ , \\ & x \geq 0 \\ \text{with} & x \in \mathbb{R}^n \ . \end{array} \tag{2.1}$$

The vector $x \in \mathbb{R}^n$ contains decision variables that describe a potential solution. Each vector that satisfies all m constraints is considered a *feasible* solution. Within the set of feasible solutions, the objective is to find the one that minimizes the function $f(x)$.

For a better interpretation of linear programming, it is helpful to explore a geometric example within two-dimensional space. To that end, we consider the following LP.

$$\begin{array}{ll} \text{minimize} & f(x,y) = x - 6y \\ \text{subject to} & -2x + 4 \leq 8 \ , \\ & 6x + 4y \leq 36 \ , \\ & x + 3y \geq 9 \\ \text{with} & x, y \in \mathbb{R}_0^+ \ . \end{array} \tag{2.2}$$

The corresponding graphical interpretation of the exemplary LP is shown in Figure 2.4a. Each inequality spans a half-space containing the solutions that satisfy this particular constraint. These half-spaces are delimited by hyper planes, each of which – for the given example – corresponds to a straight line in the two-dimensional space. Combining all inequalities yields the intersection of their respective half-spaces, forming the final solution space that satisfies all constraints. In our two-dimensional example, this space forms a convex polytope, which is illustrated by the reddish colored polygon in Figure 2.4a. The task now is to find a solution within this space that minimizes the objective function. The gray dotted lines represent level lines of equal cost of the objective function. An optimal solution is achieved by moving a level line in the direction of



(a) geometric interpretation of an LP

(b) geometric interpretation of an ILP

Fig. 2.4: Comparison of the geometric interpretation of an exemplary LP and the corresponding formulation as an ILP.

optimization (indicated by the black arrow) until it only touches the solution space at one point (indicated by the black circle).

The geometric interpretation illustrates that, due to the linearity of both the objective function and the inequalities, an optimal solution of the LP lies in one of the extreme points of the convex polytope. If the level lines of the objective function run parallel to one of the limiting hyperplanes, there can be an infinite number of optimal solutions. However, there is still an optimal solution in one extreme point.

A special method that makes use of this property is the *simplex algorithm* (Dantzig, 1963). It begins by identifying an initial feasible solution at one of the extreme points of the polytope. It then iteratively approaches the optimal solution by moving along the edges of the polytope from one extreme point to another. The process continues until an optimum is found. Note that due to the convexity of the polytope, each local optimum inherently corresponds to a global optimum of the LP (Papadimitriou and Steiglitz, 1998).

It is known that the simplex algorithm has an exponential worst-case running time (Klee and Minty, 1972; Zadeh, 1973). This complexity arises from the potential need to visit an exponential number of extreme points in the search for the optimal solution. In practice, however, the simplex algorithm often performs well and solves many linear programming problems within a polynomial running time. For this reason, the simplex algorithm forms the basis of many commercial solvers. It is important to mention that there are variants and improvements to the simplex algorithm, such as the *Bland's rule* and randomized pivot rules, which aim to mitigate some of the worst-case scenarios and improve the practical performance of the algorithm (Papadimitriou and Steiglitz, 1998).

Integer Linear Programming

A special case of linear programming is *integer linear programming* in which the solution vector $x \in \mathbb{Z}^n$ is restricted to integer values. An instance of integer linear programming is called an *integer linear program* (ILP). Due to their ability to model and handle discrete decision variables, ILPs are particularly well-suited for combinatorial problems, such as scheduling or decision making in the form of binary variables. If integerity is only required for some of the variables, a corresponding problem is referred to as *mixed-integer linear program* (MILP).

Due to the additional requirement of integer values for some or all decision variables, integer linear programming has a higher complexity than linear programming (Papadimitriou and Steiglitz, 1998). While there exist efficient algorithms with polynomial time complexity for LPs, solving ILPs is NP-hard. The higher complexity can be illustrated by transforming the exemplary LP in Equation 2.2 into an ILP. The corresponding graphical representation is shown in Figure 2.4b. By restricting the variables to integer values, the set of feasible solutions is limited to the gray dots within the polytope. Thus, the optimal solution with minimal objective value can no longer be found by exploring the extreme points of the polytope and methods such as the simplex algorithm do not lead directly to the solution.

Approaches that tackle this issue mostly rely on relaxing the ILP in a first step such that the integrality requirement is neglected and fractional values are also permitted (*LP relaxation*). One of such approaches is *branch-and-bound* (Wolsey and Nemhauser, 1999). This technique begins by solving the LP relaxation of the original ILP. Since the resulting solution may yield fractional values for the decision variables, the algorithm divides the problem into two sub-problems, each of which is formed by imposing additional constraints to force some variables to take integer values. This *branching* process continues iteratively, creating a tree structure where each node represents a sub-problem with its own constraints. After solving the LP relaxation for each sub-problem, the algorithm computes a lower bound (for maximization problems) or an upper bound (for minimization problems) as part of the *bounding* process. If this bound is lower (for maximization) or higher (for minimization) than the best-known integer solution (the *incumbent*), the sub-problem can be disregarded from further consideration. The branch-and-bound algorithm continues branching and bounding until either an integer solution is found or there is no feasible integer solution for the given constraints.

A different strategy is the *cutting-plane method* (Gomory, 1969), which gradually inserts additional inequalities that further restrict the search space. The inequalities are chosen in such a way that they separate the solution of the LP relaxation from the search space without cutting off integer solutions. This process is repeated iteratively until an optimal integer solution is found. Particularly *branch-and-cut*, which describes the combination of both branch-and-bound and the cutting-planes method, has proven to be efficient. By introducing cuts that restrict the search space of the LP relaxations, the number of branches in the branch-and-bound process can be considerably reduced.

2.5.2 Local Search

Local search is a an established heuristic strategy for solving combinatorial problems (Aarts and Lenstra, 2003). Although local search in general offers no guarantee for a level of quality, it can achieve good – often even near-optimal – solutions within a short running time. This characteristic makes this strategy particularly attractive for tackling NP-hard problems.

The basic idea behind local search is to start with an initial solution and to iteratively explore neighboring solutions by making small modifications. At each iteration, the algorithm evaluates the quality of the current solution and moves to a neighboring solution that improves the objective function. This process continues until no further improvements can be made, indicating that a local optimum has been reached. A key aspect of any local search strategy is the definition of the *neighborhood* $N(s)$ of a solution s. A neighborhood defines the set of solutions that can be reached by performing a single local modification. Correspondingly, we refer to each solution

$\bar{s} \in N(s)$ as a *neighbor* of s. A well-defined and appropriately chosen neighborhood is essential to the algorithm's ability to effectively explore the solution space.

Algorithm 1: Hill Climbing

```
1 $s \leftarrow$ initial solution
2 repeat
3     select $\bar{s} \in N(s)$ randomly                    // explore neighboring solution
      if $f(\bar{s}) < f(s)$ then
          $s \leftarrow \bar{s}$
6 until $f(\bar{s}) \geq f(s) \quad \forall \bar{s} \in N(s)$
7 return $s$
```

The most basic local search algorithm is known as *hill climbing*. For a minimization problem, its pseudocode is given in Algorithm 1. While hill climbing is characterized by its simplicity, it inherently tends to being trapped in poor local optima, failing to reach the global optimum. An extension that can mitigate this issue is *steepest ascent* hill climbing. Instead of selecting an arbitrary neighbor in each iteration (see line 3 in Algorithm 1), the neighbor with the lowest (for a maximization problem the highest) value of the objective function is selected.

Another variant of local search that aims to escape such local optima is *simulated annealing*. Algorithm 2 shows the corresponding pseudocode. Generally, this variant also adheres to the hill climbing principle. However, the algorithm diverges by accepting not only solutions that improve the objective function, but also – with a certain probability – solutions with a lower quality (Kirkpatrick et al., 1983). Analogous to the deformability of a material at decreasing temperature, this probability decreases with increasing number of iterations. Thus, the algorithm becomes more selective over time.

Algorithm 2: Simulated Annealing

```
$s \leftarrow$ initial solution
$k \leftarrow 1$
repeat
    select $\bar{s} \in N(s)$ randomly                    // explore neighboring solution
    if $f(\bar{s}) < f(s)$ then
        $s \leftarrow \bar{s}$
    else if $\exp(\frac{f(s)-f(\bar{s})}{T(k)}) > random[0,1)$ then
        $s \leftarrow \bar{s}$
    $k \leftarrow k+1$
until termination criterion
return $s$
```

The probability of accepting a lower-quality solution in the k-th iteration depends on a temperature T_k that is defined by a monotonically decreasing function. There is no universally accepted or standard definition for this function (Nouraniyand and Andresenz, 1998). However, one commonly applied definition is the *exponential cooling*

$$T_k = T_0 \cdot \alpha^k \, , \tag{2.3}$$

where T_0 is the starting temperature and $\alpha \in (0,1)$ is a constant that controls the cooling rate. Other widely used cooling functions are of *linear*, *logarithmic*, or *adaptive* nature.

The process of simulated annealing terminates when a predefined termination criterion is fulfilled. This criterion can take various forms. Common termination criteria include reaching a prescribed number of iterations, meeting a satisfactory solution quality, or falling below a temperature threshold.

2.5.3 Graph Coloring

Graph coloring is a fundamental optimization problem in combinatorial optimization used in a variety of different domains and applications. Since many of the problems introduced in this thesis are based on graph coloring, this section will highlight its key aspects, as well as important algorithmic and computational contributions.

One of the most commonly applied graph coloring problems is the *vertex coloring problem* (VCP). Given a graph $G = (V, E)$, it searches for the assignment of colors to the vertices, such that no two adjacent vertices share the same color and the number of colors is minimized. A solution of VCP with k colors is called a k-coloring of G. An optimal coloring of G is a k-coloring that uses the smallest possible value of k, known as the *chromatic number* of G. Finding the chromatic number is known to be NP-hard (Garey et al., 1974) even for graphs representing the intersection relationships among a given set of rectangles (Imai and Asano, 1983). Due to the wide range of applications like register allocation, scheduling, frequency assignment, and timetabling problems, there is a vast amount of literature on this problem. For surveys, see for example the work of Malaguti and Toth (2010) or de Lima and Carmo (2018).

The high complexity of the VCP leads algorithmic and computational contributions to mostly focus on either heuristic approaches or exact algorithms. Initial heuristic approaches were based on greedy strategies (Brélaz, 1979b; Leighton, 1979; Bollobás and Thomason, 1985), which continue to be used and refined today (Komosko et al., 2016; Kralev and Kraleva, 2023). Another well-established heuristic approach to the VCP is the use of local search strategies (Galinier and Hertz, 2006). Specifically, algorithms have been developed that implement techniques such as tabu search (Hertz and Werra, 1987; Galinier and Hao, 1999; Malaguti et al., 2008), hill climbing (Costa et al., 1995; Glass and Prügel-Bennett, 2003), and simulated annealing (Johnson et al., 1991; Morgenstern, 1996). Much of the research in this area has put particular focus on the definition of the neighborhood structure (Morgenstern, 1996; Avanthay et al., 2003; Chiarandini et al., 2003).

Exact algorithms for the VCP have been developed based on dynamic programming (Lawler, 1976; Byskov, 2002; Eppstein, 2003), branch-and-bound (Brown, 1972; Brélaz, 1979a; Sewell, 1993; Segundo, 2012), and integer linear programs (ILPs) (Jabrayilov and Mutzel, 2018; Mehrotra and Trick, 1996; Méndez-Díaz and Zabala, 2008; Campêlo et al., 2008; Hansen et al., 2009; Malaguti et al., 2011). The dynamic programming algorithms require exponential space and are therefore only of theoretical importance. In contrast, the branch-and-bound algorithms require a polynomial space and are applicable to graphs with up to 80 nodes (Segundo, 2012). However, the most efficient algorithms for large instances are ILP-based algorithms. State-of-the-art ILP models are the assignment based model (Méndez-Díaz and Zabala, 2008, 2006), the partial-ordering based model (POP) (Jabrayilov and Mutzel, 2018), the representatives model (Campêlo et al., 2008, 2004), set covering (Mehrotra and Trick, 1996; Malaguti et al., 2011), and set partitioning formulations (Hansen et al., 2009).

The assignment model, the partial-ordering based model, and the representatives model have polynomial size (Jabrayilov and Mutzel, 2018; Méndez-Díaz and Zabala, 2006; Campêlo et al., 2004) and thus they have the advantage that they can be fed directly into standard ILP solvers. Both the assignment and the partial-ordering based model have $O(|V|H)$ variables, where H is an upper bound for the chromatic number of the graph (Jabrayilov and Mutzel, 2018; Méndez-Díaz and Zabala, 2006). As the chromatic number can be rather small for sparse graphs, both models have correspondingly small memory requirements and are well-suited for such graphs. The representatives model contains $O(|V|+|\bar{E}|)$ variables (Campêlo et al., 2004), where $\bar{E}$ is the set of edges of the complement graph. As the density increases, the number $|\bar{E}|$ becomes smaller. Hence, this model is particularly suitable for dense graphs. The evaluations of Jabrayilov and Mutzel (2018) show that the partial-ordering based model outperforms both the assignment and the representatives model for sparse graphs. For dense graphs, they demonstrate that the representatives model is the most efficient. The set covering and set partitioning formulations involve a variable for each independent set and are also suited for dense graphs. Unlike the first three models, these models have an exponential number of variables and therefore can be solved with sophisticated branch-and-price algorithms (Mehrotra and Trick, 2007).

2.6 A Use-Case: Combinatorial Optimization for the Generalization of Land-Use Data

As shown in the previous sections, combinatorial optimization has become an established methodological approach for diverse cartographic problems. To provide a practical application example and conclude this chapter on automation in cartography, we will now showcase own collaborative work with Johannes Oehrlein and Jan-Henrik Haunert (Gedicke et al., 2021c) in which we apply combinatorial optimization to a map generalization task. Similar methodological approaches are later used for the development of zoomless maps, which are the focus of this thesis and are described in detail in the succeeding chapters.

2.6.1 Introduction

In many fields of application, topographic maps are used as a representation of geographical information. We focus on the generalization of topographic data, in particular areas of different land-use classes and line features representing the road network. When reducing the map scale, some areas need to be merged to larger composite *regions* within the process of *area aggregation*. Given a planar partition of areas, one usually aims to build geometrically compact regions of sufficient size while keeping class changes small (Haunert and Wolff, 2010). Technically, area features and line features are often kept in two different layers which are rendered subsequently – areas followed by lines – to obtain a visualization. However, when generalizing the map, it is important to consider geographic relationships between the different feature types (Touya et al., 2010). If the layer of the land-use classes and that of the road network were generalized separately and superimposed subsequently, narrow *sliver polygons* may be created that are too small or have unfavorable shapes. To avoid this, we address the domain of *collaborative generalization* by introducing an approach for aggregating land-use polygons that is constrained by the road network. In particular, our objective is that roads coincide with the boundaries of the output regions, i.e., to avoid that areas are aggregated across road sections. We formulate an optimization problem for area aggregation that extends the established compactness and class change criteria by our road objective, and present a heuristic based on simulated annealing to solve it.

2.6.2 Modeling

Methodologically, we base our approach on a graph model. Given a digital landscape model, we extract the planar subdivision of areas and the road center lines of the road network. The overlay of both forms the input map (see Figure 2.5a). Note that we assume the coincidence of all roads with boundaries of the input areas as precondition. We introduce an *adjacency graph* that contains a vertex for each area and an edge for each pair of areas with a common boundary (see Figure 2.5b). For each vertex, we store its area's land-use class and size. Considering our objective for the coincidence of roads with boundaries of output regions, we follow an established approach in road network generalization and generate *strokes* by combining road segments (Thomson and Richardson, 1999). In particular, we build functional road units that follow the principle of good continuation by grouping road segments based on their angle of deflection (see Figure 2.5c). The detailed procedure is described by Gedicke et al. (2021c).

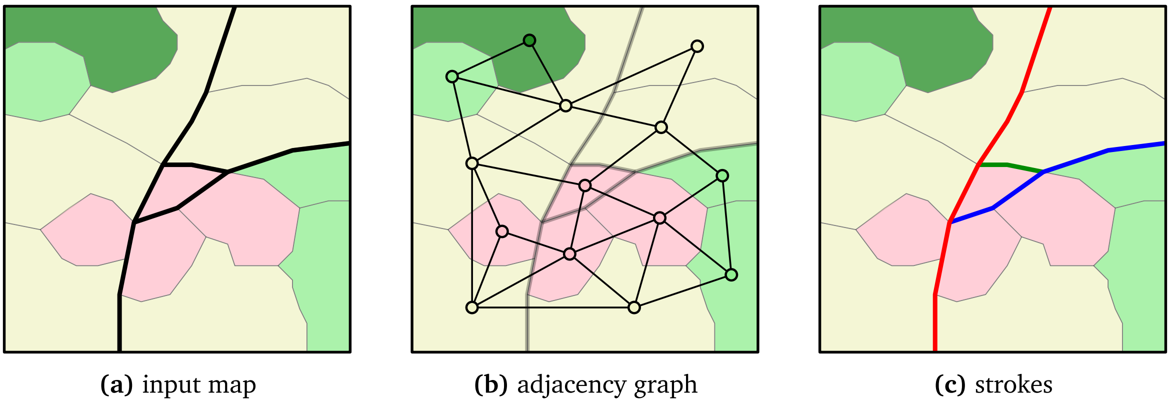

(a) input map **(b)** adjacency graph **(c)** strokes

Fig. 2.5: Illustration of different model specifications using an exemplary map with different land-use classes and roads. In (c), the strokes are indicated by the different colors.

To achieve the coincidence of roads and region boundaries, a straightforward approach would strictly prohibit the aggregation of two adjacent areas if a road runs along their common boundary. However, the prohibition of cross-road aggregations and the minimum size requirement of regions can contradict each other, so that some instances become infeasible. If small areas are completely enclosed by roads, sufficiently large regions cannot be produced without violating the constraint of the coincidence of roads and boundaries of regions. Thus, we relax this constraint and penalize road segments not satisfying the objective.

Altogether, we introduce AREAAGGREGATIONWITHROADS that aims to find a partition of output regions such that

- each region forms a set of contiguous areas,
- in each region there is at least one area with unchanged land-use class,
- the total area size of each region is larger than a specific threshold, and

the cost

$$f = \alpha \cdot f_{\text{comp}} + \beta \cdot f_{\text{class}} + \gamma \cdot f_{\text{stroke}} \tag{2.4}$$

is minimized. With f_{comp} we define a cost for non-compact regions (compactness criterion) based on a simple compactness measure (Osserman, 1978). Further, f_{class} is a cost for class changes (class change criterion) based on the semantic similarity of different classes. With regard to our road criterion, f_{stroke} is a cost for sections of strokes not coinciding region boundaries. To avoid strokes breaking into several short sections, the cost is specified such that short non-coincidental stroke sections are penalized disproportionately higher than longer ones. The factors α, β, and γ are weightings between the criteria with $\alpha + \beta + \gamma = 1$.

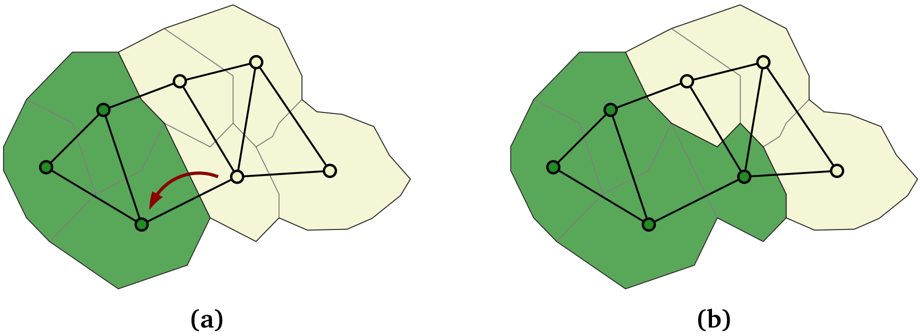


Fig. 2.6: Illustration of a swap. In (a), the upcoming swap operation is indicated by the red arrow. The solution after the swap is shown in (b).

2.6.3 Approach

For solving AreaAggregationWithRoads, we apply a combinatorial optimization method. In particular, we extend the heuristic approach of Haunert and Wolff (2010) that is based on simulated annealing. We start by finding an initial solution using an iterative approach that is called *region growing* (van Oosterom, 1995). All input areas that fall below the required minimum size are sorted in ascending order according to their size. In each iteration, the smallest area is selected and merged with a neighbor such that the cost for merging is minimal. Only for the particular areas affected by the merge, the cost is computed locally in accordance with Objective 2.4. The algorithm terminates when all composite areas (i.e., the regions) meet the minimum size condition. To explore neighboring solutions within the iterative annealing process, we define the *neighborhood* of a solution as the set of all feasible solutions created by removing an area from one region and assigning this area to an adjacent region (see Figure 2.6). We refer to this procedure as *swap*. Additionally, it may happen that a region's area size falls below the given size threshold. However, if we simply exclude such cases from potential swaps, some feasible solutions may not be reachable. Instead, we relax the minimum area size constraint by charging an extra cost for regions that fall below the threshold. If the final solution contains regions that are not sufficiently large, region growing can be reapplied to repair the solution.

2.6.4 Experimental Results

In our experiments, we used the data of a digital landscape model provided by the German topographic database ATKIS. More precisely, we extracted data at scale 1:25k from a rural area in North Rhine-Westphalia that covers a large spectrum of land-use classes and the most important roads (see Figure 2.7). To define the minimum area sizes and semantic distances, we use specifications from ATKIS that correspond to a generalization to a scale of 1:250k. Please refer to Gedicke et al. (2021c) for a detailed overview of all specifications.

Our evaluation is based on the comparison of the results of different weighting configurations of α, β, and γ. We regard the case in which only the compactness and the class changes are optimized as a baseline. If we additionally consider the road criterion, we observe that the compactness of the regions is not negatively affected. Furthermore, we find that the costs for class changes become higher with increasing influence of the road criterion, i.e., more class changes occur when avoiding cross-road aggregations. However, we can determine a *trade-off* between all criteria, i.e., a suitable balance of α, β, and γ at which substantially more roads coincide with boundaries of regions while land-use changes are still kept small.

An exemplary result obtained with such a trade-off is shown in Figure 2.7. Important structures

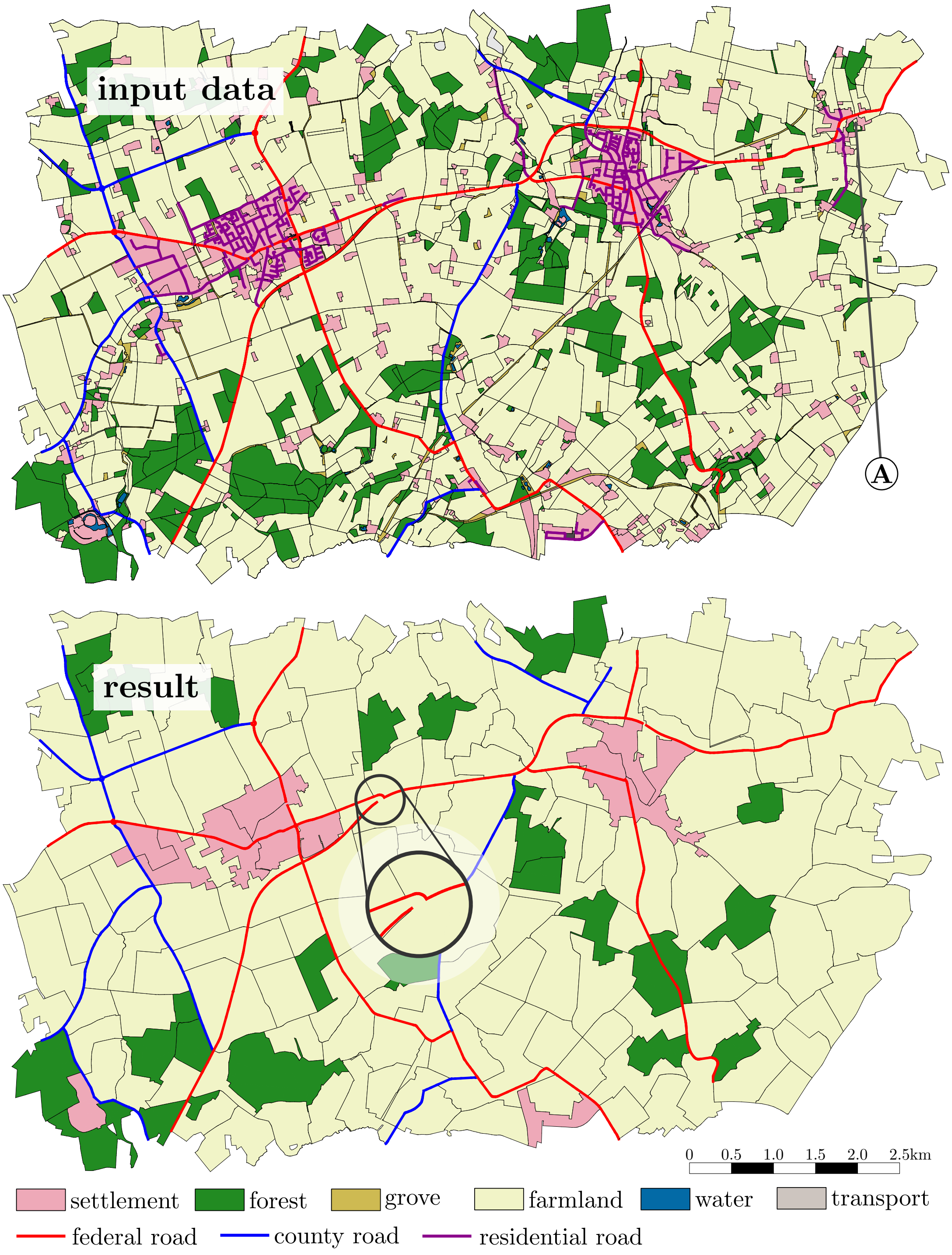


Fig. 2.7: The upper figure shows the input data extracted from ATKIS. The label Ⓐ marks a section that is addressed in our discussion. An exemplary result obtained by using a trade-off between the optimized criteria is shown in the lower figure. Note that in this visualization only roads are shown that coincide with boundaries of regions. This helps to better identify the effects of introducing the road criterion. In a final generalized map the entire road layer would be superimposed on the polygon layer.

have been preserved and most of the road segments lie on the region boundaries. In some parts of the map, however, aggregation was carried out across short road sections. In these cases, the costs for corresponding land-use changes would have been higher than the costs of the respective cross-road aggregation. Contrary to this, we observe that the settlement in the upper right part of the input map (see label Ⓐ) is no longer included in the resulting map. Due to the minimum size constraint, a cross-road aggregation would haven been necessary to preserve the land-use information of the settlement. However, this contradicts our road criterion, which dominates in this particular map section.

2.6.5 Conclusion

Overall, we showed that with an appropriate balance of all three considered criteria, substantially more roads coincide with boundaries of regions while land-use changes are still kept small and compactness is maintained. For future research, we suggest adding further criteria to our model. For example, other spatial patterns such as administrative boundaries could be taken into account. Furthermore, by considering the share of each land-use class in the total area, the preservation of land-use patterns could be enhanced.

3 INTERNAL MULTI-PAGE LABELING WITH A GRAPH-COLORING APPROACH

The following chapter mainly comprises joint work with Benjamin Niedermann and Jan-Henrik Haunert (Gedicke et al., 2019). With the support of Benjamin Niedermann, I (Sven Gedicke) developed the methodological and algorithmic content. Both Jan-Henrik Haunert and Benjamin Niedermann provided supervision and assistance in the writing process.

We present *internal multi-page labeling*, which is a concept for zoomless maps that distributes the information contained in a given area of interest on multiple pages. Each page shows the current section of the map and a selection of the features' labels (see Figure 3.1). While the map layer is kept fixed, the user can interactively navigate through the pages. Assuming an interactive map on a smart device, we consider swiping through the pages or clicking navigation buttons to be reasonable interaction strategies. From a computational point of view, the problem is to find an appropriate distribution of the labels over a certain number of pages. Addressing uncluttered visualizations, we require that no page contains overlapping labels. Further, we optimize the assignment of the labels to pages such that the number of pages is small, labels of high importance appear on the first pages, and sparsely labeled pages are avoided. Algorithmically, we reduce the label-page assignment to a weighted and constrained graph coloring problem. We propose a simple greedy heuristic that is fast enough for on-the-fly computation in interactive applications. We evaluate the quality of the obtained solutions by comparing them with optimal ones, which we get by applying integer linear programming. In our evaluations on real-world data, we particularly show that the proposed heuristic achieves high-quality solutions and that it substantially improves the distribution of the labels in comparison to the simple strategy of assigning the labels to pages solely based on the labels' weights.

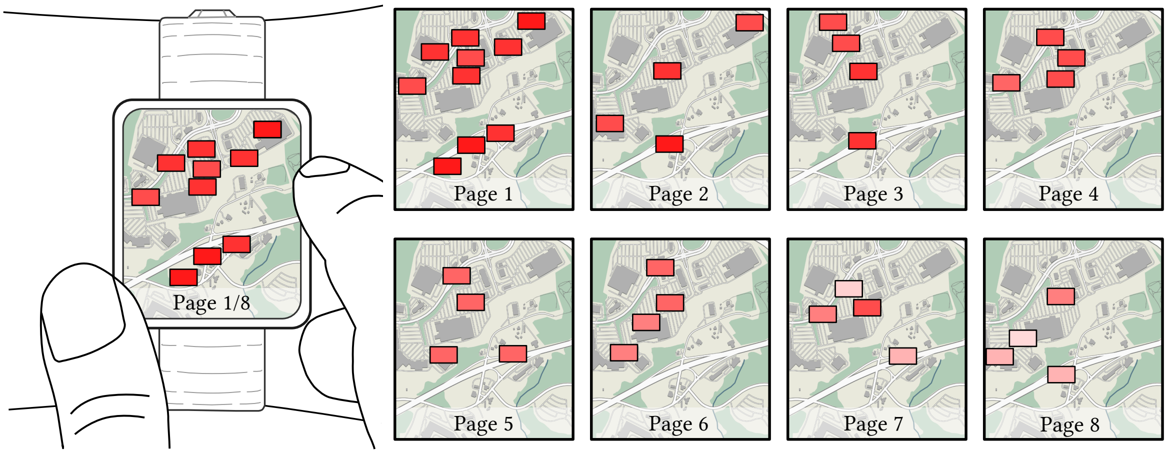


Fig. 3.1: An optimal multi-page labeling generated with our approach. The user can navigate through the pages by swiping them. The saturation of the labels' color indicates their respective weight (the more saturated, the higher the weight). *Map attribution: Carto.*

3.1 Introduction

The idea of zoomless maps essentially encompasses two objectives: (1) a clear and well-organized presentation of information, and (2) the capability of a thorough exploration of information without the need for zooming. In the strategy of *internal multi-page labeling* presented in this chapter, we visualize point features by placing labels within the map (i.e., internally). We consider the labels not only as additional information, but as a representation of the features themselves and therefore place the labels centrally on their respective features. Addressing the exploration of the information, we consider a discrete interaction space. More specifically, we aim at computing discrete states of the map, each showing the current area of interest and a selection of labels (see Figure 3.1 for an illustration). We interpret the set of map states as *pages* through which a user can interactively navigate while the underlying map layer is kept fixed. Assuming an interactive map on a mobile device, we consider swiping through the pages or clicking navigation buttons to be reasonable interaction strategies. By navigating through the pages, the user can access all information in an area of interest without the need for further zooming or panning. The computational problem behind this strategy is to find an appropriate distribution of the labels over a certain number of pages. We call such a distribution of labels a *multi-page labeling* or *labeling* for short. Although related approaches are already used in web applications of online services like Google[1], e.g., for displaying hotel search results, these approaches do not sufficiently address the legibility of the pages. In particular, overlapping labels are not prevented, which leads to both a cluttered overall visualization and information being obscured.

A simple strategy that is capable of avoiding overlaps might optimize the label placement by creating the pages sequentially, i.e., it first places labels on the first page, afterwards on the second page, and so on. Assuming that each label has a pre-defined weight representing its importance, a reasonable aim could be maximizing the total sum of weights on each page considering the labels that have not been assigned to preceding pages. Hence, for each page the problem reduces to finding a *maximum weight independent set* (MWIS) within the according conflict graph of the remaining labels, i.e., an overlap-free set of labels with maximum total weight. Finding such sets has been extensively investigated in research on label placement before (e.g., see Agarwal et al. (1998)).

Figure 3.2 shows an example that we have obtained in such a way using a simple integer linear program (ILP) for MWIS (Haunert and Wolff, 2017). We observe that the last three pages only contain one label each, while the first two pages are densely packed. The high label densities on the front pages might negatively affect both the map complexity and the perceived map readability. Users can be overwhelmed by the amount of information and important contextual information might be lost as much of the background map is covered. Contrary, swiping through the sparsely labeled pages might easily annoy users as they hardly gain new information. Hence, we argue that reusing existing techniques for label placement is not sufficient. Instead, the process of creating the pages needs to be considered in an integrated way requiring multiple criteria for the placement.

First of all, we consider accessibility a crucial aspect and thus deem the number of pages to be an important criterion to be taken into account when creating a multi-page labeling. A small number of pages not only reduces the number of interactions (e.g, swipes or clicks) necessary to navigate through the pages, but also yields a packing of the labels exploiting the available space on each page. However, merely optimizing the number of pages does not satisfactorily map the semantic information of the label's point features onto the pages. In many use cases, it is preferable to consider feature priority and place labels of important features as close to the front of the sequence of pages as possible. For example, when searching for restaurants, a user would proba-

[1] https://google.com

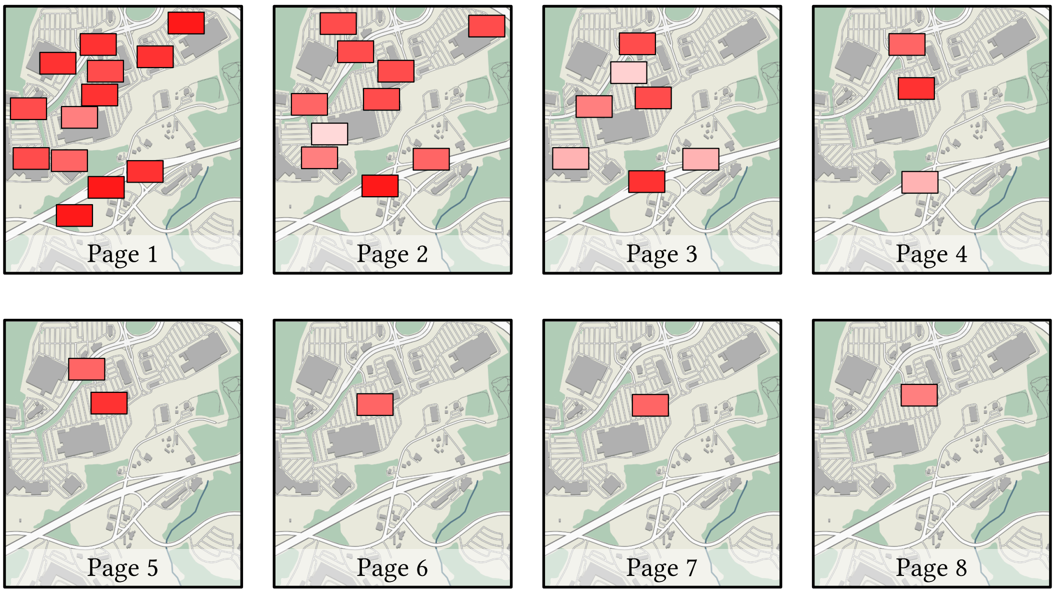


Fig. 3.2: A multi-page labeling of the same instance as in Figure 3.1 for the case that the pages are created sequentially. While the first page is densely packed with labels, the last three pages only contain one label each. Note that the labeling also corresponds to an optimal result of BiCriteria0 (see Section 3.5). *Map attribution: Carto.*

bly want high-rated restaurants to appear earlier than restaurants with lower ratings. Therefore, as a second criterion, labels of higher importance should appear on the first pages, while less important labels can be postponed to pages occurring later on in the sequence. However, if only the number of pages and the priority of features are taken into account, similar effects can occur as in the case of a sequential page creation; high label densities on front pages and sparsely labeled back pages (see Figure 3.2). Thus, we additionally aim at a balanced distribution of labels among the sequence of pages.

As not all criteria can be met simultaneously, we need to accept compromises, which we reflect in the optimization function of our formalization. Summarizing, we consider the following three criteria for multi-page labeling.

$\mathrm{C_{page}}$	The number of pages should be small.
$\mathrm{C_{prio}}$	Important labels should appear as close to the front of the sequence of pages as possible.
$\mathrm{C_{distr}}$	The distribution of labels among the sequence of pages should be balanced.

Outline We formalize the problem of computing multi-page labelings as an optimization problem based on graph coloring in Section 3.2. In particular, we introduce three problem settings with increasing generality. While the first problem setting only considers Criterion $\mathrm{C_{page}}$, the second one additionally considers Criterion $\mathrm{C_{prio}}$, and the last one considers all three criteria. In Section 3.3, we present mathematical programming formulations for all three problem settings. While these allow us to apply existing solvers for computing mathematically optimal solutions, the approach via integer linear programming is too slow for interactive applications. As we are interested in such applications that require the on-the-fly computation of labelings, we further present a simple but fast greedy heuristic in Section 3.4. Comparing our three problem settings as well as assessing the quality of the heuristic results, we present an evaluation based on real-world data in Section 3.5. We conclude the chapter by summarizing key contributions and discussing limitations of our approach in Section 3.6.

3.2 Mathematical Model

To translate our concept into a mathematical model, we assume that we are given a set F of point features within a pre-defined rectangular region $R \subset \mathbb{R}^2$. Each point feature $f \in F$ has a label that we represent by an axis-parallel rectangle with its center at f; we denote the set of all labels by L. A *multi-page labeling* with k pages is a partition $\mathcal{L} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ of L into k subsets such that no two labels $\ell, \ell' \in \mathcal{P}_i$ of the same subset overlap each other. We call $\mathcal{P}_i$ a *labeling* of the *page* i. We assume that the pages are presented to the user in that particular order. We observe that the problem of finding multi-page labelings directly translates into according graph coloring problems. To that end, let $G = (L, E)$ be the graph that we obtain by identifying each label $\ell \in L$ as a vertex of G; see Figure 3.3. Further, the set $E \in L \times L$ contains an edge $\{\ell, \ell'\}$ if and only if ℓ and ℓ' belong to two different point features and the labels overlap. Hence, a multi-page labeling with k pages corresponds to a k-coloring of G, i.e., we can assign to each vertex of G a color out of k given colors such that no two adjacent vertices of G have the same color. We call G the *conflict graph* of L. We note that even for graphs that represent intersection relationships among a given set of rectangles, the problem that asks for a coloring of G with minimal number of colors (i.e., the *vertex coloring problem*) is NP-hard (Imai and Asano, 1983).

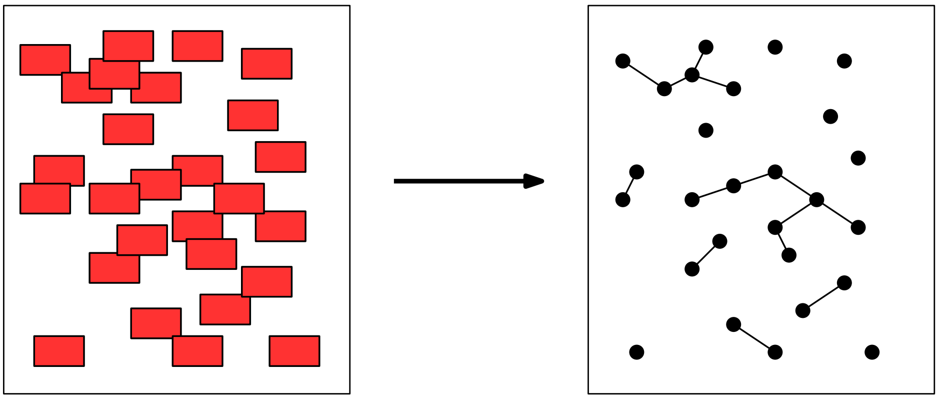


Fig. 3.3: Example of a set of labels (left) and the corresponding conflict graph (right). Each label corresponds to one vertex in the graph and two vertices are adjacent in the conflict graph if and only if the according labels intersect.

Based on these formal assumptions, we introduce different optimization problems for the label-page assignment. We start with a basic variant that minimizes the number of pages.

> **Problem** (MinNumOfPages). *Given a set of labels L, find the multi-page labeling $\mathcal{L} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ of L that minimizes the number k of pages.*

By guaranteeing a minimum number of swipes or clicks for the user when navigating through the multi-page labeling, this variant is a reasonable approach in case the labels are unweighted. However, in a weighted case, it may lead to undesired effects, e.g., important labels may be placed on pages that are shown late, while less important labels appear on early pages. Thus, we assume that we are given a weighting function $w \colon L \to \mathbb{R}^+$ that assigns to each label $\ell \in L$ a weight $w(\ell)$ that expresses the importance of the label's point feature. Further, we are given a function $h \colon \mathbb{N} \to [0, 1]$ such that $h(i)$ rates the position of the page i. When placing a label ℓ on page i the *effective weight* of ℓ is defined as $w(\ell) \cdot h(i)$. Hence, $h(i)$ describes the share of the label's weight that is taken into account when the label is placed on the i-th page. We assume that h decreases monotonically, i.e., we interpret h as the costs for navigating through the sequence of pages; the effective weight of a label decreases with the number of swipes or clicks necessary to reach the page containing the label. Assuming that the interest of the user substantially decreases with the number of performed interactions, we choose h to be an exponentially decreasing function. Altogether, this leads to the following optimization problem.

Problem (WEIGHTEDPAGES). *Given a set of labels L, a weighting function $w\colon L \to \mathbb{R}^+$ for the labels and a function $h\colon \mathbb{N} \to \mathbb{R}^+$ for the pages, find for any $k \in \mathbb{N}$ the multi-page labeling $\mathscr{L} = \{\mathscr{P}_1, \ldots, \mathscr{P}_k\}$ that maximizes the mean effective label weight*

$$\frac{1}{n}\sum_{i=1}^{k}\sum_{\ell\in\mathscr{P}_i} h(i)\cdot w(\ell)\,. \tag{3.1}$$

However, as we will show in our experiments, optimal solutions of WEIGHTEDPAGES easily yield sparsely labeled back pages, while the front pages are densely packed with labels. We therefore extend the objective such that for a multi-page labeling $\mathscr{L}$ its minimum number $n_{\mathscr{L}}$ of labels per page is also taken into account. We obtain a bi-criteria objective, which we balance linearly using a balance factor $\alpha \in [0,1]$.

Problem (BICRITERIALABELING). *Given a set of labels L, a weighting function $w\colon L \to \mathbb{R}^+$ for the labels, a function $h\colon \mathbb{N} \to \mathbb{R}^+$ for the pages and a constant $\alpha \in [0,1]$, find for any $k \in \mathbb{N}$ the multi-page labeling $\mathscr{L} = \{\mathscr{P}_1, \ldots, \mathscr{P}_k\}$ that maximizes*

$$\alpha \cdot n_{\mathscr{L}} + (1-\alpha)\cdot\frac{1}{n}\cdot\sum_{i=1}^{k}\sum_{\ell\in\mathscr{P}_i} h(i)\cdot w(\ell) \tag{3.2}$$

with $n_{\mathscr{L}} = \min\limits_{\forall i\in\{1,\ldots,k\}}\{|\,\mathscr{P}_i\,|\}\,.$

We note that all three introduced problems are NP-hard, as they contain the well-known vertex coloring problem as a special case.

3.3 Mathematical Programming

As one common approach for tackling NP-hard problems, we solve the optimization problems optimally by means of mathematical programming. More specifically, we introduce ILP formulations.

We start by presenting a formulation for the general multi-page labeling problem that asks for any assignment of the labels to pages such that no two labels overlap. Let L be a set of labels. Further, let $k \in \mathbb{N}$ be a trivial upper bound for the number of pages that a solution consists of, e.g., $k = |L|$. For each label $\ell \in L$ and each page i with $1 \le i \le k$ we introduce a binary variable

$$x_{\ell,i} = \begin{cases} 1 & \text{if } \ell \text{ is placed on the } i\text{-th page}\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.3}$$

To obtain a valid multi-page labeling we require for each page i with $1 \le i \le k$ and each pair of labels ℓ and ℓ' with $\{\ell, \ell'\} \in E$ that

$$x_{\ell,i} + x_{\ell',i} \le 1\,. \tag{3.4}$$

Thus, we prohibit two overlapping labels ℓ and ℓ' from being placed on the same page.

Further, for each label ℓ, we require that it is placed on exactly one page with

$$\sum_{i=1}^{k} x_{\ell,i} = 1\,. \tag{3.5}$$

Hence, a valid multi-page labeling is defined as $\mathscr{L} = \{\mathscr{P}_1, \ldots, \mathscr{P}_k\}$ with $\mathscr{P}_i = \{\ell \in L \mid x_{\ell,i} = 1\}$. Subject to Constraint (3.4) and Constraint (3.5), we solve MINNUMOFPAGES by maximizing the objective

$$\sum_{i=1}^{k} \sum_{\ell \in L} h(i) \cdot x_{\ell,i}$$

and we solve WEIGHTEDPAGES by maximizing the objective

$$\frac{1}{n} \cdot \sum_{i=1}^{k} \sum_{\ell \in L} h(i) \cdot w(\ell) \cdot x_{\ell,i}\,.$$

For solving BICRITERIALABELING, we introduce an additional integer variable z which we interpret as the minimum number of labels on any non-empty page. Further, for each page i with $1 \le i \le k$, we introduce a binary variable

$$y_i = \begin{cases} 1 & \text{if the } i\text{-th page is used}\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.6}$$

As *used* we denote a page that is not empty, i.e., on which at least one label has been placed. We enforce that y_i is assigned correctly by requiring

$$x_{\ell,i} \le y_i\,. \tag{3.7}$$

Further, we ensure that z does not exceed the minimum number of labels on any used page by the following constraint for each page i.

$$z \le \sum_{\ell \in L} x_{\ell,i} + (1 - y_i) \cdot n\,. \tag{3.8}$$

Subject to Constraints (3.4)–(3.8), we solve BICRITERIALABELING by maximizing

$$\alpha \cdot z + (1-\alpha) \cdot \frac{1}{n} \cdot \sum_{i=1}^{k} \sum_{\ell \in L} h(i) \cdot w(\ell) \cdot x_{\ell,i}\,,$$

where α is a pre-defined constant balancing both criteria.

In our experiments, we use a specialized solver to find optimal assignments of the variables with respect to the objective functions. This is an adequate approach to obtain results in reasonable time for our evaluation, but it can be too slow to be deployed for interactive scenarios.

3.4 Greedy Heuristic

While there exist solvers for integer linear programming that guarantee an optimal solution, computing the solution requires exponential time in the worst case. Thus, we additionally introduce a simple but fast greedy heuristic. The heuristic consists of two phases; Figure 3.4 illustrates both phases using an exemplary instance. In Phase 1, the heuristic creates a multi-page labeling optimizing the average effective label weight taking the objective of WEIGHTEDPAGES into account. Phase 2 is optional and improves that labeling with respect to the minimum number of labels per page without decreasing the value of the composed Objective (3.2). Hence, the result is a multi-page labeling that is geared towards optimizing BICRITERIALABELING.

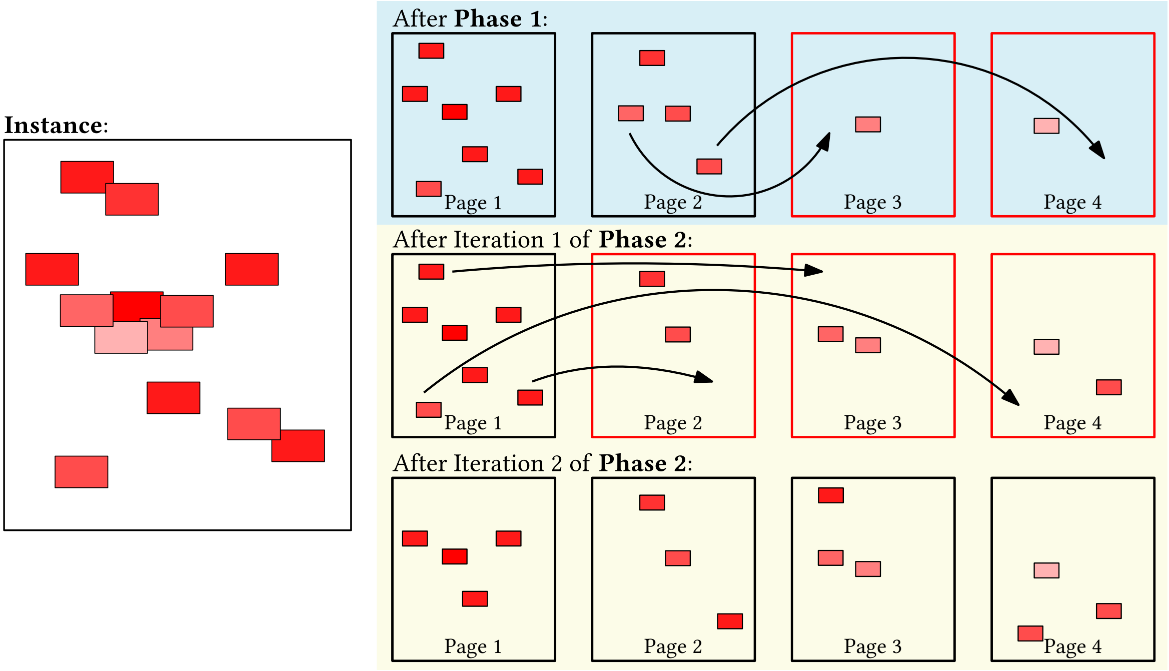


Fig. 3.4: The two phases of our heuristic using an exemplary instance (left). On the right side, labelings after each phase are presented. The first row (highlighted blue) shows the result of Phase 1. The second and third row show the iterative progress of Phase 2 (highlighted yellow). In the given case, the heuristic terminates after the second iteration of Phase 2 and, thus, the last row corresponds to the heuristic solution. Sparsest pages are framed red and arrows indicate which labels are moved in the subsequent step.

Phase 1 – First Fit In the first phase, we start with a common heuristic graph-coloring procedure (Gyárfás and Lehel, 1988; Guan and Xuding, 1997). In our context, this means that labels with large weights are preferred to be put on the first pages of the multi-page labeling. The heuristic first sorts the labels with respect to their weight in decreasing order and then iteratively adds the labels using a first-fit principle. More precisely, starting with a labeling consisting of a single page, it iterates through all labels. For each label it finds the first page in the sequence of pages that admits the placement of the label, i.e, on which placing the label yields no overlaps. If such a page does not exist, a new page is appended to the sequence and the label is placed on this page. By design, this procedure yields a valid multi-page labeling assigning each label to a page.

Phase 2 – Spreading In the second phase, we iteratively improve the multi-page labeling created in Phase 1 by increasing the minimum number of labels per page without decreasing the value of the overall Objective (3.2) of BiCriteriaLabeling. In each iteration, we first identify all pages with the minimum number of labels, which we call *sparsest pages*; Algorithm 3 shows one such iteration. All pages that contain at least two more labels are referred to as *surplus pages*. The aim is to move labels from the surplus pages to the sparsest pages while preserving a valid multi-page labeling. To that end, we go through the sequence of surplus pages in reverse order and find the label with smallest weight that can be placed on the last sparsest page. We require that moving this label to a sparsest page does not decrease the Objective (3.2). We repeat this procedure until all sparsest pages have been resolved or there is no such label. In the former case, we continue with the next iteration and in the latter case we stop Phase 2 and return the multi-page labeling after undoing the changes of the current iteration. In the example in Figure 3.4, two iterations of Phase 2 led to improvements of the Objective (3.2) of BiCriteriaLabeling.

Algorithm 3: Iteration of Phase 2

```
Data: Multi-page labeling $\mathscr{L}$
Result: Improved multi-page labeling $\mathscr{L}'$
foreach sparsest page $R \in \mathscr{L}$ in reverse order do
    foreach surplus page $Q \in \mathscr{L}$ in reverse order do
        let $S$ be the set of labels on $Q$ that have no conflicts with labels on $R$
        if $S \neq \emptyset$ then
            place label with lowest weight in $S$ on $R$
if Objective (3.2) is not improved then
    restore the input labeling
```

3.5 Experiments

In the following section, we present our experiments and discuss their outcomes. We start by describing the experimental setup followed by a detailed description and evaluation of the individual experiments. We assess our models, i.e., our different optimization problems, by comparing optimal solutions for different setups. Further, we evaluate our heuristic by comparing the quality of the heuristic solutions with those of the exact approach.

3.5.1 Experimental Setup

A data set of point features provided by the recommendation portal Yelp[2] serves as the data basis for our experiments. This set contains information about local businesses in ten metropolitan regions in Canada and the United States of America. Assuming that higher-rated businesses are more relevant to a user, we use given star ratings with values between one and five as the labels' weights; we also allow half-star ratings. Moreover, we assume that with each page, i.e., with each swipe or click, the interest of a user substantially decreases. To that end, we define h to be an exponentially decreasing function setting $h(i) = 2^{1-i}$. Hence, for the first page the entire weight of the label is taken into account as effective weight, while with each further page the effective weight of a label is halved.

Considering the visualization of the point features, we introduce an axis-parallel rectangular label for each feature. All labels have the same width and height. Taking into account existing digital maps, we choose the size of the labels large enough to be easily readable, but small enough to avoid cluttered maps. In our setup, we use labels with a width of 50 pixels and a height of 30 pixels.

Adhering to established screen sizes for mobile devices, we set the size of the screen, i.e., the size of the displayed map section, to 365×325 pixels (Jackson and Pao, 2019). Keeping in mind the use case of searching for restaurants in a particular part of a city, we choose a map scale of 1:8000, which corresponds to the representation of some blocks of a city district. To create instances of multi-page labeling for our experiments, we used these specifications and moved a corresponding map frame in a grid-wise manner over the complete data set; see Figure 3.5 for an illustration. To extract an instance, we took all labels entirely contained in the according map section. To obtain reasonably large labelings, we only considered instances with at least 20 labels. In total, we have obtained 248 instances whose sizes range from 20 to 50 labels. In our

[2] https://yelp.com

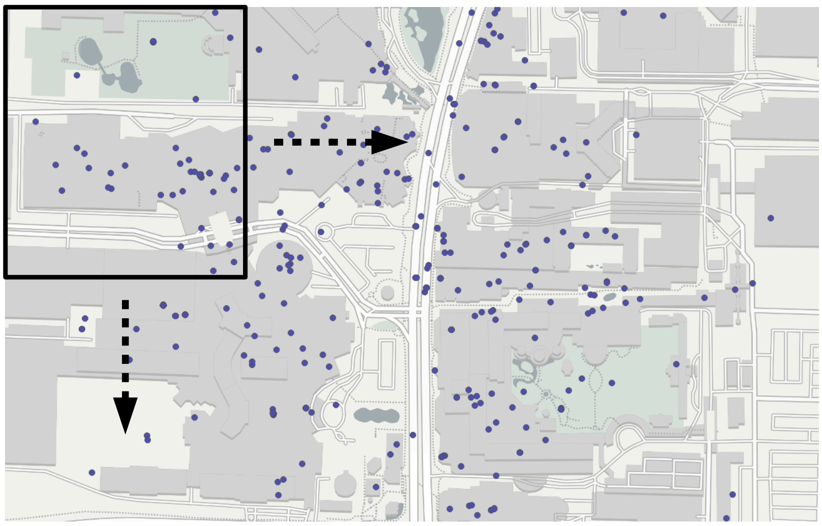


Fig. 3.5: The frame shows an exemplary map section used for sampling the data for one instance. The arrows indicate the grid-like movement of the frame. *Map attribution: Carto.*

experiments, we only considered sizes of instances (i.e., number of point features) for which at least eight instances were extracted; for each size we took the first eight of them.

We optimally solve MINNUMOFPAGES, WEIGHTEDPAGES, and BICRITERIALABELING using the ILP formulations introduced in Section 3.3. For ease of reading, we refer to MINNUMOFPAGES shortly as MINPAGE in the following. Further, we denote BICRITERIALABELING by BICRITERIAX with $X \in \{0, 25, 100\}$ where we set the balance factor $\alpha = \frac{X}{100}$. We note that BICRITERIA0 corresponds to WEIGHTEDPAGES and BICRITERIA100 only takes the minimum number of labels per page into account. In preliminary experiments that considered different values of α, we further found out that $\alpha = 0.25$ is a suitable compromise between the mean effective label weight and the minimum number of labels per page. Hence, we also consider BICRITERIA25 in our evaluation.

Our ILP implementations are written in Java and solved by the commercial Gurobi[3] Optimizer version 8.1.0. The greedy heuristic is implemented in JavaScript to provide a realistic setup that simulates a map application in a web interface. All computations were performed on an Intel Core i7-3770K CPU clocked at 3.5 GHz with 128 GB RAM.

3.5.2 Model Evaluation

To assess our mathematical models, we compare optimal solutions for the different optimization problems introduced in Section 3.2 that we gain by solving our ILP formulations. In particular, we aim to evaluate solutions of different configurations of our optimization problems with respect to the achieved quality of the three criteria C_{page}, C_{prio}, and C_{distr}. To that end, we computed the optimal solutions for MINPAGE, BICRITERIA0, BICRITERIA25, and BICRITERIA100 for all extracted instances. Figure 3.6 shows the results with regard to three different quality measures averaged for each instance size. Each of the measures reflects one of the three optimization criteria.

In the top-left plot in Figure 3.6, the average number of pages is shown which we see as a measure for Criterion C_{page}. We observe that BICRITERIA100 exceeds the minimum number of pages (see results of MINPAGE) by 1.3 on average and 3.1 in maximum. In other words: On average, 9.9% more pages are required to accommodate all labels, with a maximum of 18.9%. As BICRITERIA100 only maximizes the minimum number of labels per page, an optimal solution

[3] https://gurobi.com

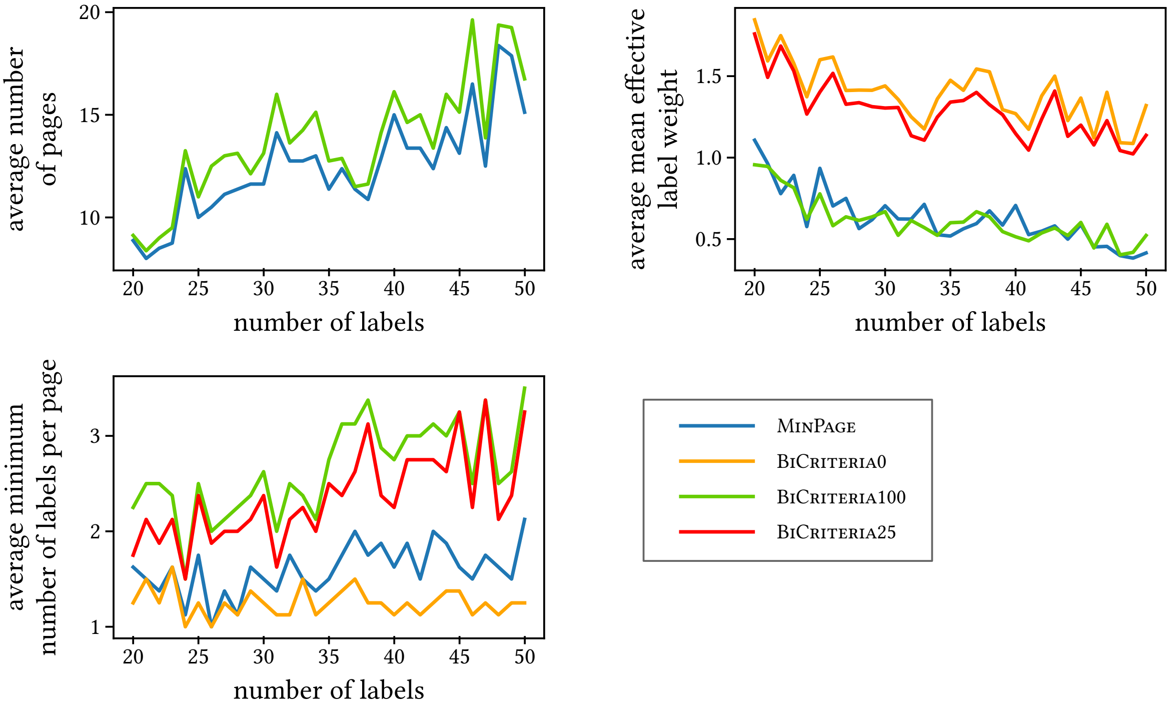


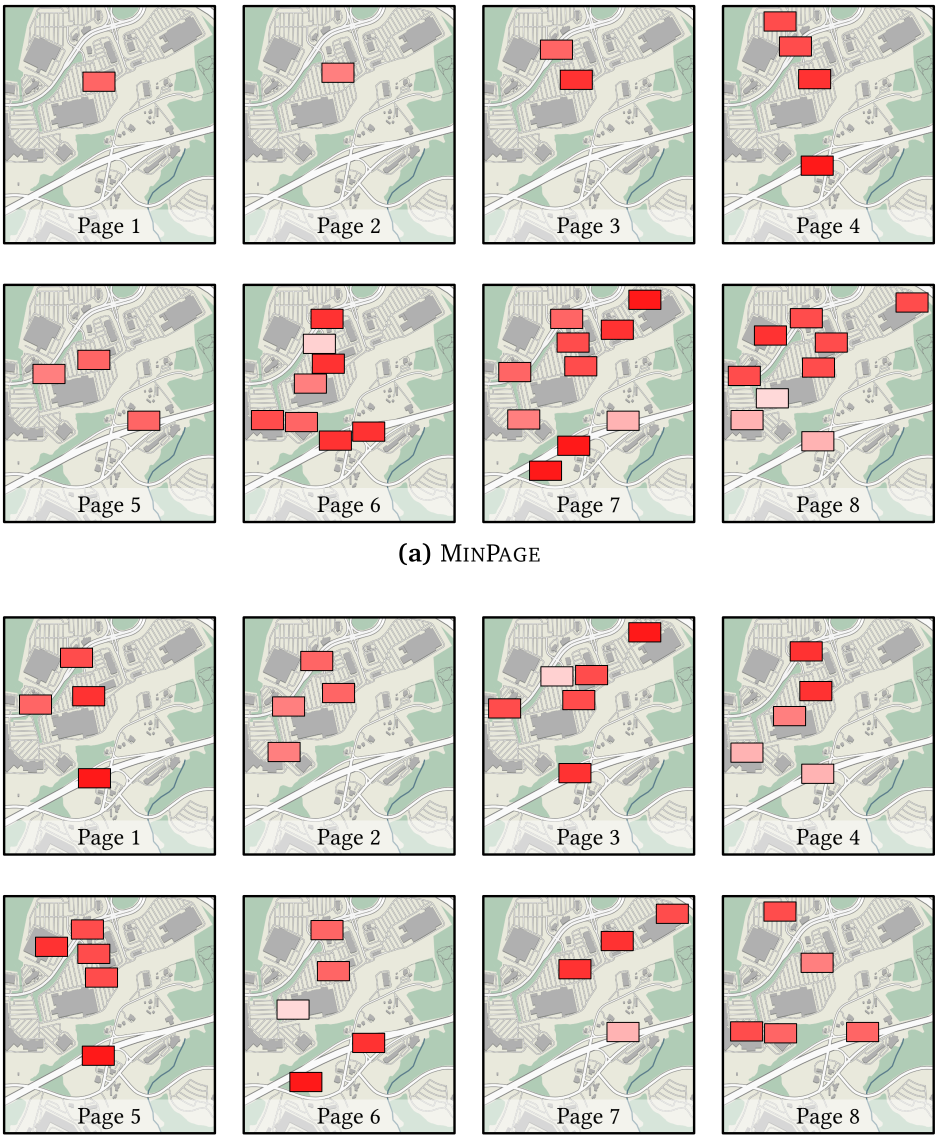
Fig. 3.6: Comparison of optimal results of different optimization problems with respect to different objectives. X-axis: Number of labels in instance (for each number of labels, eight instances have been considered). Y-axis: The average value of the eight instances with respect to the considered objective. In the top-left plot, BiCriteria0 and BiCriteria25 are not displayed, as they mostly coincide with the results of MinPage.

for BiCriteria100 might contain unnecessary pages whose labels could be redistributed on other pages. In comparison, both BiCriteria0 and BiCriteria25 achieve near optimal results. Since they mostly coincide with the results of MinPage, we do not depict these in Figure 3.6 (top-left). In particular, BiCriteria25 leads to an average number of pages that is only 0.2% above the minimum on average and 2.3% in maximum. Concerning BiCriteriaLabeling, this shows that an almost minimal number of pages can be achieved with a suitable choice of α, i.e., optimizing both Criteria C_{prio} and C_{distr} indirectly minimizes the number of pages as well.

To further explore the quality with respect to Criterion C_{prio}, we consider the mean effective label weight in the top-right plot of Figure 3.6. We observe that MinPage and BiCriteria100 achieve on average only 45% and 44% of the maximum mean effective label weight. Put differently, the mean effective label weight for MinPage and BiCriteria100 is less than half of that obtained with BiCriteria0. This was to be expected since both MinPage and BiCriteria100 do not take the weights of the labels into account. In contrast, BiCriteria25 yields solutions that come close to the maximum mean effective label weight with 92% on average and still 87% in minimum.

As a last criterion, we consider Criterion C_{distr} by comparing the average minimum number of labels per page in the bottom-left plot in Figure 3.6. On average, MinPage and BiCriteria0 lead to minimum numbers of labels per page of 1.6 and 1.3, respectively, which is only 61% and 49% of the optimal solution. A better result is achieved by solving BiCriteria25; on average, it yields 89% of the optimal solution of the minimum number of labels per page and in minimum it still reaches 76%.

To allow for a more conclusive visual assessment of these quantitative results, Figure 3.7 shows optimal labelings of an exemplary instance for MinPage and BiCriteria100. The corresponding



(a) MinPage

(b) BiCriteria100

Fig. 3.7: Optimal multi-page labelings of the same exemplary instance shown in Figure 3.1. In (a), MinPage is optimized and in (b) BiCriteria100 is optimized. *Map attribution: Carto*.

labelings of BiCriteria25 and BiCriteria0 are shown in Figure 3.1 and 3.2, respectively. We observe that the quantitative results discussed before are also reflected in the labelings. As MinPage merely minimizes the number of pages, the labels are distributed arbitrarily without considering their weights (see Figure 3.7a). BiCriteria0, which solely optimizes the average effective label weight, yields sparsely labeled pages (see Figure 3.2). For BiCriteria100, which maximizes the minimum number of labels per page, label priorities are not addressed (see Figure 3.7b).

Thus, while optimizing only one of the three optimization criteria at a time is not sufficient to ensure a satisfactory quality of all criteria, BiCriteria25 seems to produce a suitable compromise (see Figure 3.1). The labeling gained by solving BiCriteria25 also consists of eight pages, but, in contrast to the other problem variants, it also yields a good quality with regard to both Criterion C_{prio} and Criterion C_{distr}. While maintaining the minimum number of pages, high-rated labels are distributed on front pages and sparsely labeled pages are avoided. We conclude that, with a suitable choice of α, BiCriteriaLabeling is capable of producing multi-page labelings that are of high quality with respect to all three optimization criteria.

3.5.3 Evaluation of Heuristic

After assessing our different optimization problems by comparing optimal results, we now evaluate the quality of the heuristic introduced in Section 3.4. To that end, we compare heuristic solutions for the extracted instances with the optimal ones discussed in Section 3.5.2.

Since the heuristic optimizes the average effective label weight in Phase 1, we first compare heuristic results after this phase with the optimal ones of BiCriteria0 regarding the objective of WeightedPages; see the left plot in Figure 3.8. The first phase achieves 94% of the optimal solution on average and 89% in minimum. Thus, after applying Phase 1, our algorithm achieves a good approximation of WeightedPages. After additionally applying Phase 2, we compare the heuristic results with optimal ones of BiCriteria25 regarding the corresponding objective of BiCriteriaLabeling with $\alpha = 0.25$; see the right plot in Figure 3.8. The heuristic achieves 96% of the optimal solution on average and 93% in minimum.

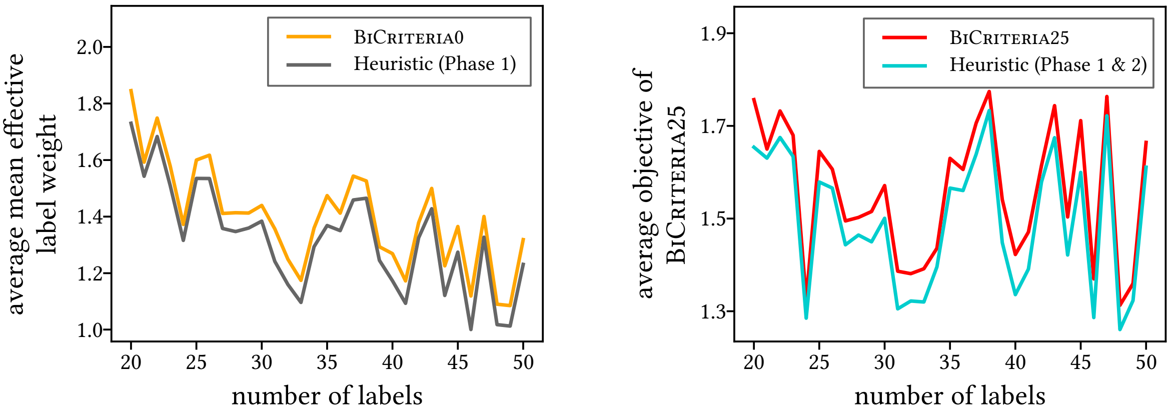


Fig. 3.8: Comparison of heuristic results to exact results with respect to different objectives. X-axis: Number of labels in instance (for each number of labels, eight instances have been considered). Y-axis: The average value of the eight instances with respect to the considered objective.

In summary, our greedy heuristic provides results that are close to optimal for both WeightedPages and BiCriteriaLabeling. In particular, for BiCriteriaLabeling this is to be emphasized since all criteria are taken into account here.

To further investigate the applicability of the greedy heuristic in web applications, we consider its running time. Especially in the development of interactive applications, it is known that response times of more than one second cause users to perceive the response as sluggish and to lose the feeling of a smooth interaction (Card et al., 1991; Nielsen, 1993). However, for all instances, solely running Phase 1 took only 0.13 ms on average and 3.15 ms in maximum. Running both Phase 1 and Phase 2 took 0.16 ms on average and 4.32 ms in maximum. We also ran the heuristic on a smartwatch, on which it also operates smoothly. Thus, we argue that our heuristic can generate labeling on demand and is suitable even for interactive applications.

3.6 Conclusion

We have introduced internal multi-page labeling: a concept for zoomless maps that distributes all labels in an area of interest on multiple pages. By clicking or swiping through the sequence of pages, this concept allows a user to access all the information without resorting to zooming or panning. From a modeling perspective, we have shown that finding a multi-page labeling directly translates into according graph coloring problems. We have introduced different problem formulations that model the general problem of finding a suitable label-page assignment taking into account different optimization criteria. To solve these problems, we have presented both an exact approach and a greedy heuristic. From our experiments based on real-world data, we can draw the following conclusions.

- Many of the existing mathematical concepts for map labeling can be transferred to multi-page labeling; by adding basic criteria such as minimizing the number of pages, we obtain our multi-criteria model.
- The number of pages remains close to minimal when we simultaneously optimize for effective label weights (C_{prio}) and a balanced label distribution (C_{distr}).
- By appropriately balancing all considered optimization criteria, we are able to compute solutions that satisfy all criteria without substantial compromises.
- Our simple heuristic yields solutions that are close to optimal with respect to our mathematical model and is fast enough for application in interactive maps.

Overall, we conclude that with internal multi-page labeling we have presented a suitable concept for zoomless maps. Our multi-criteria model addresses *accessibility* by both minimizing the number of pages and prioritizing the placement of relevant information. Thus, it provides the potential to access information with few interactions. Moreover, it considers a *manageable organization* of information by balancing the label distribution among the sequence of pages.

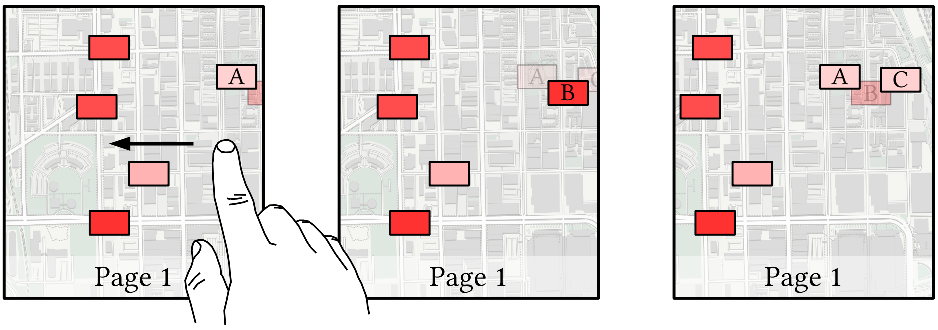


Fig. 3.9: Example of the effect of repeatedly appearing and disappearing labels while panning. In the initial map section on the left, *label A* is assigned to the first page and thus shown to the user. When panning the map to the left, *label B* moves into the map section and is placed instead of *label A* due to its higher weight. Panning the map further to the left, *label C* moves in, which can be placed simultaneously with *label A* without overlapping. Hence, *label A* reappears and *label B* is placed on another page. *Map attribution: Carto.*

However, although our experiments both validated the fitness of our model and proved the quality of the heuristic solutions, we see potential for further improvement. In our approach, we consider each map section separately. In the context of an interactive map, this means that we compute a new multi-page labeling each time a user pans the map section. This can lead to a particular label not being placed on the same page in two successive visualizations, meaning that the label-page assignment is not consistent. An example of such an effect is shown in Figure 3.9. When panning the map, labels repeatedly appear and disappear, which can quickly be perceived as annoying and irritating by the user. We will address this limitation in the following Chapter 4 and present an approach for generating multi-page labelings that ensure consistent label-page assignments.

4 ADVANCED INTERNAL MULTI-PAGE LABELING WITH CONSISTENCY

The following chapter mainly comprises joint work with Adalat Jabrayilov, Benjamin Niedermann, Petra Mutzel, and Jan-Henrik Haunert (Gedicke et al., 2021b). Benjamin Niedermann, Jan-Henrik Haunert, and I (Sven Gedicke) collaboratively developed the presented conceptual framework. Adalat Jabrayilov and Petra Mutzel contributed their expertise in mathematical programming, offering valuable insights that shaped the employed methodology. With the support of Benjamin Niedermann and Adalat Jabrayilov, I carried out the algorithmic implementation and evaluation. Both Jan-Henrik Haunert and Petra Mutzel provided supervision and assistance during the writing process.

We adhere to the general concept of *multi-page labeling*, in which all labels in an area of interest are distributed on multiple pages, but extend it to include a criterion that ensures a consistent label-page assignment when panning the map. More specifically, we propose a two-phase strategy that ensures that each label is placed on the same page in successive visualizations, i.e., in successive map sections. Instead of computing a solution for each map section separately, we use an optimization approach in a *pre-processing phase* to pre-compute global labelings on the level of a whole city. To retrieve individual labelings at a more local level, i.e., for map sections covering a small area of interest, we present a second *query phase* that allows the on-demand retrieval of labelings. As with internal multi-page labeling presented in Chapter 3, we optimize the label-page assignment such that the number of pages is small, a user has quick access to important information, and the distribution of labels on pages is balanced. With respect to the distribution of labels, however, we not only aim for a balanced distribution over all pages, but also consider a spatially balanced distribution on each page. In experiments on real-world data, we analyze different parameter settings and show that our mathematical model yields high-quality solutions.

4.1 Introduction

In the approach presented in the preceding Chapter 3, we focused on computing multi-page labelings on the fly. Following this strategy, each map section is considered separately. Hence, when a user pans the map, two successive map sections are not labeled consistently, i.e., the label-page assignment is not consistent during panning. This easily leads to distracting effects, such as labels repeatedly appearing and disappearing. Often, this effect is referred to as *flickering*. When interacting with the map, flickering is not only perceived as irritating and annoying, but it can also cause a user to lose track of the changes in successive visualizations.

This chapter presents an approach that addresses this issue. Contrary to our on-the-fly approach presented in Chapter 3, we introduce a concept that ensures a consistent label-page assignment when a user pans the map. We propose a strategy with two phases: a *pre-processing phase* and a *query phase*. In the pre-processing phase, we use an optimization approach to pre-compute labelings on the level of a whole city. We call such a solution a *global labeling*. The query phase is used for the on-demand retrieval of individual labelings at a more local level that can be accessed by a user on the fly. We refer to these solutions as *local labelings*. Figure 4.1 illustrates a global labeling and two exemplary local labelings marked with Ⓐ and Ⓑ. By querying local labelings from pre-computed global labelings, we ensure that a certain label is placed on the same page

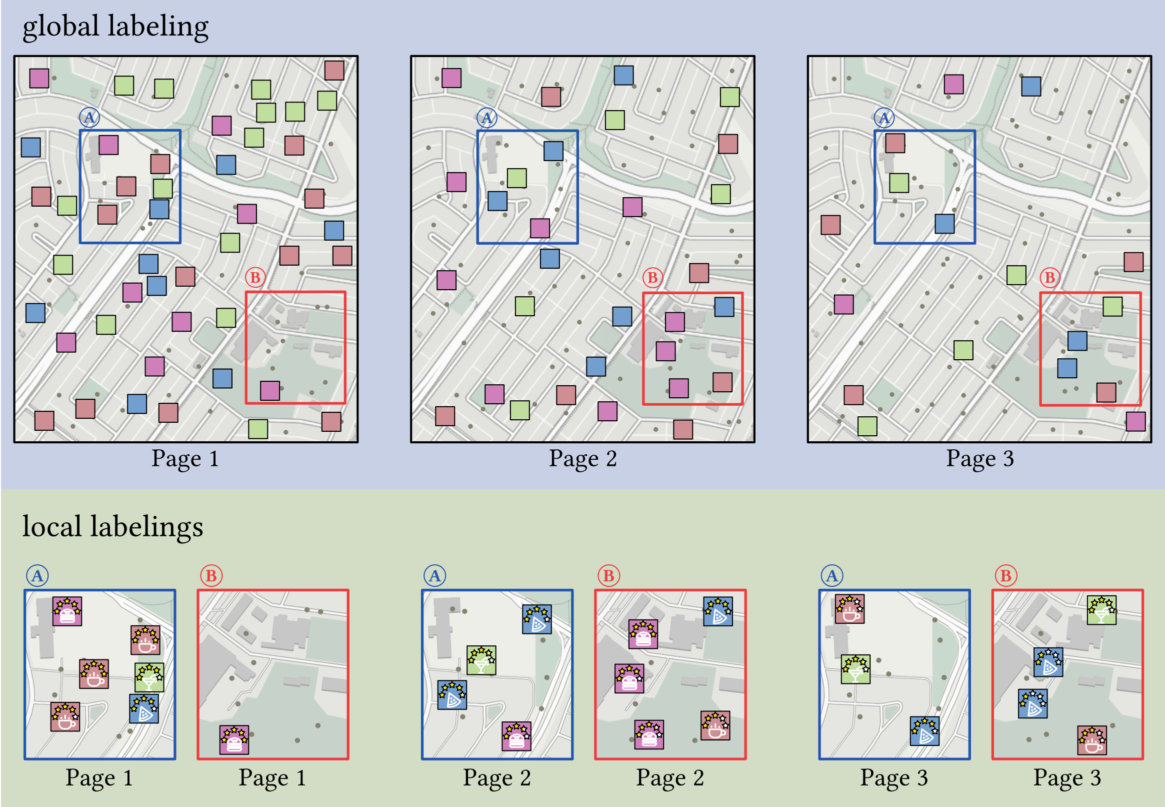


Fig. 4.1: Example of a global labeling consisting of three pages. Two exemplary smaller sections of the map are marked with Ⓐ and Ⓑ. The pages of the local labelings of Ⓐ and Ⓑ are shown below the corresponding pages of the global labeling. *Map attribution:* Carto.

for all map sections in which it is contained. Thus, when a user pans the map, the label is placed on the same page in two successive map sections, helping the user to keep track while exploring information.

We emphasize that we only need to compute a global labeling once. Instead of requiring a time-consuming computation for each map section, we can rapidly extract local labelings in the query phase. Assuming server-client communication, our strategy relieves the client in terms of computational overhead. Further, local labelings can be extracted for variable screen sizes and formats without the necessity of re-computing the underlying global labeling. To get an impression of the look-and-feel of our approach, we provide an illustrative demo video[1]. In this demonstration, we employ a real-world scenario of a user searching for an ice cream shop on their smartphone. We contrast our approach with a typical labeling method and underscore the advantages of our strategy.

For the computation of global labelings in the pre-processing phase, we consider multiple criteria that each labeling should satisfy. More precisely, we adopt the criteria from internal multi-page labeling that consider *accessibility*: a small number of pages and label priority. Keeping the number of pages small and distributing important information on front pages provides a user with quick access to relevant information. However, considering solely a small number of pages and feature priority can lead to a high label density on front pages. This can negatively affect both the map legibility and the perceived map complexity. Thus, to avoid both densely packed and only sparsely occupied pages, we additionally aim at a balanced distribution of labels on pages. Not only do we aim for each page to contain a similar number of labels, but we also take into

[1] `https://youtu.be/O0f34_7BriU`

account the spatial distribution on each page. Especially when keeping in mind that we obtain local labelings by extracting them from global labelings, a spatially balanced label distribution takes on special importance. If not sufficiently taken into account, pages in the local labelings can be empty or only sparsely occupied although there are many labels placed on later pages. An example of such a labeling is marked with Ⓑ in Figure 4.1. While the first page only contains one label, the following pages contain five and four labels, respectively. For a user browsing through such a labeling this can quickly become cumbersome and misleading. Summarizing, we consider the following three optimization criteria.

C_{page} The number of pages should be small.

C_{prio} Important labels should appear as close to the front of the sequence of pages as possible.

$\text{C}_{\text{distr}^s}$ The distribution of labels should be spatially balanced.

We note that that the distinction between C_{distr} considered in the model described in Chapter 3 and $\text{C}_{\text{distr}^s}$ here lies in the fact that we not only strive for a balanced distribution of labels among all pages, but also for a spatially balanced distribution of labels on each individual page.

Outline We transfer our concept into a mathematical model and use it to define the optimization problem of finding a multi-page labeling satisfying all our criteria in Section 4.2. Switching to the algorithmic design, we present an integer linear program (ILP) in Section 4.3 and give a detailed description of our two-phase approach in Section 4.4. In our experiments based on real-world data in Section 4.5, we determine a sweet spot that describes a suitable balance between our optimization criteria. Using different parameter settings, we compute global labelings and sample local labelings of different sizes, i.e., with different numbers of labels. These local labelings are then compared to reference labelings that are optimal according to our criteria. Our findings indicate that the quality of the local labelings aligns well with the optimal benchmarks. Further, we show that a majority of all labels can be placed on a small set of front pages. Section 4.6 discusses remaining limitations of our overall approach, particularly in the context of application in an interactive map. To conclude, Section 4.7 summarizes our key contributions and gives a brief outlook on future research.

4.2 Mathematical Model

To set up a formal model for our concept, we use similar basic assumptions as for internal multi-page labeling in Chapter 3. For the sake of completeness, we restate the key specifications.

We assume that we are given a set F of point features within a pre-defined map section $R \subset \mathbb{R}^2$ that we refer to as *frame*. We label each point feature $f \in F$ with an axis-parallel rectangle that we refer to as *label*. As we consider a label not only as an additional annotation, but also as a representation of a point feature itself, we place it centrally on p. We denote the set of all labels by L. Further, we introduce a weight function $w\colon L \to \mathbb{R}^+$ that assigns a weight $w(\ell)$ to each label $\ell \in L$ expressing the importance of the label's point feature. A multi-page labeling with κ pages is a partition $\mathscr{L} = \{\mathscr{P}_1, \ldots, \mathscr{P}_\kappa\}$ of L into κ subsets such that no two labels $\ell, \ell' \in \mathscr{P}_i$ of the same subset overlap each other. Finding a multi-page labeling can be modeled as a graph coloring problem as follows. We introduce a graph $G = (L, E)$ that contains a vertex for each label $\ell \in L$. Further, let $E \in L \times L$ be the set of edges containing an edge $\{\ell, \ell'\}$ for each two different labels $\ell, \ell' \in L$ that overlap. We refer to G as the *conflict graph* of L. We observe that distributing all labels L on κ pages without any overlaps is equivalent to assigning each vertex of G one of κ given colors, such that no two adjacent vertices have the same color. In other words, a multi-page labeling with κ pages corresponds to a κ-coloring of G.

Following Criterion C_{page} and Criterion C_{prio}, we aim for a small number of pages and prefer important labels to appear on the first pages. We assign a rating $h(i)$ to each page i, which is expressed with the function $h\colon \mathbb{N} \to [0,1]$. As we assume that the users' interest decreases substantially with the number of pages they have to browse through, we choose h to be monotonically decreasing. Further, we define $h(i) \cdot w(\ell)$ as *effective label weight* assuming that the label ℓ is placed on page i. It specifies how much of each weight $w(\ell)$ of a label placed on page i is considered. As the position i of the pages increases, the effective label weight decreases.

Unlike in internal multi-page labeling in Chapter 3, we do not only balance the number of labels over the sequence of pages, but also distribute them on the pages spatially; see Criterion C_{distr^s}. To that end, we use an approach from space geodesy in which the distribution of space observations in the sky is determined by hierarchically partitioning the local sky into cells (Corbin et al., 2020). Each cell containing at least one observation contributes a certain score to an overall sky coverage score. Transferring this concept to multi-page labeling, for each page i we subdivide the frame into m grids $\mathcal{G}_1, \dots, \mathcal{G}_m$ of different levels of partitioning; see Figure 4.2. Assuming a square frame, the total number N_j of cells contained in a grid $\mathcal{G}_j$ increases exponentially with increasing j so that $N_j = N_1 \cdot 4^{j-1}$. Let $\mathcal{C}$ be the set of all cells over all m levels of partitioning. Each cell $c \in \mathcal{C}$ is assigned a score $s(c)$, which depends on the level of partitioning of the corresponding grid $\mathcal{G}_j$. More precisely, the score $s(c)$ of a cell c is defined as $s(c) = \frac{1}{N_j}$. Thus, the cells of the coarsest partition contribute the highest score while the score of a cell decreases with increasing level of partitioning. We refer to a cell c as *empty* if no label is placed in it, i.e., there is no label whose point feature is located in c. Further, for a multi-page labeling $\mathcal{L} = \{\mathcal{P}_1, \dots, \mathcal{P}_\kappa\}$ we define $\mathcal{E}(\mathcal{P}_i)$ as the set of empty cells on the i-th page if $\mathcal{P}_i$ contains at least one label. Otherwise, if $\mathcal{P}_i$ does not contain a label, we define $\mathcal{E}(\mathcal{P}_i) = \emptyset$.

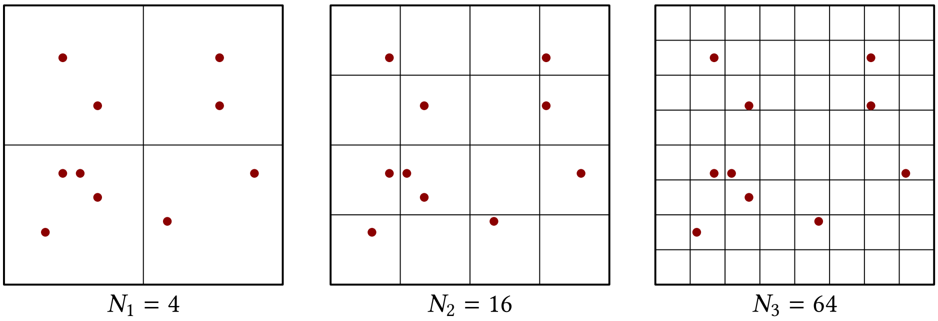


Fig. 4.2: Example of dividing a given map section containing several point features (red dots) into three grids with different levels of partitioning. The fineness of the grids and thus the number N_j of cells increases with increasing level of partitioning (from left to right).

We aim for a multi-page labeling with κ pages that satisfies all criteria C_{page}, C_{prio}, and C_{distr^s}. However, we do not necessarily require these criteria to be considered on all κ pages. Instead, we keep the number of optimized pages variable with a parameter λ. In the following, we refer to λ as the *optimization scope* and to all optimized pages as *focus pages*. We define a tuple $\Upsilon = (L, \mathcal{E}, w, s, h, \lambda)$ to be an instance of our mathematical model (see Table 4.1).

Tab. 4.1: Parameters of an instance Υ.

L	set of labels	s	cell scores	w	label weights
$\mathcal{E}$	set of empty cells	h	page rating	λ	optimization scope

We define the following optimization problem.

Problem (MAPCOVERAGELABELING). *Given an instance* $\Upsilon = (L, \mathcal{E}, w, s, h, \lambda)$ *and a constant* $\alpha \in [0,1]$*, find for any* $\kappa \geq \lambda$ *the multi-page labeling* $\mathcal{L} = \{\mathcal{P}_1, \dots, \mathcal{P}_\kappa\}$ *that maximizes*

$$(1-\alpha) \cdot f_\Upsilon(\mathcal{L}) \; - \; \alpha \cdot g_\Upsilon(\mathcal{L}), \tag{4.1}$$

with

$$f_\Upsilon(\mathcal{L}) = \frac{1}{n} \cdot \sum_{i=1}^{\lambda} \sum_{\ell \in \mathcal{P}_i} h(i) \cdot w(\ell)\,, \tag{4.2}$$

$$g_\Upsilon(\mathcal{L}) = \frac{1}{\omega(G)} \cdot \sum_{i=1}^{\lambda} \sum_{c \in \mathcal{E}(\mathcal{P}_i)} s(c)\,, \tag{4.3}$$

where n *is the number of labels,* G *is the conflict graph of* L*, and* $\omega(G)$ *is the number of vertices in a maximum clique in* G*.*

By maximizing f_Υ we aim for both a small number of pages (Criterion $\mathrm{C}_{\mathrm{page}}$) and important labels to appear as close to the front of the sequence of pages as possible (Criterion $\mathrm{C}_{\mathrm{prio}}$). As we maximize the sum $\sum_{i=1}^{\lambda} \sum_{\ell \in \mathcal{P}_i} h(i) \cdot w(\ell)$ of effective label weights divided by the number n of labels, we denote f_Υ as the *mean effective label weight*. With g_Υ we consider Criterion $\mathrm{C}_{\mathrm{distr}^s}$ by avoiding the occurrence of empty cells on non-empty pages. We divide the sum $\sum_{i=1}^{\lambda} \sum_{c \in \mathcal{E}(\mathcal{P}_i)} s(c)$ of scores by the size $\omega(G)$ of the maximum clique in G in order to adjust g_Υ to a similar order of magnitude as f_Υ. Note that the maximum clique in G can be found in polynomial time since G represents an intersection graph of axis-parallel rectangles (Imai and Asano, 1983). In the following, we refer to g_Υ as *map coverage score*.

4.3 Mathematical Programming

Generally, multi-page labeling asks for an assignment of labels to pages such that no two labels placed on the same page are in conflict with each other. This problem can be considered as a graph coloring problem. We can transfer a set of labels into a graph that contains a vertex for each label and an edge for each label-label conflict. Distributing the labels on κ pages without introducing graphical conflicts then corresponds to a κ-coloring of the graph. Even for graphs representing the conflicts among a given set of rectangles, the problem of finding a κ-coloring with minimal κ is known to be NP-hard (Imai and Asano, 1983). A common approach for tackling coloring problems is the use of integer linear programming. There exist a vast number of state-of-the-art ILP approaches, which differ in their suitability for different types of graphs. Especially for sparse graphs, the *partial-ordering* based model of Jabrayilov and Mutzel (2018) has proven to be favorable with respect to the running time (see Section 2.5.3). It considers the vertex coloring problem as a partial-ordering problem (POP). Instead of directly assigning a color to each vertex, a partial order is generated from the union of the set of vertices and the set of ordered colors. Since we aim to solve large city-wide instances on a rather small scale, we deem the graphs for MAPCOVERAGELABELING to be rather sparse. Hence, we formalize an ILP formulation that models MAPCOVERAGELABELING as POP.

We assume that κ pages $(1, \dots, \kappa)$ are linearly ordered, where κ is defined as an upper bound of the number of pages which is at most $|L|$. With $\kappa = |L|$ we can place each label on its own page. We refer to a page on which at least one label is placed as *non-empty*. To identify the last page that is non-empty, we introduce an additional dummy label τ. We determine the relative order of each label with respect to each page in the page ordering. For every page i and every label $\ell \in L \cup \{\tau\}$ our model provides the information if ℓ is assigned to a page after the i-th page or not. We denote this relation by $\ell \succ i$. The pages and the labels build a partially ordered set in which all pairs of the form (ℓ, i) with $\ell \in L \cup \{\tau\}$ and $i = 1, \ldots, \kappa$ are comparable.

To express the partial order, we define the binary variable

$$y_{i,\ell} = \begin{cases} 1 & \ell \succ i\,, \\ 0 & \text{otherwise}\,. \end{cases} \qquad \text{for all} \quad \ell \in L \cup \{\tau\}, 1 \le i \le \kappa \tag{4.4}$$

For each label ℓ we require with the following constraints that it is placed on exactly one page.

$$y_{i-1,\ell} \ge y_{i,\ell} \qquad \text{for all} \qquad \ell \in L \cup \{\tau\}, 2 \le i \le \kappa, \tag{4.5}$$

$$y_{\kappa,\ell} = 0 \qquad \text{for all} \qquad \ell \in L \cup \{\tau\}. \tag{4.6}$$

To ensure that no two conflicting labels are placed on the same page, we further require

$$y_{1,\ell} + y_{1,\ell'} \ge 1 \qquad \text{for all} \qquad \{\ell, \ell'\} \in E\,, \tag{4.7}$$

$$(y_{i-1,\ell} - y_{i,\ell}) + (y_{i-1,\ell'} - y_{i,\ell'}) \le 1 \qquad \text{for all} \qquad \{\ell, \ell'\} \in E, \; 2 \le i \le \kappa\,. \tag{4.8}$$

If a label ℓ is placed on the first page $y_{1,\ell} = 0$ must hold. With Constraint (4.7) we ensure that this can only be true for at most one of two conflicting labels ℓ and ℓ'. Further, if a label ℓ is placed on a page $i \ge 2$, then $(y_{i-1,\ell} - y_{i,\ell}) = 1$ must hold. With Constraint (4.8) this cannot be the case for two conflicting labels on the same page.

To identify the last non-empty page by placing the dummy label τ on that page, the following conditions must hold.

$$y_{i,\tau} \ge y_{i,\ell} \qquad \text{for all} \qquad \ell \in L, 1 \le i \le \kappa\,, \tag{4.9}$$

$$y_{i,\tau} \le \sum_{\ell \in L} y_{i,\ell} \qquad \text{for all} \qquad 1 \le i \le \kappa\,. \tag{4.10}$$

With Constraint (4.9) we ensure that there is no label $\ell \in L$ that is placed on a later page than the one on which τ is placed. Additionally, Constraint (4.10) enforces that there is at least one other label on the page containing τ.

To consider Criterion $\mathrm{C}_{\mathrm{distr}^s}$ in our ILP formulation, for each cell $c \in \mathscr{C}$ we introduce a set $L_c \subseteq L$ of labels whose point features are geometrically located in c. To obtain for each cell whether it is empty on a page that does not succeed the page containing τ (i.e., the last page containing at least one label), we introduce a binary variable $q_{c,i} \in \{0,1\}$ for each cell c and page i. More precisely, we interpret $q_{c,i}$ such that $q_{c,i} = 1$ if cell c is empty on a page i that does not succeed the page containing τ, i.e, there is no label placed on page i, whose point feature is located in c. We ensure that $q_{c,i}$ is assigned correctly by requiring

$$1 - \sum_{\ell \in L_c} (1 - y_{1,\ell}) \le q_{c,1} \qquad \text{for all} \qquad c \in \mathscr{C}\,, \tag{4.11}$$

$$y_{i-1,\tau} - \sum_{\ell \in L_c} (y_{i-1,\ell} - y_{i,\ell}) \le q_{c,i} \qquad \text{for all} \qquad 2 \le i \le \kappa, c \in \mathscr{C}\,. \tag{4.12}$$

If no label $\ell \in L_c$ is placed on the first page, the sum $\sum_{\ell\in L_c}(1-y_{1,\ell})$ equals 0 in Constraint (4.11) and therefore $q_{c,1} = 1$ must hold. As $y_{i-1,\tau} = 1$ holds for all non-empty pages $i \geq 2$, Constraint (4.12) ensures that $q_{c,i} = 1$ must hold if no label $\ell \in L_c$ is placed on the i-th page.

Satisfying all constraints, we search for a multi-page labeling that maximizes

$$\frac{\beta}{n}\cdot\underbrace{\sum_{\ell\in L}\left((1-y_{1,\ell})\cdot h(1)\cdot w(\ell)+\sum_{i=2}^{\lambda}(y_{i-1,\ell}-y_{i,\ell})\cdot h(i)\cdot w(\ell)\right)}_{\text{I}}$$
$$-\frac{\alpha}{\omega(G)}\cdot\underbrace{\sum_{i=1}^{\lambda}\sum_{c\in\mathscr{C}}s(c)\cdot q_{c,i}}_{\text{II}}\,, \tag{4.13}$$

with $\beta = (1-\alpha)$. Each solution of this ILP formulation corresponds to a valid multi-page labeling. When maximizing Objective (4.13), the occurrence of empty pages between non-empty pages in the solution is avoided. More precisely, for $\alpha < 1$, labels are placed preferably on front pages by maximizing Part I. Moreover, for $\alpha = 1$, empty pages are avoided with Part II, since empty cells on an empty page preceding the page containing τ would lower the overall objective score.

4.4 A Two-Phase Approach

Solving MapCoverageLabeling for each map section separately can be a time-consuming task that is not suitable for interactive applications. Further, it would not guarantee a consistent label-page assignment. Hence, we pre-compute global labelings in a first *pre-processing phase* and extract local labelings in a subsequent *query phase*. In the following, we describe both phases in more detail. Figure 4.3 shows a flowchart of the processes in both phases.

Pre-Processing Phase

We compute global labelings on the level of an entire city in two steps. In the first step, we compute a labeling optimizing a certain number λ of focus pages, which become the *front pages*. In the second step, we consider all labels that could not be placed on the front pages and distribute them on additional pages, which become the *back pages* of the labeling. At the core of both steps we solve MapCoverageLabeling with our ILP formulation (see Section 4.3). We emphasize that we could consider the front and back pages in an integrated way. However, this can lead to local labelings with an unnecessarily large number of pages as we explain in the following.

The minimum number of pages necessary to distribute all labels without any overlaps is determined by the size $\omega(G)$ of the maximum clique of labels. Hence, if an instance contains a large clique of labels (e.g., labels of restaurants in a shopping mall), the global labeling consists of an accordingly large number of pages. A fictive example is illustrated in Figure 4.4. Due to the size $\omega(G) = 8$ of the maximum clique in the input, a corresponding global labeling would consist of at least eight pages. As the map coverage score g_Υ aims at a spatially balanced distribution of labels on all focus pages (see Objective (4.1)), the number of labels is kept balanced among all eight pages. Hence, it is likely that an extracted local labeling has a large number of pages with only few labels per page. In the example at hand, a possible local labeling of frame Ⓐ is shown on the right. Although all labels contained in frame Ⓐ could be placed on one single page, the local labeling comprises four pages. Additionally, both the distribution of important labels and the spatial distribution of the labels could be improved by placing all labels on one single page.

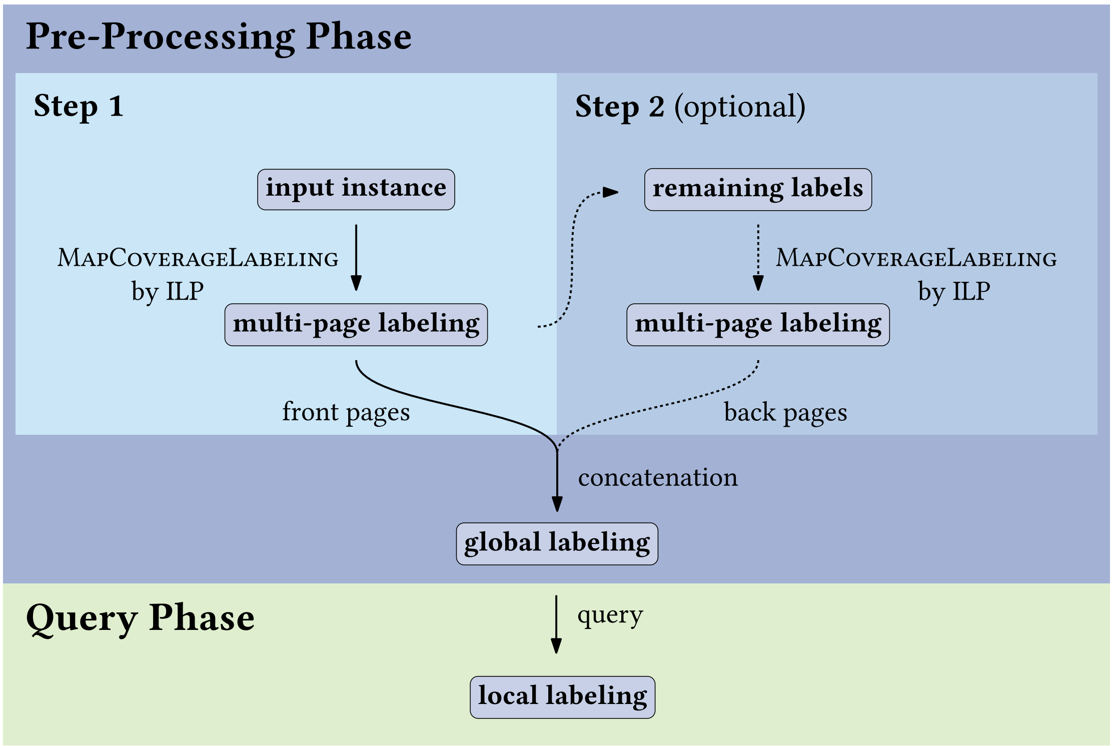


Fig. 4.3: A schematic flowchart of the two-phase approach. Since the second step of the first phase is optional, the connections to the first step are shown as dashed lines.

Assuming that a user's interest decreases with the number of explored pages, it is especially important to optimize the considered criteria on the first pages of a multi-page labeling. Hence, we argue that it can be reasonable to consider only a small number i of focus pages (i.e., $\lambda = i$) that become the front pages of the labeling.

Therefore, we propose to create the front and back pages in two separate steps (see the flowchart in Figure 4.3). In the first step, we solve MapCoverageLabeling for all given labels with λ set to a certain number i of pages. These pages become the front pages of the labeling. We refer to all labels that could not be placed on front pages as *remaining labels*. In the second step, only the remaining labels are used as input for solving MapCoverageLabeling a second time. This time, we set λ to the number of remaining labels to make sure that all remaining labels are assigned to focus pages. We consider the resulting pages to be the back pages of the labeling. Finally, we concatenate the front pages from the first step and the back pages from the second step and obtain our global labeling. We emphasize that applying the second step is optional. Assuming that a user does not want to browse through more than the front pages anyway, one can simply neglect the remaining labels after the first step.

For the reader's convenience, we denote a global labeling with λ set to a certain number i of front pages as *global* i*-labeling*. Thus, in a global κ-labeling, all pages necessary for placing the labels are front pages. We note that global κ-labelings are obtained without applying the second step as there are no remaining labels.

Query Phase

The query phase is used for extracting local labelings from the global labelings that we obtain after the pre-processing phase. This allows us to retrieve local labelings for any region of interest covered by the global labeling, flexibly accommodating different screen sizes and formats. More

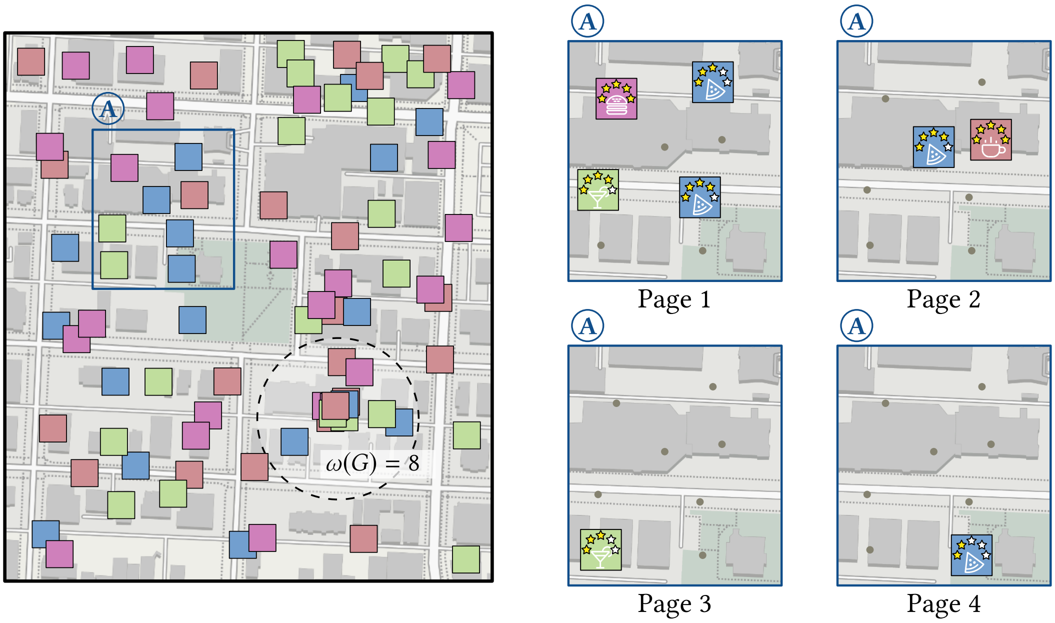


Fig. 4.4: A fictive input instance for MapCoverageLabeling is shown on the left. The maximum clique of labels with a size of $\omega(G) = 8$ is highlighted with the dashed circle. A possible local labeling (marked with A) is shown on the right. *Map attribution: Carto*.

precisely, we can extract a local labeling by sampling for each page all labels contained in the frame representing a current region of interest (see Figure 4.1). In case the frame on a page is empty and there is no more information on subsequent pages, we neglect that page. We denote local labelings that we extract from a global i-labeling as *local i-labelings*.

4.5 Experiments

In the following, we explain, evaluate, and discuss our experiments in which we assess multi-page labelings obtained with our two-phase approach.

4.5.1 Experimental Setup

We consider the use case of a pedestrian searching for restaurants within an unfamiliar city. For this purpose, we use data of restaurants from four metropolitan areas in the USA and Canada provided by the review portal Yelp[2]. More precisely, we use data of restaurants in *Las Vegas (USA)*, *Pittsburgh (USA)*, *Calgary (Canada)*, and *Montréal (Canada)*. Assuming the use of a mobile device, we set the size of the screen, i.e., the size of the frame of each local labeling, to 300 × 300 pixels. This corresponds to a common resolution on smartwatches (Jackson and Pao, 2019). Since our assumed use case addresses applications for pedestrians, we choose a rather large map scale of 1:4000 (see Figure 4.5). Using a spherical mercator projection (EPSG:3857), the frames comprise different area sizes depending on the latitude. For each city, Table 4.2 gives these area sizes. We compute one global labeling for every city, each with its frame centered on the respective city center. As we intend global labelings to cover the entire city we choose them to comprise an extent of 9600 × 9600 pixels at scale 1:4000. The area coverage for each city is

[2] https://yelp.com

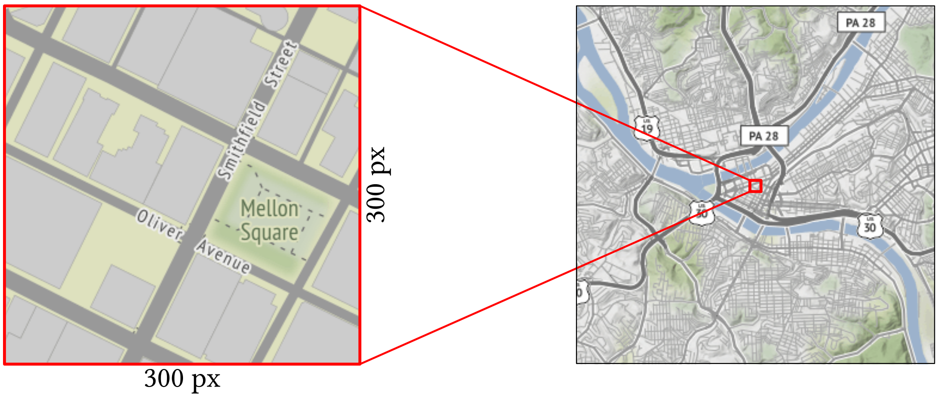


Fig. 4.5: Map scale and resolution used in the experiments. An exemplary frame with a resolution of 300 × 300 pixels at scale 1:4000 that we use for the local labelings is shown on the left. Shown on the right is the map extent that is covered when computing a global labeling. Solely for the purpose of illustrating the whole map extent we display this frame with a smaller scale. *Map attribution: Stamen.*

shown in Table 4.2. The number of restaurants contained in the cities' frames ranges from 2070 in Calgary to 5349 in Las Vegas.

For each restaurant, a user-given star rating between one and five is provided that we use as the weighting $w(\ell)$ of each restaurant's label ℓ. We note that half-star ratings are also taken into account. We choose the label size as a compromise between a sufficiently large space for filling the labels with information and a rather small size to occlude as little map background as possible. Based on these considerations, each label has a size of 40 × 40 pixels.

Tab. 4.2: Extents of frames for local and global labelings for each considered city [m].

	Las Vegas	Pittsburgh	Calgary	Montréal
local	290 × 290	272 × 272	225 × 225	250 × 250
global	9250 × 9250	8717 × 8717	7203 × 7203	8019 × 8019

We express the weight $h(i)$ of each page i by the function $h(i) = 2^{1-i}$ which decreases exponentially. Thus, with the effective label weight $h(i) \cdot w(\ell)$, the entire weight of a label is considered on the first page, half of a label's weight is considered on the second page, and so on.

We divide each frame into multiple grids of different levels of partitioning. In particular, we consider eight levels of partitioning. Let $\mathcal{G}_1$ be the grid with the lowest level of partitioning, we divide the frame into four equally sized squares such that $N_1 = 4$; see Figure 4.2. Following the definitions of Section 4.2, each cell c of this grid is assigned a score $s(c) = 1/4$. Correspondingly, the finest grid $\mathcal{G}_8$ has a total number of $N_8 = 65{,}536$ cells each with a score of $s(c) = 1/65{,}536$. As the frame of each global labeling covers 9600 × 9600 pixels, each cell of this grid has a size of 37.5 × 37.5 pixels. Keeping in mind our label size of 40 × 40 pixels, we note that a finer grid would not be reasonable, since each cell can contain at most one label.

For all our experiments, we used the two-phase approach introduced in Section 4.4. We implemented our approach in Java and used the software Gurobi[3] 8.1.0 for solving the ILP formulations. We ran our experiments on an Intel(R) Xeon(R) W-2125 CPU clocked at 4.00GHz with 128 GiB RAM. We were able to solve most instances within a few minutes up to one hour at maximum. We note that only the pre-computation phase requires such a high-performance setting to

[3] https://gurobi.com

gain the global labelings. For the query phase, we only extract the local labelings from the global labelings, which can be done on the fly.

4.5.2 Balancing the Objective

In a first evaluation step, we assess our model and determine a suitable balance between the mean effective label weight f_Υ and the map coverage score g_Υ (see Objective (4.1)). We aim for a value of the balancing factor α that describes a sweet spot between the optimization criteria.

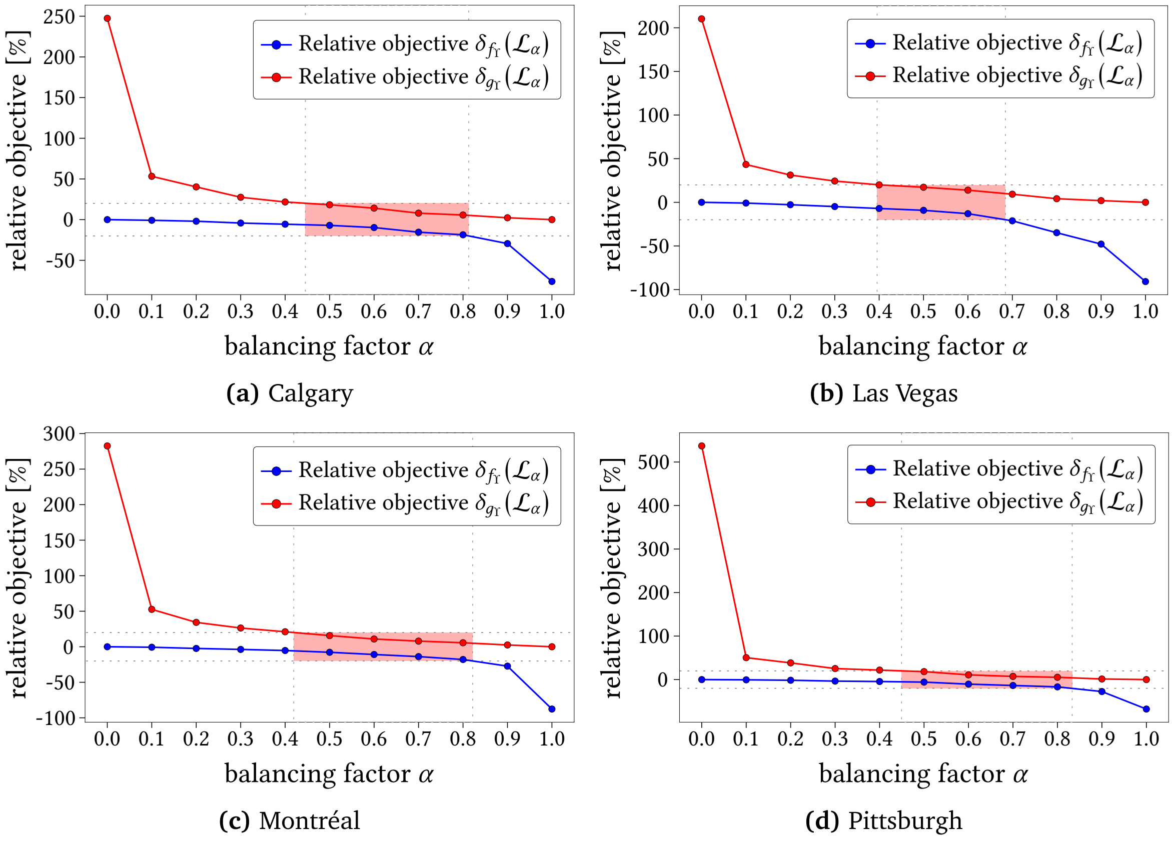
For this purpose, we denote a multi-page labeling with a specific α by $\mathscr{L}_\alpha$ and define

$$\delta_{f_\Upsilon}(\mathscr{L}_\alpha) = \frac{f_\Upsilon(\mathscr{L}_\alpha) - f_\Upsilon(\mathscr{L}_0)}{f_\Upsilon(\mathscr{L}_0)} \cdot 100\%, \tag{4.14}$$

$$\delta_{g_\Upsilon}(\mathscr{L}_\alpha) = \frac{g_\Upsilon(\mathscr{L}_\alpha) - g_\Upsilon(\mathscr{L}_1)}{g_\Upsilon(\mathscr{L}_1)} \cdot 100\% \tag{4.15}$$

as the *relative objectives* of a labeling $\mathscr{L}_\alpha$ compared to a labeling $\mathscr{L}_0$ and $\mathscr{L}_1$, respectively. We note that $\mathscr{L}_0$ is optimal with respect to f_Υ and $\mathscr{L}_1$ is optimal with respect to g_Υ.

Using our two-phase approach, we compute global κ-labelings and hence consider all pages of a labeling to be front pages. We sample the value for α with a step width of 0.1 and compute the global κ-labeling $\mathscr{L}_\alpha$ for each sampled α and each city. The resulting relative objectives for the four considered cities are shown in Figure 4.6.



(a) Calgary

(b) Las Vegas

(c) Montréal

(d) Pittsburgh

Fig. 4.6: Relative objectives δ_{f_Υ} and δ_{g_Υ} of the four considered city instances. For each instance, the relative objectives of sampled values for $\alpha \in [0,1]$ with a step width of 0.1 are shown. Red zones indicate ranges of α for which δ_{f_Υ} is not below 20% and δ_{g_Υ} is not above 20%.

We observe that the relative objective δ_{f_Υ} decreases monotonically with increasing value for α, which implies a decrease in quality regarding the mean effective label weight. In contrast, the quality regarding the map coverage score decreases with an increasing relative objective δ_{g_Υ}. This is because we maximize the negative $-g_\Upsilon$ of the map coverage score in our model; see Objective (4.1). Consequently, an appropriate value for α that balances both f_Υ and g_Υ should yield values close to 0 for both relative objectives. The red colored sections in Figure 4.6 indicate the range for α for which δ_{f_Υ} is not below 20% and δ_{g_Υ} is not above 20%. Thus, choosing a value for α within this range yields a possible compromise between the mean effective label weight and the map coverage score. Although the considered cities differ in both structure and density of labels, the ranges of an appropriate value for α are similar. For all four considered cities, a common intersection of the ranges can be observed between $\alpha = 0.45$ and $\alpha = 0.69$. Thus, for our following experiments, we set $\alpha = 0.6$ as an intermediate value that sufficiently balances both the mean effective label weight and the map coverage score. However, we note that this specification is only a suggestion. Depending on which criterion should be given a higher priority, other values of α can be reasonable.

4.5.3 Evaluation of Local κ-Labelings

In the following, we want to assess the quality of local κ-labelings, i.e., labelings that result when all pages are front pages and thus optimized in one step. To that end, we compare them with respect to locally optimal *reference labelings* that we obtain by applying our ILP formulation to each instance of the corresponding local κ-labeling. We use the sweet spot of $\alpha = 0.6$ obtained in the preceding Section 4.5.2 and extract local κ-labelings from global κ-labelings for each considered city.

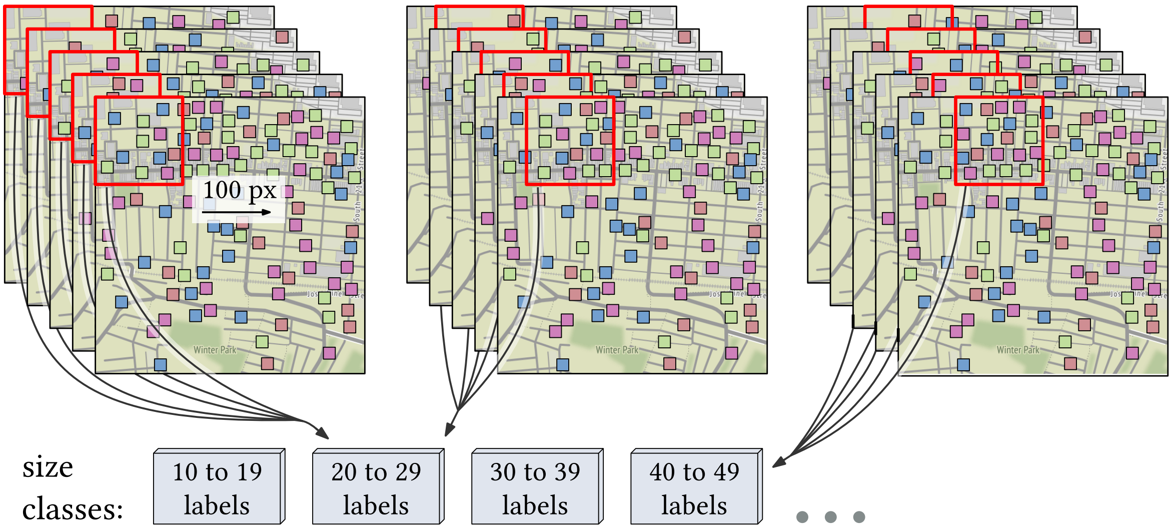


Fig. 4.7: Extraction of local labelings from a global labeling. The five stacked layers represent the pages of the global labeling. For each position of the frame (shown in red), a local labeling is extracted by sampling for each page all labels contained in the frame. Depending on the total number of labels, local labelings are categorized into size classes. *Map attribution: Stamen.*

Extracting Local κ-Labelings For each city, a frame of 300 × 300 pixels is moved over the global κ-labelings in a grid-like manner with a step size of 100 pixels; see Figure 4.7. For each position of the frame, we extract a local κ-labeling by using our approach of the query phase (see Section 4.4). As we are interested in labelings of reasonably dense point sets, we neglect local κ-labelings with a total number of less than ten labels. We denote a local κ-labeling by $\mathcal{L}^{\kappa}_{\text{loc}}$. Further, to analyze the local κ-labelings in a differentiated way, we categorize them into different *size classes*. More precisely, we define size classes that contain labelings with 10 to 19 labels, 20 to 29 labels, and so on. To compare the results of all cities, we only consider the size classes that contain at least one local κ-labeling for all cities. Hence, the size class containing labelings with 70 to 79 labels is the class with the largest number of labels that we consider in the following.

Computing Reference Labelings To investigate the quality of the local κ-labelings, we compare them to reference labelings that are optimal with regard to the optimization criteria. We compute such a reference labeling by sampling all labels contained in a local κ-labeling and using these labels as input for our ILP formulation (see Section 4.3). Figure 4.8 gives an illustration of this procedure. Since the frames comprise only 300 × 300 pixels, fewer levels of partitioning are necessary to provide a grid coverage that is equivalent to the one from the global labelings. Hence, we use the same experimental setup that we use for computing global labelings with the exception that we consider only three levels of partitioning. We denote a resulting reference labeling by $\mathscr{L}_{\text{ref}}$. Solving each local instance optimally can take a certain amount of time that is not necessarily suitable for interactive scenarios. We emphasize that we use the reference labelings for evaluation purposes only.

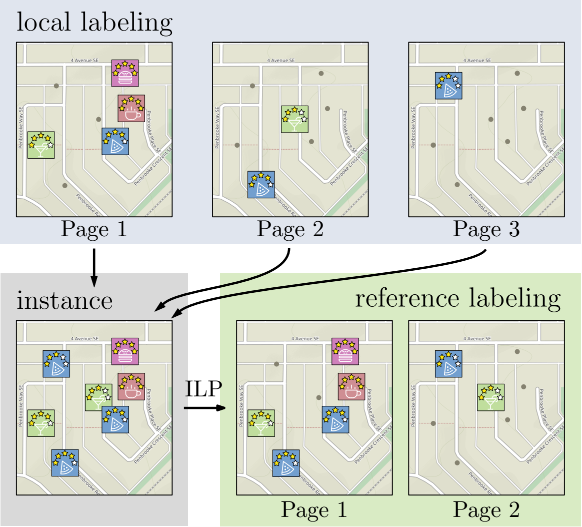


Fig. 4.8: Obtaining the reference labeling of a local labeling. Labels contained in the local labeling form an instance that is solved with our ILP. *Map attribution: Stamen.*

Comparison to Reference Labelings We compare the local κ-labelings with the reference labelings only with respect to Criterion C_{page}. As the total number of pages also affects the quality of the other two criteria C_{prio} and $\mathrm{C}_{\text{distr}^s}$ (see Section 4.4), we focus on this evaluation. We denote the number of pages of a labeling that are non-empty, i.e., that contain at least one label, as ρ. Further, we define

$$\delta_\rho\left(\mathscr{L}^\kappa_{\text{loc}}, \mathscr{L}_{\text{ref}}\right) = \frac{\rho\left(\mathscr{L}^\kappa_{\text{loc}}\right) - \rho\left(\mathscr{L}_{\text{ref}}\right)}{\rho\left(\mathscr{L}_{\text{ref}}\right)} \cdot 100\% \tag{4.16}$$

as the *relative page difference* between a local κ-labeling $\mathscr{L}^\kappa_{\text{loc}}$ and the corresponding reference labeling $\mathscr{L}_{\text{ref}}$. For each considered city, we average the relative page differences over all labelings in one size class. Figure 4.9 shows the resulting mean relative page differences for each city and size class. We observe that, on average, the number ρ of pages is between 31% and 105% higher for the local κ-labelings than for the reference labelings. In other words, a user may have to browse through twice as many pages to access all information contained in a certain area of interest. It can be seen that the relative page difference tends to increase with the amount of information to be presented, i.e., the number of labels in a size class. As more pages are used than necessary, both the quality in terms of the placement of important labels and the spatial distribution of the labels is reduced. We note that setting $\alpha = 0$ would soften the negative effect on Criterion C_{page} and Criterion C_{prio}, since with this setup only the mean effective label weight would be optimized. However, this would cause the spatial distribution of the labels (Criterion $\mathrm{C}_{\text{distr}^s}$) to be

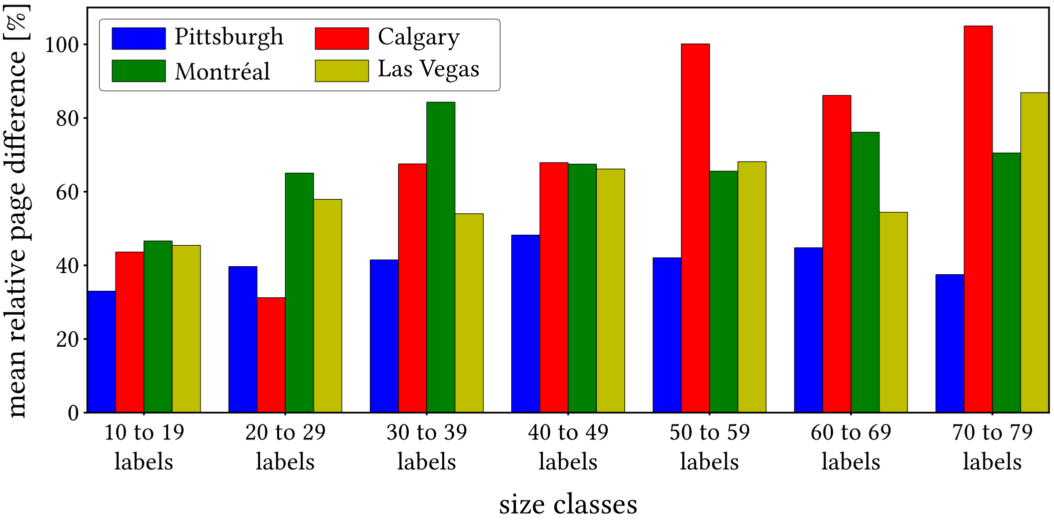


Fig. 4.9: Relative differences δ_ρ of the number of pages between local κ-labelings and reference labelings for different size classes of four instances of different cities.

completely disregarded. We aim at local labelings that sufficiently satisfy all three criteria. Thus, in a next evaluation step, we consider a smaller number i of front pages.

4.5.4 On the Choice of the Number of Front Pages

In the preceding evaluation step, we determined that local κ-labelings do not sufficiently satisfy all optimization criteria. As it can be reasonable to consider only a smaller number i of front pages, we perform further experiments with different choices of the optimization scope λ. We evaluate both global and local i-labelings in the following.

Evaluation of Global i-Labelings For different values of the optimization scope λ, we investigate how many remaining labels are created, i.e., how many labels cannot be placed on front pages. We sample the values for the optimization scope λ in a range of one to ten, i.e., we optimize the first one to ten pages. For each $\lambda \in [1, 10]$ we compute the global i-labeling for each city. Figure 4.10 shows the share of remaining labels in the total number of labels for each sampled value of the optimization scope λ. As λ increases, the share of remaining labels in the total number of labels decreases monotonically. This was to be expected, since an increasing optimization scope corresponds to an increase in the number of front pages on which the labels can be placed. Especially in the range of a small optimization scope (about between one and five), we observe that the number of remaining labels decreases rapidly. Already with an optimization scope of $\lambda = 5$, less than 10% of the labels are remaining for Calgary and Montréal and about 10% are remaining for Pittsburgh. In other words, about 90% of all labels can be placed on front pages

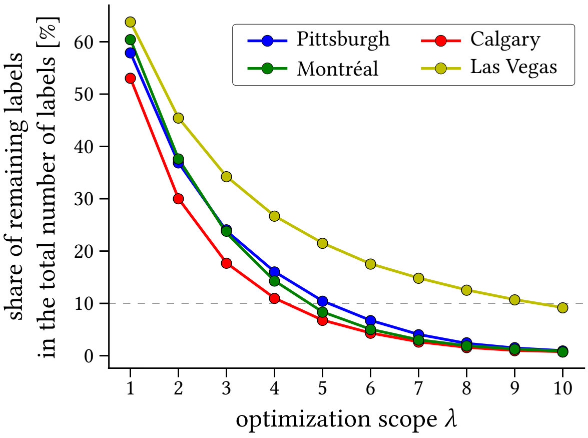


Fig. 4.10: Share of remaining labels in the total number of labels.

for three out of four cities when setting $\lambda = 5$. Assuming that a user may not want to browse through more than the front pages anyway, we note that the second step of the pre-processing phase can be omitted. However, the second step provides the advantage that even the remaining labels are placed on pages that optimize the considered criteria. Reaching the last page, a user can be sure that all information was shown.

Evaluation of Local i-Labelings So far, we showed that an optimization scope of $\lambda = 5$ is reasonable for at least three of the four tested cities. Using global i-labelings obtained with $\lambda = 5$, we now verify that the reduced number of front pages has a positive impact on the quality of local labelings. In the following, we use the notations *global 5-labeling* and *local 5-labeling* and denote a local 5-labeling by $\mathscr{L}^5_{\text{loc}}$. Similarly to the procedure in Section 4.5.3, we extract local 5-labelings from the global 5-labelings in a grid-wise manner. We compare the local 5-labelings with corresponding reference labelings in terms of the relative page difference $\delta_\rho(\mathscr{L}^5_{\text{loc}}, \mathscr{L}_{\text{ref}})$. Figure 4.11 shows the resulting mean relative page differences for each city and size class. We observe that the relative page difference slightly increases with the size of the instances. However, even for the largest considered size class with 70 to 79 labels, the number ρ of pages is at most 25% higher for the local 5-labelings than for the reference labelings. For all other considered classes, the relative page difference is even smaller. Compared to the relative page differences between the local κ-labelings and the reference labelings in Figure 4.9, we obtain a substantial reduction in the number of non-empty pages. While the local κ-labelings consist of up to twice the number of pages as the reference labelings, the number of pages of the local 5-labelings is only a quarter higher at maximum. As already stated in Section 4.4, a small number of pages not only improves the quality in terms of Criterion C_{page}, but also regarding the other two criteria C_{prio} and $\text{C}_{\text{distr}^s}$.

In summary, we argue that by choosing an appropriate optimization scope λ (e.g., $\lambda = 5$) instead of optimizing all κ pages, a majority of all labels can be placed on front pages. This not only reduces the number of pages of the local labelings, but also improves the quality in terms of all optimization criteria.

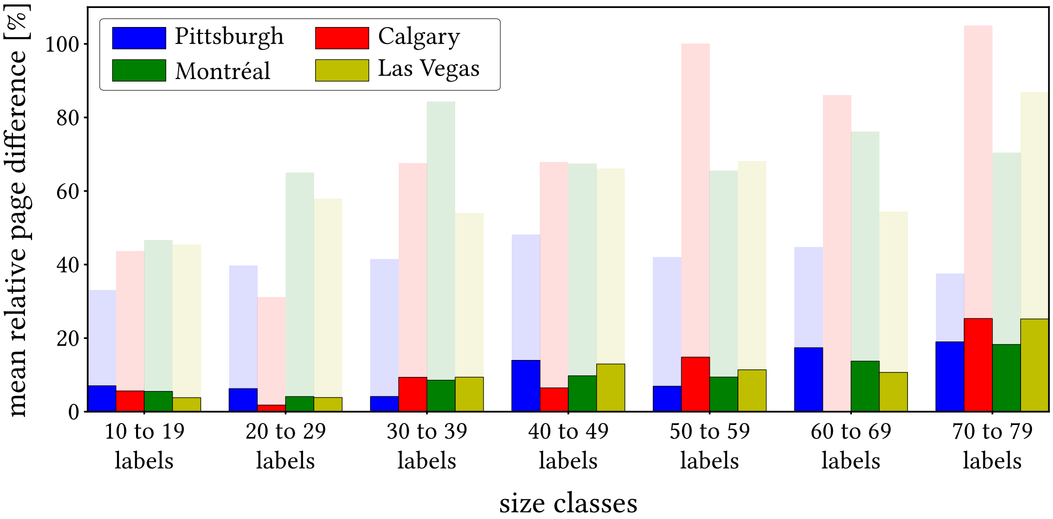


Fig. 4.11: Relative differences δ_ρ of the number of pages between local 5-labelings and reference labelings for different size classes of four instances of different cities. For comparison, the relative differences between local κ-labelings and reference labelings from Figure 4.9 are shown in the background.

4.6 Limitations of Approach

The presented two-phase approach adds a consistency property to the strategy of distributing information across multiple pages introduced in Chapter 3. Although our extended method avoids the flickering of labels as a user pans the map, there are remaining limitations in terms of its application in an interactive scenario.

When querying local labelings in the second phase of our approach, it can happen that the query frame is empty for a given page, but labels are contained on subsequent pages (see bottom row of Figure 4.12). By including such an empty page in the local labeling, a consistent label-page assignment is ensured. However, it can be confusing for users if they first have to navigate through an empty page before information is displayed again on the following page. To avoid this issue, one could simply omit empty pages when extracting local labelings. However, this procedure causes the global consistency to be lost, which becomes apparent with the example in Figure 4.12. Before the user pans the map, the local labeling (marked with Ⓐ) consists of four non-empty pages. However, when the map is panned upwards, the third page becomes empty. If the empty page is omitted and the fourth page is displayed instead, the previous visualization cannot be restored without causing either flickering or changing the page. Hence, to ensure consistency across all pages and over an entire global labeling, empty pages in the local labelings must be accepted, at least if information follows on subsequent pages. We note, however, that such empty pages rarely occurred in our experiments, and when they did, it was late in the sequence of pages with less relevant information.

A related issue can emerge when for two neighboring map frames the query phase results in a different number of pages of the corresponding local labelings. For example, if a user is currently exploring information on the second page and now pans the map to a consecutive frame that

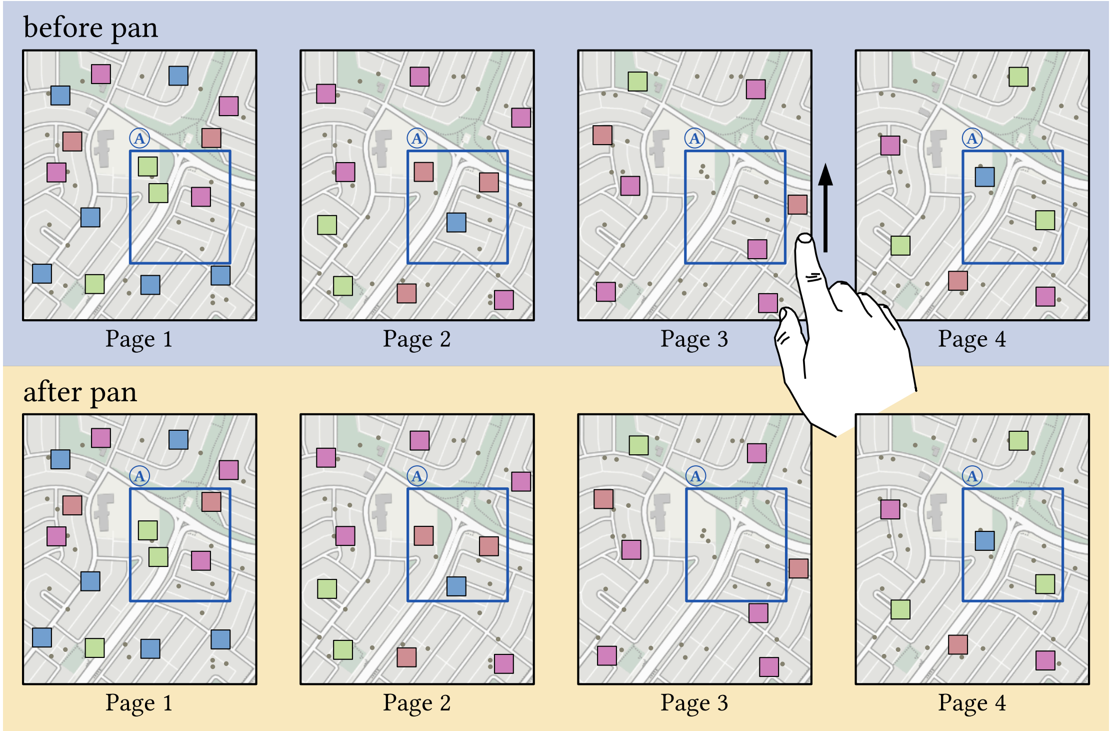


Fig. 4.12: Example of the occurrence of empty intermediate pages in local labelings. If a user pans the map section upwards on the third page (see top row), this page is empty in the subsequent local labeling (see bottom row). *Map attribution: Stamen.*

only contains information on the first page, there are two methodological options; either the interface keeps staying on the global second page, consequently showing an empty map section, or it automatically switches to the first page, as only this page contains information. As with the latter variant the user is no longer able to easily undo actions performed and thus to recreate prior visualizations, we advocate retaining the current global page. We argue that the user can be made aware that information is contained on previous pages by means of appropriate visual indications.

As our approach focuses on maintaining consistency when panning the map rather than zooming, an additional limitation arises when combining our strategy with established zooming functionalities. For a zoom-capable application, we need to compute a global labeling for each specific zoom level within a pre-determined set of discrete zoom levels. Consequently, as labels may be placed on different pages at different zoom levels, zooming the map can yield to flickering effects. We believe, however, that maintaining consistency during panning is a priority, since zoomless maps are primarily intended to reduce the reliance on zooming. Yet, for future work, it could be promising to combine our approach with established methods that ensure consistent labeling during zooming, such as activity-interval frameworks (Been et al., 2006).

4.7 Conclusion

By adapting the general concept of multi-page labeling, i.e., resolving label overlaps by distributing them on multiple pages, we have introduced an advanced strategy that provides a label-page assignment that is consistent during panning operations. Our optimization criteria ensure both quick access to important information and consider legibility by spatially balancing the distribution of labels. By pre-computing global solutions on the level of an entire city, we allow the on-demand query of local labelings. The latter can be flexibly adapted to different screen sizes and formats. Instead of requiring a time-consuming computation for each frame, we only need to compute a global labeling once. In conclusion, we

- have determined a sweet spot that describes a suitable balance between our considered optimization criteria,
- have shown that, in terms of the criteria quality, it is preferable not to optimize all pages in one step,
- have shown that a majority of all labels can be placed on a small set of front pages,
- have shown that when optimizing only a few front pages, the queried labelings have a similar quality compared to corresponding optimal reference labelings.

With the additional consideration of a consistency criterion, we regard the presented strategy as a meaningful extension of the internal multi-page labeling presented in Chapter 3. However, as highlighted in Section 4.6, we still see potential for further progress in implementing our strategy in interactive applications. In particular, we see value in introducing additional visual cues to highlight hidden information and in maintaining consistency across different map scales as part of future development.

5 EXTERNAL LABELING STRATEGIES FOR ZOOMLESS MAPS

The following chapter mainly comprises joint work with Annika Bonerath, Benjamin Niedermann, and Jan-Henrik Haunert (Gedicke et al., 2021a). The methodological and algorithmic contribution was provided in equal parts and in close collaboration by Annika Bonerath, Benjamin Niedermann, and myself (Sven Gedicke). Annika Bonerath and Benjamin Niedermann were primarily responsible for the concepts presented in Sections 5.4 and 5.6, while the concept introduced in Section 5.5 was under my responsibility. Jan-Henrik Haunert provided support and supervision both in the conception of the methodology and in writing the sections.

When labels are placed directly in the map (i.e., internally), it cannot be avoided that they cover the underlying map and thus obscure contextual information to some degree. Therefore, in this chapter, we switch from internal to external label placement and present different concepts for zoomless maps which shift the labels to the boundary of the map (see Figure 5.1). In particular, the presented strategies place labels at the bottom side of the map and visually associate them with their corresponding features via connecting lines called *leaders*. Since the space on mobile screens is limited, showing all labels at the same time is impractical. Therefore, at any time, we label a subset of the features. We offer different interaction techniques to change the current selection of features systematically and, thus, give the user access to all information. We distinguish three strategies, which allow the user either to slide the labels along the bottom side of the map or to browse the labels based on pages or stacks. We present a generic algorithmic framework that provides us with the possibility of expressing the different variants of interaction techniques as optimization problems in a unified way. We propose both exact algorithms and one fast and simple heuristic that solve the optimization problems taking into account different criteria such as the importance of the labels, the total leader length, as well as the distance between leaders. In experiments on real-world data, we evaluate these algorithms and discuss the three variants with respect to their strengths and weaknesses proving the flexibility of the presented algorithmic framework.

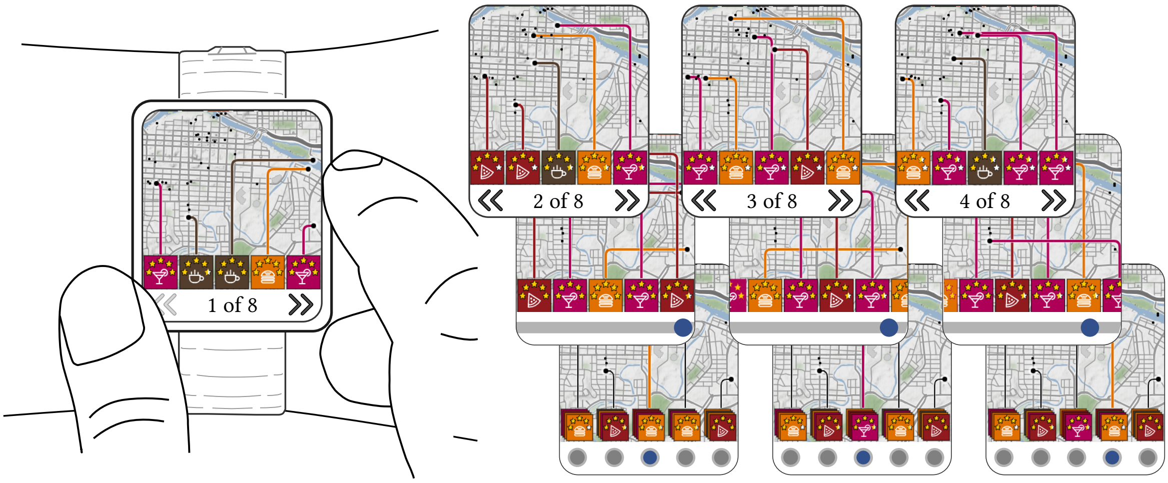


Fig. 5.1: The three labeling concepts that we present in this chapter. The rows of displays show these concepts; *multi-page boundary labeling* topmost, *sliding boundary labeling* centrally, and *stacking boundary labeling* bottommost. *Map attribution: Stamen.*

5.1 Introduction

From a cartographic perspective, the internal placement of labels in the map inevitably leads to obscured map content. Especially when using pictograms or even small images as labels, internal placement can quickly reach its limits in terms of legibility. Contents of the background map, such as streets or buildings, can be covered by labels so that essential contextual information is not visible to the user. Moreover, internal labeling does not provide the possibility of presenting labels in a specific order, e.g., with respect to the relevance of the labels' point features. Addressing these disadvantages and suggesting alternatives to the internal labeling strategies presented in the preceding Chapters 3 and 4, we follow the idea of shifting labels to the bottom side of the map. To maintain an unambiguous assignment of labels to point features, we visually associate the labels with their features by connecting them with thin lines called *leaders*. Placing the labels alongside the boundary of a rectangular map or figure is known in literature as *boundary labeling* which is a special case of *external labeling* (Bekos et al., 2019). For devices with small screens, the number k of labels that can be placed at the bottom of the map is inherently small. In our strategies, we set the number of labels displayed simultaneously to $k = 5$. In contrast to previous work on boundary labeling, which has mostly considered static labelings (Bekos et al., 2019), we create and optimize the labelings for user interaction. We consider different labeling concepts that offer the possibility of interactively exploring the information by browsing through the labels. The rows in Figure 5.1 illustrate these concepts. We note that the control bars are intended to illustrate the type of interaction. In actual applications, they would be omitted and a user can perform the interaction via a corresponding touch interaction (e.g., swiping the labels). Each concept offers advantages that we deem to be beneficial in different use cases. In particular, we introduce the following three concepts.

Multi-page boundary labeling Similar to the internal concepts introduced in Chapter 3 and Chapter 4, *multi-page boundary labeling* distributes the labels on multiple pages. Each page consists of a boundary labeling; see the topmost row in Figure 5.1. The user can navigate through the sequence of pages displaying at most k labels in each step. Hence, after $\lceil \frac{n}{k} \rceil$ steps the user has obtained all pages and labels. We present an algorithm for creating labelings optimizing both the order of the labels with respect to the importance of the features as well as the leader length. The distribution of labels on pages allows a user to gather information with a small number of interactions and to quickly locate individual labels.

Sliding boundary labeling This labeling method arranges the labels in a single row that can be continuously slid along the bottom of the map; see the middle row in Figure 5.1. Only the labels directly below the map are displayed to the user. However, this may easily lead to leaders that intersect or closely run in parallel. Using a graph-based model, we therefore optimize the order of the labels in the sequence taking into account the importance of the labels, the number of crossings, as well as the vertical distance between two leaders. Since sliding boundary labeling supports continuous animation, users can easily trace changes in the labeling.

Stacking boundary labeling In *stacking boundary labeling*, k stacks of labels below the map are created; see the bottommost row in Figure 5.1. We distribute the labels such that each label belongs to exactly one stack and the leader length is minimized. The topmost label of the stack is connected to its feature via a leader. When the user clicks on the stack, the topmost label is pushed underneath the bottommost label. Hence, the second topmost label moves up and is then connected to its feature. Since each stack can be operated independently, stacking boundary labeling provides the possibility of creating customized labelings.

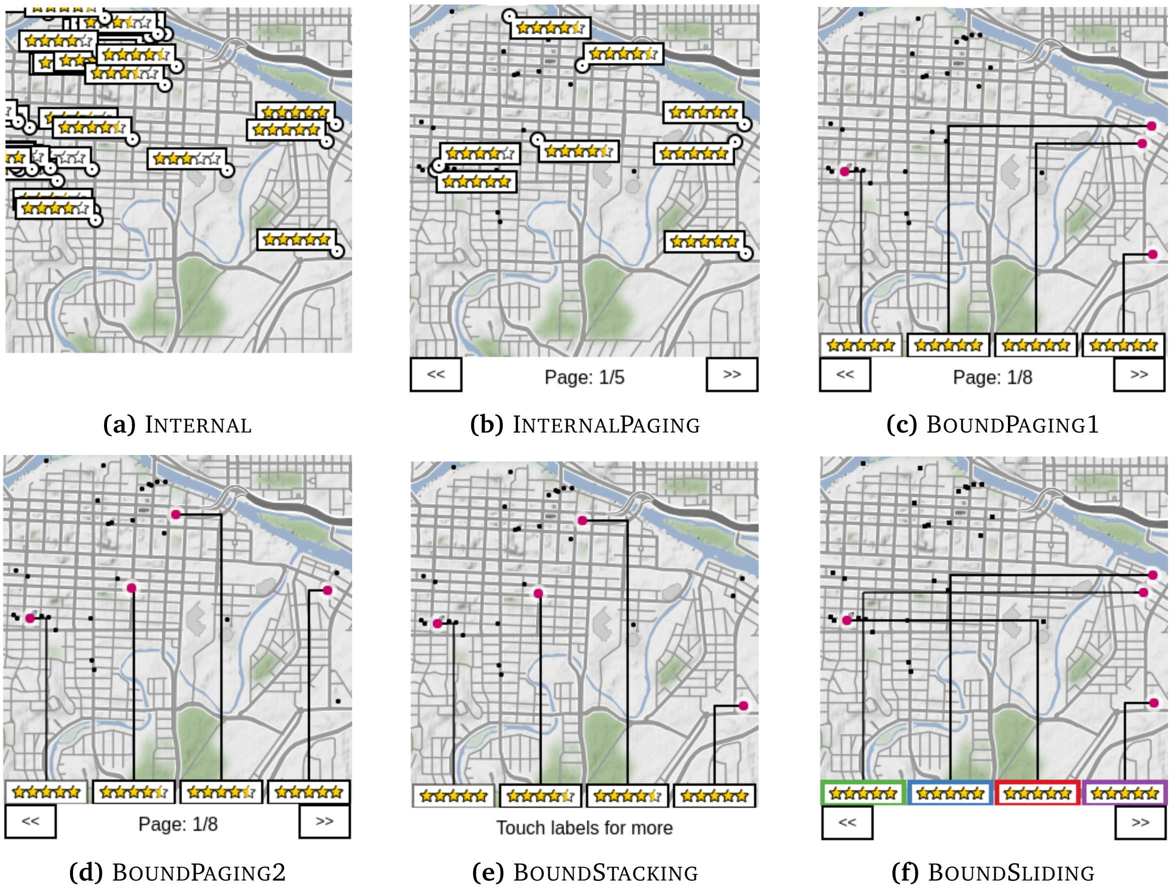
Outline In our three suggested concepts, we consider different design criteria from existing literature of cartography and information visualization. To verify these criteria and to more generally prove the suitability of our labeling strategies, we discussed the concepts with domain experts in an early stage of the development process (see Section 5.2). Conducting such an expert survey allowed us to identify inconsistencies early on and take them into account when fine-tuning our models. From a more technical point of view, we provide a generic framework in Section 5.3 that subsumes all three labeling concepts in one unified model. In contrast to previous work on boundary labeling, which mostly assumes specialized settings (e.g., only L-shaped leaders are allowed, or the algorithms work only for single optimization criteria such as total leader length), our framework allows for an easy adaptation to any boundary labeling style and the consideration of multi-criteria objectives. Further, we can use it to define mathematical optimization problems expressing the three labeling concepts in a consistent way. In Sections 5.4 – 5.6, we introduce these problem formulations and, for each of the three variants, present algorithms that yield optimal solutions. This provides us with the possibility of evaluating the underlying labeling models independently from the applied algorithms. Since for sliding boundary labeling the exact approach is rather slow, we also introduce a simple heuristic that is fast enough for interactive operations. In our evaluation in Section 5.7, we assess our three labeling concepts based on real-world data focusing on quantitative criteria and performance. In particular, we determine sweet spots that describe suitable balances between the considered optimization criteria, discuss our mathematical models, and evaluate the quality of heuristic solutions. We point out that none of the labeling strategies prevail the others, but have their pros and cons depending on the application scenario. We analyze them in greater detail to give the reader a simple way to select an appropriate technique for the specific purposes. Finally, Section 5.8 gives a summary of our key findings.

5.2 Preliminary Expert Survey

Based on preliminary implementations of our three labeling concepts, we conducted an expert survey to receive constructive feedback on our strategies. In particular, we aimed at improving our models and verifying whether the design criteria from literature are valid for the proposed interactions. By discussing our concepts with domain experts in an early stage, we identified inconsistencies before fine-tuning the algorithms. The survey involved nine experts in the fields of information visualization, cartography, and computational geometry. We presented the different concepts as preliminary interfaces in a web browser simulating a smartwatch; see Figure 5.2. We asked the experts to use this simulation envisioning the following scenario.

A user searches for restaurants in the city of Calgary with their mobile device. The restaurants are labeled with user-given star ratings. To maintain orientation in the city, as much map context as possible should be visible and zooming should be avoided.

We presented six different interfaces that address this problem. The first interface, which we call INTERNAL, simply places all labels with the bottom right corner at their features in increasing order regarding their star rating. Thus, when labels overlap, the label with the highest rating is always displayed in the foreground. The second interface, which we call INTERNALPAGING, is an adaptation of internal multi-page labeling presented in Chapter 3. This approach distributes the labels on multiple pages and places them internally. We applied a four-position model for the placement of the labels, i.e., either the upper left, upper right, lower left, or lower right corner of the label coincides with its feature. For the optimization we took into account both the distribution of labels in the sequence of pages and each label's position. We placed high rated labels on front pages and further maximized the minimal number of labels on each page. The third interface, which we call BOUNDPAGING1, and the fourth one, which we call BOUNDPAGING2, use the



(a) INTERNAL **(b)** INTERNALPAGING **(c)** BOUNDPAGING1

(d) BOUNDPAGING2 **(e)** BOUNDSTACKING **(f)** BOUNDSLIDING

Fig. 5.2: The six interfaces presented in the expert survey. *Map attribution: Stamen.*

algorithms for multi-page boundary labeling presented in this chapter. BOUNDPAGING1 purely optimizes the order of the labels with regard to their ratings, while BOUNDPAGING2 optimizes a balance between the labels' ratings and the total leader length. The fifth interface, which we call BOUNDSTACKING, and the sixth one, which we call BOUNDSLIDING, present a stacking and sliding boundary labeling, respectively.

For the simulation, we chose a screen size of 300 × 300 pixels which is a common size for smartwatches (Jackson and Pao, 2019). We extracted data of 32 restaurants from the review portal Yelp[1]. In contrast to the labels presented in Figure 5.1, the labels in the survey only contained the restaurants' ratings but no additional symbols. We note that all methods were implemented without continuous animations so that BOUNDSLIDING could not develop its full potential concerning the traceability of changes. However, to support the user tracing individual labels during interaction we colored the labels' outlines differently.

For each interface, we asked the experts to rate the following statements from 1 (*disagree*) to 4 (*fully agree*).

S1 The interface gives me quick access to well-rated restaurants.

S2 The visual association of labels and point features is sufficiently clear.

S3 The map content necessary for orientation is not too much covered.

Further, we asked to rate two additional statements only with regard to BOUNDPAGING1 and BOUNDPAGING2:

[1] https://yelp.com

S4 The assignment of restaurants to front or back pages sufficiently reflects the labels' ratings.

S5 The shorter guide lines in BoundPaging2 lead to an improved visualization compared to BoundPaging1.

For each statement, we computed the average rating over all experts. For Statements *S1*–*S4*, the results are shown in Table 5.1. With regard to Statement *S5*, we obtained an average rating of 2.56 ± 1.51.

Tab. 5.1: Averaged values and standard deviations of the nine experts' ratings for Statements *S1*–*S4*.

Experiment	*S1*	*S2*	*S3*	*S4*
Internal	2.44 ± 1.24	2.22 ± 1.09	1.44 ± 0.53	-
InternalPaging	3.56 ± 0.88	3.56 ± 0.73	3.11 ± 0.78	-
BoundPaging1	3.00 ± 1.22	2.78 ± 0.83	3.83 ± 0.35	3.67 ± 0.71
BoundPaging2	2.11 ± 0.60	3.11 ± 0.78	3.94 ± 0.17	2.72 ± 0.67
BoundStacking	1.78 ± 0.67	3.11 ± 0.93	3.83 ± 0.35	-
BoundSliding	2.56 ± 1.24	2.78 ± 0.97	3.83 ± 0.35	-

Considering Statement *S1*, we observe that the experts rated InternalPaging best. We justify this by the fact that labels with high ratings are placed on front pages in this strategy. Although this also applies to the other page-based interfaces, i.e., BoundPaging1 and BoundPaging2, more pages need to be explored because the external placement of labels only allows five labels to be displayed at a time. Also with respect to Statement *S2*, InternalPaging is rated best by the experts, while Internal is least preferred here. We argue that the label-feature association is generally more intuitive in internal strategies since each label is placed directly next to the corresponding feature. However, when there are a lot of overlaps among the labels as in Internal, the visual association is no longer unambiguous. When placing labels externally, you have to visually follow the leaders for associating a label to its feature. For all our external labeling concepts, the experts gave rather positive ratings on average. In terms of Statement *S3*, the interfaces with external labeling performed best. This is probably due to the fact that the labels are placed outside the map and the leaders cover the map only slightly. We see this fact as one of the biggest advantages of boundary labeling. Statements *S4* and *S5* only concern BoundPaging1 and BoundPaging2. In the experts' opinion, BoundPaging1 fulfills *S4* substantially better than BoundPaging2. This was to be expected, since in BoundPaging1 the pages are optimized solely with respect to the ratings of the labels, while in BoundPaging2 the leader lengths are also taken into account. However, even for BoundPaging2, we see approval regarding Statement *S4* in the experts' assessments. As a final assessment, we asked the experts to rate on the statement that the shorter leaders in BoundPaging2 lead to an improved visualization compared to BoundPaging1 (Statement *S5*). The experts' opinions varied the most for this statement, and with an average rating of 2.56, there was neither clear agreement nor disagreement. We believe that it was difficult to answer this question clearly on the basis of a single example.

Besides rating the statements, we asked the experts to further comment on the different interfaces. Most comments have been less about the underlying models and more about the visuals. Among other things, a more concise coloring of the labels and thicker leaders were suggested. While we agree with these design suggestions in principle, we emphasize that we primarily intended to gather opinions about the general concepts and the models. Besides design issues, many of the experts highlighted the interface they preferred most and least. However, we observed that these opinions drift far apart. Except for BoundStacking, each of the interfaces was highlighted as the best model by at least one expert. We conclude that each interface, and thus each underlying model, has its advantages and disadvantages, which are perceived differently depending on the user. Generally, the results of the study show that the concepts using boundary labeling were accepted well by the experts. In comparison to INTERNALPAGING, the boundary labeling in general performed better regarding Statement *S3*, while INTERNALPAGING performed better for both Statement *S1* and Statement *S2*. We emphasize that the aim of this survey was not to compare INTERNALPAGING against the concepts implementing boundary labeling. Both internal and external labeling have advantages and disadvantages but non outperforms the other.

In summary, we are applying the following improvements taken from the ratings and comments of the experts.

I1 We visualize the labels as actual stacks in BOUNDSTACKING.

I2 We enrich the information displayed in a label by additionally visualizing each restaurant's category (as in the internal labeling concepts in Chapters 3 and 4).

I3 We strongly focus on a strict order of the labels according to their ratings.

Besides these general adaptations, we further made two adaptations to BOUNDSLIDING. Aiming to improve the label-feature association, we maximize the vertical distances between the horizontal segments of leaders and provide a continuous sliding animation.

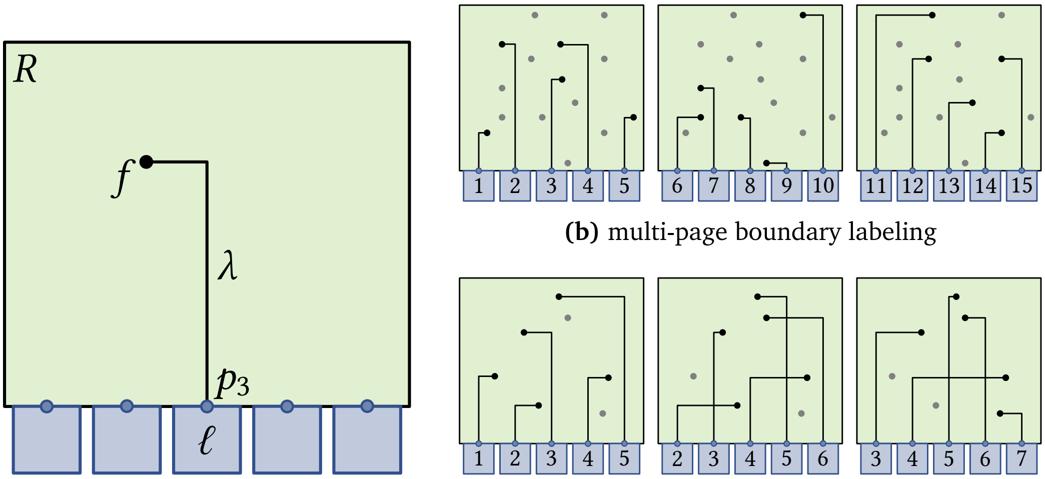
Considering these improvements, we obtain the labeling concepts *multi-page boundary labeling*, *sliding boundary labeling*, and *stacking boundary labeling* described in Section 5.1.

5.3 Mathematical Model

We introduce a unified algorithmic framework, i.e., a mathematical model that serves as a basis for our three labeling concepts. It is assumed that we are given a set F of point features within a pre-defined rectangular region $R \subset \mathbb{R}^2$ that we refer to as *map* in the following; see Figure 5.3a. We introduce a weight function $w\colon F \to [0,1]$ that assigns a weight $w(f)$ to each feature $f \in F$. Each feature in the map is allocated to a rectangular label ℓ that describes the feature. We denote the set of all labels by L and assume that all labels are uniformly sized. As the available space on a small-screen device is limited, we simultaneously place k labels at the bottom side of the map. More precisely, we define k fixed positions at the bottom side of the map, which we call *ports*. We denote the set of ports by $P = \{p_1, \dots, p_k\}$ and assume that they are ordered from left to right. We *attach* a label ℓ to a port p_j by placing the label such that the midpoint of its upper side coincides with p_j. The ports are arranged such that the attached labels do not overlap. Moreover, we visually associate a feature f and the port p of an attached label by connecting f with p with a leader. Although the framework covers any kind of leader, we use so-called *po*-leaders (Bekos et al., 2019) in our experiments. Such leaders consist of two line segments. The first starts at the feature and is parallel (p) to the bottom side of the map. The second ends at the port and is orthogonal (o) to the bottom side of the map.

We model the assignment between features and ports by a *state* $s\colon P \to F$ that maps each port on a feature. We denote the set of all states by S. Further, a state s *contains* a feature $f \in F$ if a port $p \in P$ exists with $s(p) = f$. A state is *crossing-free* if the leaders that connect the ports with the contained features of s do not intersect each other.

In the proposed labeling concepts, we describe a *labeling* as a sequence $\mathcal{L} = (s_1, \dots, s_l)$ of l states. We require that each feature f is contained in at least one state of $\mathcal{L}$. Further, we call a labeling *crossing-free* if all its states are crossing-free. For the computation of labelings, we



(a) general labeling model

(b) multi-page boundary labeling

(c) sliding boundary labeling

Fig. 5.3: In (a), our general labeling model is shown. The leader that connects feature f with port p_3 is denoted as λ. Examples for multi-page and sliding boundary labeling are shown in (b) and (c), respectively.

consider different criteria that a labeling should satisfy. We base these criteria both on existing literature and on the feedback from the expert surveys. Considering accessibility, we aim to distribute important features to states appearing early so that a user has quick access to relevant information. Further, to maintain legible visualizations, we follow the guidlines of Bekos et al. (2019), who provide a list of criteria that apply cartographic principles to external labeling. They identify both the avoidance of crossing leaders and the preference for short leaders as the most prominent criteria. Both reduce visual clutter and facilitate an unambiguous association between a map feature and its label. Thus, in our model, we aim to avoid leader crossings and keep the overall length of the leaders short. To further reduce ambiguities between leaders and thus to improve the label-feature association, we maximize the vertical distances between the horizontal segments of leaders. Not only have we taken this aspect from the feedback of the expert survey, but it is also one of the criteria listed by Bekos et al. (2019). Summarizing, we consider four criteria for our labelings:

$\mathrm{C}_{\mathrm{prio}}$ Important features should be contained in states appearing first.

$\mathrm{C}_{\mathrm{cross}}$ Crossings of leaders should be avoided.

$\mathrm{C}_{\mathrm{len}}$ Leaders should be short.

$\mathrm{C}_{\mathrm{dist}}$ Vertical distances between horizontal segments of different leaders in the same state should be large.

Depending on the labeling concept, we strictly enforce these criteria or express them as cost functions rating each state s_i with $1 \leq i \leq l$ separately. We discuss the cost functions in the following.

For a labeling $\mathscr{L}$, Criterion $\mathrm{C}_{\mathrm{prio}}$ is described as *priority cost* $c_{\mathrm{prio}}(\mathscr{L}) = \sum_{i=1}^{l} c_{\mathrm{prio}}(s_i)$ with

$$c_{\mathrm{prio}}(s_i) = \frac{1}{k \cdot 2^i} \sum_{p \in P} \Big(1 - w\big(s_i(p)\big)\Big).$$

Important features preferably occur in a state with a small index i, while less important features are assigned to states with higher indices.

Criterion $\mathrm{C}_{\text{cross}}$ is described as *crossing cost* $c_{\text{cross}}(\mathcal{L}) = \sum_{i=1}^{l} c_{\text{cross}}(s_i)$ with

$$c_{\text{cross}}(s_i) = \frac{\operatorname{cross}(s_i)}{\binom{k}{2}}.$$

The term $\operatorname{cross}(s_i)$ denotes the number of crossings between leaders in state s_i. We normalize the number of crossings by the maximum number $\binom{k}{2}$ of possible crossings per state.

Criterion C_{dist} is described as *distance cost* $c_{\text{dist}}(\mathcal{L}) = \sum_{i=1}^{l} c_{\text{dist}}(s_i)$ with

$$c_{\text{dist}}(s_i) = \frac{1}{\binom{k}{2}} \sum_{\{f,q\} \in H(s_i)} \frac{1}{|y(f) - y(q)|}.$$

The set $H(s_i)$ contains all pairs $\{f, q\}$ of features whose leaders in s_i have horizontal segments partially running above each other. Hence, states with horizontal leader segments running close above each other have higher distance costs than states with well separated leaders. We normalize the cost by dividing it with the maximal size $\binom{k}{2}$ of $H(s_i)$.

Criterion C_{len} is described as *length cost* $c_{\text{len}}(\mathcal{L}) = \sum_{i=1}^{l} c_{\text{len}}(s_i)$ with

$$c_{\text{len}}(s_i) = \frac{1}{k \cdot 2^i} \sum_{p \in P} \frac{\operatorname{length}\big(p, s_i(p)\big)}{\text{screen width} + \text{screen height}}.$$

The term $\operatorname{length}\big(p, s_i(p)\big)$ denotes the length of the leader that connects the port p with the feature $s_i(p)$. We normalize the length of the leaders by the screen width and height. The relevance of a state decreases exponentially with increasing index i, so that short leader lengths are given more priority on front states than on back states.

For each of the three labeling concepts, we use c_{prio}, c_{cross}, c_{dist}, and c_{len} to compose a bi-criteria cost function c that balances two of those cost functions by means of a factor $\alpha \in [0, 1]$. We call $c(\mathcal{L})$ the *total cost* of a labeling $\mathcal{L}$. We restricted ourselves to bi-criteria cost functions to be capable of controlling the effects of the optimization on the resulting labeling. More precisely, we use the two criteria that affect each model most as *soft* constraints and express the others as *hard* constraints if this is reasonable. We argue for the actual choice in the following sections.

5.4 Multi-Page Boundary Labeling

In multi-page boundary labeling, the labels are distributed on multiple pages such that each page labels a distinct set of features (see Figure 5.3b). Each page corresponds to a state in a labeling $\mathcal{L} = (s_1, \ldots, s_l)$ with $l = \lceil n/k \rceil$. In the case that $n/k \notin \mathbb{N}$ we introduce dummy features that do not influence the overall solution. We say $\mathcal{L}$ is a multi-page boundary labeling if each feature is contained in exactly one state and it is crossing-free (Criterion $\mathrm{C}_{\text{cross}}$). Among all multi-page boundary labelings, we search for one that optimizes the priority cost (Criterion C_{prio}) and the length cost (Criterion C_{len}). We do not consider the vertical distances between leaders (Criterion C_{dist}) as we deem small distances mostly relevant in combination with crossings. Based on c_{prio} and c_{len}, we define the cost of a multi-page boundary labeling $\mathcal{L}$ as

$$c_{\text{MPL}}(\mathcal{L}, \alpha) = \alpha \cdot c_{\text{len}}(\mathcal{L}) + (1 - \alpha) \cdot c_{\text{prio}}(\mathcal{L}),$$

where $\alpha \in [0, 1]$ balances the length and priority cost of $\mathcal{L}$. For $\alpha = 0$ and $\alpha = 1$, only the priority cost and the length cost are considered, respectively. For a single state s_i of $\mathcal{L}$ we further define $c_{\text{MPL}}(s_i, \alpha) = \alpha \cdot c_{\text{len}}(s_i) + (1 - \alpha) \cdot c_{\text{prio}}(s_i)$. Due to the linearity we obtain $c_{\text{MPL}}(\mathcal{L}, \alpha) = \sum_{i=1}^{l} c_{\text{MPL}}(s_i, \alpha)$.

5.4.1 Algorithm

We show that the problem of finding a multi-page boundary labeling can be solved optimally by transforming it into finding a perfect matching in a bipartite graph. We observe that – for a set of k features and k ports – the problem reduces to computing a one-sided boundary labeling with minimum total leader length for a single page. Such a labeling can be computed in $O(k \log k)$ time (Benkert et al., 2009), no matter whether the length of the leader is normalized or not. We reformulate this as follows.

Observation 1. *Let $f_1, \dots, f_k$ be features and let $p_1, \dots, p_k$ be ports. Further, let s be a not necessarily crossing-free state with minimal costs $c_{\mathrm{MPL}}(s, \alpha)$ over all states of $f_1, \dots, f_k$ and $p_1, \dots, p_k$. There exists a crossing-free state s' with $c_{\mathrm{MPL}}(s, \alpha) = c_{\mathrm{MPL}}(s', \alpha)$.*

Observation 1 holds because the number of features corresponds to the number of ports. Hence, only the length cost is considered, but not the priority cost. Using Observation 1, we divide the computation into two steps. First, we find a cost-minimal possibly non-crossing-free solution, and in the second step we resolve the crossings on each page.

The first step corresponds to finding a perfect matching in a bipartite graph; see Figure 5.4. In general, let $G = (V, E)$ be a weighted bipartite graph, i.e., V is partitioned into two disjoint sets V_1 and V_2 and each edge $e \in E$ connects a vertex $v_1 \in V_1$ with a vertex $v_2 \in V_2$. A *perfect matching* in G is a subset of E such that each vertex is incident to exactly one edge of the subset. A *minimum-weight* perfect matching minimizes the sum of the weights over all selected edges.

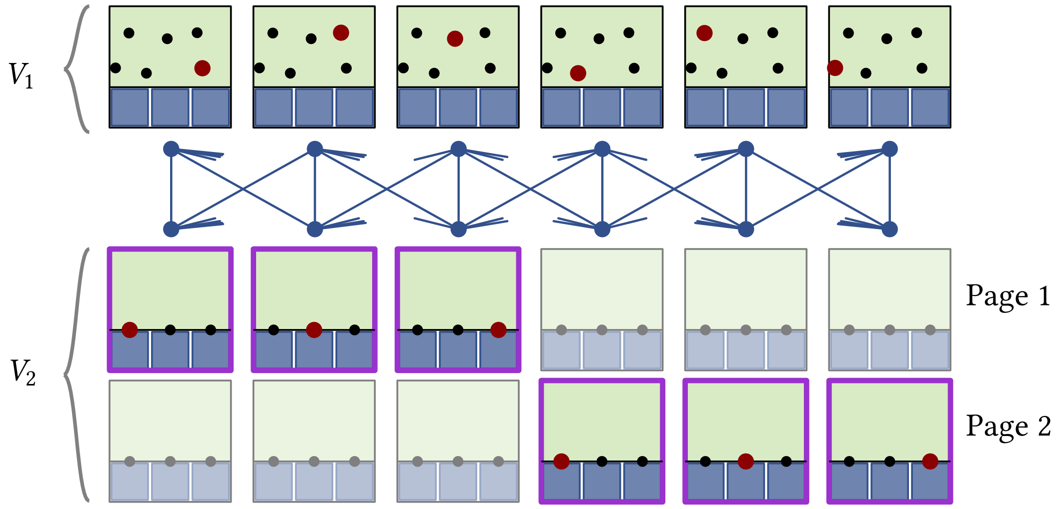


Fig. 5.4: Multi-page boundary labeling visualized as a perfect matching. The vertices contained in V_1 correspond to the point features (marked in red). The vertices in V_2 correspond to the pairs of a port and a state (shown in purple).

For multi-page boundary labeling, the vertex set V_1 consists of the vertices v_f where f is a feature and the set V_2 consists of the vertices $v_{p,s}$ where $\{p, s\}$ is a pair of a port p and a state s. Further, G is a complete bipartite graph. For each edge e that connects the vertices v_f and $v_{p,s}$, we set the weight equal to

$$\frac{1}{2^i} \cdot \big(1 - w(f)\big) + \alpha \cdot \frac{1}{2^i} \cdot \frac{\mathrm{length}\big(p, s_i(p)\big)}{\text{screen width} + \text{screen height}} .$$

To solve multi-page boundary labeling allowing crossings, we compute a perfect minimum-weight matching in G, which can be done in $O(n^3)$ time (Kuhn, 1955; Munkres, 1957). In our implementation, we used a linear programming approach which is an established strategy (Grötschel and Holland, 1985).

In the second step, we apply the algorithmic approach presented by Benkert et al. (2009) for each state to find a crossing-free labeling with minimal leader length. We also use this approach for the stacking boundary labeling and discuss it in Section 5.6 in greater detail.

5.5 Sliding Boundary Labeling

In sliding boundary labeling, the labels can be slid along the bottom side of the map. From a technical point of view, we assume that the labeling is a sequence $(s_1, \ldots, s_l)$ of states; see Figure 5.3c. The interaction of sliding the labels then corresponds to an animated transition between consecutive states. To guarantee a continuous flow of labels, we require that from one state s_i to the next state s_{i+1} the labels are shifted to the left by one port and a new label appears at the rightmost port. More formally, we say that the transition from s_i to s_{i+1} is *valid*, if $s_i(p_j) = s_{i+1}(p_{j-1})$ for all $1 < j \leq k$ and $s_i(p_1) \neq s_{i+1}(p_k)$.

Since, except for the last $k-1$ labels, each label is shifted once from the rightmost to the leftmost port, we do not consider the leader length (Criterion $\mathrm{C_{len}}$). Instead, we aim for a labeling that minimizes both the crossing cost (Criterion $\mathrm{C_{cross}}$) and the distance cost (Criterion $\mathrm{C_{dist}}$). For $\alpha \in [0,1]$, we define the cost of a sliding boundary labeling $\mathcal{L}$ as

$$c_{\mathrm{Slid}}(\mathcal{L}) = \alpha \cdot c_{\mathrm{cross}}(\mathcal{L}) + (1-\alpha) \cdot c_{\mathrm{dist}}(\mathcal{L})\,.$$

In the following, we present an exact approach that yields an optimal solution as well as a fast and simple heuristic.

5.5.1 Exact Approach

For obtaining an optimal sliding boundary labeling, we use a graph-based method that models our optimization problem as a constrained *orienteering problem* (Vansteenwegen et al., 2011). Given a graph in which each vertex has a score and each edge a length, and additionally given a source and a target vertex, the orienteering problem asks for a path from the source to the target such that the total score along the path is maximized and a given length is not exceeded. We observe that with some adaptations on the original problem definition, this path represents a labeling. We define an integer linear program (ILP) that expresses the problem of finding an optimal labeling as a linear objective function subject to a set of linear constraints. In particular, our formulation models the problem as a *flow network*. The idea is to send a unit from a source to a target vertex via the graph. By controlling the flow of the unit with appropriate constraints, we ensure the connectivity of the desired path. We now present the details of our approach. Let $G = (S \cup \{\sigma, \tau\}, E)$ be a directed graph containing a vertex for each state $s \in S$. Further, G contains a source vertex σ and a target vertex τ. Each state $s \in S$ is connected with the source by an edge (σ, s) and with the target by an edge (s, τ). Moreover, a pair of states $s, t \in S$ is connected by a directed edge $(s, t) \in E$ if and only if the transition from state s to state t is valid. We aim for a path from σ to τ representing a labeling; see Figure 5.5.

For each directed edge $(s,t) \in E$, we introduce a binary variable

$$x_{s,t} = \begin{cases} 1 & \text{if and only if the transition of state } s \text{ to state } t \text{ is used,} \\ 0 & \text{otherwise .} \end{cases} \tag{5.1}$$

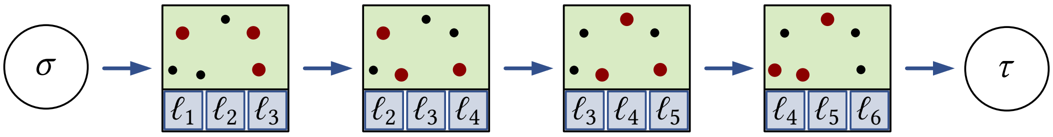


Fig. 5.5: Illustration of an exemplary path from σ to τ in the graph G.

Additionally, we introduce a variable $y_s \in \{0,\dots,n\}$ for each state s with n being the number of states. We use y_s to ensure that the sub-graph defined by the edges $e \in E$ with $x_e = 1$ is actually one connected component forming a path from σ to τ. We denote the set of all states that contain a feature f by S_f. Further, we define $E_f \subseteq E$ to be the set that contains any edge $e = (s,t)$ whose source s does not contain f, but whose target t contains f, i.e., $E_f = E \cap \big((S \setminus S_f) \times S_f\big)$. To model a path in G that represents a labeling, we require

$$\sum_{t \in S} x_{\sigma,t} = 1\,, \tag{5.2}$$

$$\sum_{s \in S} x_{s,\tau} = 1\,. \tag{5.3}$$

Constraint (5.2) and Constraint (5.3) ensure that the path begins at the source σ and ends at the target τ by enforcing that exactly one transition from σ and one transition to τ is used. Hence, in terms of the flow network, exactly one unit leaves the source and one unit reaches the target.

Further, we ensure that each state can be the origin of at most one transition, i.e., at most one unit leaves each state. To that end, we require

$$\sum_{\substack{t \in S\\ t \neq s}} x_{s,t} \leq 1 \;\forall s \in S\,. \tag{5.4}$$

To preserve the flow, the following constraint implies that if a unit reaches one state, it also leaves this state.

$$\sum_{\substack{s \in S\\ s \neq t}} x_{s,t} = \sum_{\substack{u \in S\\ u \neq t}} x_{t,u} \;\forall t \in S \tag{5.5}$$

As we aim for one connected path through G, we further require

$$y_s + 1 \leq y_t + (n-1)\cdot\big(1 - x_{s,t}\big) \;\forall s,t \in S, s \neq t\,, \tag{5.6}$$

and

$$\sum_{(s,t) \in E_f} x_{s,t} = 1 \;\forall f \in F\,. \tag{5.7}$$

Constraint (5.6) guarantees that the states are numbered in ascending order along the path. Hence, no cycles can be created. To ensure that the path corresponds to a labeling, Constraint (5.7) guarantees that each feature $f \in F$ is labeled at least in one state s and that it only appears in consecutive states. Subject to Constraints (5.2) – (5.7), we minimize

$$\Big(\alpha \cdot \sum_{s \in S} c_{\text{cross}}(s) + (1-\alpha) \sum_{s \in S} c_{\text{dist}}(s)\Big) \cdot \sum_{\substack{t \in S\\ t \neq s}} x_{s,t}\,. \tag{5.8}$$

Let $Q = \{e \in E \mid x_e = 1\}$ be the path from σ to τ. Following Q, we obtain a sequence $\mathscr{L} = (s_1,\dots,s_l)$ of states which forms a sliding boundary labeling such that $c_{\text{Slid}}(\mathscr{L})$ is minimal.

In our experiments, we use a state-of-the-art ILP solver to gain mathematically optimal solutions to the ILP formulation.

5.5.2 Heuristic Approach

While solving the ILP guarantees an optimal solution, the computation requires exponential time in the worst case. Thus, we additionally introduce a simple and fast greedy heuristic for sliding boundary labeling. It is based on the local search strategy *hill climbing*; see Section 2.5.2.

To get an initial solution, we apply a simple first-fit strategy. We create an arbitrary order $f_1, \ldots, f_n$ of the features and define a state s_i as $s_i(p_1) = f_i, \ldots, s_i(p_k) = f_{i+k-1}$ for $1 \leq i \leq n-k$. Hence, sliding the labels from right to left, the features are labeled in that order. We iteratively improve that labeling by performing local changes. We obtain a neighboring solution by *swapping* the assignments $s_i(p) = f$ and $s_j(p') = q$ of two randomly chosen features f and q, i.e., after this *swap* we have $s_i(p) = q$ and $s_j(p') = f$. For an illustration see Figure 5.6. If such a swap improves the value of the objective function, it is applied and a new solution is obtained. The algorithm continues with performing such swaps until a pre-defined number of iterations is reached or no more neighboring solutions of higher quality exist.

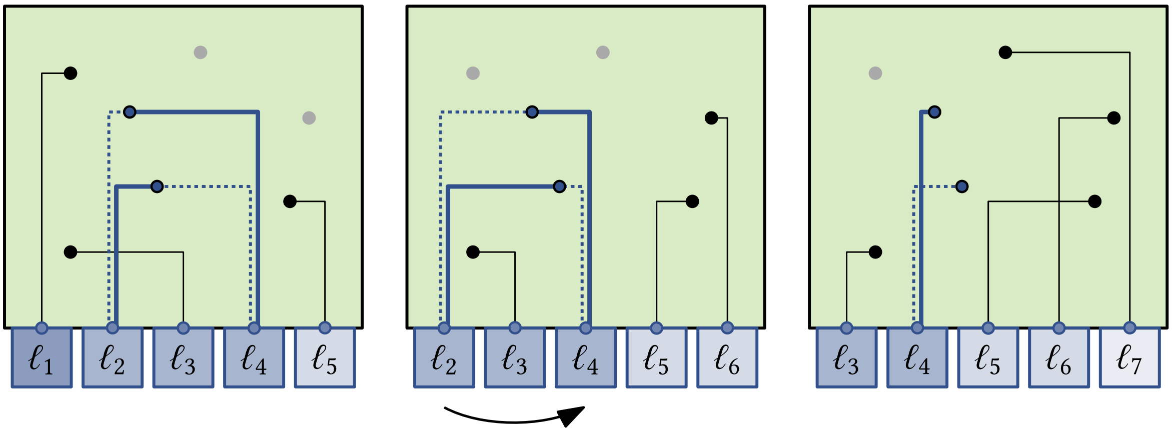


Fig. 5.6: Illustration of a swap. The saturation of the labels' colors represent the weight of the features. The leaders affected by the swap are shown in blue. The dashed blue leaders correspond to the solution before and the solid blue leaders belong to the solution after the swap.

5.6 Stacking Boundary Labeling

Stacking boundary labeling is characterized by its interaction technique. Instead of having one pre-defined sequence of states, the features are partitioned into k groups; each assigned to one stack. Hence, we have an own sequence of states for each single stack, where each state consists of a single feature. By clicking on one label, its port is connected to the next feature in the stack. We call $t_p : i \rightarrow F$ with $1 \leq i \leq l$ the *stack* of port p. A feature f is *contained* in a stack t at position i if $t(i) = f$. A stacking boundary labeling consists of the set of stacks $\mathcal{T} = \{t_{p_1}, \ldots, t_{p_k}\}$. We require that the features are evenly distributed over all stacks[2], i.e., $l = n/k$. Further, we require that each feature is contained in exactly one stack at one position, and that the labeling is crossing-free (Criterion $\mathrm{C_{cross}}$). Among all stacking boundary labelings, we search for the one that optimizes the weights of the features on the front stack positions (Criterion $\mathrm{C_{prio}}$) and the length of the leaders (Criterion $\mathrm{C_{len}}$). As there exists a state for every two features of different stacks in which they are labeled simultaneously, we do not consider the vertical distance (Criterion $\mathrm{C_{dist}}$). The user can improve the visualization by clicking on the labels of features that are vertically close together.

[2] We assume $n/k \in \mathbb{N}$. If this is not the case, we insert dummy feature points.

As the primary objective for partitioning the features into k groups, we optimize the leader length globally for $\mathcal{T}$ as follows.

$$\min\ c(\mathcal{T}) = \min \sum_{p\in P} \sum_{i=1}^{l} \operatorname{length}\big(p, t_p(i)\big)$$

As the secondary objective for each sequence, we optimize Criterion C_{prio} by ordering the features of each stack by their weight.

5.6.1 Algorithm

We compute optimal stacking boundary labelings by transforming the corresponding optimization problem into a static boundary labeling problem. We introduce each port of our stacking boundary labeling problem l times; see the left part of Figure 5.7. Further, we adapt the definition of a crossing; we say that two leaders that start at the same port cannot cross. A solution of the first step of the stacking boundary labeling problem is a crossing-free assignment with minimal leader length between the labels' ports and features. The approach by Benkert et al. (2009) solves this problem in $O(n \log n)$ time. In this approach, the map is partitioned into strips induced by vertical lines through each label's port and feature; see the right part of Figure 5.7. The strips are categorized into leftwards and rightwards directed strips by a comparison of the number of labels and the number of features to the right and left. Each consecutive set of leftwards directed strips is solved with a sweep-line approach from right to left. Each time the sweep line passes a feature, it is inserted into a list W which is ordered by the features' y-coordinates. Every time the sweep line intersects a port, this port is assigned to the lowest feature contained in W. The rightwards directed strips are solved with a symmetric approach. After performing this approach, we sort each stack t_p by the weight of the features. We note that a stacking boundary labeling corresponds to a multi-page boundary labeling by assigning the i-th label of each stack to the i-th page, i.e., $s_i(p) = t_p(i)$.

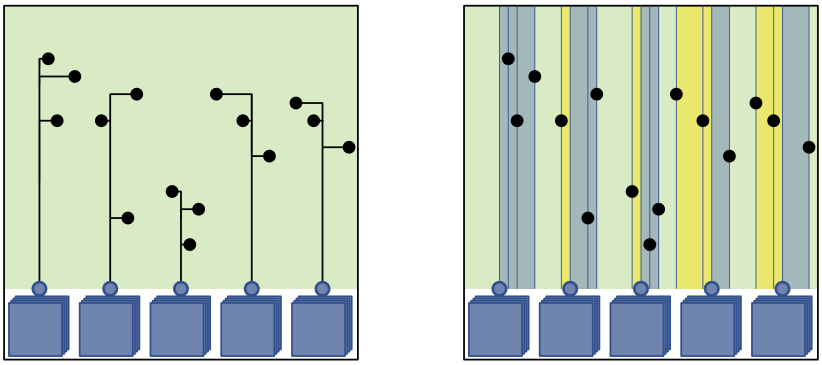


Fig. 5.7: Methodology for solving stacking boundary labeling. On the left, all leaders of the stacking boundary labeling are shown. On the right, the map is partitioned into strips (yellow strips: rightwards, blue strips: leftwards) following the approach of Benkert et al. (2009).

5.7 Experiments

Following, we present experiments evaluating our three labeling concepts on real-world data focusing on quantitative criteria and performance.

We assume the use case of querying and visualizing restaurants with a mobile map application on a smartwatch. Based on common resolutions on such devices, we set the screen size to 300 × 300 pixels (Jackson and Pao, 2019). We choose a map scale of 1:70,000 which comprises the extent of about a small village or suburb. We deem this extent to be a compromise between a rough overview of a certain part of a city and the possibility of identifying the location of a restaurant. Using a spherical mercator projection (EPSG:3857), the maps comprise different area sizes depending on the latitude. For examples see Figure 5.1 or Figure 5.2. We note that other scales, e.g., representing even smaller areas of interest such as city blocks, could also be used.

We use data of restaurants from five metropolitan areas in the USA and Canada provided by the review portal Yelp. More precisely, we consider data in *Las Vegas (USA)*, *Pittsburgh (USA)*, *Calgary (Canada)*, and *Montréal (Canada)* and *Toronto (Canada)*. Each restaurant is given with its location and a user-given rating. The rating varies between one and five stars; half-stars are also allowed. We normalize the ratings obtaining weights in the interval $[0,1]$. We create 100 instances of different map sections each having 30 randomly sampled features. In each of our concepts, we show five labels ($k = 5$) at a time, each with a size of 60×60 pixels. For an application-oriented visualization, each label displays the restaurant's rating as star symbols and indicates its category in the form of a pictogram and through a specific color scheme. We note that this has no impact on the algorithms nor the evaluation.

We evaluate the model of each labeling concept individually. In particular, we aim to find a suitable choice of α describing the sweet spot between the two respective criteria that are optimized. For the different criteria, we also show that it is worth optimizing them by comparing the results of the algorithms with respective worst and best cases. For a cost function c and two labelings $\mathscr{L}$ and $\mathscr{L}'$ we define their *relative cost* as

$$\delta(c,\mathscr{L},\mathscr{L}') = \frac{c(\mathscr{L}) - c(\mathscr{L}')}{c(\mathscr{L}')} \cdot 100\%.$$

We use the relative cost to compare a labeling $\mathscr{L}$ with another labeling $\mathscr{L}'$ that is optimal with respect to one or multiple of the considered criteria. We conclude our experimental evaluation by assessing the running times of our algorithms and comparing each concept's strengths and weaknesses.

Our approaches are implemented in Java, and the ILP and LP formulations were solved using the commercial Gurobi[3] Optimizer version 8.1.0. We ran the experiments on an Intel(R) Xeon(R) W-2125 CPU clocked at 4.00GHz with 128 GiB RAM. Considering server-client communication as the use case for our algorithms, we performed the computations on a server system.

5.7.1 Multi-Page Boundary Labeling

For multi-page boundary labeling, we focus on assessing the priority cost (Criterion C_{prio}), the length cost (Criterion C_{len}), and the distance cost (Criterion C_{dist}) for different values of α. We note that only the priority cost and length cost are considered in the objective where α balances those. For the experiments, we sampled α in the range $[0,1]$ with a step width of 0.025. We denote the set of the resulting 41 values by $\mathscr{A}$ and a multi-page boundary labeling for α by $\mathscr{L}_\alpha$.

[3] https://gurobi.com

Optimized Criteria C_{prio} and C_{len}

We take into account the relative priority cost $\delta_{prio}(\mathscr{L}_\alpha) := \delta(c_{prio}, \mathscr{L}_\alpha, \mathscr{L}_0)$ of a labeling $\mathscr{L}_\alpha$ and the labeling $\mathscr{L}_0$ with minimal priority cost; see Figure 5.8. Similarly, we define the relative length cost $\delta_{len}(\mathscr{L}_\alpha) := \delta(c_{len}, \mathscr{L}_\alpha, \mathscr{L}_1)$ for a labeling $\mathscr{L}_\alpha$ and the labeling $\mathscr{L}_1$ being the solution with minimal length cost c_{len}. Considering Figure 5.8, we observe that with an increasing value for α the relative length cost δ_{len} monotonically decreases while the relative priority cost δ_{prio} monotonically increases. We can see that the difference between the relative length cost for $\alpha = 0$ and the next larger step $\alpha = 0.025$ is remarkably large. At the same time, the relative priority cost is still optimal for $\alpha = 0.025$. Hence, we are able to optimize c_{len} to some extent without decreasing the quality in terms of c_{prio}.

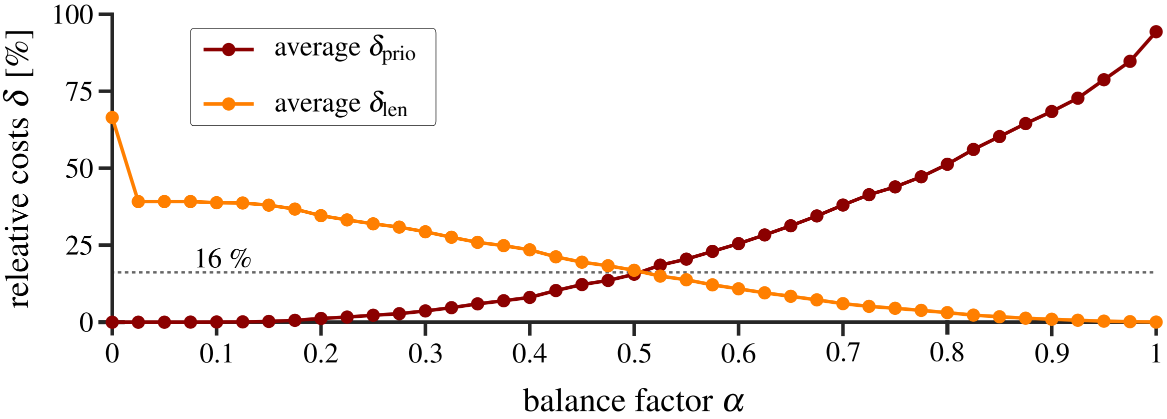


Fig. 5.8: Average relative priority cost δ_{prio} and average relative length cost δ_{len} for each $\alpha \in \mathscr{A}$ of multi-page boundary labeling.

Further, we argue that a suitable sweet spot between the two criteria is where the plots of the relative costs cross. To determine the corresponding value for α, we interpret the plots as line objects consisting of multiple segments and compute the geometric point of intersection. Following this procedure, we obtain a suitable balance at $\alpha = 0.5$. Using this sweet spot, both the priority cost and the length cost are only 16% higher compared to the respective optimal solutions. Hence, we deem $\alpha = 0.5$ to be a suitable compromise between optimizing the priority cost and length cost.

Non-Optimized Criterion C_{dist}

In multi-page boundary labeling, we do not optimize the distance cost c_{dist} . Hence, horizontal segments of leaders might run very close in parallel, making it difficult to visually distinguish them. In the following, we aim to investigate whether this is actually a concern that is present in our solutions. We analyze all pairs H of leaders which are labeled on the same page and for which the horizontal parts of the leaders run above each other. At first, we evaluate the size of H in relation to the overall number of leader pairs on the same page. On average, 20.7% of the leader pairs on the same page horizontally run above each other. We note that this ratio is similar for different values of α. Moreover, we analyze the ratio of leader pairs in H with a vertical distance smaller than five pixels with respect to the size of H. This ratio varies between 11.0% for $\alpha = 0$, and 38.4% for $\alpha = 1$. This means that at most 38.4% of the 20.7% of leader pairs which have horizontal segments lying above each other have a vertical distance under five pixels. Since multi-page boundary labeling provides static pages, we conclude from these results that leaders are sufficiently distinguishable.

5.7.2 Sliding Boundary Labeling

In our preliminary expert survey, experts generally recommended to strictly label the features in descending weight order. Hence, we consider a special case of sliding boundary labeling in which we enforce Criterion C_{prio} as hard constraint. While the overall weight order is preserved, we optimize the order within groups of features having the same weight regarding the cost function $c_{\text{Slid}}(\mathscr{L})$. We modify our exact approach such that the graph $G = (S', E)$ only contains states satisfying Criterion C_{prio}. Hence, $S' \subseteq S$ is the set of states in which the features are assigned to the ports from left to right in descending weight order.

We also adapt the heuristic as follows. When constructing an initial solution, we sort the given features in descending weight order. Further, when obtaining neighboring solutions, we only swap the assignments $s_i(p) = f$ and $s_j(p) = q$ of two randomly chosen features f and q if $w(f) = w(q)$. We use 5000 iterations as termination criterion, which has proven to be a suitable choice in preliminary experiments. For each instance and each considered weighting α, we run our heuristic five times and build the arithmetic mean of the results.

Evaluation of Model

We first evaluate our model with respect to both optimized criteria, i.e., the crossing cost (Criterion C_{cross}) and the distance cost (Criterion C_{dist}). To that end, we run the exact approach for $\alpha \in \{0, 0.5, 1\}$. We note that for $\alpha = 0$ and $\alpha = 1$ we only optimize C_{dist} and C_{cross}, respectively. We use $\alpha = 0.5$ as an additional intermediate value that balances both criteria. When running our exact approach with a number of $k = 5$ labels per state, the storage capacity of the used system is exceeded. By reducing the number of labels to $k = 4$, we are able to successively run the exact approach on all 100 instances. We argue that limiting the number of labels to four still models a realistic setting in which a high number of leader crossings and small vertical distances may occur in the worst case.

We aim to investigate the potential of optimization with respect to both considered criteria. To that end, we additionally determine the maximum possible costs c_{cross} and c_{dist} of both criteria. More specifically, instead of minimizing the objective function in our ILP formulation, we maximize it for $\alpha = 0$ and $\alpha = 1$, respectively. We denote the results by $\text{MaxC}_{\text{CROSS}}$ for $\alpha = 1$ and by $\text{MaxC}_{\text{DIST}}$ for $\alpha = 0$. Further, we created for each instance a labeling in which features of the same weight group are ordered randomly; we refer to them as RANDOM. The results are shown in Figure 5.9. The crossing costs are displayed without normalization so that they show the absolute number of crossings.

We observe that, on average, the number of leader crossings has a minimum of 25.8 ($\alpha = 1$) and a maximum of 73.5 ($\text{MaxC}_{\text{CROSS}}$). Thus, by choosing $\alpha = 1$, the number of leader crossings can be reduced by $\delta(c_{\text{cross}}, \mathscr{L}_1, \mathscr{L}_{\text{MaxC}_{\text{CROSS}}}) = 64.8\%$ in maximum. When considering the trade-off with $\alpha = 0.5$, we obtain near-optimal results with respect to the number of crossings. Compared to $\text{MaxC}_{\text{CROSS}}$, the number of leader crossings can be reduced by $\delta(c_{\text{cross}}, \mathscr{L}_{0.5}, \mathscr{L}_{\text{MaxC}_{\text{CROSS}}}) = 62.9\%$ on average. Compared to a randomly generated labeling (RANDOM), the number of leader crossings is still reduced by $\delta(c_{\text{cross}}, \mathscr{L}_{0.5}, \mathscr{L}_{\text{RANDOM}}) = 44.3\%$ on average.

Further concerning the distance cost of the optimal results for $\alpha \in \{0, 0.5\}$, $\text{MaxC}_{\text{DIST}}$, and RANDOM, we make a similar observation (see the right plot in Figure 5.9). The average distance cost is 4.0 in maximum and 0.5 in minimum, which means that the distance cost can be reduced by a maximum of $\delta(c_{\text{dist}}, \mathscr{L}_0, \mathscr{L}_{\text{MaxC}_{\text{DIST}}}) = 87.5\%$. For $\alpha = 0.5$, the distance cost is still reduced by $\delta(c_{\text{dist}}, \mathscr{L}_{0.5}, \mathscr{L}_{\text{MaxC}_{\text{DIST}}}) = 72.8\%$ on average. Compared to a randomly generated labeling, the distance cost is reduced by $\delta(c_{\text{dist}}, \mathscr{L}_{0.5}, \mathscr{L}_{\text{RANDOM}}) = 33.5\%$ on average with $\alpha = 0.5$.

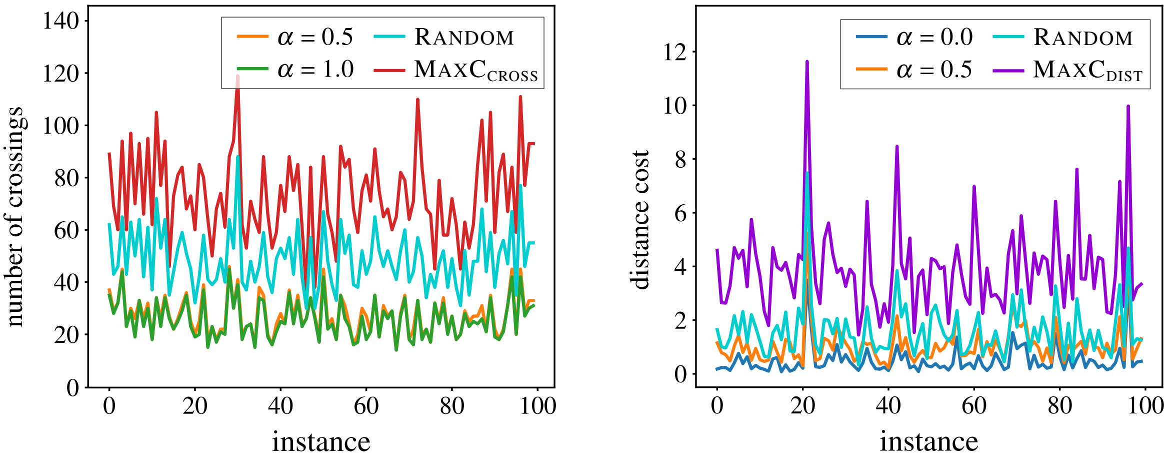


Fig. 5.9: Comparison of optimal results of different models of sliding boundary labeling for 100 instances. The left plot shows the crossing cost c_{cross} (without normalization, i.e., total number of crossings). The right plot shows the distance cost c_{dist}.

Summarizing, we conclude that there is a high optimization potential for both considered Criteria C$_{\text{cross}}$ and C$_{\text{dist}}$. With a suitable choice of α both criteria can be substantially optimized simultaneously. Specifically with $\alpha = 0.5$, the number of leader crossings can be reduced by a maximum of 62.9% and the distance cost can be reduced by a maximum of 72.8%.

Evaluation of Heuristic

We evaluate the quality of our heuristic by comparing its results with the optimal ones obtained from the exact approach. For each instance and each $\alpha \in \{0, 0.5, 1\}$, we compute the relative cost $\delta_H(\mathscr{L}) := \delta(c_{\text{Slid}}, \mathscr{L}, \mathscr{L}_{opt})$ where $\mathscr{L}$ is a labeling produced by the heuristic and $\mathscr{L}_{opt}$ is the optimal labeling for the same instance. Figure 5.10 shows these relative costs.

For $\alpha = 0$ and for 68% of the instances, the relative cost is less than 20%. Further, 92% and 93% of the instances have relative costs less than 20% for $\alpha = 0.5$ and $\alpha = 1.0$, respectively.

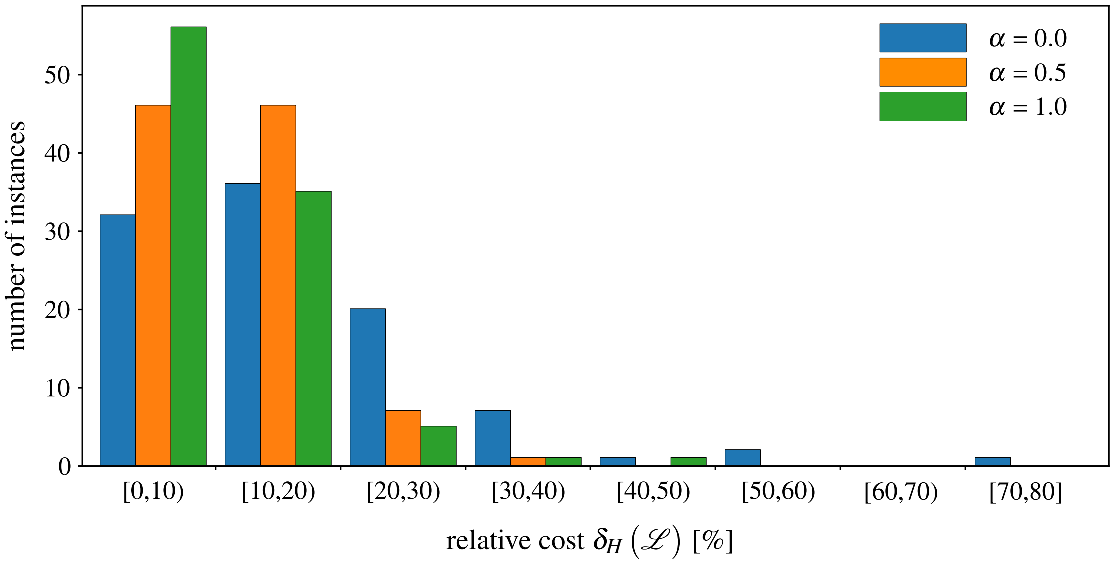


Fig. 5.10: The relative costs $\delta_H(\mathscr{L})$ for labelings $\mathscr{L}$ of the heuristic and according optimal solutions for $\alpha \in \{0, 0.5, 1\}$ of sliding boundary labeling.

On average, the relative costs of the heuristic solutions are 17.1%, 11.3%, and 9.5% for $\alpha = 0$, $\alpha = 0.5$, and $\alpha = 1.0$, respectively. In particular, for $\alpha \in \{0.5, 1.0\}$ about 90% of the quality of the optimal solution was reached on average. Hence, apart from some instances, we argue that our heuristic provides results that are sufficiently close to optimal.

We further aim to systematically analyze how the heuristic behaves for each $\alpha \in \mathscr{A}$, i.e., α sampled in the range [0, 1] with a step width of 0.025 (see Section 5.7.1). For a certain value of $\alpha \in \mathscr{A}$, we denote a corresponding labeling by $\mathscr{L}_\alpha$. As we do not take the exact approach into account, we use five labels for each state as also done for the other labeling concepts.

To find a suitable sweet spot that balances Criteria C_{cross} and C_{dist}, we consider for each labeling $\mathscr{L}_\alpha$ the *relative crossing cost* $\delta_{\text{cross}}(\mathscr{L}_\alpha) := \delta(c_{\text{cross}}, \mathscr{L}_\alpha, \mathscr{L}_1)$ where $\mathscr{L}_1$ only takes Criterion C_{cross} into account. Similarly, we consider the *relative distance cost* $\delta_{\text{dist}}(\mathscr{L}_\alpha) := \delta(c_{\text{dist}}, \mathscr{L}_\alpha, \mathscr{L}_0)$, where S_0 only takes Criterion C_{dist} into account. Figure 5.11 shows both considered relative costs $\delta_{\text{cross}}(\mathscr{L}_\alpha)$ and $\delta_{\text{dist}}(\mathscr{L}_\alpha)$. With increasing α, we observe that the relative crossing cost monotonically decreases and the relative distance cost increases. For $\alpha = 0$, the relative crossing cost is highest with 59.3% on average. Symmetrically, the relative distance cost is highest for $\alpha = 1$ with a value of 234.3% on average. Interpreting the plots as line objects consisting of multiple segments and computing the geometric point of intersection, we observe a suitable sweet spot between both criteria at $\alpha = 0.14$. In particular, for $\alpha = 0.14$, the relative cost of both c_{cross} and c_{dist} is 25% on average.

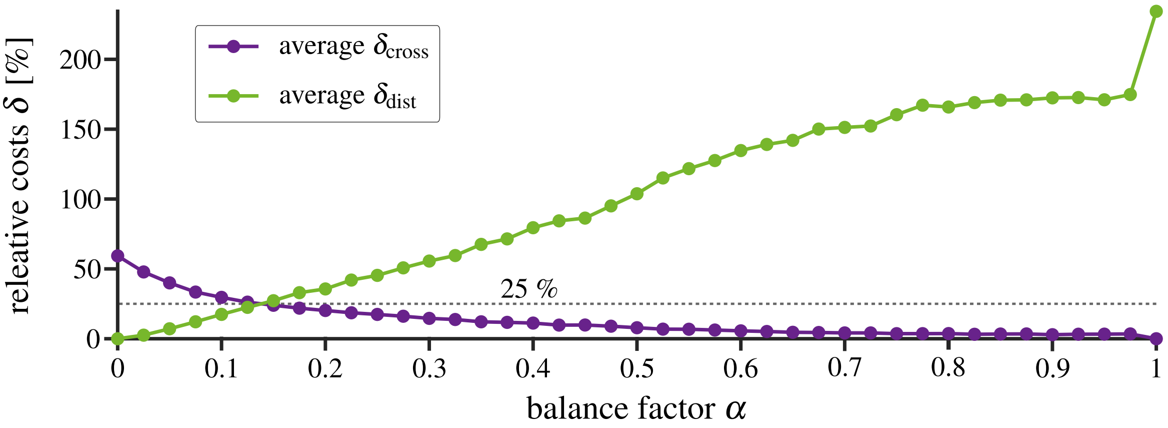


Fig. 5.11: Average relative crossing cost δ_{cross} and average relative distance cost δ_{dist} for each $\alpha \in \mathscr{A}$ of sliding boundary labeling.

Aiming to further evaluate the extent to which Criterion C_{dist} is optimized, we consider the distribution of vertical leader distances within a range of $[0, 50]$ pixels for $\alpha \in \{0, 0.14, 1\}$; see Figure 5.12. We obtain that compared to the results in which only Criterion C_{cross} is optimized ($\alpha = 1$), we can reduce the number of vertical leader distances smaller than five pixels by 77.2% when solely optimizing C_{dist} ($\alpha = 0$). When balancing both criteria with $\alpha = 0.14$, we can still reduce these smallest leader distances by 72.1%.

5.7.3 Stacking Boundary Labeling

Similar to the evaluation for multi-page boundary labeling, we analyze both the priority cost (Criterion C_{prio}) and the distance cost (Criterion C_{dist}) for stacking boundary labeling. Recall that a stacking boundary labeling $\mathscr{L}$ can be interpreted as a multi-page boundary labeling. Hence, we can evaluate the relative priority costs $\delta(c_{\text{prio}}, \mathscr{L}, \mathscr{L}_0)$ where $\mathscr{L}$ is a stacking boundary labeling and $\mathscr{L}_0$ is the multi-page boundary labeling with minimal priority cost. On average, the relative

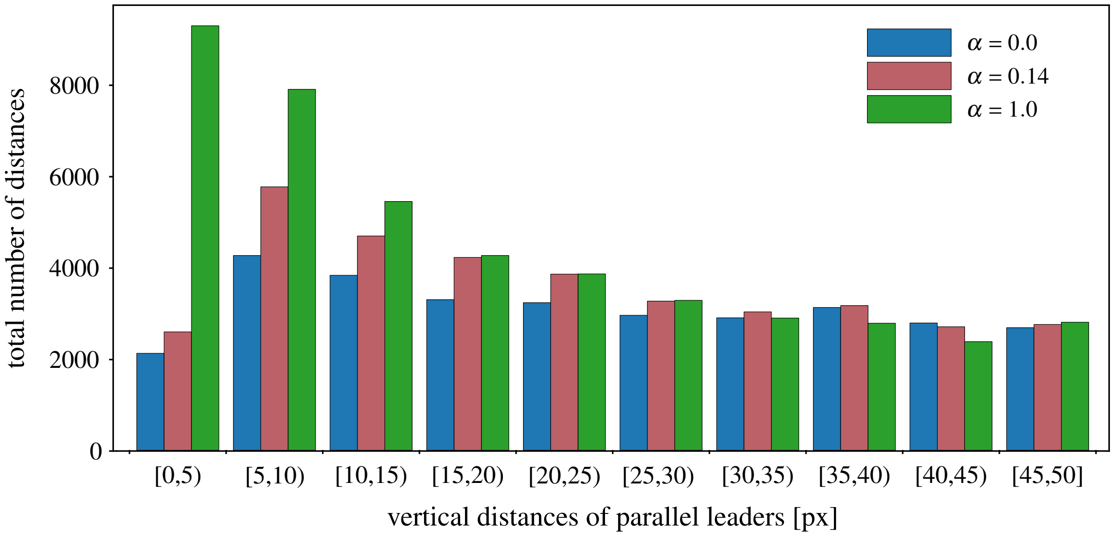


Fig. 5.12: Distribution of small vertical leader distances in the range of $[0, 50]$ pixels for $\alpha \in \{0, 0.14, 1\}$ of sliding boundary labeling.

priority cost is 11.6% with a standard deviation of 8.6%. Hence, high-rated features are nearly as equally distributed on the stacks as for the optimal multi-page boundary labeling. Put differently, we obtain a small number of interactions necessary for exploring them. Moreover, we evaluate the vertical distances of leaders that run horizontally above each other. Let H be the set of all pairs of leaders with horizontal segments lying above each other and for which the leaders start at different ports. The ratio between the size of H and the overall number of leader pairs starting at different ports is 17.6% on average with a standard deviation of 17.6%. The number of leader pairs in H with a vertical distance smaller than five pixels is 7.9%. Hence, on average, we obtain a smaller number of leader pairs with horizontal segments running above each other and a smaller number of leader pairs with vertical distances under five pixels than for multi-page boundary labeling (see evaluation in Section 5.7.1).

5.7.4 Comparison of Models

To provide decision support for deploying our three different concepts in practice, we compare their key properties (see Table 5.2).

Tab. 5.2: Key properties of the three labeling concepts. The considered criteria are either enforced as *hard* constraint or optimized as *soft* constraint if reasonable.

Key Property	Multi-Page	Sliding	Stacking
C_{prio}	soft	hard	soft
C_{cross}	hard	soft	hard
C_{dist}	–	soft	–
C_{len}	soft	–	soft
animation	discrete	continuous	discrete
change	k labels	1 label	1 label

In all concepts, one criterion is expressed as a hard constraint and two criteria are additionally optimized, i.e., modeled as soft constraints. Hence, the enforced criterion can be interpreted as the primary design rule. For example, for crossing-free labelings multi-page and stacking boundary labeling are preferable.

Further, the concepts differ in their interaction techniques. Both multi-page and stacking boundary labeling are designed for discrete interaction, so that switching between two states involves no transitional animation. In contrast, sliding boundary labeling provides continuous animation, which can support the user in tracing changes.

Another difference is the amount of visual change between two interaction steps. In multi-page boundary labeling, all k labels are exchanged in each interaction step. In contrast, in the two other models, only one label is exchanged with each interaction. Hence, for multi-page boundary labeling, less interaction is necessary to explore all information while the other concepts provide the possibility of adjusting the visualization in small steps. In sliding boundary labelings, the interaction is still strongly bounded, i.e., the labels are slid along in a specific order. Contrary, stacking boundary labeling provides less restricted interaction. The user can easily customize the displayed labeling by changing the topmost labels on the stacks. In particular, the user can explore features that lie closely together in at most $\lceil \frac{n}{k} \rceil$ steps.

For all concepts, we measured the running time for $k = 5$ ports and $n = 30$ point features; see Table 5.3. We note that the running times for sliding boundary labeling refer to the heuristic approach. To further assess the impact of the number of ports and features on the running time, we also considered a setup with $k = 10$ and $n = 100$. Since the running times are similar for all values of α, we averaged them for each setup and concept. We observe that for both multi-page and stacking boundary labeling, the running time is strongly dependent on the number of features but not on the number of ports. In contrast, the running time of the sliding boundary labeling increases only slightly with a higher number of features but more strongly with the number of ports. However, for each setting the running time is less than a tenth of a second, which – in the field of application design – corresponds to the limit at which the user feels that the system reacts immediately (Nielsen, 1993). Therefore, we consider the algorithms of all concepts suitable in terms of their actual applicability.

Tab. 5.3: Averaged running times with standard deviation of multi-page, sliding (heuristic) and stacking boundary labeling in milliseconds.

Setup		Multi-Page	Sliding	Stacking
$k = 5$	$n = 30$	10.58 ± 2.89	44.30 ± 0.52	10.51 ± 3.51
	$n = 100$	83.53 ± 11.18	53.53 ± 0.60	71.10 ± 4.92
$k = 10$	$n = 30$	7.67 ± 3.09	75.47 ± 0.93	8.72 ± 3.93
	$n = 100$	85.58 ± 12.53	96.20 ± 1.82	74.06 ± 5.40

Overall, we argue that no concept is generally better than the others. Which of the three concepts is more suitable depends largely on the specific application. With Table 5.2, we have provided a decision aid that highlights the strengths and weaknesses of each concept.

5.8 Conclusion

In this chapter, we have introduced and evaluated three external labeling concepts for zoomless maps. Instead of placing labels directly onto their corresponding point features, the presented concepts shift them to the bottom side of the map. For all three concepts, we have provided algorithms that run fast enough for applications on mobile devices such as smartwatches (assuming server-client communication). They enforce and optimize different criteria which we have derived from both existing literature and a preliminary survey with domain experts. With a systematic and quantitative evaluation, we have validated our mathematical models and proved the performance of our algorithms. Altogether, this chapter provides the following novel contributions.

- Three new external labeling concepts for zoomless maps.
- Design decisions for external labeling based on expert advice.
- Generalization of the three labeling concepts into a unified algorithmic framework.
- Exact algorithms and one fast heuristic for the optimization problems.
- Decision support for deploying the different concepts in practice.

By placing the labels externally instead of internally as in the concepts presented in Chapter 3 and Chapter 4, we have aimed to improve the legibility of the background map. Placing the labels at the bottom side of the map prevents them from obscuring essential contextual information such as roads. Further, our external labeling concepts allow for a more structured presentation of information, as they allow labels to be presented in a specific order, e.g., descending by relevance. However, we point out that we do not place the external strategies over the internal ones. Both internal and external label placement have their pros and cons. With our three external concepts, we have presented alternatives to the internal multi-page strategies. The decision which of the strategies to choose finally depends on the preferences of a user and the particular use case. To assess the usability and utility of the different concepts of zoomless maps introduced, an empirical user study is presented in the following Chapter 6.

6 AN EMPIRICAL STUDY ASSESSING THE USABILITY AND UTILITY OF ZOOMLESS INTERFACES FOR MOBILE MAPS

The following chapter mainly covers joint work with Jan-Henrik Haunert (Gedicke and Haunert, 2023) in which we present an empirical study assessing the usability and utility of different concepts of zoomless maps. The study was designed, conducted, and assessed by me (Sven Gedicke), with valuable support from Jan-Henrik Haunert during the different phases.

In particular, we implement three of our previously introduced concepts as actual user interfaces and compare them against an established zoom-and-pan interface. As a primary objective, we aim to validate the central hypothesis of the thesis for the common task of browsing through a large collection of features to find an object that suits a user's preferences best. As part of a user-centered design, we investigate how well the specialized interactions are adopted by users and whether they provide advantages over the established combination of zooming and panning. As the participants took part in the study, we recorded different metrics to analyze their behavior. Furthermore, we asked the participants for their personal opinions in a questionnaire after the study was completed. Our evaluation shows that participants perform significantly fewer zooming and panning operations using our novel interfaces than using the zoom-and-pan interface. We observe advantages in terms of search accuracy and the extent of the visible map area. Generally, the participants are open-minded about our novel interfaces.

6.1 Introduction

The previous chapters introduced different concepts for zoomless maps that rely on either internal or external labeling strategies. For each proposed concept, we discussed its theoretical merits and performed quantitative evaluations of the underlying models and algorithms. However, when developing new applications and interfaces, another crucial aspect is the verification of their usability and utility. In this context, *user-centered design* (UCD) describes an early focus and active integration of the user in the design process (Abras et al., 2004; Robinson et al., 2005).

In this chapter, we aim to assess both the actual usability and utility of our proposed concepts and present an empirical study with test users. In particular, we focus on three of our concepts which we implement as executable user interfaces: *internal multi-page labeling* with consistency (see Chapter 4), *multi-page boundary labeling* (see Chapter 5), and *sliding boundary labeling* (see Chapter 5). In this exact order, for the sake of simplicity, we will refer to the interfaces as MULTIPAGEINT, MULTIPAGEEXT, and SLIDING in the following. We subsume all three interfaces under the term *zoomless interfaces*; an illustration is shown in Figure 6.1. In our study, we compare the zoomless interfaces against an established zoom-and-pan interface which we refer to as BASE (see Figure 6.2). With respect to the multi-stage development of zoomless interfaces (see Chapter 1), we view this study as an integral step completing one cycle of the development process.

Our primary aim is to validate the central hypothesis of the thesis for the common task of browsing through a large collection of features to find an object that best matches specific attributes, i.e., a ranking task within a visual search. Based on this task, we assess how well the specialized interactions of the zoomless interfaces are adopted by the participants and whether they actually lead participants to zoom and pan less when searching for specific information. Further, we

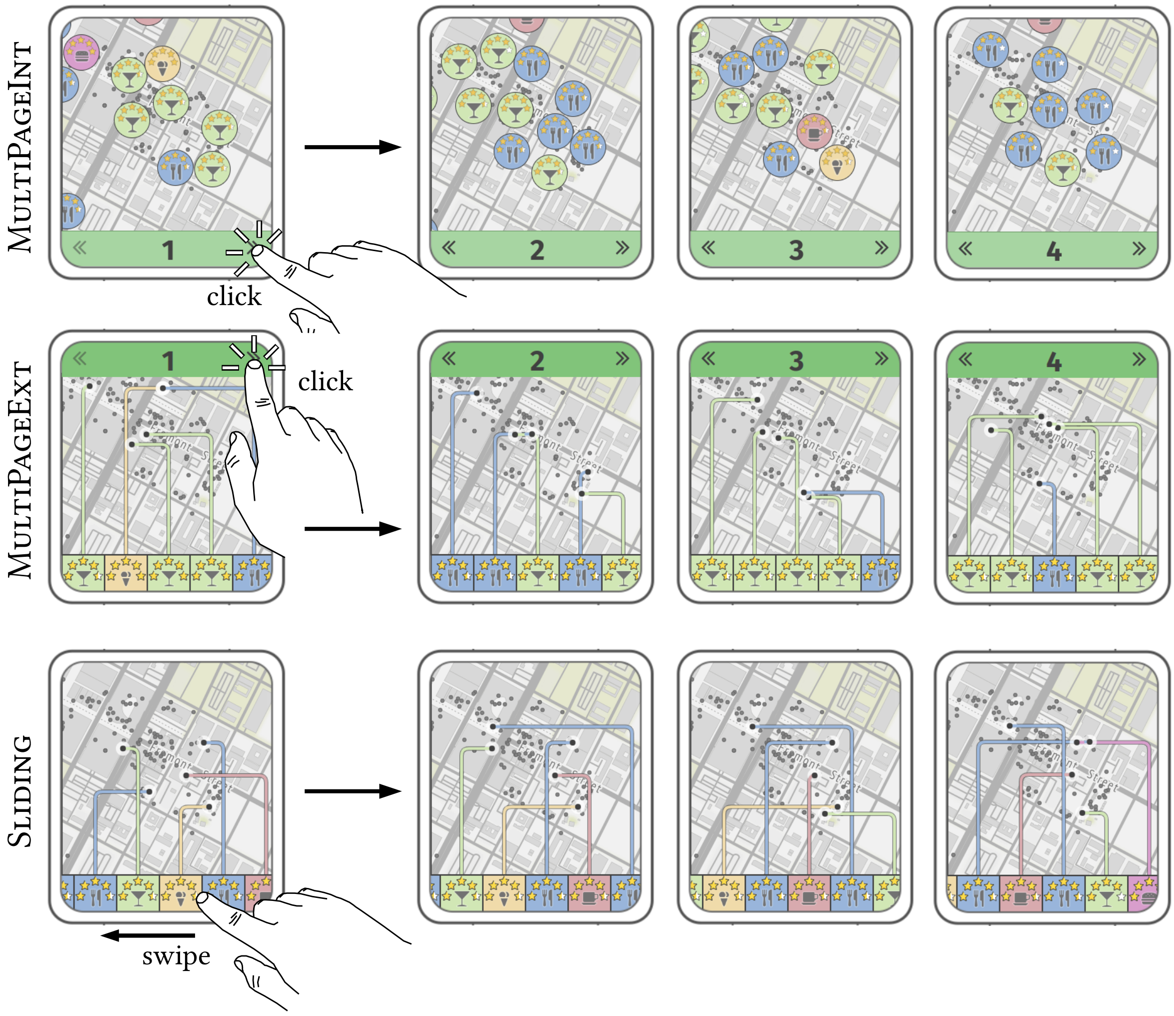


Fig. 6.1: Implementation of the three zoomless interfaces MULTIPAGEINT, MULTIPAGEEXT, and SLIDING in the desktop environment. For each interface, it is shown how the displayed information changes through the interactions.

aim to investigate to what extent the novel interactions support participants in their search with respect to the search accuracy and the amount of visible contextual information, i.e., the extent of the visible map area. Specifically, we pose three research questions. We hypothesize that, when searching for information that aligns with specific criteria, the three zoomless interfaces

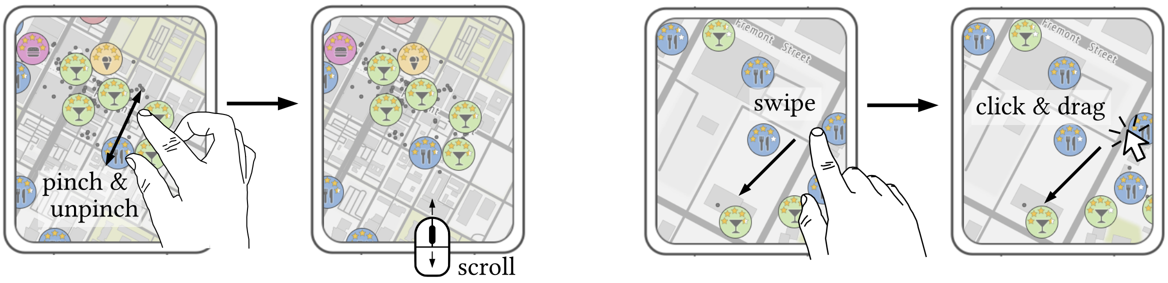
H1 reduce the number of both zooming and panning operations,

H2 improve the search accuracy,

H3 allow more map context to be visible when exploring the map,

compared to the established zoom-and-pan interface.

To test these hypotheses, we conducted an online study in which we deployed the four tested interfaces in a desktop environment. For each interface, we simulated its use on a smartwatch. All interactions were implemented mouse-based. We used point features with different attributes and visualized the features with symbols showing two of these attributes in the form of pictograms; a category and a user rating. Assuming an application where touching a symbol reveals more information, such as addresses or photos appearing in the form of pop-up windows, we chose symbols large enough to be unambiguously touchable. Recording different metrics during the processing of each task, we were able to quantitatively compare the tested interfaces regard-



(a) zooming

(b) panning

Fig. 6.2: Implementation of the basic operations (a) zooming and (b) panning in the desktop environment. Shown is the zoom-and-pan interface BASE.

ing their utility. We also gained insights into the participants' perceptions by asking them for their personal opinions in a questionnaire after completing the tasks.

Outline The structure and design of the user study is described in detail in Section 6.2. In particular, we provide a report about the participants, a short description of the used interfaces, and explain the technical setup and the tasks. Subsequently, we give an overview of the procedure and an explanation of our analytic strategies. With the methods of our study outlined, we provide a detailed presentation of the study's outcomes in Section 6.3. In particular, we assess different quantitative measures that we recorded during the completion of the tasks and evaluate the ratings and comments given by the participants. In a comprehensive discussion (Section 6.4), we address the outcomes in the context of the stated hypotheses and address limitations of the study. Finally, we summarize our main findings and give a brief outlook in Section 6.5.

6.2 Study Design

We begin with a report on the participants and then go into detail about the materials used. In particular, we explain the four map interfaces under test, describe the technical setup and the tasks and stimuli. Further, we address the procedure that the participants followed when taking part in the study. We conclude the section with a description of our analytic strategies.

6.2.1 Participants

To reach a large number of participants, the study was promoted through emails sent to university-wide mailing lists, requests to friends and family, and posts on social media. The study was accessible via a URL and participation was free of compensation. In total, 130 people participated. With 67%, the majority of participants indicated they were male while 29% stated they were female. While 25% of the participants indicated that they use interactive maps professionally, 75% only use such maps for leisure. Considering the age distribution of participants in Table 6.1, we were able to cover a wide range of age groups. With 41%, the age group of 25-35 years accounted for the largest share.

Tab. 6.1: Age distribution of the participants.

	under 25	25-35	36-45	46-55	56-65	over 65
number	24 (18%)	53 (41%)	18 (14%)	19 (15%)	12 (9%)	4 (3%)

6.2.2 Interfaces under Test

We tested four different map interfaces in our study. We used one baseline interface that relies on zooming and panning only and three different interfaces that implement additional specialized zoomless interactions. For an illustration of the baseline interface see Figure 6.2 and for the zoomless interfaces see Figure 6.1. To get an impression of how to use the interfaces, we provide a demo video[1]. We recall that the idea of zoomless maps is not to exclude zooming and panning, but to reduce the need for relying on these interactions. Thus, in all four interfaces, zooming and panning was available. Since we deployed the interfaces within a desktop environment, we implemented all interactions mouse-based.

We assume that we are given a set P of point features within a pre-defined rectangular region $R \subset \mathbb{R}^2$ that we refer to as *map*. We visualize each feature p with a *label* ℓ that describes the feature. We denote the set of all labels by L. Further, we introduce a weight function $w \colon L \to \mathbb{R}^+$ that assigns a weight $w(\ell)$ to each label $\ell \in L$ expressing the importance of its feature. Although the underlying concept and the mathematical model of each of the interfaces tested in the study has already been explained in detail in the previous chapters, for the sake of completeness, we provide a brief description of each interface below.

Baseline Interface (Base) In this interface, we place the labels internally. We regard a label not only as additional information, but also as a representation of a feature itself. Hence, rather than introducing multiple candidate positions, we centrally place a label on its corresponding point feature. We follow an established approach of selecting a maximum weighted independent set (*MWIS*) of all labels such that no two labels overlap. Only the selected labels are visualized in the map (see Figure 6.2). We solve MWIS by using a simple integer linear programming formulation (ILP) of Haunert and Wolff (2017). We prevent labels from continuously appearing and disappearing when panning the map by computing a MWIS for all labels contained in a given data set at each zoom level. To reveal labels that are not contained in the current set of visualized labels, the user needs to zoom and pan.

Internal Multi-Page Labeling (MultiPageInt) This interface implements the approach presented in Chapter 4. Similar to Base, we visualize each feature with a label that is placed centrally on its feature. All labels are distributed on multiple pages such that each page consists of an overlap-free subset of labels. The user can navigate through the sequence of pages while the background map remains fixed. Specifically, we display a panel at the bottom of the map on which there are two buttons for navigating back and forth (see Figure 6.1). The current page number is displayed in the center of the panel. For distributing the labels on pages, we use an optimization approach that provides a consistent label-page assignment, i.e, the method prevents labels from repeatedly appearing and disappearing when panning the map. The labels are distributed on pages in such a way that the number of pages is small, high-rated labels appear on front pages, and the spatial distribution of labels is balanced on each page.

Multi-Page Boundary Labeling (MultiPageExt) For this interface, we adapt the external multi-page approach introduced in Chapter 5. Instead of placing the labels internally, this interface shifts them to the bottom side of the map (see Figure 6.1). We arrange a prescribed number k of labels next to each other on one horizontal level such that they do not overlap. We visually associate the labels with their features by connecting them with leaders. We use so-called

[1] https://youtu.be/inBRqezTOAA

po-leaders, which consist of one segment running parallel (p) and one segment running orthogonal (o) to the bottom edge of the map; see Bekos et al. (2019). Similar to MULTIPAGEINT, we distribute the labels on multiple pages such that on each page a distinct set of features is presented. With each new page, up to k new labels are shown. We note that the last page can contain fewer than k labels if $n/k \notin \mathbb{N}$ with n being the total number of features located in the map. As the bottom of the map is used for placing the labels, we display a panel with navigation buttons at the top of the map in this interface. By clicking the buttons, the user can navigate back and forth through the sequence of pages. Our approach optimizes both the order of labels with respect to their importance and the length of the leaders. By construction, the approach ensures a minimal number of pages and that leaders do not intersect each other.

Sliding Labeling (SLIDING) This interface implements another external strategy presented in Chapter 5. We place the labels in a single row that can be continuously slid along the bottom side of the map (see Figure 6.1). Only the labels directly below the map are shown to the user. Similar to MULTIPAGEEXT, *po*-leaders connect the labels with their features in the map. No additional navigation controls are needed to slide the labels, as the row of labels itself can be shifted. We use a heuristic approach in which the labels are ordered in descending weight order. The approach minimizes the number of intersecting leaders and avoids leaders that run closely in parallel.

6.2.3 Technical Setup

To avoid possible incompatibilities on different mobile devices and operating systems, we designed an online study in which we simulated the use of a smartwatch within a desktop environment. For each of the interfaces, we showed the same schematized sketch of a smartwatch with a rectangular screen that has a resolution of 300 × 355 pixels. We note that this corresponds to a reasonable resolution for smartwatches (Jackson and Pao, 2019). Depending on the interface, either the entire screen is used to display the map or parts of it are used to place the labels externally and/or provide interaction controls (see Section 6.2.2). Demonstrating a realistic map application, we used map tiles provided by *Stamen Design* which rely on ©*OpenStreetMap* data as a background map. We decided on tiles in the *terrain* style because they use unobtrusive colors that we consider suitable for a background map that does not distract the focus from the labels. We obtained our experimental data from the review portal Yelp[2]. We used point data of food services in five metropolitan areas in the USA and Canada that we grouped in five categories; *restaurant*, *bar*, *ice cream parlor*, *fast food*, and *cafe*. For each feature, Yelp provides a user-given rating from one to five that we used as the weighting $w(\ell)$ of each feature's label ℓ.

For each of the five categories to which a label's feature can belong, we use a different color for the labels and leaders. Additionally, each label contains a centrally placed pictogram that also represents the category (e.g., cocktail glass for category *bar*). To further visualize the weight of a feature, we represent its rating in the form of radially arranged golden stars above the pictogram. We note that half stars, e.g., for a weight of 3.5, are also possible. Considering the shape and size of the labels, we distinguished between the interfaces that place the labels internally (BASE, MULTIPAGEINT) and those that place the labels at the bottom side of the map (MULTIPAGEEXT, SLIDING). If the labels are placed internally they should provide enough space to convey basic information, but not cover too much of the background map. Assuming an application where selecting (touching or clicking) a label reveals more information – such as addresses or photos appearing in the form of pop-up windows – the labels should further be large enough to be unambiguously touchable. Therefore, we decided to use rather large round

[2] `https://yelp.com`

symbols with a diameter of 50 px as labels. When placing the labels at the bottom of the map, rectangular labels are more suitable because they can be arranged in one line without creating visual gaps. With a screen width of 300 px, we chose a size of 50 × 60 px for the labels, so that five labels can be arranged next to each other without producing overlaps. To indicate the location of features that are not labeled in the current visualization, we represented such features by small grey dots. We note that we had to make design decisions for the implementation of the interfaces (such as shapes, sizes, and colors), but that these are not subject of our scientific contribution. We do not put our assumptions above other possible designs.

We used HTML as the structural foundation, which allowed us to easily design the layout, record different metrics, and embed external contents. To incorporate map data, we used the open-source JavaScript library OpenLayers which provides an API for creating dynamic maps with built-in map features such as zooming and panning. Considering BASE and MULTIPAGEINT, we implemented our approaches in Java and solved the corresponding ILP formulations with the commercial Gurobi[3] Optimizer version 9.5.1. For MULTIPAGEEXT and SLIDING, we implemented the corresponding algorithms in JavaScript.

6.2.4 Tasks and Stimuli

Using data of food services in five metropolitan areas in the USA and Canada, we build a separate *data set* for each of these regions. One of these data sets was used for a tutorial which was the same for each participant (see Section 6.2.5). We used the remaining four data sets of the regions *Las Vegas*, *Calgary*, *Phoenix*, and *Pittsburgh* for the actual tasks the participants had to complete. To each of the data sets, we assigned one specific task. The structure of the task is always the same:

Given a hypothetical user position marked in the map, we asked to select a label of a feature of a certain category which is rated as high as possible and is close to the marked position.

For convenience, we refer to the different tasks according to their associated data set regions.

In our experimental design, we deem the interfaces the independent variable that we are interested in examining to see how they impact the dependent variables (e.g., number of zooming and panning operations). Within the context of a repeated measures design, each participant tests each interface making them a *within-subjects factor*. The combination of our four levels of factors (i.e. the different interfaces) and the four tasks yields a total of 16 different stimuli. We randomly assigned each participant to one of four groups A, B, C, or D based on which a specific assignment to the stimuli was determined. Table 6.2 shows this assignment depending on a user's group. The number of participants in the groups came out to $||A|| = 29$, $||B|| = 35$, $||C|| = 35$, and $||D|| = 31$. With a maximum difference of six, we deem the distribution of participants to groups to be sufficiently balanced.

Tab. 6.2: The combinations building the 16 stimuli used in the study. The cells contain the assignment of the four different participant groups to the stimuli.

	Las Vegas	Calgary	Phoenix	Pittsburgh
BASE	Group A	Group B	Group C	Group D
MULTIPAGEINT	Group D	Group A	Group B	Group C
MULTIPAGEEXT	Group C	Group D	Group A	Group B
SLIDING	Group B	Group C	Group D	Group A

[3] https://gurobi.com

6.2.5 Procedure

As mentioned in Section 6.2.3, we conducted an online study in a desktop environment to avoid incompatibilities on different mobile devices and operating systems. We provided the study in both English and German between which the participants could choose in a first step. We provided some basic information about the scope and the estimated duration of the study and asked the participants to agree to a consent form. Using an exemplary illustration, we then explained what is shown in the map and what kind of information is provided by the labels. We gave the participants a listing of all occurring pictograms with an explanation of which category they represent (see Section 6.2.3). After this introduction, each participant needed to complete a tutorial that explained how to solve a task with each of the interfaces under test. For each interface, we explained the available interaction options.

After finishing the tutorial, the actual study started. Depending on the group that each participant was assigned to, a certain task had to be completed for each interface one after another (see Table 6.2). We actively chose to assign each participant only one task per interface to avoid a decrease in attention. By using alternating tasks, we also counteracted a learning and transfer effect between the interfaces. With the start of each task, we centered the map on the marked user position. By default, we set the map to a scale of 1:8000 (zoom level 16), which roughly corresponds to the representation of a small number of city blocks. To avoid participants selecting a label by mistake, the selection had to be confirmed by pressing a button.

To gain additional insight into the participants' perceptions, a short questionnaire followed the completion of the four tasks. For each of the four interfaces, we asked the participants to rate three statements from 1 (disagree) to 5 (fully agree). The statements thereby read:

S1 The interaction with the map is intuitive.

S2 Using this type of map, solving the given task was easy for me.

S3 I can imagine using this type of map for personal use.

In addition, we gave each participant the opportunity to provide an individual comment on each interface. After the questionnaire, the study ended by asking the participants to fill out a personal information form.

6.2.6 Analytic Strategies

To validate our posed hypotheses *H1* – *H3*, we recorded different metrics during the participants' task processing. To check whether the zoomless interfaces reduce the number of zooming and panning interactions when searching for information (*H1*), we captured how often a participant panned the map section and how often they zoomed in and out. To further validate whether the zoomless interfaces improve the ability to search for information that meets certain criteria (*H2*), for each task we collected the selected label's position, category, rating, and euclidean distance to the marked position. Aiming at assessing how much map context can be seen when selecting a label (*H3*), we recorded at what zoom level a label has been selected. Considering the size of the display and depending on the latitude of the shown map, we computed the size of the visible map area at task completion. Additionally, we tracked the time each participant needed for completing each task.

To check for statistical significance, we apply statistical tests for the recorded quantitative metrics. Since each participant tested each interface, i.e., we have repeated measurements, and our measurements are not normally distributed, we use a Friedman test for a non-parametric analysis (Purchase, 2012). In case the Friedman test yields a significant p-value ($p \leq 0.05$), we further apply a Nemenyi test for a post-hoc pairwise comparison between the four interfaces. The application of a Nemenyi test allows us to examine the pairwise differences without the need of applying statistical corrections.

6.3 Results of the Study

To test the hypotheses of the study, we evaluate the different quantitative measures that we recorded during the completion of the tasks. Subsequently, we additionally evaluate the ratings and comments of the participants. We note that we use the term *average* when we refer to the arithmetic mean.

6.3.1 Analysis of Interaction Behavior

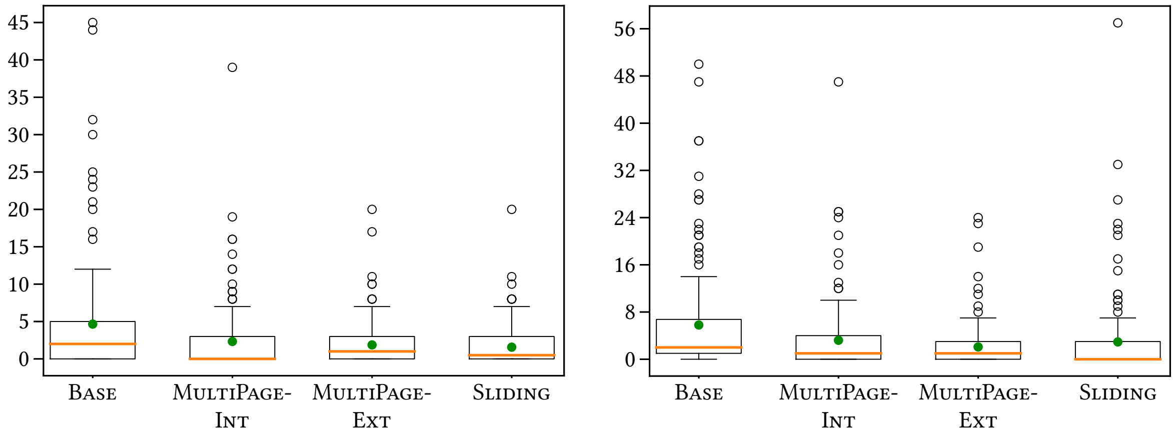
In the following, we assess the behavior of the participants in terms of the interactions they performed while working on the tasks. More specifically, we analyze the acceptance of the new interactions provided and evaluate whether the zoomless interfaces reduced the number of zooming and panning operations compared to the established interface.

Acceptance of Novel Interactions

We aim to find out whether the novel interaction techniques in our zoomless interfaces were actually accepted and used by the participants or whether they stuck to the established zoom-and-pan operations. To that end, we consider the number of performed page changes for MULTIPAGEINT and MULTIPAGEEXT, and the number of swipes for SLIDING. While for MULTIPAGEINT only 43% of the participants made any page changes at all, for MULTIPAGEEXT almost three quarters of all participants (71%) used the paging interaction. We assume that the rather low percentage for MULTIPAGEINT is related to the fact that the interface is visually similar to the established interface. The labels are also placed internally and so it can be used in the same way as the zoom-and-pan interface. This assumption is supported by the comments of some participants who stated that they relied on zooming and panning when using MULTIPAGEINT. However, the average number of page changes is still 8.1 for MULTIPAGEINT. For MULTIPAGEEXT it is even 13.6 on average and 4.0 in median. Considering SLIDING, we observe that 75% of all participants used the sliding interaction. With 14.0 slides on average and 4.0 in median, a majority of the participants embraced the novel interaction and used it for completing the task. Overall, we consider the user behavior regarding our novel interactions to be satisfactory. It was to be expected that some participants would retain their usual interaction behavior and continue to rely on zooming and panning. Generally, a certain period of familiarization is required before new interfaces and applications are fully accepted by users. Nevertheless, we observed that especially for interfaces that place labels externally, a large majority of participants employed our interactions.

Zooming and Panning

We validate whether the zoomless interfaces can reduce the necessity for both zooming and panning (see *H1*). To that end, we compare how often participants zoomed in and out and how often they panned the map section while processing a task. Figure 6.3 shows these measurements as box plots and Table 6.3 shows the distribution of participants with respect to the number of zoom and pan operations they performed per task for each interface.



(a) zooming operations

(b) panning operations

Fig. 6.3: Number of (a) zooming and (b) panning operations per task for each interface. Medians are shown as orange bars and average values are indicated by green circles. The boxes range from the lower to the upper quartile values with its length referred to as interquartile range (IQR). The whiskers above and below the box describe the range that is at most 1.5 times the IQR. Whiskers do not end exactly after this length, but at the value from the data that is still within this limit. Outliers are indicated by circles.

We observe that, on average and median, both fewer zooming and panning operations were applied when using the zoomless interfaces. More than half of the participants used both less than two zooming operations and less than two panning operations with the zoomless interfaces (see Table 6.3). For MULTIPAGEINT and SLIDING, half of the participants even applied no zooming at all. For SLIDING, this additionally holds for the panning operations. Especially when considering the higher numbers of operations, the differences between BASE and the zoomless interfaces become even more apparent. While 14.6% of the participants performed more than 10 zooming operations with BASE, this was only the case for between 1.5% and 5.4% of the participants using the zoomless interfaces. With respect to panning, over 15% of the participants panned the map more than twelve times using BASE. This holds for only between 3.1% and 6.2% of the participants when using the zoomless interfaces. Additionally, we observe that for both the number of zooming and panning operations, BASE has a higher number of upward outliers compared to the zoomless interfaces (see Figure 6.3).

Tab. 6.3: Distribution of participants with respect to the number of zooming and panning operations they performed per task [%].

# zooming operations	0	< 2	> 4	> 6	> 8	> 10	> 12
BASE	43.1	48.5	33.8	23.8	17.7	14.6	8.5
MULTIPAGEINT	53.8	60.0	18.5	10.8	7.7	5.4	3.8
MULTIPAGEEXT	48.5	55.4	13.1	7.7	3.8	2.3	1.5
SLIDING	50.0	63.1	10.8	6.9	2.3	1.5	0.8
# panning operations	0	< 2	> 4	> 6	> 8	> 10	> 12
BASE	24.6	39.2	30.8	25.4	18.5	16.2	15.4
MULTIPAGEINT	32.3	54.6	21.5	14.6	10.0	7.7	6.2
MULTIPAGEEXT	43.8	63.1	16.9	10.8	5.4	4.6	3.1
SLIDING	54.6	67.7	19.2	16.1	10.0	7.7	6.2

When applying statistical testing, we observe that for all zoomless interfaces the number of zooming operations is significantly lower than for BASE. Further, both between BASE and MULTIPAGEEXT and between BASE and SLIDING the difference in panning operations is statistically significant. In between the zoomless interfaces there are no significant differences for both the number of zooming and panning operations.

6.3.2 Analysis of Accuracy and Time

Recall that in each task we asked the participants to select a high-rated label that is close to an indicated user position. Considering the rating and the distance to the user position as objectives and each label as a possible solution, we determine the Pareto front of non-dominated solutions for each task. In other words, for each task we determine the set of labels for which it is not possible to improve one objective without simultaneously degrading the other. To verify whether participants were able to find appropriate labels matching the task, we analyze whether they selected a Pareto-optimal label or not. For each tested interface, Figure 6.4 shows the share of participants who did not select a Pareto-optimal label. We observe that 77% of the participants did not choose a Pareto-optimal label when using BASE. Thus, for more than three quarters of the selections, there would have been a label that is better rated or closer or both. For all zoomless interfaces, the share of not Pareto-optimal selections is lower. In particular, for the two interfaces that place the labels externally (MULTIPAGEEXT and SLIDING), the share of participants who did not select a Pareto-optimal label is only 35% and 23%, respectively.

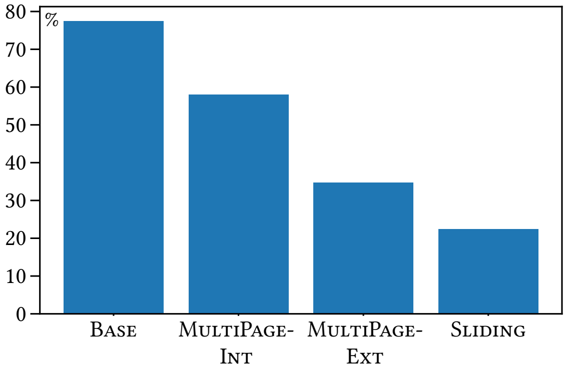


Fig. 6.4: Share of participants that did not select a Pareto-optimal label.

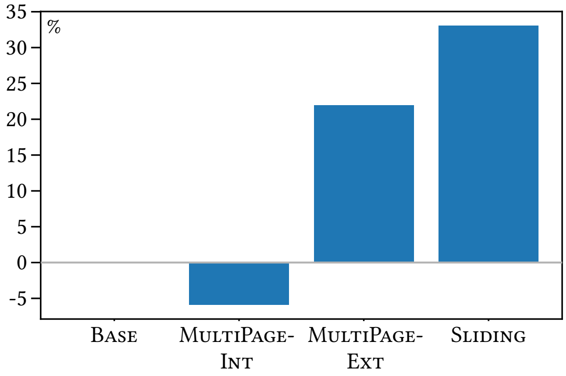


Fig. 6.5: Percentage difference of time needed to complete a task compared to BASE.

It is worth to consider these results in relation to the actual employment of the novel interactions (see Section 6.3.1). When using interface MULTIPAGEINT, MULTIPAGEEXT, and SLIDING, respectively, 43%, 71%, and 75% of the participants used the specialized interaction technique we proposed. It is noticeable that the higher the usage rate of our interaction techniques, the higher the quality of the selected labels.

We interpret these results as a confirmation that the tested zoomless interfaces provide easy accessibility to information. They allow users to search information more effectively and accurately with respect to specific criteria (see *H2*). Instead of zooming and panning to arbitrary locations, the user can easily browse through all labels within an area of interest.

In the context of these observations, the time taken by the participants to complete the tasks is an interesting measurement as well. For each zoomless interface, we compare the average time needed to complete a task with the corresponding time when using BASE. Figure 6.5 shows the percentage difference in these times. We observe that, on average, participants completed a task faster using MULTIPAGEINT than using BASE. Although the participants might be more familiar interacting with BASE, 6% less time was needed with MULTIPAGEINT on average. However, we also observe that completing a task with MULTIPAGEEXT and SLIDING took the participants more time than using BASE (22% and 33%, respectively). Reconsidering the share of Pareto-optimal selections in Figure 6.4, we see that the slightly longer processing times might correlate with a positive effect on the quality of the selected labels. Participants took more time to search for a label and as a result they mostly chose a more suitable one. We further see a reason for the longer processing times in the positioning of the labels in MULTIPAGEEXT and SLIDING. When searching for a suitable label, the user's gaze must continuously switch between the map and the labels. This is a fact that participants mentioned in their additional comments multiple times (see Section 6.3.4). We deem this to be a general problem when placing labels externally. However, considering the times for MULTIPAGEINT, we argue that the specialized interaction strategies themselves might have no negative influence on the time needed for solving the task. To investigate this theory further, eye-tracking would be a suitable method to capture the actual eye movements of the users. In the context of subsequent cycles in the development process of zoomless interfaces, studies with eye-tracking would thus be of interest.

Considering the times needed to complete one task, there is no significant difference between BASE and the zoomless interfaces. However, there is statistical significance between MULTIPAGEINT and MULTIPAGEEXT and between MULTIPAGEINT and SLIDING.

6.3.3 Analysis of Shown Map Context

In comparison to the zoom-and-pan interface BASE, we want to investigate whether the use of zoomless interfaces leads to more map context being visible to a user when searching for information (see *H3*). For each interface and participant, we recorded at what zoom level a label was selected to complete the task. Considering the size of the display and depending on the latitude of the shown region, we used the zoom level to compute the size of the visible map area. On average, the participants selected a label at a smaller zoom level with all zoomless interfaces than with BASE. Accordingly, a larger map area and thus more context was visible to participants when using zoomless interfaces (see Figure 6.6). This additional context can, for example, support users to spatially associate their own position with that of the selected information. In a direct comparison, MULTIPAGEINT, MULTIPAGEEXT, and SLIDING resulted in an average of 44%, 72%, and 40% more area being visible at the time a label was selected than BASE. Particularly between BASE and MULTIPAGEINT the difference is statistically significant.

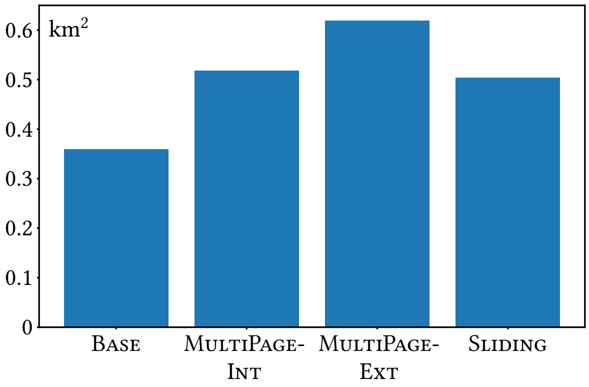


Fig. 6.6: Average size of the area that is shown to a participant when selecting a label.

6.3.4 Assessing Ratings and Comments of the Participants

After the participants finished the tasks, we asked them to rate three statements from 1 (disagree) to 5 (fully agree) with regard to the four interfaces under test. Table 6.4 shows the average and median scores given by the participants. When the participants were asked to rate how intuitive the interaction with the interfaces was (*S1*), on average they gave BASE the highest score. However, this was to be expected as it implements an established strategy used in many desktop and mobile applications. Although the zoomless interfaces were new to the participants, they rated the intuitiveness of the interactions with 3.22 to 3.70 on average. Thus, assuming

Tab. 6.4: Average and median scores for each interface and statement (median in parentheses).

Statement	BASE	MULTIPAGEINT	MULTIPAGEEXT	SLIDING
S1: intuitiveness	4.66 (5)	3.70 (4)	3.22 (4)	3.40 (4)
S2: utility for task	4.00 (4)	3.85 (4)	3.28 (4)	3.34 (4)
S3: personal use	4.15 (4)	3.14 (3)	2.68 (3)	2.91 (3)

that a rating of 3 is to be understood as neutral, participants tend to rate zoomless interfaces as rather intuitive. With regard to the median of the scores, we further observe that half of the participants even rated the intuitiveness of the zoomless interfaces 4 or better.

Considering the ratings on how easy the participants found it to solve a task with the different interfaces (*S2*), we observe that the median rating is the same (score of 4) for all interfaces. Further, we can see that particularly the average score for MULTIPAGEINT is close to the one for BASE. Participants seem to see benefits in the specialized interactions when searching for specific information. Both interfaces that place the labels externally (MULTIPAGEEXT and SLIDING) have been assigned a similar average rating with 3.28 to 3.34, respectively. Comparing these ratings with the actual quality of the selected labels in Figure 6.4, an interesting discrepancy becomes apparent. Although participants rated BASE as the best in terms of the ease of accomplishing the tasks, they selected the fewest Pareto-optimal labels with this interface. In contrast, with the two lowest-rated interfaces MULTIPAGEEXT and SLIDING, the highest number of Pareto-optimal selections was made. Thus, we assume a certain overconfidence and a bias of the participants, which we attribute to the familiarity of the participants with the baseline interface.

When the participants were asked whether they could imagine using the different interfaces in their everyday lives (*S3*), the zoomless interfaces received average scores between 2.68 and 3.14. Compared to the average score 4.15 of BASE, these scores are rather low. However, considering the additional comments that participants gave us, we received much positive and approving feedback on the zoomless interfaces. Many participants liked the integration of novel interactions. Especially swiping through the labels in SLIDING was often commented as a *cool* and *useful* interaction for exploring information. We see one reason for the overall low ratings of the zoomless interfaces with regard to *S3* in the fact that the participants are not familiar with the novel interactions. Many participants described the baseline interface BASE as *intuitive*, *a common approach*, or argued that they are *used to it*. Contrary, they indicated that they found the novel interactions to be *unusual* and *not straight forward*. When using MULTIPAGEINT, some participants reported that they only used zooming and panning as interactions because they found it *odd* to navigate through pages. Particularly concerning MULTIPAGEEXT and SLIDING, we received many comments on the external placement of the labels. Participants pointed out that they found it *annoying* and *time-consuming* to follow the leaders in order to associate features with their labels. Participants mentioned that after following a leader they needed to *reorient* themselves in the map. Further, it was commented multiple times that the leaders cover too much of the map background. On a more general level, we also received many comments on the design of our interfaces regarding suggestions on colors, shapes, and arrangements of elements, which we greatly appreciate as feedback.

6.4 Discussion

In the previous section, we presented the key observations on both the quantitative measures and the participants' feedback. In the following, we assess these results in the context of the posed hypotheses *H1* – *H3* and discuss limitations that we identified.

6.4.1 Assessment of the Study's Hypotheses

Considering the task of selecting information that meets specific criteria, we hypothesized that the tested zoomless interfaces can reduce the necessity for both zooming and panning (*H1*). The analysis of the participants' interaction behavior showed that both zooming and panning operations were used less with all zoomless interfaces compared to BASE (see Section 6.3.1). For all zoomless interfaces, the number of zooming and panning operations were significantly smaller to the ones of BASE. Hence, we argue that our hypothesis *H1* holds. With the reduced use of zooming and panning, the visualization of the underlying map changes less in terms of both extent and scale. This means that users do not have to reorient themselves often and can focus more on the exploration of information.

This assumption is supported with our observations on the quality of the selected labels (see Section 6.3.2). We showed that using the zoomless interfaces, a greater proportion of participants selected a Pareto-optimal label than with BASE. In other words, zoomless interfaces have proven to be helpful in finding information that best fits given criteria. Especially with the interfaces that place the labels externally, the participants showed a high performance with regard to the quality of the selected labels. Even though the participants needed up to one third more time to complete the tasks, the tasks were solved more accurately with zoomless interfaces than with BASE. Contrasting with this observation are the ratings of the participants who considered BASE to be most suitable for accomplishing the task. However, we attribute this assessment to a bias of the participants, who are more familiar with the zooming-and-panning-only strategy of BASE. Summarizing, we argue that zoomless interfaces indeed improve the accuracy when searching for information that meets certain criteria. Therefore, we also consider hypothesis *H2* to be confirmed.

The analysis of the size of the map area visible when selecting a label showed that, on average, more area is visible with all zoomless interfaces than with BASE (see Section 6.3.3). On average, between 40% and 72% more area is shown and hence more contextual information is available. This additional context can help a user with spatial orientation. Similar to our evaluation of the zooming and panning behavior, we argue that this allows users to focus more on the actual exploration of information. A statistical evaluation of the visible map areas for each interface showed that the difference is statistically significant only between BASE and MULTIPAGEINT. Hence, we argue that hypothesis *H3* holds partially.

6.4.2 Limitations of the Study

Instead of actually testing our interfaces on mobile devices, we decided to conduct an online study in which we simulated the use of a smartwatch within a desktop environment. While this offers the advantage of easier implementation and acquisition of participants, it also has two key limitations. The first concerns the realization of the specialized interactions. All interactions were implemented in a mouse-based manner, whereas in applications on mobile devices fingers would be used for touch-based interactions. Especially in terms of user experience (UX), but also user behavior and performance, this difference might bias the study's outcomes. Further, the study was not conducted in a mobile environment, which is inconsistent with the intended use of the interfaces. In mobile use, environmental factors would influence the attention and behavior of the participants. For these reasons, testing the interfaces on actual mobile devices in real-world scenarios is an intriguing step within follow-up studies on zoomless interfaces.

Another aspect worth discussing is the tasks and stimuli. We intentionally chose only one type of task and a small number of stimuli to keep the participants' full attention throughout the whole study. However, a larger number of stimuli could be helpful to differentiate results more precisely and to analyze further strengths and weaknesses of zoomless interfaces. Data sets of different information densities could be used to investigate how these affect the participants' behavior. Moreover, we used only one type of task, that is, a ranking as part of a visual search. However, to better elaborate the potentials and scopes of the different interfaces, subsequent studies should include further tasks for mobile maps, such as identifying, comparing, associating, and delineating.

Even though there was only one type of task, we made sure that each participant had a different particular task to solve with each interface. Which task to accomplish with which interface was randomized by assigning each participant to one of four user groups at the beginning of the study (see Table 6.2). However, we did not take into account that the four interfaces are presented to the participants in different orders. Thus, the interfaces were shown to each participant in the same order; BASE first, then MULTIPAGEINT, then MULTIPAGEEXT, and finally SLIDING. Generally, not varying the order of the interfaces can bias the outcomes of the study. However, the average times required to complete the tasks show that the longest times were spent on the interfaces tested last, i.e., MULTIPAGEEXT and SLIDING (see Figure 6.5). Thus, we infer that neither a learning effect nor a loss of interest was evident. Furthermore, we do not see a trend of a decreasing attention, as the interfaces that were tested last achieved better results in terms of the quality of the selected labels (see Figure 6.4).

Another source of potential bias in the results could be in the design choices made. Even though our contribution is not to promote one particular design over others, we had to make design decisions for the implementation of the interfaces. Especially the presentation of the labels, i.e., the symbolization, is a factor that can influence user preferences and performance. To determine whether certain study outcomes are biased by the chosen representations, different designs should be compared in the ongoing development of zoomless interfaces.

Further, we believe that both the quantitative results and the feedback from the participants are biased on the participants' habituated map usage. Since established maps rely largely on zooming and panning, most participants are familiar with these basic interactions. Thus, new interactions are initially counterintuitive and some people tend to reject them immediately. Participants stated that they also relied on zooming and panning when using zoomless interfaces rather than testing the provided specialized interactions. Not only does this directly affect the recorded measurements, but also has an influence on the participants' ratings of the interfaces. In all three statements *S1* – *S3*, BASE was rated best although it did not perform best or even worst in the quantitative evaluations. We feel that the results presented in the previous sections are strongly influenced by the bias and habit of the participants. To gain user acceptance for new ideas and applications, a longer introductory phase could aid in familiarizing users with them.

As one last limitation, we address the recorded times it took participants to complete the tasks. Since the participants were not in a controlled environment, it cannot be excluded that they were distracted by external influences and split their attention. This would have caused a direct impact on the response times and, thus, the time measurements are less reliable than in controlled settings. Thus, for continuing research, it would be interesting to consider controlled setups.

6.5 Conclusion

As an integral part of the development process of map interfaces, we have embraced the idea of a user-centered design by conducting an empirical user study. Our study aimed at validating the usability and utility of different zoomless interfaces that rely on the models and algorithms presented in the previous chapters. To that end, we had participants perform a ranking task within a visual search using three different zoomless interfaces and one established zoom-and-pan interface. Assessing different quantitative measurements as well as the participants' ratings and comments, we were able to show that the tested zoomless interfaces support users in their task. Compared to the baseline zoom-and-pan interface, we have proved that the use of our interfaces

- significantly reduces the number of zooming and panning operations,
- improves the search accuracy,
- results in a larger map area and, thus, more context being visible,

when searching for specific information. Although many participants rated the established interface best, a majority was open to the novel interactions and indicated that they appreciate the zoomless interfaces. This general interest in zoomless interfaces was also reflected in a large number of specific design suggestions that we received. We have identified evidence of a certain overconfidence of the participants in the established zoom-and-pan interface, which we attribute to the familiarity with zooming and panning. Generally, one should keep in mind that habit is a substantial factor in the subjective perception. It is human nature to often resist change and new experiences at first. This phenomenon can lead to people tending to reject unfamiliar interfaces or applications. Established novelty needs time. Overall, however, we have confirmed the merits of the tested zoomless interfaces and have learned through the direct feedback that a large number of users consider the novel interactions to be helpful. We note that a detailed discussion regarding the degree to which the outcomes of the study confirm the central hypothesis of the thesis is provided in Section 8.1.

Besides these findings, we have also identified and discussed different limitations of our study. Since the online study was limited to a simulation of maps for mobile use, further empirical research should involve the actual use of mobile devices. Both to investigate user behavior with respect to touch-based interactions and to consider possible environmental influences, field studies with a smartphone or smartwatch should be subsequent steps in the development process of zoomless interfaces. Furthermore, additional limitations should be addressed by varying the number and type of tasks, comparing effects of different design decisions, and including eye tracking techniques.

Overall, we see our study as an important milestone that confirms the value of zoomless maps in its key characteristics. The main findings of our evaluation suggest that the anticipated benefits of zoomless maps apply to the common use case of searching for specific information. However, we do not consider the study as a conclusive endpoint but as a crucial step forward in ongoing empirical research within subsequent stages of the development process.

7 A CASE OF APPLICATION: EXTERNAL LABELING OF POINT FEATURES IN SITUATION MAPS FOR RELIEF FORCES

The following chapter mainly comprises joint work with Lukas Arzoumanidis and Jan-Henrik Haunert (Gedicke et al., 2023a) which builds upon the bachelor thesis of Lukas Arzoumanidis. Lukas Arzoumanidis and I (Sven Gedicke) extended the original mathematical model, adapted the existing exact algorithm, and developed a heuristic method. I took primary responsibility for writing the text, with support from Jan-Henrik Haunert.

While the previous chapters focused on general concepts for zoomless maps that are flexible in their application, we now aim to shift the attention to a practical use case. In particular, we focus on so-called *situation maps* which are used in the domain of disaster and emergency management. Such maps present mission-relevant information within the emergency area, providing situational awareness to the responders (see Figure 7.1 for an example). Usually, the information is communicated through the visualization of descriptive symbols which are still predominantly placed manually. With the intention of saving valuable time and relieving the burden on emergency personnel, we present an approach that automates this process. Based on an established layout of situation maps used in Germany that distributes the symbols to the map boundaries and following general principles from literature, we formalize the symbol placement as an optimization problem. We take into account the relevance of tactical symbols as well as short and crossing-free leaders and allow the grouped representation of symbols of similar semantics and spatially close map features. To generate solutions to our problem, we present one exact and one fast heuristic approach. In experiments with real-world data, we determine a balance between the optimization criteria and show that our heuristic generates high-quality results in less than a second. In an assessment by an expert, we get confirmation that our maps are suitable for use in emergency scenarios.

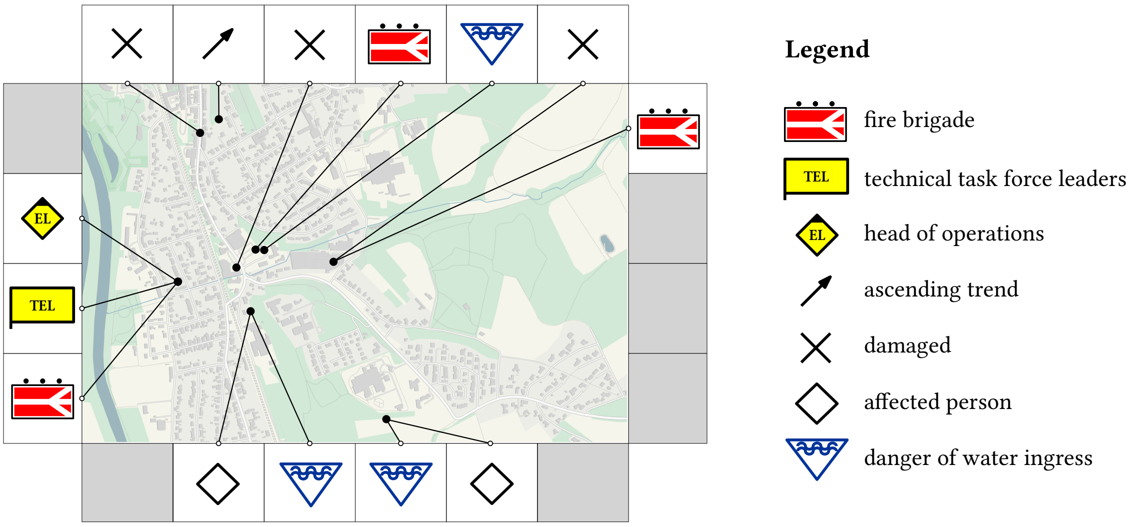


Fig. 7.1: Example of a situation map generated with our heuristic based on real mission data. The tactical symbols are chosen according to the guidelines of the German Federal Agency for Technical Relief (THW). *Map attribution: Carto.*

7.1 Introduction

In the previous Chapters 3 – 6, we have introduced, implemented, and evaluated concepts for zoomless maps that describe flexible and easily adaptable strategies for presenting information in interactive maps. These concepts are not specifically tailored to specific use cases, making them applicable to a wide spectrum of applications and tasks. In this chapter, however, we consider interactive maps for a more concrete area of application, i.e., for the domain of disaster and emergency management. More precisely, we present both exact and fast algorithms for presenting mission-relevant information within an interactive map interface.

When it comes to an emergency like a major fire or flooding, maintaining situational awareness is an important issue in emergency response. For an effective organization of emergency forces, it is essential to perceive and understand all relevant aspects of the situation and to be able to assess its further development (Endsley, 1995). Since a large number of people with different professional backgrounds usually work together, an extensive collection of information must be coordinated and communicated in a well-organized manner. To facilitate the communication of geographic information, so-called *situation maps* provide an overview of the operational area. Standardized descriptive symbols are placed to convey both the location and semantic information of important features, such as available assets, affected areas, squad positions, or hazards (see Figure 7.1). In the following, we refer to such symbols as *tactical symbols*.

Although robust and powerful mobile devices are increasingly used in emergency response, tactical symbols are still predominantly placed manually (Bernsdorf and Fritze, 2016). This process takes up both human and time resources, which are especially valuable in emergency situations. To get a quick overview of the situation and to coordinate tasks efficiently, a fast provision of a situation map is essential. Especially addressing the work of platoon leaders on site, we see great potential for saving time and increasing efficiency in changing from manual to automatic symbol placement.

Addressing this potential, we present both exact and fast algorithms for the automatic placement of tactical symbols in situation maps, i.e., for presenting mission-relevant information within an interactive map interface. More specifically, we consider situation maps commonly used in emergency response in Germany. We adopt an established map layout used by emergency services in North Rhine-Westphalia that is based on a rectangular map section showing the operational area (Lamers and Denker, 2022). The tactical symbols are placed externally at all four boundaries of the map and are visually connected to their positions in the map via thin connecting lines called *leaders*. Figure 7.1 shows an example of such a situation map generated with our approach. Within the context of map labeling, this type of label placement is known as *boundary labeling* (Bekos et al., 2019). We note that it is not part of our scientific contribution to place this map layout and labeling strategy above other possible designs. Our objective is to decrease the workload of emergency forces that use these maps by reducing the need for manual working steps. By providing algorithms for placing the symbols, we aim to contribute to the automatic generation of situation maps.

Mathematically, we formalize the symbol placement as an optimization problem that prohibits crossing leaders and prioritizes short leaders. It takes into account the operational relevance of the symbols and allows grouping them according to the semantics and spatial proximity of their features. In an informal survey at an early stage of the algorithmic development, these design decisions were generally supported by domain experts.

Outline To place our contribution in the context of disaster and emergency management, Section 7.2 discusses map-based communication in emergency response. We will then establish a mathematical framework in Section 7.3. Based on principles and observations from existing literature in the fields of cartography, information visualization, and emergency response, we derive the constraints and optimization criteria that we express within an optimization problem. We introduce both an exact approach using integer linear programming (see Section 7.4) and a heuristic method based on simulated annealing (see Section 7.5) that is fast enough for application in emergency scenarios. In experiments with real-world data that we discuss in Section 7.6, we determine an appropriate balance between the selected optimization criteria. We show that the results of the heuristic are of high quality with respect to the criteria. In a final assessment by an expert, we receive confirmation that the automation of the symbol placement is of high value and that our maps are suitable for use in practice. In Section 7.7, we summarize the contributions of our work and provide directions for future research.

7.2 Map-Based Communication in Emergency Response

Establishing situational awareness is one of the most crucial issues in emergency and disaster response (Karagiannis and Synolakis, 2016). Generally, situational awareness is understood as the perception and comprehension of all relevant aspects of the environment and the ability to assess its further development (Endsley, 1995). Especially when a large number of people with different professional backgrounds cooperate in dynamic and complex environments, providing a shared overview of the situation is essential for an efficient emergency response (Seppänen et al., 2013). Usually, a large amount of information needs to be coordinated and communicated in a well-organized manner. In this context, the *common operational picture* (COP) is known as a visual representation of relevant data from different sources that is shared among all involved actors (McNeese et al., 2006). Despite its prevalent use in emergency response, there is no unique definition for the structure and the included elements of a COP (Wolbers and Boersma, 2013). Both Sophronides et al. (2017) and Steen-Tveit and Munkvold (2021) provide a comprehensive review of different COP specifications established in literature. Regarding the development of COPs, much effort has been invested in identifying challenges and requirements (Blandford and Wong, 2004; Copeland, 2008; Seppänen et al., 2013; Hwang and Yoon, 2020; Steen-Tveit, 2020) as well as designing rules and general frameworks (Artinger et al., 2012; Boukhtouta and Berger, 2014; Steen-Tveit and Munkvold, 2021). Bringing such frameworks to actual implementations, systems relying on the use of handheld devices (Björkbom et al., 2013; Bernsdorf and Fritze, 2016; Šterk and Praprotnik, 2017) and even smart glasses (Demir et al., 2017; Conover and Dammann, 2017) have been proposed.

In the context of emergency and disaster response, COPs typically incorporate maps to facilitate the communication of geographic information (Kuvedžić Divjak and Lapaine, 2014) and enable situational awareness (Opach et al., 2023). More specifically, such maps provide an overview of the operational area (Karagiannis and Synolakis, 2016) and indicate the locations of relevant spatial features such as available assets, affected areas, squad positions, or hazards (Opach and Rød, 2022). Often, these maps are referred to as *situation maps* (Tomaszewski et al., 2015). Kuvedžić Divjak and Lapaine (2014) assess situation maps from a cartographic perspective and emphasize the need to account for cartographic principles to maintain an effective communication of information. Commonly, descriptive symbols or signs are used to annotate the features in the map (Lamers and Denker, 2022). Investigating different designs of symbols used in emergency response in Canada and America, Bianchetti et al. (2012) point out that establishing a standardization of the map symbology can further improve communication efficiency. With regard to this observation, many authors addressed the need for standardized symbols in the last decades (Opach and Rød, 2022; Kostelnick et al., 2008; Marinova, 2018; Robinson et al., 2011, 2013; Kuvežđić Divjak et al., 2020; Waidyanatha and Frommberger, 2022).

While much work has been invested in the design of symbols in situation maps, only little attention has been given to their placement. Commonly, symbols are placed directly at the position of their respective feature in the situation map. However, this can easily lead to important elements of the map being obscured, symbols overlapping, and a cluttered overall appearance. The more symbols are shown, the higher the perceived complexity (Liao et al., 2019) and correspondingly lower the readability (Harrie et al., 2015). In addition, internal placement does not allow symbols to be arranged in a particular order, i.e., according to their relevance or similarity. In German emergency response, situation maps often place the symbols at the boundaries of a rectangular map section (Bernsdorf and Fritze, 2016; Lamers and Denker, 2022). Symbols are visually associated with their positions in the map by thin connecting lines. Although numerous GIS-based software solutions for visualizing situation maps have been developed in recent years, these still predominantly require the placement of symbols by hand (Doeweling et al., 2013; Ramchurn et al., 2016; Bernsdorf and Fritze, 2016; Šterk and Praprotnik, 2017). This process takes up both human and time resources (Morrison, 1980), which are especially valuable in emergency situations. The acquisition of situational awareness is also known to be limited by the time-consuming manual labeling process (Kedia et al., 2022). In our literature review, we only identified one article addressing this issue in the context of emergency response: Considering multiple optimization criteria, Kaneider et al. (2013) present an approach for automatically placing labels in situation maps. They place the labels in such a way that they obscure as little relevant map content as possible and do not overlap each other. We also address the need for automating the process of placing labels in situation maps. Instead of placing the labels internally, we adapt the layout that is established in Germany and shift the labels to the map's boundaries. Put differently, we apply *boundary labeling* (Bekos et al., 2019).

7.3 Mathematical Model

In this section, we describe the algorithmic framework that we use for automating the placement of tactical symbols in situation maps. Based on related literature, we establish constraints and criteria that we consider for the symbol placement. Subsequently, we introduce a formal model based on which we define a mathematical optimization problem.

7.3.1 Defining Constraints & Criteria

We address situation maps within an established map layout used by the emergency services in Germany (Lamers and Denker, 2022). The layout consists of a rectangular map view and a fixed number of positions around the map at which tactical symbols can be placed (see Figure 7.1). The symbols are connected to their positions in the map by leaders. More precisely, sticking to the established design, we use leaders that consist of one straight line. To automate the symbol placement, we need to decide on specific constraints and optimization criteria that we formalize into a mathematical problem formulation. We base our choices on principles and observations from existing literature in cartography, information visualization, and emergency response.

To ensure an effective communication of information, maps for emergency response should generally be designed according to cartographic principles (Kuvedžić Divjak and Lapaine, 2014). With respect to map labeling, Imhof (1975) has established such principles and requirements that are intended to assure clarity and legibility. Based on a comprehensive review of literature, Bekos et al. (2019) provide a list of criteria that apply these principles to external labeling. They identify both the avoidance of crossing leaders and the preference for short leaders as the most prominent criteria. Both criteria reduce visual clutter and facilitate an unambiguous association between a map feature and its label. Thus, in our model, we require that leaders do not intersect and minimize the overall length of the leaders.

Assuming a fixed number of symbol positions around the map, it can happen that it is not possible to place symbols for all features contained in an operational area. In this case, a selection of features must be made. Addressing maps for disaster management, Wang et al. (2010) state that important emergency information should be visible to map readers at first glance. Thus, we suggest to assign each symbol a weight that expresses the operational relevance of its feature. If there is not enough space to place all tactical symbols, we follow a common strategy in map labeling (see e.g., Mauri et al. (2010); Löffler et al. (2015); Li et al. (2016)) and preferably place symbols with a high relevance. We point out that the relevance of tactical symbols can vary depending on the particular emergency. Assuming an interactive map design, the user can reveal symbols of currently unlabeled features by zooming further into the map. We note that it would also be an option to adapt the multi-page strategy from the previous chapters, i.e., to distribute all information across multiple pages. The user would then have the possibility to switch between the different pages to view all features. However, in an informal survey that we conducted early in the model development, domain experts indicated that they would prefer to stick with a zoom-and-pan strategy.

Another aspect worth considering concerns the placement of the symbols in relation to each other. In external labeling, grouping or clustering labels based on semantics (Bekos et al., 2019; Niedermann et al., 2017) as well as based on the spatial proximity of their features (Fink et al., 2012) are common design criteria. In this regard, we refer to the *Gestalt laws of grouping* introduced by Wertheimer (1938), which describe how humans perceive objects, simplify complex structures, and recognize patterns. According to the *law of proximity*, objects that are close to each other are perceived as one group. Applied to situation maps, we argue that grouping certain symbols, i.e., placing them next to each other, can facilitate a more focused overview of the emergency situation. More precisely, placing semantically similar symbols next to each other can help specialized units to quickly get an overview of all locations relevant to their operational expertise (e.g., storage locations of hazardous materials in the event of a fire). Further, grouping tactical symbols of features that are spatially close to each other can help to capture operational focal points more quickly. Thus, our model provides the ability to group symbols by placing either semantically or spatially similar ones next to each other. We note that the external placement of symbols allows the necessary flexibility to place the symbols in a grouped manner, i.e., placing them next to each other. A semantic grouping would be difficult to realize if the symbols were placed internally.

Summarizing, our model places tactical symbols such that crossing leaders are prohibited and the following optimization criteria are considered.

C_{prio} Symbols of highly relevant features should be placed with priority.

C_{len} Leaders should be short.

C_{group} Symbols of features with similar semantics or spatial proximity should be placed next to each other.

To confirm our literature-based decisions, we additionally conducted an informal survey with domain experts at an early stage of the model development. The surveys involved seven German experts in the fields of flood protection, emergency response, and cartography. We asked the experts to rate statements on different aspects and representations of situation maps. Generally, the assessments showed that the decisions we made were supported both from a cartographic perspective and in terms of use in disaster management. In particular, all experts agreed that it is possible to rank tactical symbols according to their importance during an emergency operation. Therefore, we consider it reasonable to give priority to particularly relevant symbols (C_{prio}). There was also unanimity on the importance of avoiding long leaders (C_{len}) and leader intersections to improve map clarity and readability. With regard to a grouped placement of symbols, there were different assessments whether symbols should be grouped according to their semantics or to the location of their map positions. Since arguments can be made for both strategies, we allow for both variants in our model (C_{group}). However, we emphasize that our criteria are primarily based on established design rules and research in the field of emergency response, and that we conducted the survey to identify potential deficiencies early in the process.

7.3.2 Optimization Problem

We assume that we are given a rectangular region $R \subset \mathbb{R}^2$ which we denote as *map*; see Figure 7.2.

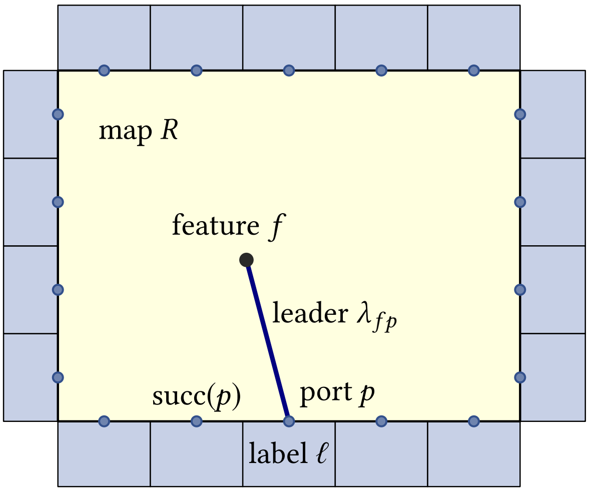


Fig. 7.2: Technical sketch of our labeling model.

The set F contains all point features (e.g., assembly points or locations of hazards) within the map. For convenience, we refer to a point feature $f \in F$ as a *feature*. Given a weighting function $w\colon F \to \mathbb{R}^+$, we assign each feature a weight $w(f)$ expressing its operational relevance. We group the features based on a set of different classes $\mathscr{G}$, which are established depending on either semantic similarity or spatial proximity. Each feature f is assigned to at least one class $g(f)$ using a function $g\colon F \to \mathscr{G}$ or we set $g(f) \neq \emptyset$ if f is not part of any group. For each feature, we introduce a rectangular *label* ℓ that describes the feature with a tactical symbol. We denote the set of all labels as L. Based on the established design of situation maps (see Section 7.1), we specify k fixed positions at all four boundaries of the map to which we attach the labels; see Figure 7.2. We call such a fixed position a *port* p and arrange the set P of ports in such a way that the same number of ports face each other and that the attached labels do not overlap. Starting from the upper left corner of the map, we order the ports clockwise and call the port that succeeds a port p its successor succ(p). To associate a label with its corresponding feature, we connect the port p to which the label is attached and the feature f with a leader λ_{fp}. The set Λ contains all leaders. Although we use leaders consisting of one straight (s) line element, i.e., s-leaders (Bekos et al., 2019), we note that other types of leaders could also be integrated into our framework. We define two leaders λ and λ' to be *crossing* if the interior $\mathscr{I}$ of both leaders intersect, i.e., $\mathscr{I}(\lambda) \cap \mathscr{I}(\lambda') \neq \emptyset$ holds.

We define a *labeling* as a mapping $\mathscr{L}\colon P \to F$ of each port on a feature and refer to a feature for which $f = \mathscr{L}(p)$ holds as *assigned* to port p. We further denote a port p with $\mathscr{L}(p) = \emptyset$ as *unused*. We aim for a labeling that preferentially places symbols of high-weighted features (C_{prio}), keeps the overall leader length short (C_{len}), and places symbols of semantically similar or spatially close features together (C_{group}).

Based on the formal model described above, we can define an *instance* as the tuple $\Upsilon = (F, P, w, g, \Lambda)$. A formal optimization problem is defined as follows.

Problem (EMERGENCYLABELING).
Given an instance $\Upsilon = (F, P, w, g, \Lambda)$ and the constants $\alpha, \beta, \gamma \in [0,1]$, find a labeling $\mathscr{L}$ such that no two leaders are crossing and that maximizes

$$\varphi(\mathscr{L}) = \alpha \cdot \varphi_{\text{prio}}(\mathscr{L}) - \beta \cdot \varphi_{\text{len}}(\mathscr{L}) + \gamma \cdot \varphi_{\text{group}}(\mathscr{L}) \tag{7.1}$$

with

$$\varphi_{\text{prio}}(\mathscr{L}) = \frac{1}{w_{\max}} \cdot \sum_{p \in P} w\big(\mathscr{L}(p)\big)\,, \tag{7.2}$$

$$\varphi_{\text{len}}(\mathscr{L}) = \frac{1}{\text{length}(\lambda_{\max})} \cdot \sum_{p \in P} \text{length}\big(\lambda_{\mathscr{L}(p),p}\big)\,, \tag{7.3}$$

$$\varphi_{\text{group}}(\mathscr{L}) = \sum_{p \in P} \text{score}\Big(\mathscr{L}\big(p\big), \mathscr{L}\big(\text{succ}(p)\big)\Big)\,, \tag{7.4}$$

where $\varphi_{\text{prio}}(\mathscr{L})$, $\varphi_{\text{len}}(\mathscr{L})$, and $\varphi_{\text{group}}(\mathscr{L})$ relate to criteria C_{prio}, C_{len} *and* C_{group}.

In EMERGENCYLABELING, the term $\text{length}\big(\lambda_{\mathscr{L}(p),p}\big)$ describes the length of the leader connecting a port p with its assigned feature $\mathscr{L}(p)$. With the term $\text{score}\Big(\mathscr{L}\big(p\big), \mathscr{L}\big(\text{succ}(p)\big)\Big)$ we define a reward if two neighboring ports p and $\text{succ}(p)$ are mapped to two features $\mathscr{L}\big(p\big)$ and $\mathscr{L}\big(\text{succ}(p)\big)$ with the same class (i.e., that are either semantically or spatially similar). We set $\text{score}\Big(\mathscr{L}\big(p\big), \mathscr{L}\big(\text{succ}(p)\big)\Big) = 1$. To bring all three parts of the objective to a similar order of magnitude, we scale the sum $\sum_{p \in P} w\big(\mathscr{L}(p)\big)$ with the maximum occurring weight $w_{\max}$ and the sum $\sum_{p \in P} \text{length}\big(\lambda_{\mathscr{L}(p),p}\big)$ with the length of the longest possible leader $\lambda_{\max}$. We note that we use pixels as the unit of leader lengths.

7.4 Mathematical Programming

To solve EMERGENCYLABELING optimally, we make use of combinatorial optimization and express the optimization problem as an integer linear program (ILP). For each feature $f \in F$ and each port $p \in P$ we introduce a binary variable

$$x_{fp} = \begin{cases} 1 & \text{if } f \text{ is assigned to } p, \\ 0 & \text{otherwise}\,. \end{cases} \tag{7.5}$$

To obtain a feasible labeling, we prevent a feature f from being labeled multiple times by requiring that it can be assigned to at most one port p with

$$\sum_{p \in P} x_{fp} \leq 1 \qquad \forall f \in F. \tag{7.6}$$

Further, we enforce that each port p must be mapped on exactly one feature f with the constraint

$$\sum_{f \in F} x_{fp} = 1 \qquad \forall p \in P. \tag{7.7}$$

If the number k of ports exceeds the number n of features and hence Constraint (7.7) is not feasible, we introduce a set $\mathscr{T}$ of $k-n$ additional *dummy* features. Since we do not want dummies to be assigned to ports instead of other features, we set the weight $w(\tau)$ of each dummy $\tau \in \mathscr{T}$ to 0. Further, we set the length $\lambda_{\tau p}$ of each dummy to a constant that is larger than the longest possible real leader $\lambda_{\max}$ and the class to $g(\tau) = \emptyset$.

To ensure an unambiguous association between a feature f and its label ℓ, we do not allow crossing leaders in EMERGENCYLABELING. We prevent leaders from crossing with the constraint

$$x_{fp} + x_{f'p'} \leq 1 \qquad \begin{array}{l} \forall f, f' \in F,\ \ p, p' \in P \quad \text{with} \\ f \neq f', p \neq p', \mathscr{I}(\lambda_{fp}) \cap \mathscr{I}(\lambda_{f'p'}) \neq \emptyset. \end{array} \tag{7.8}$$

To reward the case that two features f and f' with the same class $g(f) = g(f')$ are assigned to two neighboring ports p and $\operatorname{succ}(p)$ (see Criterion $\mathrm{C}_{\text{group}}$), we introduce another binary variable

$$z_p = \begin{cases} 1 & \text{if } g\Big(\mathscr{L}(p)\Big) = g\Big(\mathscr{L}(\operatorname{succ}(p))\Big)\,, \\ 0 & \text{otherwise}\,, \end{cases}$$

for each $p \in P$. We define $F_{\text{group}} \subseteq F$ to be the set that contains any feature f to which a class is assigned, i.e., $g(f) \neq \emptyset$. We enforce that $z_p = 0$ holds if two succeeding ports p and $\operatorname{succ}(p)$ are mapped to two features f and f' with different classes $g(f) \neq g(f')$ with

$$z_p + x_{fp} + x_{f'\operatorname{succ}(p)} \leq 2 \qquad \begin{array}{l} \forall p \in P,\ \ f, f' \in F_{\text{group}} \quad \text{with} \\ f \neq f', g(f) \neq g(f'). \end{array} \tag{7.9}$$

Further, $z_p = 0$ must hold if p or $\operatorname{succ}(p)$ is mapped to a feature that is not assigned to any class:

$$z_p \leq 1 - \sum_{f \in F \setminus F_{\text{group}}} x_{fp} \qquad \forall p \in P, \tag{7.10}$$

$$z_p \leq 1 - \sum_{f \in F \setminus F_{\text{group}}} x_{f\operatorname{succ}(p)} \qquad \forall p \in P. \tag{7.11}$$

Constraints (7.9)–(7.11) ensure that $z_p = 1$ can only hold, if p and $\operatorname{succ}(p)$ are mapped to features of the same class.

Satisfying all Constraints (7.6)–(7.11), we maximize

$$\frac{\alpha}{w_{\max}} \sum_{p \in P} \sum_{f \in F} w(f) \cdot x_{fp} - \frac{\beta}{l(\lambda_{\max})} \sum_{p \in P} \sum_{f \in F} l(\lambda_{fp}) \cdot x_{fp} + \gamma \sum_{p \in P} z_p\,, \tag{7.12}$$

where α, β, and γ balance the three objective parts, $w_{\max}$ is the maximum feature weight, $l(\lambda_{\max})$ is the length of the longest leader, and $l(\lambda_{fp})$ is the length of a leader λ_{fp}.

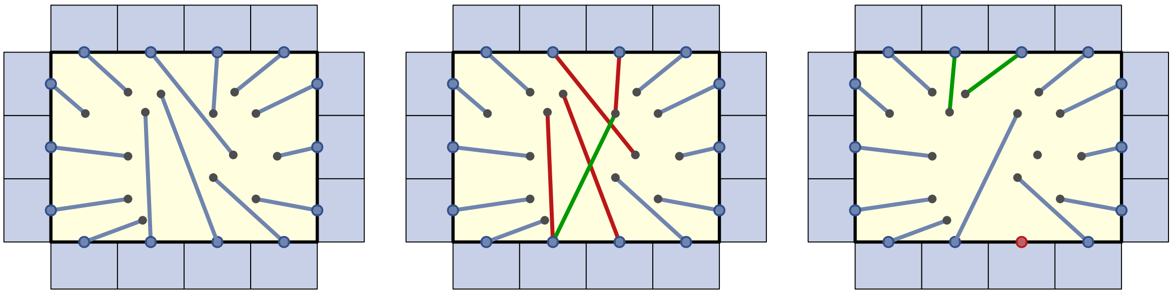
To achieve mathematically optimal solutions to our ILP formulation, we apply a specialized solver in our experiments.

7.5 Heuristic Approach

For interactive applications, the on-the-fly capability of the underlying algorithms is a crucial aspect. It becomes even more important when, as in our case, the interface is intended for time-critical tasks. Since with the automatic symbol placement in situation maps we address one part of such interfaces, we also have to consider the speed of our algorithms. While there exist solvers for integer linear programming that guarantee an optimal solution, computing the solution requires exponential time in the worst case. Thus, for solving EMERGENCYLABELING, we additionally introduce a fast heuristic that is based on *simulated annealing*.

Before starting the iterative process, we need to define an initial solution. Similar to Fink et al. (2012), we transform the problem of finding a labeling into finding a maximum weighted matching in a bipartite graph. We introduce a bipartite graph $G = (V_F \cup V_P, E)$ in which V_F contains all features F and V_P contains all ports P. We connect each pair of a vertex $v_f \in V_F$ and a vertex $v_p \in V_P$ with an edge $e \in E$. Hence, each edge corresponds to a potential leader. We assign each edge e a weight that we set to $l(\lambda_{\max}) - l(\lambda_e)$, i.e., the difference in the lengths of the longest possible leader and the leader of e. To obtain an initial solution, we compute a maximum weighted matching in G using the approach described by Mehlhorn et al. (1997). Since we maximize the sum of the length differences between the longest possible leader and the edges' leaders, the matching corresponds to a labeling that is optimal with respect to Criterion C_{len}. We note that the labeling is inherently free of crossing leaders.

Starting from this initial solution, a neighboring solution is considered in each iteration. To define the neighborhood, we introduce a set $\mathscr{C}$ of candidates which contains a candidate $c_{fp} \in \mathscr{C}$ for each possible assignment of a feature f to a port p. We split $\mathscr{C}$ into two disjoint sets $\mathscr{C}_{\text{act}} \cup \mathscr{C}_{\text{inact}} = \mathscr{C}$. The set $\mathscr{C}_{\text{act}}$ contains all candidates that are selected in a current solution, i.e., candidates that are *active*. Accordingly, the set $\mathscr{C}_{\text{inact}}$ contains all *inactive* candidates. In each iteration, we perform a *swap* of one randomly picked inactive candidate with all conflicting active candidates. In a subsequent fill-up step, we attempt to map unused ports to features by moving non-conflicting inactive candidates to the set of active candidates. An example of a swap is shown in Figure 7.3 and an algorithmic description is given in Algorithm 4.



(a) solution before swap **(b)** swap of candidates **(c)** solution after swap

Fig. 7.3: Example of a swap. The solution before the swap (active candidates) is shown in (a). In (b), the green line depicts the new candidate to be inserted into the set of active candidates. The red lines conflict with this candidate and hence the corresponding candidates are moved to the set of inactive candidates. In (c), the solution after the swap is shown. The green lines show the candidates that were inserted in the fill-up step. The red dot represents a port that is not mapped on any feature after the swap.

After a swap has been performed, one or more ports may no longer be mapped to features (see Figure 7.3c). Since we want to avoid unused ports, we add a penalty θ to the Objective 7.1 for each unused port. The penalty should be defined such that it is low enough to allow intermediate

Algorithm 4: Performing a swap.

```
Input  : set $\mathscr{C}_{\text{act}}$, set $\mathscr{C}_{\text{inact}}$
Output: updated set $\mathscr{C}'_{\text{act}}$, updated set $\mathscr{C}'_{\text{inact}}$
randomly pick $c_{fp} \in \mathscr{C}_{\text{inact}}$
delete $c_{fp}$ from $\mathscr{C}_{\text{inact}}$
for each $c_{f'p'} \in \mathscr{C}_{\text{act}}$ with $f = f'$ or $p = p'$ or $\mathscr{I}(\lambda_{fp}) \cap \mathscr{I}(\lambda_{f'p'}) \neq \emptyset$ do
    delete $c_{f'p'}$ from $\mathscr{C}_{\text{act}}$
    add $c_{f'p'}$ to $\mathscr{C}_{\text{inact}}$
end
add $c_{fp}$ to $\mathscr{C}_{\text{act}}$
for each $c_{f''p''} \in \mathscr{C}_{\text{inact}}$ do
    if there is no $c_{f'p'} \in \mathscr{C}_{\text{act}}$ with $f'' = f'$ or $p'' = p'$ or $\mathscr{I}(\lambda_{f''p''}) \cap \mathscr{I}(\lambda_{f'p'}) \neq \emptyset$ then
        delete $c_{f''p''}$ from $\mathscr{C}_{\text{inact}}$
        add $c_{f''p''}$ to $\mathscr{C}_{\text{act}}$
    end
end
$\mathscr{C}'_{\text{act}} \leftarrow \mathscr{C}_{\text{act}}$, $\mathscr{C}'_{\text{inact}} \leftarrow \mathscr{C}_{\text{inact}}$
```

solutions with unused ports in early iterations, but high enough so that all ports are mapped to features at the end of the algorithm. With the set $P_{\mathscr{E}}$ containing all unused ports, we get the updated objective

$$\varphi(\mathscr{L})' = \varphi(\mathscr{L}) - |P_{\mathscr{E}}| \cdot \theta. \tag{7.13}$$

In simulated annealing, the probability of accepting a lower-quality solution in the i-th iteration depends on a certain temperature T_i. Given a starting temperature T_0, a final temperature T_{E}, and the total number N of iterations, we define

$$T_i = T_{i-1} \cdot \sqrt[N]{T_{\text{E}}/T_0}\,. \tag{7.14}$$

Following Kirkpatrick et al. (1983), the probability P with which a new solution is accepted is

$$P = \begin{cases} 1 & \text{if } \Delta c \geq 0\,, \\ \exp(\Delta c/T_i) & \text{if } \Delta c < 0\,, \end{cases} \tag{7.15}$$

with $\Delta c = \varphi(\mathscr{L}_i) - \varphi(\mathscr{L}_{i-1})$ being the difference in the objective function between the current and the previous solution.

7.6 Experiments

In the following, we first describe the experimental setup and then explain and evaluate the individual experiments carried out. In particular, we assess our model by visually comparing results of our exact approach and determining a suitable balance between the considered optimization criteria. Subsequently, we evaluate the quality of the heuristic results, elaborate on running times, and discuss an expert's assessment of situation maps generated with our heuristic.

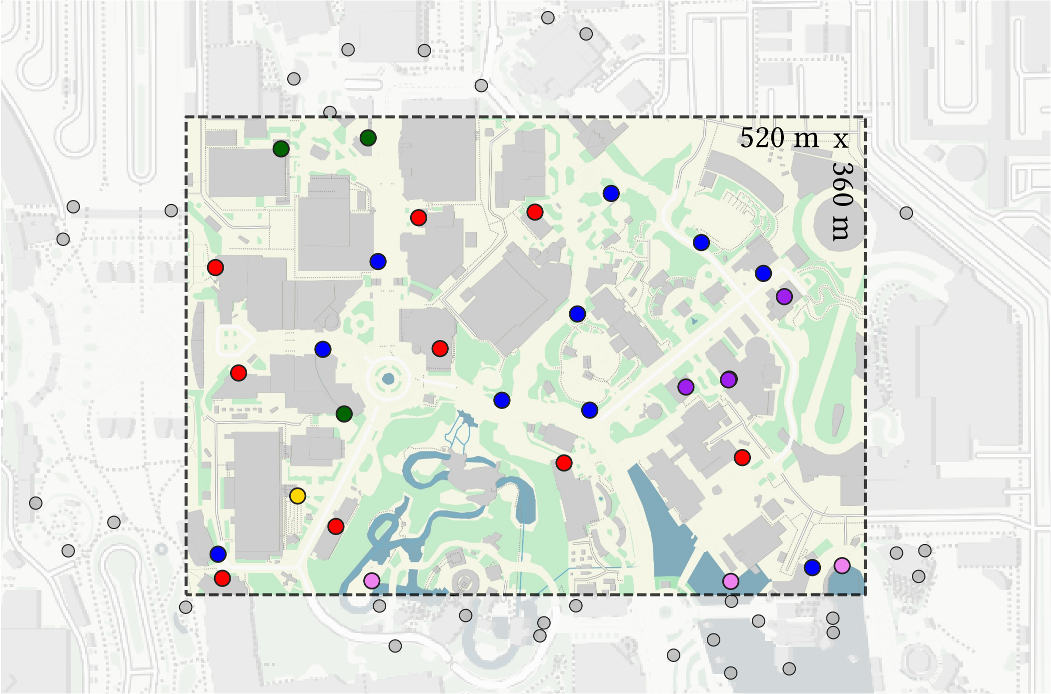


Fig. 7.4: Example of an extracted input instance. The colors of the point features represent different semantic classes. The gray dots indicate other features that are not within the frame. *Map attribution: Stamen.*

7.6.1 Experimental Setup

Adapting the map layout used by platoon leaders in Germany, we define the map to have the aspect ratio of an A4 format, i.e., $1 : \sqrt{2}$ (Lamers and Denker, 2022). To provide enough space for placing tactical symbols, but at the same time to not overload the map, we use a total of 20 ports. We place six ports on each of the longer sides of the map and four ports on the shorter sides (see, e.g., Figure 7.1). We obtain our experimental data from ©OpenStreetMap[1]. To retrieve data that is realistic to be annotated in an emergency setting, we query point features with the key attribute *emergency*. We restrict the query to the area of the *Disneyland Resort* in Los Angeles, which contains a total of 216 matching features. The query results in six different emergency classes $\mathcal{G} = \{\text{fire hydrant, fire extinguisher, fire hose, exit, life ring, defibrillator}\}$. We assign each feature f its corresponding semantic class $g(f) \in \mathcal{G}$.

To assess our model in terms of the interplay of the three optimization criteria and quantitatively evaluate the quality of our heuristic, we extract smaller instances intended to match the spatial scale of realistic emergency scenarios. We move a frame of 360 by 520 meters in a grid-wise manner over the data so that each frame overlaps 50 percent with the previous one. For each position of the frame, we extract one instance by sampling all features contained in that frame. An example of such an instance is shown in Figure 7.4, which we will use as a running example in the following. In total, we obtain 30 instances containing a number of 22 to 30 features.

Simulating realistic emergency scenarios, we assume the event of an outbreak of fire. We define the weight $w(f)$ of each feature f subject to its semantic class $g(f)$ such that it reflects the tactical relevance in the given scenario (see Table 7.1). In addition to the semantic classes $\mathcal{G}$, we introduce proximity-based classes $\mathcal{G}'$ for each instance. Aiming to delineate mission focal points, we set a threshold of $\epsilon = 25$ meters and assign two features f and f' to the same proximity group

[1] https://www.openstreetmap.org

$g'(f) = g'(f')$ if their distance does not exceed ϵ. We note that our specifications are based on own considerations and that other definitions are possible.

Tab. 7.1: Specified feature weights depending on the emergency classes.

	fire hydrant	fire extinguisher	fire hose	exit	life ring	defibrillator
weight	$w = 3$	$w = 3$	$w = 3$	$w = 2$	$w = 1$	$w = 1$

Considering the quantitative assessment of our experiments, we compare two labelings $\mathcal{L}$ and $\mathcal{L}'$ with respect to either $\mathrm{C}_{\mathrm{prio}}$, $\mathrm{C}_{\mathrm{len}}$, or $\mathrm{C}_{\mathrm{group}}$. We define the *relative objectives*

$$\delta_{\mathrm{prio}}(\mathcal{L}, \mathcal{L}') = \frac{\varphi_{\mathrm{prio}}(\mathcal{L}) - \varphi_{\mathrm{prio}}(\mathcal{L}')}{\varphi_{\mathrm{prio}}(\mathcal{L}')} \cdot 100\%, \tag{7.16}$$

$$\delta_{\mathrm{len}}(\mathcal{L}, \mathcal{L}') = -\frac{\varphi_{\mathrm{len}}(\mathcal{L}) - \varphi_{\mathrm{len}}(\mathcal{L}')}{\varphi_{\mathrm{len}}(\mathcal{L}')} \cdot 100\%, \tag{7.17}$$

$$\delta_{\mathrm{group}}(\mathcal{L}, \mathcal{L}') = \frac{\varphi_{\mathrm{group}}(\mathcal{L}) - \varphi_{\mathrm{group}}(\mathcal{L}')}{\varphi_{\mathrm{group}}(\mathcal{L}')} \cdot 100\%. \tag{7.18}$$

The relative objectives express by how many percent the quality of φ_{prio}, φ_{len}, and φ_{group} of a labeling $\mathcal{L}$ deviates from that of another labeling $\mathcal{L}'$. According to this definition, a positive value indicates a better quality and a negative value indicates a lower quality for $\mathcal{L}$.

We implemented both our ILP formulation and our heuristic in Java and used Gurobi[2] 9.1.1 to solve the ILP. All computations were performed using a server system with an AMD Ryzen™ Threadripper™ 3970X clocked at 3.7GHz and 128 GB of ECC memory.

7.6.2 Evaluation of Model

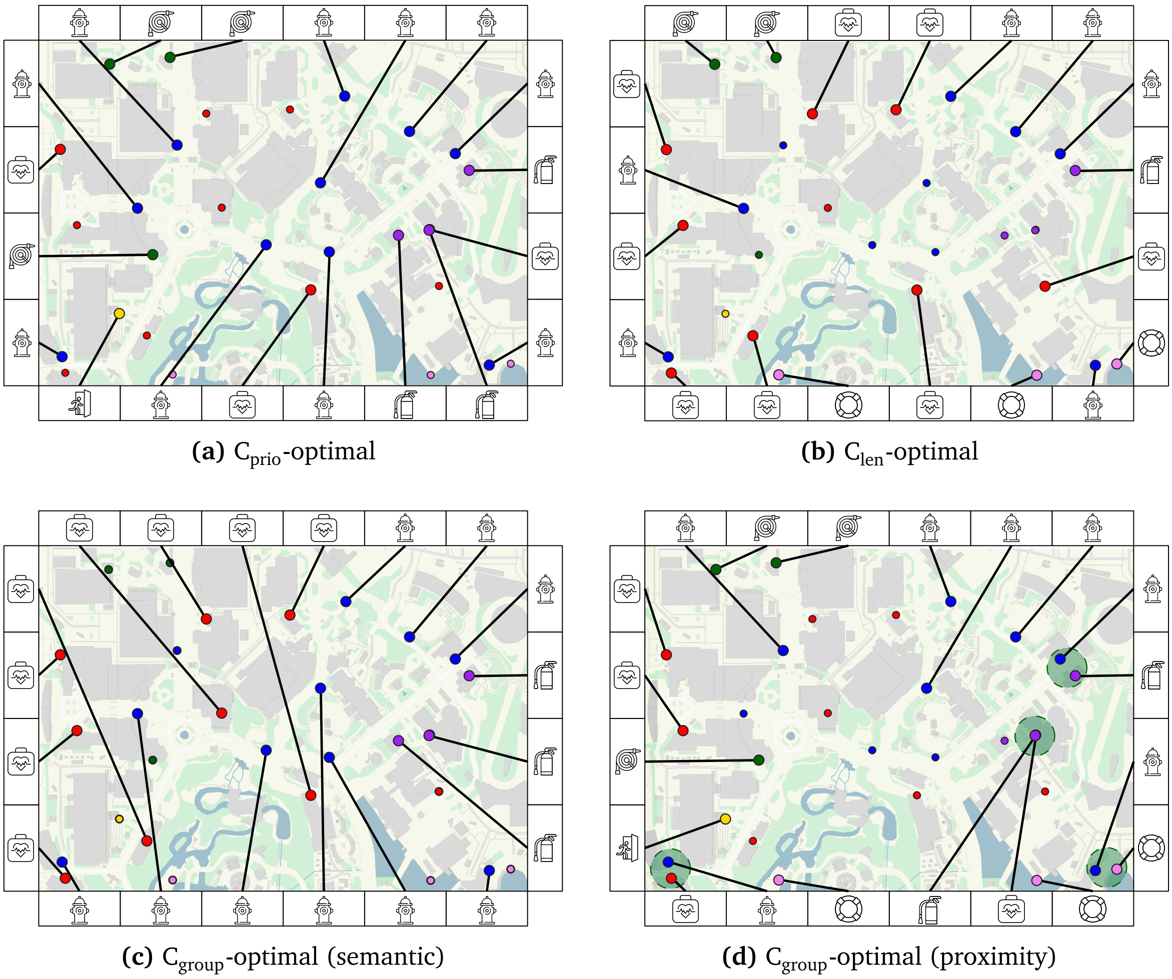
The optimization function in EMERGENCYLABELING is composed of the three objectives φ_{prio}, φ_{len}, and φ_{group}. The three parameters α, β, and γ can be used to control how strongly a respective objective is taken into account in the optimization.

Visual Assessment

For the exemplary instance in Figure 7.4, optimal results gained with our ILP are shown in Figure 7.5. In each of the results, only one of the criteria was optimized, i.e., a result is either optimal with respect to $\mathrm{C}_{\mathrm{prio}}$, $\mathrm{C}_{\mathrm{len}}$, or $\mathrm{C}_{\mathrm{group}}$. We observe that if only feature weights are optimized (see Figure 7.5a), a symbol has been placed for all occurring features with the highest weight (fire extinguisher, hydrant, and fire hose). Only less relevant features are not labeled with a symbol. However, taking the most relevant features into account results in long leaders that belong to features in the center of the map. If instead only the length of the leaders is considered (see Figure 7.5b), more features that lie closer to the boundaries of the map are labeled, resulting in shorter leaders.

As a third criterion, we introduced the grouping of symbols, i.e., placing similar symbols next to each other. With respect to this criterion, Figure 7.5c and Figure 7.5d represent optimal solutions in which grouping is based on semantic classes and proximity-based classes, respectively.

[2] https://gurobi.com



(a) C_{prio}-optimal

(b) C_{len}-optimal

(c) C_{group}-optimal (semantic)

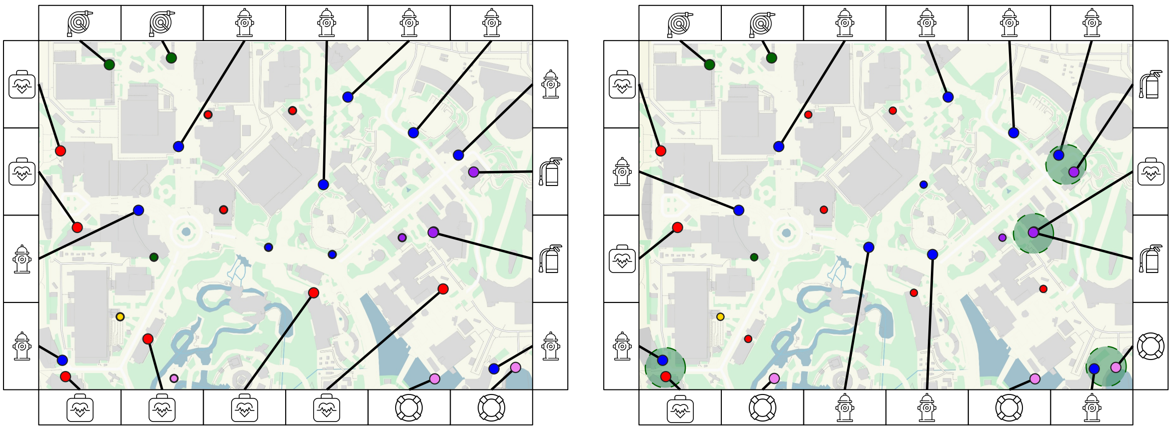
(d) C_{group}-optimal (proximity)

Fig. 7.5: Results of the exact approach for the exemplary instance in Figure 7.4. Each map shows a result in which only one criterion is optimized. The symbols and colors of the point features represent the different semantic classes. In (d), the green circles indicate the proximity-based classes. *Map attribution: Stamen.*

When optimizing a semantic-based grouping, four sequences of consecutive identical symbols were generated. Symbols for hydrants, fire extinguishers, and defibrillators are placed next to each other. Conversely, if the grouping is based on the spatial proximity of the features, the semantic classes are not taken into account and all symbols whose features belong to one proximity class are placed next to each other. Due to the distribution of the features over the complete map area, there are only four proximity classes with two features each (see green circles in Figure 7.5d).

Balancing the Objective

The evaluation of the visual results of our exact approach shows how the individual criteria affect the placement of symbols. However, we conclude that an interplay of multiple criteria could be useful to combine the individual benefits. Thus, we aim to find a suitable sweet spot between all three optimization criteria. In other words, we try to find a balance between α, β, and γ such that all three criteria achieve a satisfactory quality. To that end, we set $\alpha + \beta + \gamma = 1$ and apply our exact approach with different value combinations. We vary α, β, and γ with a step width of 0.05 within a range of $[0, 1]$ and compute the optimal solution for each possible combination



(a) balanced solution with semantic grouping **(b)** balanced solution with proximity grouping

Fig. 7.6: Results of the exact approach for the exemplary instance in Figure 7.4 with (a) the identified sweet spot for semantic grouping and (b) the sweet spot for proximity-based grouping. *Map attribution: Stamen.*

and each of the 30 instances. We denote a labeling obtained with a specific value combination by $\mathscr{L}_{\alpha\beta\gamma}$.

For each labeling $\mathscr{L}_{\alpha\beta\gamma}$, we investigate to what extent the achieved quality for φ_{prio}, φ_{len}, and φ_{group} deviates from the one of the respective best possible solution. For this purpose, we determine the relative objectives $\delta_{\text{prio}}(\mathscr{L}_{\alpha\beta\gamma}, \mathscr{L}_{\text{opt}})$, $\delta_{\text{len}}(\mathscr{L}_{\alpha\beta\gamma}, \mathscr{L}_{\text{opt}})$, and $\delta_{\text{group}}(\mathscr{L}_{\alpha\beta\gamma}, \mathscr{L}_{\text{opt}})$, where $\mathscr{L}_{\text{opt}}$ describes the labeling that achieves the optimal solution for the respective objective. Thus, for a labeling $\mathscr{L}_{\alpha\beta\gamma}$, the relative objectives describe the percentage deviation in the quality of a criterion compared to the best possible quality.

To find a suitable sweet spot between the three criteria, we average the relative objectives over all 30 instances. We search for the combination of α, β, and γ for which the lowest of the three relative objectives is the highest among all combinations. In other words, we aim to find the combination of α, β, and γ that maximizes the lowest quality of φ_{prio}, φ_{len}, and φ_{group}. For the case of a semantic grouping, we obtain such a sweet spot with $\alpha = 0.05, \beta = 0.8$, and $\gamma = 0.15$. With this combination, we achieve a reduction in quality of only 10.7%, 15.0%, and 17.0% in terms of φ_{prio}, φ_{len}, and φ_{group}, respectively. Further, when we group symbols according to the spatial proximity of their features, we obtain a slightly different sweet spot with $\alpha = 0.1, \beta = 0.7$, and $\gamma = 0.2$. Here, the reduction in quality is even less with 7.8%, 7.4%, and 4.2%. These higher qualities compared to the case of a semantic grouping can be justified by the fact that the instances contain only a few features building proximity groups. This allows a high flexibility of the symbol placement such that high qualities for φ_{prio} and φ_{len} can be achieved without compromising φ_{group}.

To provide a visual impression, Figure 7.6 shows the results for the exemplary instance (see Figure 7.4) for the two sweet spots identified. For the case of semantic grouping (see Figure 7.6a), we observe that most of the highest weighted features were labeled and, at the same time, many symbols of the same semantic classes were placed next to each other. A similar structure can be seen in the case of proximity-based grouping in Figure 7.6b. All symbols belonging to proximity groups were placed next to each other and at the same time the most relevant features were labeled. In both maps, shorter leaders are evident compared to Figure 7.5a and Figure 7.5c.

7.6.3 Assessing the Quality of the Heuristic

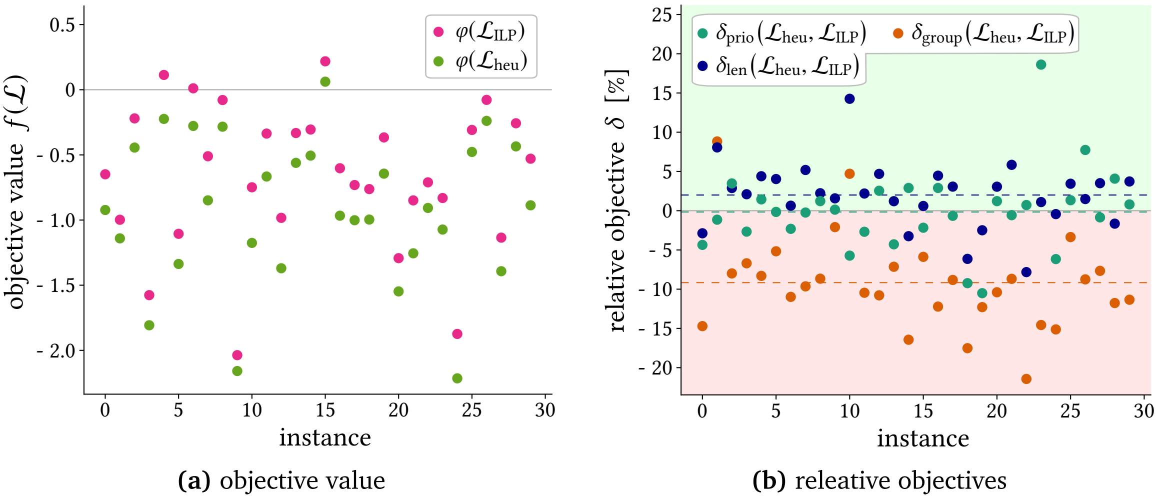
To apply our heuristic, it is necessary to specify initial parameters (see Section 7.5). We have determined these parameters experimentally and summarize them in an annealing schedule in Table 7.2. With $\theta = 1$, we set the penalty for an unused port equal to the reward for assigning a feature f with the highest possible weight to a port, i.e., $^{w(f)}/_{w_{\max}} = 1$.

Tab. 7.2: Annealing schedule that specifies parameters used in the heuristic.

starting temperature	$T_0 = 100$	final temperature	$T_E = 10^{-6}$
number of iterations	$N = 8000$	penalty for unused ports	$\theta = 1$

To asses the quality of labelings gained using our heuristic approach, we compare them to optimal labelings computed with the exact approach. We stick to the case of a semantic grouping and use the sweet spot at $\alpha = 0.05, \beta = 0.8$, and $\gamma = 0.15$ that we determined in the previous Section 7.6.2. We apply both our exact and our heuristic approach to all 30 instances and average the heuristic results over 50 runs for each instance. We denote a labeling gained with the heuristic by $\mathscr{L}_{\text{heu}}$ and a labeling gained with the exact approach by $\mathscr{L}_{\text{ILP}}$.

For each instance, Figure 7.7a shows the objective values $\varphi(\mathscr{L}_{\text{heu}})$ and $\varphi(\mathscr{L}_{\text{ILP}})$. We observe that the values of the objective achieved by the exact approach are higher, and thus of higher quality than those achieved by the heuristic. This was to be expected, since the exact approach yields optimal solutions. For a more detailed assessment, we evaluate the three optimization criteria independently. More precisely, we consider the relative objectives $\delta_{\text{prio}}(\mathscr{L}_{\text{heu}}, \mathscr{L}_{\text{ILP}})$, $\delta_{\text{len}}(\mathscr{L}_{\text{heu}}, \mathscr{L}_{\text{ILP}})$, and $\delta_{\text{group}}(\mathscr{L}_{\text{heu}}, \mathscr{L}_{\text{ILP}})$, which express the percentage deviation of the quality of φ_{prio}, φ_{len}, and φ_{group} obtained with the heuristic to that obtained with the exact ILP.



(a) objective value

(b) releative objectives

Fig. 7.7: Comparison of heuristic and optimal results gained by using 30 instances with the identified sweet spot for semantic grouping. The dashed lines in (b) indicate average values.

Figure 7.7b shows these relative objectives for each instance. We observe that the heuristic produces solutions for which the quality of φ_{prio} is, on average, approximately equal to the quality of the optimal labelings (−0.1%). The heuristic even achieves at most a 18.7% higher and at least only a 10.5% lower quality of φ_{prio}. When further considering φ_{len}, we can see that the average quality of the heuristic results is 2.0% higher compared to the ones gained with the exact approach. In other words, when applying the heuristic, in most cases we obtain labelings with shorter leaders than when using the exact approach. The quality ranges from a 7.9% lower to

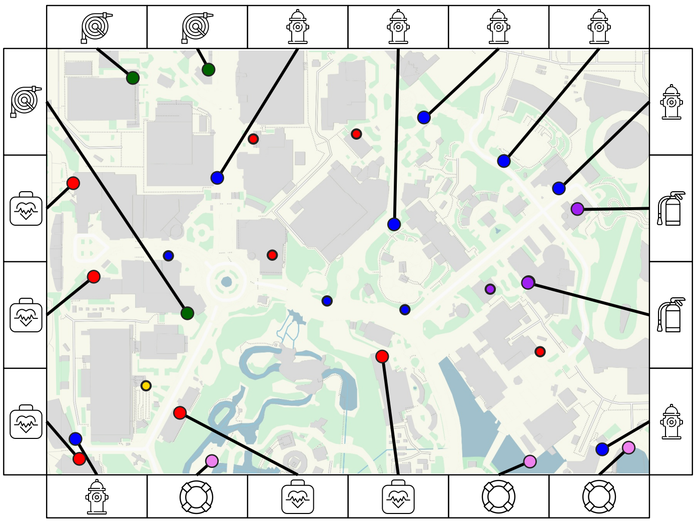


Fig. 7.8: Exemplary result of the heuristic for the instance in Figure 7.4 with the identified sweet spot for semantic grouping. *Map attribution: Stamen.*

a 14.1% higher quality. Considering φ_{group}, we find that the resulting quality of the heuristic is only 9.2% lower on average. At maximum, the quality is 21.4% lower.

To better place these numerical results in the visual context, Figure 7.8 shows one solution of the example instance (see Figure 7.4) generated with the heuristic. Compared to the optimal result in Figure 7.6a, we observe an equal quality with regard to φ_{prio} ($\delta_{\text{prio}}(\mathcal{L}_{\text{heu}}, \mathcal{L}_{\text{ILP}}) = 0.0\%$). In other words, the sum of the weights of the placed symbols in the heuristic solution is equal to that in the optimal solution. While the quality of φ_{len} is even slightly higher in the heuristic solution ($\delta_{\text{len}}(\mathcal{L}_{\text{heu}}, \mathcal{L}_{\text{ILP}}) = 2.5\%$), that of φ_{group} is slightly lower ($\delta_{\text{group}}(\mathcal{L}_{\text{heu}}, \mathcal{L}_{\text{ILP}}) = -8.3\%$). Thus, the overall length of the leaders is shorter, but one less pair of symbols sharing the same semantic class has been placed next to each other than in the optimal solution.

In summary, we argue that the heuristic produces results of high quality. On average, the quality of φ_{prio} and φ_{len} is not lower than that of the optimal results and in many cases even higher. Only for the grouping criterion, a loss of quality can be observed. However, even for φ_{group}, the quality still reaches over 90% of the optimal one on average.

7.6.4 Running Time

Running the exact approach on our server system took between five seconds and ten minutes for the tested instances. Considering an interactive map design, each time a user zooms or pans the map, they would need to wait up to ten minutes to get a new visualization. Especially in the development of interactive applications, it is known that response times of more than one second cause users to perceive the response as sluggish and to lose the feeling of a smooth interaction (Card et al., 1991; Nielsen, 1993).

Unlike the exact approach, our heuristic generated a solution in under a second for each of the 30 instances. More precisely, we measured running times between 0.44 and 0.82 seconds. Assuming client-server communication, this means that the heuristic can generate maps on demand and is suitable even for interactive applications. However, a stable connection to a server cannot always be guaranteed, e.g., in emergencies in tunnels or in areas that are far away from populated areas. To account for this case, where situation maps need to be computed directly on mobile devices, we additionally ran our heuristic on a standard tablet, i.e., a Samsung Galaxy Tab A T515 from 2019. Generating maps took between 3.8 and 5.7 seconds on the tablet. Although these running times are above the threshold of one second, we state that even without a server connection, maps can be created in a few moments without the need for special software or large processing capacities.

7.6.5 An Expert's Assessment

So far, we evaluated our approaches visually and quantitatively. To further find out whether our situation maps are suitable for real scenarios, we asked an expert in German emergency management to provide us with point data from an emergency operation. We used this data as input for our heuristic and ran it both for the case of a semantic grouping and for a proximity-based grouping with the balancing factors set to the respective sweet spots determined in Section 7.6.2. To visualize the two resulting situation maps, we adopted tactical symbols as defined by the guidelines of the German Federal Agency for Technical Relief (THW). The resulting map with the proximity-based grouping is shown in Figure 7.1.

In an online interview, we asked the expert to comment on the visualizations with respect to the placement of the symbols. Overall, he considered the arrangement of the symbols to be appropriate in both maps. He stated that the maps looks clear and it is easy to associate the symbols to their features. However, he perceived the grouping of symbols by semantics as rather unusual. In his opinion, the grouping according to the location of the symbols' features is more sensible, as it allows for a better differentiation of operational focal points. Especially when the map features lie exactly on top of each other, he argued that with such grouping, leaders could be combined to reduce clutter and map coverage. We consider this to be an interesting aspect that can be implemented in continuing work. The expert emphasized that the distance threshold, which indicates whether two features are grouped into one proximity group, depends strongly on the particular scene and the addressed operational level. For this reason, setting the threshold should be flexible and done by the user. Further, the expert suggested to combine both grouping by semantics and by location.

Commenting more generally on automating the placement of symbols in situation maps, the expert emphasized its value concerning all operational levels in emergency response. For the emergency forces on site as well as for coordinating forces at a higher level, the automatic placement of symbols reduces the workload and thus constitutes a relief for the personnel. The expert stated that it also enables situation maps to be automatically generalized with respect to the different requirements of operation levels. For example, while it is relevant for on-scene responders that each fire engine is labeled, for higher operational levels they could be automatically grouped into one fire brigade. We consider the automatic generalization of situation maps to be another interesting topic for continuing research.

7.7 Conclusion

Within the context of interactive map interfaces, this chapter has highlighted a particular use-case: presenting operational information to emergency responders. By automating the process of placing symbols in such map interfaces, we have contributed to saving valuable time during emergency operations and to reduce the workload of the staff. Accounting for both cartographic aspects and the actual demands in emergency response, we have derived constrains and optimization criteria from existing literature which have been confirmed by experts in a preliminary survey. We have introduced a multi-objective optimization problem for which we have provided both an exact approach and a fast heuristic. Our experiments based on real-world data led to the following key findings.

- We have determined a suitable balance between all considered optimization criteria.
- We have shown that the heuristic yields high-quality labelings in short time.
- An expert confirmed that the automation of the labeling process is of high value and that our labelings are suitable for use in practice.

We see our main contribution in the heuristic approach that achieves high-quality solutions with respect to the chosen objectives. Assuming client-server communication, it provides labelings in less than a second, which makes it suitable for interactive applications. Even in scenarios where a stable server connection can not be guaranteed, our heuristic generates situation maps within a few seconds when executed on a standard tablet. When realizing our approach as part of an interactive application, we do not consider the labelings generated by our heuristic as a final and immutable solution. Depending on the requirements of the emergency and the operator, the solution should be adaptable, allowing for manual adjustment of individual symbols as needed.

For future development, we believe that adapting our model in terms of combining semantic and proximity-based grouping of symbols is worth considering. Instead of grouping by either semantics or proximity, both variants could be merged. Further, with regard to the clarity of the presentation, different variants of leaders could be integrated and compared. In case features are located at the same map position, leaders could be combined to one line connected to the corresponding group of symbols. Concerning the use of situation maps in different levels of action, we think that automatically adapting and generalizing these maps is another promising prospect for future research. Additionally, we regard field tests of our methods under real conditions as an important step towards user-centered design. In an emergency simulation, relief forces on site could assess the actual usability and provide recommendations for possible improvements.

Summarizing, we think that we have made a contribution to increasing efficiency in emergency response and gave a thought-provoking impulse for further research.

8 CONCLUSION AND FUTURE PROSPECTS

This thesis has addressed interfaces for the presentation of large and detailed geographic data sets on mobile devices. In terms of a thorough exploration of the information in a given area of interest, we have emphasized limitations of established zoom-and-pan strategies. Most notably, we have stressed the issue that map context is lost when zooming in to a very large scale – which is often necessary to reveal map features when the information density is high. Following the assumption that progress beyond the common zoom-and-pan strategies can be achieved by shifting the focus to other specialized interaction techniques, flexible models and algorithms have been presented that are tailored towards a new type of map interface: *zoomless maps*. As part of the investigation into the usability and utility of these introduced maps, the key objective of this thesis was to validate the following central hypothesis.

Central Hypothesis. *Organizing the content of a dense map on a small screen into manageable and easily accessible pieces improves the interaction with the map in such a way that users find the information they search for at a smaller scale and, thus, zooming to a larger scale can be avoided.*

8.1 Assessment of the Central Hypothesis

To be able to verify our central hypothesis, we have conducted an extensive empirical user study in which participants tested our concepts implemented as interactive interfaces (see Chapter 6). Based on a ranking task as part of a visual search, we have compared three of our zoomless concepts with one established zoom-and-pan strategy with respect to different metrics that we captured during task processing.

We have seen that using our novel interfaces has a positive effect on the accuracy with respect to the specified search criteria. Furthermore, we were able to determine that significantly less zooming and panning was used, and that objects were selected with more map area being visible. With respect to the central hypothesis, these main findings provide initial support for its validity. Using our specialized interaction capabilities, users were able to find information that better matched the search criteria than with the baseline strategy. From this we deduce that our novel interfaces allow easy access to information, which is in line with the basic assumption of the hypothesis. For our tested interfaces, we were able to show that users rely less on zooming and find suitable information at a smaller scale, i.e., with more map context being visible. Thus, for the specific conditions tested, the outcomes of the study suggest that the central hypothesis holds true.

However, it is important to emphasize that the findings obtained should be treated with caution in terms of their general validity. While the observed merits are promising, these results may not be universally applicable to all variations of map interfaces. Essentially, we identify three key aspects that constrain the generalizability of the results of the user study:

(1) the limitation to one specific user task

(2) the use of a desktop environment

(3) the design choices made

To assess the usability and utility of zoomless maps, we have considered one specific user task, i.e., a ranking task within a visual search. The extent to which the observed results can be transferred to other map tasks remains to be investigated. Apart from confining to a singular type of task, our implementations simulated mobile use in a desktop environment. Compared to actual mobile use, this setting differs both in the implementation of the interaction capabilities and in the absence of external influences. Mouse-based interactions of the desktop environment would be implemented touch-based on mobile devices, which can have an impact on user behavior. Under real-world conditions, users may also face challenges such as distractions and the need for hands-free operation, which can affect interaction and usability. Furthermore, specific design decisions were made such as the size, color, and symbolization of displayed information. The success of the interfaces tested could be influenced by their unique features and elements, which may not transfer to other design specifications.

In light of these considerations, we emphasize the iterative character of the development process that we take into account for zoomless maps. Summarizing the work presented in this thesis, we deem one full cycle of the development complete. We have validated our central hypothesis for specific conditions and specifications, but further development steps are needed to generalize our findings (see Section 8.3).

8.2 Summary of Contribution

Overall, we see our concepts and implementations of zoomless maps as a successful and pioneering contribution to the domain of location-based services. Our interfaces broaden the horizon of interactive map applications in terms of both the presentation and exploration of dense geospatial information. By providing novel interaction techniques, our interfaces tackle prevailing issues with established zoom-and-pan strategies that require a user to repeatedly zoom and pan to reveal all information. Unlike prior research, we have shifted the focus from enhancing zooming capabilities to developing alternative interaction types. These specialized interactions enable a comprehensive data exploration without the need for additional zooming, thereby maintaining a spatial overview. Since only one type of interaction is required for assessing the information, our interfaces provide the ability to limit interaction complexity, which is known to reduce cognitive load (Wang et al., 2018) and increase task performance and user confidence (Vincent et al., 2019).

In experimental evaluations, we were able to verify our mathematical models and to demonstrate the high quality of the results of our algorithms with respect to these models. To ensure a smooth execution of our interfaces, we have presented efficient algorithms that allow the integration into highly interactive map environments – even without a connection to an external server. Within the scope of validating the central hypothesis of the thesis, we were able to generally confirm the anticipated merits of zoomless maps through an empirical user study. Despite a certain degree of habituation to established zoom-and-pan interfaces, most users were open to the novel interactions. Both user acceptance and the confirmed utility suggest that zoomless maps hold promise for the future.

However, we have also indicated remaining limitations of the different strategies we have presented. We consider zoomless maps as an ongoing project and still see much potential for further development. In this context, we again emphasize the cyclical nature of the development process and its scope for continuous improvement.

8.3 Future Research and Directions

As one key perspective for own future work, we deem it important to continue the development cycle of zoomless maps through further iterations. Gathering the findings we gained so far, we should adapt the concepts, models, algorithms, and experiments in a meaningful way. Requests and suggestions of the test users should be taken into account and further studies should be conducted. To ascertain generalizability and robustness of the central hypothesis of this thesis, it is necessary to further test the (revised) interfaces in different settings.

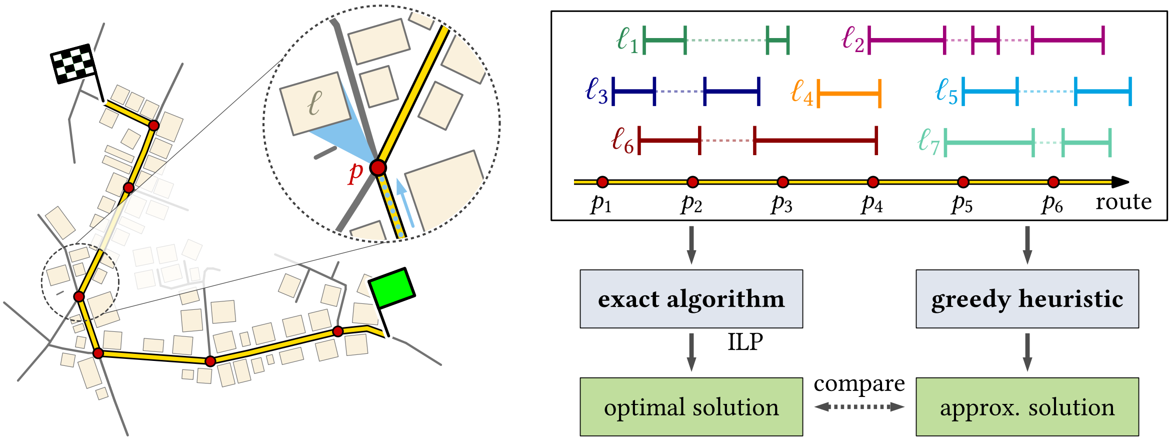
Aiming at investigating the interfaces under realistic conditions, the primary focus should be on conducting field testing for different user tasks in diverse outdoor environments. In this way, it can be investigated whether external influences and environmental factors associated with mobile use affect the attention and behavior of users. Switching to the actual use of mobile devices also offers the opportunity to assess the extent to which touch-based interactions influence user behavior compared to the mouse-based interactions that have already been tested. Understanding how interfaces perform and adapt in various contexts and scenarios facilitates more focused advancements in their development.

Furthermore, efforts should be invested in comparing and establishing particular design choices. So far, the design decisions have been based on personal reasoning and literature references, but have not been compared with alternative designs. However, design considerations for user interfaces are an own complex and evolving area of research across multiple disciplines that must receive careful attention as we move toward well-engineered map applications (Roth, 2017; White, 2017).

Besides revising the already introduced concepts, we see further motivation in the development of more flexible models. For example, one could think of a clustering of features that are spatially close to each other. The challenge is to simultaneously aggregate features into clusters, to divide the clusters into accessible parts (e.g., to distribute them on pages), and to represent the clusters in a meaningful way. When displaying each cluster as one symbol, an additional layer of interaction could allow the individual features contained in a cluster to be revealed. In this case, the placement and representation of the features would be an additional challenge.

From a methodological point of view, combinatorial optimization has proven to be well-suited for our developed concepts. Expanding beyond the focus on interfaces for presenting geospatial information, we see substantial potential for applying mathematical optimization to other types of location-based services as well. As one recent example from own research, Gedicke et al. (2023b) consider the selection of landmarks used as visual cues for wayfinding assistance, modeling the task as an optimization problem; see Figure 8.1. Aiming to mitigate the cognitive load caused by the inclusion of a large number of different landmarks in routing instructions, the optimization model minimizes the number of selected landmarks while ensuring visibility at decision points along the route. As an additional criterion, landmarks are preferably chosen that are already clearly visible when approaching a decision point, thus optimizing *advance visibility*.

Emerging technologies like augmented reality (AR) introduce new types of location-based services, yet they also introduce novel challenges that can be addressed through the application of combinatorial optimization. For example, ensuring a precise registration within augmented reality systems is crucial to maintain the visual coherence of the augmented environment (Furht, 2011). By means of optimization approaches, AR systems can tackle the problem of aligning virtual objects with real-world environments. This includes solving optimization problems such as the *correspondence problem*, which aims to determine the best match between the features extracted from the real environment and the corresponding features in the virtual scene. Moreover, combinatorial optimization techniques can also be applied to other AR-related problems, such as



(a) sketch of approach

(b) methodology

Fig. 8.1: Presentation of own research published in Gedicke et al. (2023b). In (a), a sketch is shown that supports the explanation of the general approach. It is assumed that a route (yellow line) and a set of potential landmarks (beige polygons), such as building footprints, are given as input. To provide navigation guidance through routing instructions at each intersection along the route (red dots), a visible landmark should be selected for each such decision point. The enlarged section shows the visibility of a particular landmark ℓ from a decision point p. Possible visual obstructions due to obstacles (e.g., walls or other landmarks) are taken into account. The selection of landmarks is modeled as an optimization problem. In addition to the visibility criterion, the number of selected landmarks is minimized to decrease the cognitive load associated with landmark-based route descriptions. Furthermore, preference is given to landmarks that are already clearly visible when approaching a decision point (*advance visibility*). The upper part in (b) sketches how the visibility of the landmarks along the route can be modeled as multi-intervals. If only the number of landmarks is to be minimized so that one landmark is visible from each decision point, the task becomes the c-INTERVAL COVER PROBLEM, which is known to be NP-hard (Ding et al., 2011). Due to the high problem complexity, combinatorial optimization is applied both through an exact method based on integer linear programming and through a greedy heuristic approach. The optimal results of the ILP are used to evaluate the quality of the heuristic solutions.

object placement and scene understanding. For example, optimization approaches can be used to arrange virtual objects in the real environment optimizing given visual criteria while avoiding occlusions.

In light of this perspective and the contributions that have been presented in this thesis, we would like to conclude with the following quote.

> Without change, there is no innovation,
> creativity or incentive for improvement.
>
> *William G. Pollard*

BIBLIOGRAPHY

Aarts, E. and Lenstra, J. K. *Local Search in Combinatorial Optimization*. Princeton University Press, 2003. doi:10.1515/9780691187563. 27

Abras, C., Maloney-Krichmar, D., and Preece, J. User-Centered Design. *Encyclopedia of Human-Computer Interaction*, 37(4):445–456, 2004. URL `https://www.academia.edu/1012299/User_centered_design`. 5, 11, 89

Adamaszek, A. and Wiese, A. Approximation Schemes for Maximum Weight Independent Set of Rectangles. In *IEEE Annual Symp. on Foundations of Computer Science (FOCS'13)*, pp. 400–409. IEEE, 2013. doi:10.1109/FOCS.2013.50. 20

Afonso, A. P., Carmo, M. B., Costa, P., and Pereira, T. ARWithDistance: Distance Awareness in Off-Screen Visualization Techniques for AR Applications. In *27th Int. Conf. Info. Vis. (IV'23)*, pp. 340–345. IEEE, 2023. doi:10.1109/IV60283.2023.00064. 13

Agarwal, P. K., Van Kreveld, M., and Suri, S. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, 11(3-4):209–218, 1998. doi:10.1016/S0925-7721(98)00028-5. 20, 36

Ahn, J. and Freeman, H. A Program for Automatic Name Placement. In *Proc. of the Int. Symp. on Auto. Carto. (Auto-Carto VI)*, pp. 444–453, 1983. 19

Akpinar, E., Yeşilada, Y., and Temizer, S. The Effect of Context on Small Screen and Wearable Device Users' Performance - A Systematic Review. *ACM Computing Surveys (CSUR)*, 53(3):1–44, 2020. doi:10.1145/3386370. 11

Ali, K., Hartmann, K., and Strothotte, T. Label Layout for Interactive 3D Illustrations. *Journal of WSCG*, 13:1–8, 2005. 23

Alinhac, G. *Cartographie: Théorique et Technique*. Institut Géographique National, 1963. 19

Artinger, E., Maier, P., Coskun, T., Nestler, S., Mähler, M., Yildirim-Krannig, Y., Wucholt, F., Echtler, F., and Klinker, G. Creating a common operation picture in realtime with user-centered interfaces for mass casualty incidents. In *Proc. of the Int. Conf. on Pervasive Comp. Technologies for Healthcare (PervasiveHealth'12)*, pp. 291–296. IEEE, 2012. doi:10.4108/icst.pervasivehealth.2012.248598. 107

Avanthay, C., Hertz, A., and Zufferey, N. A variable neighborhood search for graph coloring. *European Journal of Operational Research*, 151(2):379–388, 2003. doi:10.1016/S0377-2217(02)00832-9. 29

Bae, W. D., Alkobaisi, S., Narayanappa, S., Vojtechovsky, P., and Bae, K. Y. Optimizing map labeling of point features based on an onion peeling approach. *Journal of Spatial Information Science (JOSIS)*, 2: 3–28, 2011. doi:10.5311/JOSIS.2011.2.4. 21

Barth, L., Gemsa, A., Niedermann, B., and Nöllenburg, M. On the readability of leaders in boundary labeling. *Information Visualization*, 18(1):110–132, 2019. doi:10.1177/1473871618799500. 23

Bartling, M., Robinson, A. C., Resch, B., Eitzinger, A., and Atzmanstorfer, K. The role of user context in the design of mobile map applications. *Cartography and Geographic Information Science*, 48(5):432–448, 2021. doi:10.1080/15230406.2021.1933595. 10, 11

Baudisch, P. and Rosenholtz, R. Halo: a technique for visualizing off-screen objects. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'03)*, pp. 481–488. ACM, 2003. doi:10.1145/642611.642695. 12

Beard, M. Constraints on rule formation. In *Map Generalization: Making Rules for Knowledge Representation*, pp. 32–58. Longman's Inc, 1991. 16

Been, K., Daiches, E., and Yap, C. Dynamic Map Labeling. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):773–780, 2006. doi:10.1109/TVCG.2006.136. 2, 22, 65

Been, K., Nöllenburg, M., Poon, S.-H., and Wolff, A. Optimizing Active Ranges for Consistent Dynamic Map Labeling. *Computational Geometry: Theory and Applications*, 43(3):312–328, 2010. doi:10.1016/j.comgeo.2009.03.006. 2, 22

Bekos, M., Niedermann, B., and Nöllenburg, M. External Labeling Techniques: A Taxonomy and Survey. *Computer Graphics Forum*, 38(3):833–860, 2019. doi:10.1111/cgf.13729. 23, 24, 68, 72, 73, 93, 106, 108, 109, 110

Bekos, M. A., Kaufmann, M., Symvonis, A., and Wolff, A. Boundary Labeling: Models and Efficient Algorithms for Rectangular Maps. In *Graph Drawing (GD'04)*, volume 3383 of *Lecture Notes in Computer Science*, pp. 49–59. Springer, 2004. doi:10.1007/978-3-540-31843-9_7. 23

Bekos, M. A., Kaufmann, M., Nöllenburg, M., and Symvonis, A. Boundary Labeling with Octilinear Leaders. *Algorithmica*, 57(3):436–461, 2010. doi:10.1007/s00453-009-9283-6. 23

Bekos, M. A., Cornelsen, S., Fink, M., Hong, S.-H., Kaufmann, M., Nöllenburg, M., Rutter, I., and Symvonis, A. Many-to-One Boundary Labeling with Backbones. *Journal of Graph Algorithms and Applications*, 19 (3):779–816, 2015. doi:10.7155/jgaa.00379. 23

Benkert, M., Haverkort, H. J., Kroll, M., and Nöllenburg, M. Algorithms for Multi-Criteria Boundary Labeling. *Journal of Graph Algorithms and Applications*, 13(3):289–317, 2009. doi:10.7155/jgaa.00189. 23, 75, 76, 79

Bernsdorf, B. and Fritze, H. Digitale Karten und eine geodatenbasierte Lageskizze für den Feuerwehreinsatz. *Zeitschrift für Forschung, Technik und Management im Brandschutz*, 65(4):171–183, 2016. 106, 107, 108

Bertini, E., Rigamonti, M., and Lalanne, D. Extended Excentric Labeling. *Computer Graphics Forum*, 28 (3):927–934, 2009. doi:10.1111/j.1467-8659.2009.01456.x. 24

Bianchetti, R. A., Wallgrün, J. O., Yang, J., Blanford, J., Robinson, A. C., and Klippel, A. Free Classification of Canadian and American Emergency Management Map Symbol Standards. *The Cartographic Journal*, 49(4):350–360, 2012. doi:10.1179/1743277412Y.0000000022. 107

Bill, R., Blankenbach, J., Breunig, M., Haunert, J.-H., Heipke, C., Herle, S., Maas, H.-G., Mayer, H., Meng, L., Rottensteiner, F., et al. Geospatial Information Research: State of the Art, Case Studies and Future Perspectives. *Journal of Photogrammetry, Remote Sensing and Geoinformation Science (PFG)*, 90(4): 349–389, 2022. doi:10.1007/s41064-022-00217-9. 7

Biswas, N., Singh, A., and Bhattacharya, S. Augmented 3D arrows for visualizing off-screen Points of Interest without clutter. *Displays*, 79, 2023. doi:10.1016/j.displa.2023.102502. 13

Björkbom, M., Timonen, J., Yiğitler, H., Kaltiokallio, O., García, J. M. V., Myrsky, M., Saarinen, J., Korkalainen, M., Cuhac, C., Jäntti, R., et al. Localization Services for Online Common Operational Picture and Situation Awareness. *IEEE Access*, 1:742–757, 2013. doi:10.1109/ACCESS.2013.2287302. 107

Blana, N., Kavadas, I., and Tsoulos, L. A Constraint-Based Generalization Model Incorporating a Quality Control Mechanism. *Geographies*, 3(2):321–343, 2023. doi:10.3390/geographies3020017. 17

Blandford, A. and Wong, B. W. Situation awareness in emergency medical dispatch. *International Journal of Human-Computer Studies*, 61(4):421–452, 2004. doi:10.1016/j.ijhcs.2003.12.012. 107

Bobák, P., Čmolík, L., and Čadík, M. Video Sequence Boundary Labeling with Temporal Coherence. In *Advances in Computer Graphics: Proc. of the Computer Graphics Int. Conf. (CGI'19)*, volume 11542 of *Lecture Notes in Computer Science*, pp. 40–52. Springer, 2019. doi:10.1007/978-3-030-22514-8_4. 23

Bobák, P., Čmolík, L., and Čadík, M. Temporally stable boundary labeling for interactive and non-interactive dynamic scenes. *Computers & Graphics*, 91:265–278, 2020. doi:10.1016/j.cag.2020.08.005. 23

Bojko, A. *Eye Tracking the User Experience: A Practical Guide to Research*. Rosenfeld Media, Brooklyn, New York, USA, 2013. 11

Bollobás, B. and Thomason, A. Random Graphs of Small Order. In *Random Graphs '83*, volume 118 of *North-Holland Mathematics Studies*, pp. 47–97. North-Holland, 1985. doi:10.1016/S0304-0208(08)73612-0. 29

Boring, S., Ledo, D., Chen, X., Marquardt, N., Tang, A., and Greenberg, S. The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction. In *Proc. of the Int. Conf. on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'12)*, pp. 39–48. ACM, 2012. doi:10.1145/2371574.2371582. 10

Bose, P., Carmi, P., Keil, J. M., Mehrabi, S., and Mondal, D. Boundary Labeling for Rectangular Diagrams. In *Proc. of the Scandinavian Symp. and Workshops on Algorithm Theory (SWAT'18)*, volume 101 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 12:1–12:14. Leibniz Center for Informatics, 2018. doi:10.4230/LIPIcs.SWAT.2018.12. 23

Bose, P., Mehrabi, S., and Mondal, D. Faster Multi-sided One-Bend Boundary Labelling. In *Proc. of the Int. Workshop on Algorithms and Computation (WALCOM'21)*, pp. 116–128. Springer, 2021. doi:10.1007/978-3-030-68211-8_10. 23

Boukhtouta, A. and Berger, J. A Framework of Recognized Operational Support Picture for Asset Visibility. *Procedia - Social and Behavioral Sciences*, 147:251–259, 2014. doi:10.1016/j.sbspro.2014.07.168. 107

Boutoura, C. Logarithmic Urban Thematic Mapping in MIS Environment. *Cartographica*, 31(1):41–53, 1994. doi:10.3138/C0XG-1260-5808-34HX. 13

Brélaz, D. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979a. doi:10.1145/359094.359101. 29

Brélaz, D. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979b. doi:10.1145/359094.359101. 29

Brown, J. R. Chromatic Scheduling and the Chromatic Number Problem. *INFORMS Management Science*, 19(4):456–463, 1972. doi:10.1287/mnsc.19.4.456. 29

Burghardt, D., Schmid, S., and Stoter, J. Investigations on Cartographic Constraint Formalisation. In *Workshop of the ICA Commission on Generalisation and Multiple Representation*, volume 19, pp. 2, 2007. 16

Burigat, S. and Chittaro, L. Location-aware visualization of VRML models in GPS-based mobile guides. In *Proc. of the Int. Conf. on 3D Web Technology (Web3D'05)*, pp. 57–64. ACM, 2005. doi:10.1145/1050491.1050499. 9

Burigat, S. and Chittaro, L. Visualizing references to off-screen content on mobile devices: A comparison of Arrows, Wedge, and Overview+ Detail. *Interacting with Computers*, 23(2):156–166, 2011. doi:10.1016/j.intcom.2011.02.005. 2, 12

Burigat, S., Chittaro, L., and Gabrielli, S. Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. In *Proc. of the Int. Conf. on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'06)*, pp. 239–246. ACM, 2006. doi:10.1145/1152215.1152266. 12

Buttenfield, B. P. and Mark, D. M. Expert systems in cartographic design. In *Modern Cartography Series*, volume 1, pp. 129–150. Elsevier, 1991. doi:10.1016/B978-0-08-040277-2.50015-5. 7

Byskov, J. Chromatic Number in Time O(2.4023^n) Using Maximal Independent Sets. *BRICS Report Series*, 9(45), 2002. doi:10.7146/brics.v9i45.21760. 29

Campêlo, M. B., Corrêa, R. C., and Frota, Y. Cliques, holes and the vertex coloring polytope. *Information Processing Letters*, 89(4):159–164, 2004. doi:10.1016/j.ipl.2003.11.005. 29, 30

Campêlo, M. B., Campos, V. A., and Corrêa, R. C. On the asymmetric representatives formulation for the vertex coloring problem. *Discrete Applied Mathematics*, 156(7):1097–1111, 2008. doi:10.1016/j.dam.2007.05.058. 29

Cano, R. G., de Souza, C. C., and de Rezende, P. J. Solving dynamic labeling problems to optimality using solution space reductions. *Theoretical Computer Science*, 789:77–92, 2019. doi:10.1016/j.tcs.2018.08.010. 22

Cao, W., Xu, J., Peng, F., Tong, X., Wang, X., Zhao, S., and Liu, W. A point-feature label placement algorithm based on spatial data mining. *Mathematical Biosciences and Engineering*, 20(7):12169–12193, 2023. doi:10.3934/mbe.2023542. 22

Card, S. K., Robertson, G. G., and Mackinlay, J. D. The Information Visualizer, an Information Workspace. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'91)*, pp. 181–186. ACM, 1991. doi:10.1145/108844.108874. 8, 46, 120

Carmo, M. B., Afonso, A. P., Melo, M., Rocha, B., and Botan, V. Augmented Reality with Maps for Off-Screen POI Awareness. In *24th Int. Conf. Info. Vis. (IV'20)*, pp. 454–459. IEEE, 2020. doi:10.1109/IV51561.2020.00079. 13

Chalermsook, P. and Chuzhoy, J. Maximum Independent Set of Rectangles. In *Proc. of the Annual ACM-SIAM Symp. on Discrete Algorithms (SODA'09)*, pp. 892–901. SIAM, 2009. doi:10.1137/1.9781611973068.97. 20

Chalermsook, P. and Walczak, B. Coloring and Maximum Weight Independent Set of Rectangles. In *Proc. of the Annual ACM-SIAM Symp. on Discrete Algorithms (SODA'21)*, pp. 860–868. SIAM, 2021. doi:10.1137/1.9781611976465.54. 21

Chan, T. M. A note on maximum independent sets in rectangle intersection graphs. *Information Processing Letters*, 89(1):19–23, 2004. doi:10.1016/j.ipl.2003.09.019. 20

Chan, T. M. and Har-Peled, S. Approximation algorithms for maximum independent set of pseudo-disks. In *Proc. of the Annual Symp. on Comp. Geom. (SCG'09)*, pp. 333–340. ACM, 2009. doi:10.1145/1542362.1542420. 21

Cheliotis, K., Liarokapis, F., Kokla, M., Tomai, E., Pastra, K., Anastopoulou, N., Bezerianou, M., Darra, A., and Kavouras, M. A systematic review of application development in augmented reality navigation research. *Cartography and Geographic Information Science*, 50(3):249–271, 2023. doi:10.1080/15230406.2023.2194032. 9

Chiarandini, M., Dumitrescu, I., and Stützle, T. Local Search for the Colouring Graph Problem. A Computational Study. Technical Report AIDA-03-01, FG Intellektik, TU Darmstadt, 2003. 29

Chimani, M., van Dijk, T. C., and Haunert, J.-H. How to eat a graph: Computing selection sequences for the continuous generalization of road networks. In *Proc. of the SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (SIGSPATIAL'14)*, pp. 243–252. ACM, 2014. doi:10.1145/2666310.2666414. 2, 18

Chittaro, L. Visualizing information on mobile devices. *IEEE Computer*, 39(3):40–45, 2006. doi:10.1109/MC.2006.109. 1, 8, 10, 12

Christensen, J., Marks, J., and Shieber, S. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232, 1995. doi:10.1145/212332.212334. 19, 20, 21

Chuzhoy, J. and Ene, A. On Approximating Maximum Independent Set of Rectangles. In *IEEE Annual Symp. on Foundations of Computer Science (FOCS'16)*, pp. 820–829. IEEE, 2016. doi:10.1109/FOCS.2016.92. 20

Čmolík, L., Pavlovec, V., Wu, H.-Y., and Nöllenburg, M. Mixed Labeling: Integrating Internal and External Labels. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1848–1861, 2022. doi:10.1109/TVCG.2020.3027368. 24

Conover, D. M. and Dammann, J. J. F. Three-Dimensional Sensor Common Operating Picture (3-D Sensor COP). Technical report, US Army Research Laboratory, Adelphi, MD, 2017. URL `https://apps.dtic.mil/sti/citations/AD1026246`. 107

Copeland, J. Emergency Response: Unity of Effort Through a Common Operational Picture. Technical report, US Army War College, Carlisle Barracks, PA, 2008. 107

Corbin, A., Niedermann, B., Nothnagel, A., Haas, R., and Haunert, J.-H. Combinatorial optimization applied to VLBI scheduling. *Journal of Geodesy*, 94(2):1–22, 2020. doi:10.1007/s00190-020-01348-w. 52

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms*. MIT Press, fourth edition, 2022. 24

Costa, D., Hertz, A., and Dubuis, C. Embedding a sequential procedure within an evolutionary algorithm for coloring problems in graphs. *Journal of Heuristics*, 1:105–128, 1995. doi:10.1007/BF02430368. 29

Courtial, A., El Ayedi, A., Touya, G., and Zhang, X. Exploring the Potential of Deep Learning Segmentation for Mountain Roads Generalisation. *ISPRS International Journal of Geo-Information*, 9(5):338, 2020. doi:10.3390/ijgi9050338. 18

Courtial, A., Touya, G., and Zhang, X. Constraint-Based Evaluation of Map Images Generalized by Deep Learning. *Journal of Geovisualization and Spatial Analysis*, 6(1):13, 2022. doi:10.1007/s41651-022-00104-2. 18

Courtial, A., Touya, G., and Zhang, X. Deriving map images of generalised mountain roads with generative adversarial networks. *International Journal of Geographical Information Science*, 37(3):499–528, 2023. doi:10.1080/13658816.2022.2123488. 18

Cranmer, E. E., tom Dieck, M. C., and Fountoulaki, P. Exploring the value of augmented reality for tourism. *Tourism Management Perspectives*, 35:100672, 2020. doi:https://doi.org/10.1016/j.tmp.2020.100672. 9

Danciger, J., Devadoss, S. L., Mugno, J., Sheehy, D., and Ward, R. Shape deformation in continuous map generalization. *Geoinformatica*, 13:203–221, 2009. doi:10.1007/s10707-008-0049-0. 18

Dantzig, G. *Linear Programming and Extensions*. Princeton University Press, 1963. 26

de Berg, M. and Gerrits, D. H. Approximation algorithms for free-label maximization. *Computational Geometry: Theory and Applications*, 45(4):153–168, 2012. doi:10.1016/j.comgeo.2011.10.004. 20, 21

de Lima, A. M. and Carmo, R. Exact Algorithms for the Graph Coloring Problem. *Revista de Informática Teórica e Aplicada*, 25(4):57, 2018. doi:10.22456/2175-2745.80721. 29

DeLucia, A. and Black, T. A comprehensive approach to automatic feature generalization. In *Proc. of the Int. Cartographic Conf. (ICC'87)*, pp. 173–191, 1987. 14

Demir, F., Ahmad, S., Calyam, P., Jiang, D., Huang, R., and Jahnke, I. A Next-Generation Augmented Reality Platform for Mass Casualty Incidents (MCI). *Journal of Usability Studies*, 12(4):193–214, 2017. 107

Deng, M. and Peng, D. Morphing linear features based on their entire structures. *Transactions in GIS*, 19 (5):653–677, 2015. doi:10.1111/tgis.12111. 18

Ding, L., Fu, B., and Zhu, B. Minimum Interval Cover and Its Application to Genome Sequencing. In *Combinatorial Optimization and Applications: Selections of the Int. Conf. on Combinatorial Optimization and Applications (COCOA'11)*, volume 6831 of *Lecture Notes in Computer Science*, pp. 287–298. Springer, 2011. doi:10.1007/978-3-642-22616-8_23. 126

Doddi, S., Marathe, M. V., Mirzaian, A., Moret, B. M. E., and Zhu, B. Map Labeling and Its Generalizations. In *Proc. of the Annual ACM-SIAM Symp. on Discrete Algorithms (SODA'97)*, pp. 148–157. SIAM, 1997. doi:10.5555/314161.314250. 20, 21

Doerschler, J. S. and Freeman, H. An expert system for dense-map name placement. In *Proc. Auto-Carto*, volume 9, pp. 215–224. Citeseer, 1989. 7

Doeweling, S., Tahiri, T., Sowinski, P., Schmidt, B., and Khalilbeigi, M. Support for collaborative situation analysis and planning in crisis management teams using interactive tabletops. In *Proc. of the ACM Int. Conf. on Interactive Tabletops and Surfaces (ITS'13)*, pp. 273–282. ACM, 2013. doi:10.1145/2512349.2512823. 108

Dong, W., Tian, Y., and Zhang, Y. Automatic generalization of metro maps based on dynamic segmentation. *Science China: Technological Sciences*, 53:158–165, 2010. doi:10.1007/s11431-010-3204-4. 18

Du, J., Wu, F., Xing, R., Gong, X., and Yu, L. Segmentation and sampling method for complex polyline generalization based on a generative adversarial network. *Geocarto International*, 37(14):4158–4180, 2022. doi:10.1080/10106049.2021.1878288. 18

Duchêne, C., Baella, B., Brewer, C. A., Burghardt, D., Buttenfield, B. P., Gaffuri, J., Käuferle, D., Lecordix, F., Maugeais, E., Nijhuis, R., Pla, M., Post, M., Regnauld, N., Stanislawski, L. V., Stoter, J. E., Tóth, K., Urbanke, S., van Altena, V., and Wiedemann, A. Generalisation in Practice Within National Mapping Agencies. In *Abstracting Geographic Information in a Data Rich World*, Lecture Notes in Geoinformation and Cartography, pp. 329–391. Springer, 2014. doi:10.1007/978-3-319-00203-3_11. 17

Duchêne, C., Touya, G., Taillandier, P., Gaffuri, J., Ruas, A., and Renard, J. Multi-Agents Systems for Cartographic Generalization: Feedback from Past and On-going Research. Technical report, Institut national de l'information géographique et forestière, 2018. doi:10.13140/RG.2.2.35489.92006. 8, 17

Duchowski, A. T. *Eye Tracking Methodology: Theory and Practice*. Springer, third edition, 2017. doi:10.1007/978-3-319-57883-5. 11

Dumont, M., Touya, G., and Duchêne, C. Designing multi-scale maps: lessons learned from existing practices. *International Journal of Cartography*, 6(1):121–151, 2020. doi:10.1080/23729333.2020.1717832. 12, 22

Edmondson, S., Christensen, J., Marks, J., and Shieber, S. A General Cartographic Labelling Algorithm. *Cartographica*, 33(4):13–24, 1996. doi:10.3138/U3N2-6363-130N-H870. 21

Ehrliholzer, R. Quality assessment in generalization: integrating quantitative and qualitative methods. In *Proc. of the Int. Cartographic Conf. (ICC'95)*, pp. 2241–2250, 1995. 17

Ellis, G. and Dix, A. A Taxonomy of Clutter Reduction for Information Visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007. doi:10.1109/TVCG.2007.70535. 11

Elmqvist, N. and Fekete, J.-D. Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2009. doi:10.1109/TVCG.2009.84. 12

Endsley, M. R. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors*, 37(1): 32–64, 1995. doi:10.1518/001872095779049543. 106, 107

Eppler, M. J. and Mengis, J. The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines. *The Information Society*, 20(5): 325–344, 2004. doi:10.1080/01972240490507974. 11

Eppstein, D. Small Maximal Independent Sets and Faster Exact Graph Coloring. *Journal of Graph Algorithms and Applications*, 7(2):131–140, 2003. doi:10.7155/JGAA.00064. 29

Erlebach, T., Hagerup, T., Jansen, K., Minzlaff, M., and Wolff, A. Trimming of graphs, with application to point labeling. *Theory of Computing Systems*, 47:613–636, 2010. doi:10.1007/s00224-009-9184-8. 21

Fairbairn, D. and Taylor, G. Developing a variable-scale map projection for urban areas. *Computers & Geosciences*, 21(9):1053–1064, 1995. doi:10.1016/0098-3004(95)00041-6. 13

Fekete, J.-D. and Plaisant, C. Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'99)*, pp. 512–519. ACM, 1999. doi:10.1145/302979.303148. 24

Feng, Y., Thiemann, F., and Sester, M. Learning Cartographic Building Generalization with Deep Convolutional Neural Networks. *ISPRS International Journal of Geo-Information*, 8(6):258, 2019. doi:10.3390/ijgi8060258. 18

Feng, Y., Huang, X., and Sester, M. Extraction and analysis of natural disaster-related VGI from social media: review, opportunities and challenges. *International Journal of Geographical Information Science*, 36(7):1275–1316, 2022. doi:10.1080/13658816.2022.2048835. 9

Fink, M., Haunert, J.-H., Schulz, A., Spoerhase, J., and Wolff, A. Algorithms for Labeling Focus Regions. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2583–2592, 2012. doi:10.1109/TVCG.2012.193. 24, 109, 113

Fisher, P. F., Sester, M., and Brenner, C. Continuous Generalization for Visualization on Small Mobile Devices. In *Developments in Spatial Data Handling*, pp. 355–368. Springer, 2005. doi:10.1007/3-540-26772-7_27. 18

Formann, M. and Wagner, F. A Packing Problem with Applications to Lettering of Maps. In *Proc. of the Annual Symp. on Comp. Geom. (SCG'91)*, pp. 281–288. ACM, 1991. doi:10.1145/109648.109680. 19, 20, 21

Forsch, A., Amann, F., and Haunert, J.-H. Visualizing the Off-Screen Evolution of Trajectories. *KN - Journal of Cartography and Geographic Information*, 72(3):201–212, 2022. doi:10.1007/s42489-022-00106-6. 13

Fowler, R. J., Paterson, M. S., and Tanimoto, S. L. Optimal packing and covering in the plane are NP-complete. *Information Processing Letters*, 12(3):133–137, 1981. doi:10.1016/0020-0190(81)90111-3. 20

Fu, C., Zhou, Z., Feng, Y., and Weibel, R. Keeping walls straight: data model and training set size matter for deep learning in building generalization. *Cartography and Geographic Information Science*, 51(1): 1–16, 2023. doi:10.1080/15230406.2023.2264757. 18

Furht, B. *Handbook of Augmented Reality*. Springer, 2011. doi:10.1007/978-1-4614-0064-6. 125

Furnas, G. W. Generalized Fisheye Views. *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'86)*, 17(4):16–23, 1986. doi:10.1145/22339.22342. 13

Galinier, P. and Hao, J.-K. Hybrid Evolutionary Algorithms for Graph Coloring. *Journal of Combinatorial Optimization*, 3:379–397, 1999. doi:10.1023/A:1009823419804. 29

Galinier, P. and Hertz, A. A survey of local search methods for graph coloring. *Computers & Operations Research*, 33(9):2547–2562, 2006. doi:10.1016/j.cor.2005.07.028. 29

Gálvez, W., Khan, A., Mari, M., Mömke, T., Pittu, M. R., and Wiese, A. A 3-Approximation Algorithm for Maximum Independent Set of Rectangles. In *Proc. of the Annual ACM-SIAM Symp. on Discrete Algorithms (SODA'22)*, pp. 894–905. SIAM, 2022. doi:10.1137/1.9781611977073.38. 20

Gao, A., Li, J., and Chen, K. A morphing approach for continuous generalization of linear map features. *PLOS ONE*, 15(12), 2020. doi:10.1371/journal.pone.0243328. 2, 18

Garey, M. R., Johnson, D. S., and Stockmeyer, L. Some Simplified NP-Complete Problems. In *Proc. of the Annual ACM Symp. on Theory of Computing (STOC'74)*, pp. 47–63. ACM, 1974. doi:10.1145/800119.803884. 29

Gedicke, S., Niedermann, B., and Haunert, J.-H. Multi-page Labeling of Small-screen Maps with a Graph-coloring Approach. In *Advances in Cartography and GIScience of the ICA: Proc. of the Int. Conf. on Location Based Services (LBS'19)*, volume 2, pp. 4–11. Springer, 2019. doi:10.5194/ica-adv-2-4-2019. 35

Gedicke, S. and Haunert, J.-H. An Empirical Study on Interfaces for Presenting Large Sets of Point Features in Mobile Maps. *The Cartographic Journal*, 60(1):25–42, 2023. doi:10.1080/00087041.2023.2182354. 89

Gedicke, S., Bonerath, A., Niedermann, B., and Haunert, J.-H. Zoomless Maps: External Labeling Methods for the Interactive Exploration of Dense Point Sets at a Fixed Map Scale. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1247–1256, 2021a. doi:10.1109/TVCG.2020.3030399. 67

Gedicke, S., Jabrayilov, A., Niedermann, B., Mutzel, P., and Haunert, J.-H. Point feature label placement for multi-page maps on small-screen devices. *Computers & Graphics*, 100:66–80, 2021b. doi:10.1016/j.cag.2021.07.019. 49

Gedicke, S., Oehrlein, J., and Haunert, J.-H. Aggregating land-use polygons considering line features as separating map elements. *Cartography and Geographic Information Science*, 48(2):124–139, 2021c. doi:10.1080/15230406.2020.1851613. 30, 31, 32

Gedicke, S., Arzoumanidis, L., and Haunert, J.-H. Automating the External Placement of Symbols for Point Features in Situation Maps for Emergency Response. *Cartography and Geographic Information Science*, 50(4):385–402, 2023a. doi:10.1080/15230406.2023.2213446. 105

Gedicke, S., Tomko, M., Winter, S., and Haunert, J.-H. Selecting Landmarks for Wayfinding Assistance Based on Advance Visibility. In *Proc. of the SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (SIGSPATIAL'23)*, pp. 1–10. ACM, 2023b. doi:10.1145/3589132.3625605. 125, 126

Gemsa, A., Haunert, J.-H., and Nöllenburg, M. Boundary-labeling algorithms for panorama images. In *Proc. of the SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (GIS'11)*, pp. 289–298. ACM, 2011a. doi:10.1145/2093973.2094012. 23

Gemsa, A., Nöllenburg, M., and Rutter, I. Sliding labels for dynamic point labeling. In *Proc. of the Annual Canadian Conf. on Computational Geometry (CCCG'11)*, pp. 205–2010. University of Toronto Press, 2011b. 22

Gemsa, A., Haunert, J.-H., and Nöllenburg, M. Multirow Boundary-Labeling Algorithms for Panorama Images. *Transactions on Spatial Algorithms and Systems (TSAS)*, 1(1):1–30, 2015. doi:10.1145/2794299. 23

Gemsa, A., Nöllenburg, M., and Rutter, I. Consistent Labeling of Rotating Maps. *J. Comput. Geom.*, 7(1): 308–331, 2016a. 22

Gemsa, A., Nöllenburg, M., and Rutter, I. Evaluation of Labeling Strategies for Rotating Maps. *Journal of Experimental Algorithmics*, 21(1):1–21, 2016b. doi:10.1145/2851493. 22

Gemsa, A., Niedermann, B., and Nöllenburg, M. A Unified Model and Algorithms for Temporal Map Labeling. *Algorithmica*, 82(10):2709–2736, 2020. doi:10.1007/s00453-020-00694-7. 22

Glass, C. A. and Prügel-Bennett, A. Genetic Algorithm for Graph Coloring: Exploration of Galinier and Hao's Algorithm. *Journal of Combinatorial Optimization*, 7:229–236, 2003. doi:10.1023/A:1027312403532. 29

Göbel, F., Kurzhals, K., Schinazi, V. R., Kiefer, P., and Raubal, M. Gaze-Adaptive Lenses for Feature-Rich Information Spaces. In *Proc. of the ACM Symp. on Eye Tracking Research and Applications (ETRA'20)*, pp. 1–8. ACM, 2020. doi:10.1145/3379155.3391323. 14

Gomes, S. P., Ribeiro, G. M., and Lorena, L. A. N. Dispersion for the Point-Feature Cartographic Label Placement Problem. *Expert Systems with Applications*, 40(15):5878–5883, 2013. doi:10.1016/j.eswa.2013.04.035. 21

Gomes, S. P., Lorena, L. A. N., and Ribeiro, G. M. A Constructive Genetic Algorithm for Discrete Dispersion on Point Feature Cartographic Label Placement Problems. *Geographical Analysis*, 48(1):43–58, 2016. doi:10.1111/gean.12082. 21

Gomory, R. E. Some polyhedra related to combinatorial problems. *Linear Algebra and its Applications*, 2 (4):451–558, 1969. 27

Goodchild, M. F. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007. doi:10.1007/s10708-007-9111-y. 9

Gould, N. and Mackaness, W. From taxonomies to ontologies: Formalizing generalization knowledge for on-demand mapping. *Cartography and Geographic Information Science*, 43(3):208–222, 2016. doi:10.1080/15230406.2015.1072737. 15, 16, 18

Griffin, A. L., White, T., Fish, C., Tomio, B., Huang, H., Sluter, C. R., Bravo, J. V. M., Fabrikant, S. I., Bleisch, S., Yamada, M., et al. Designing across map use contexts: A research agenda. *International Journal of Cartography*, 3(sup1):90–114, 2017. doi:10.1080/23729333.2017.1315988. 10

Grinstein, G., Kobsa, A., Plaisant, C., Shneiderman, B., and Stasko, J. T. Which comes first, usability or utility? In *Proc. of the IEEE Conf. on Visualization*, pp. 605–606. IEEE, 2003. doi:10.1109/VISUAL.2003.1250426. 11

Grötschel, M. and Holland, O. Solving matching problems with linear programming. *Mathematical Programming*, 33(3):243–259, 1985. doi:10.1007/BF01584376. 76

Gruenefeld, U., Ali, A. E., Boll, S., and Heuten, W. Beyond halo and wedge: Visualizing out-of-view objects on head-mounted virtual and augmented reality devices. In *Proc. of the Int. Conf. on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'18)*, pp. 1–11. ACM, 2018. doi:10.1145/3229434.3229438. 13

Gruget, M., Touya, G., and Muehlenhaus, I. Missing the city for buildings? A critical review of pan-scalar map generalization and design in contemporary zoomable maps. *International Journal of Cartography*, 9(2):255–285, 2023. doi:10.1080/23729333.2022.2153467. 2

Guan, D. and Xuding, Z. A coloring problem for weighted graphs. *Information Processing Letters*, 61(2): 77–81, 1997. doi:10.1016/S0020-0190(97)00002-1. 41

Guedira, Y., Kolski, C., and Lepreux, S. Pedestrian Navigation through Pictograms and Landmark Photos on Smart Glasses: A Pilot Study. In *Proc. of the Int. Conf. on Human-Computer Interaction (RoCHI'22)*, pp. 13–20, 2022. doi:10.37789/rochi.2022.1.1.4. 9

Guerra, F. and Boutoura, C. An electronic lens on digital tourist city-maps. In *Proc. of the Int. Cartographic Conf. (ICC'01)*, pp. 1151–1157, 2001. 13

Gustafson, S., Baudisch, P., Gutwin, C., and Irani, P. Wedge: clutter-free visualization of off-screen locations. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'08)*, pp. 787–796. ACM, 2008. doi:10.1145/1357054.1357179. 13

Gustafson, S. G. and Irani, P. P. Comparing visualizations for tracking off-screen moving targets. In *Extended Abstracts on Human Factors in Computing Systems (CHI EA'07)*, pp. 2399–2404. ACM, 2007. doi:10.1145/1240866.1241014. 12

Gyárfás, A. and Lehel, J. On-line and first fit colorings of graphs. *Journal of Graph Theory*, 12(2):217–227, 1988. doi:10.1002/jgt.3190120212. 41

Hake, G., Grünreich, D., and Meng, L. *Kartographie - Visualisierung raum-zeitlicher Informationen*. De Gruyter, Berlin, Boston, 1994. ISBN 9783110870572. doi:10.1515/9783110870572. 15

Hansen, P., Labbé, M., and Schindl, D. Set covering and packing formulations of graph coloring: Algorithms and first polyhedral results. *Discrete Optimization*, 6(2):135–147, 2009. doi:10.1016/j.disopt.2008.10.004. 29

Harrie, L. and Weibel, R. Modelling the Overall Process of Generalisation. In *Generalisation of Geographic Information: Cartographic Modelling and Applications*, pp. 67 – 87. Elsevier, 2007. doi:10.1016/B978-008045374-3/50006-5. 8, 16, 17

Harrie, L., Sarjakoski, L. T., and Lehto, L. A Mapping Function for Variable-Scale Maps in Small-Display Cartography. *Journal of Geospatial Engineering*, 4(2):111–124, 2002. 13

Harrie, L., Stigmar, H., and Djordjevic, M. Analytical Estimation of Map Readability. *ISPRS International Journal of Geo-Information*, 4(2):418–446, 2015. doi:10.3390/ijgi4020418. 11, 108

Harrie, L., Oucheikh, R., Nilsson, Å., Oxenstierna, A., Cederholm, P., Wei, L., Richter, K.-F., and Olsson, P. Label Placement Challenges in City Wayfinding Map Production—Identification and Possible Solutions. *Journal of Geovisualization and Spatial Analysis*, 6(1):16, 2022. doi:10.1007/s41651-022-00115-z. 19

Harrie, L., Touya, G., Oucheikh, R., Ai, T., Courtial, A., and Richter, K.-F. Machine learning in cartography. *Cartography and Geographic Information Science*, 51(1):1–19, 2024. doi:10.1080/15230406.2023.2295948. 9

Hasan, K., Kim, J., Ahlström, D., and Irani, P. Thumbs-Up: 3D Spatial Thumb-Reachable Space for One-Handed Thumb Interaction on Smartphones. In *Proc. of the Symp. on Spatial User Interaction (SUI'16)*, pp. 103–106. ACM, 2016. doi:10.1145/2983310.2985755. 10

Hassenzahl, M. The Thing and I: Understanding the Relationship Between User and Product. In *Funology: From Usability to Enjoyment*, volume 3, pp. 31–42. Springer, 2005. doi:10.1007/1-4020-2967-5_4. 11

Haunert, J.-H. and Hermes, T. Labeling Circular Focus Regions Based on a Tractable Case of Maximum Weight Independent Set of Rectangles. In *Proc. of the SIGSPATIAL Int. Workshop on Interacting with Maps (MapInteract'14)*, pp. 15–21. ACM, 2014. doi:10.1145/2677068.2677069. 24

Haunert, J.-H. and Sering, L. Drawing Road Networks with Focus Regions. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2555–2562, 2011. doi:10.1109/TVCG.2011.191. 13

Haunert, J.-H. and Wolff, A. Area aggregation in map generalisation by mixed-integer programming. *International Journal of Geographical Information Science*, 24(12):1871–1897, 2010. doi:10.1080/13658810903401008. 18, 30, 32

Haunert, J.-H. and Wolff, A. Beyond maximum independent set: an extended integer programming formulation for point labeling. *ISPRS International Journal of Geo-Information*, 6(11):342, 2017. doi:10.3390/ijgi6110342. 21, 36, 92

Heidmann, F., Hermann, F., and Peissner, M. Interactive Maps On Mobile, Location-Based Systems: Design Solutions And Usability Testing. In *Proc. of the Int. Cartographic Conf. (ICC'03)*, pp. 1299–1305, 2003. 12

Heinsohn, N., Gerasch, A., and Kaufmann, M. Boundary Labeling Methods for Dynamic Focus Regions. In *Proc. of the IEEE Pacific Visualization Symp. (PacificVis'14)*, pp. 243–247. IEEE, 2014. doi:10.1109/PacificVis.2014.20. 24

Hertz, A. and Werra, D. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, 1987. doi:10.1007/BF02239976. 29

Higashikawa, Y., Imai, K., Shiraga, T., Sukegawa, N., and Yokosuka, Y. Minimum point-overlap labelling*. *Optimization Methods and Software*, 36(2-3):316–325, 2021. doi:10.1080/10556788.2020.1833880. 21

Hirsch, S. A. An Algorithm for Automatic Name Placement Around Point Data. *The American Cartographer*, 9(1):5–17, 1982. doi:10.1559/152304082783948367. 19

Hornbæk, K., Bederson, B. B., and Plaisant, C. Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview. *Transactions on Computer-Human Interaction (TOCHI)*, 9(4): 362–389, 2002. doi:10.1145/586081.586086. 12

Huang, B.-C., Hsu, J., Chu, E. T.-H., and Wu, H.-M. ARBIN: Augmented Reality Based Indoor Navigation System. *Sensors*, 20(20):5890, 2020. doi:10.3390/s20205890. 9

Huang, H. and Gartner, G. A Technical Survey on Decluttering of Icons in Online Map-Based Mashups. In *Online Maps with APIs and WebServices*, Lecture Notes in Geoinformation and Cartography, pp. 157–175. Springer, 2012. doi:10.1007/978-3-642-27485-5_11. 12

Huang, H., Gartner, G., Krisp, J. M., Raubal, M., and Van de Weghe, N. Location based services: ongoing evolution and research agenda. *Journal of Location Based Services*, 12(2):63–93, 2018. doi:10.1080/17489725.2018.1508763. 10

Huang, L., Ai, T., Van Oosterom, P., Yan, X., and Yang, M. A Matrix-Based Structure for Vario-Scale Vector Representation over a Wide Range of Map Scales: The Case of River Network Data. *ISPRS International Journal of Geo-Information*, 6(7):218, 2017. doi:10.3390/ijgi6070218. 18

Huang, Q. and Wong, D. W. Activity patterns, socioeconomic status and urban spatial structure: what can social media data tell us? *International Journal of Geographical Information Science*, 30(9):1873–1898, 2016. doi:10.1080/13658816.2016.1145225. 9

Huffman, F. T. and Cromley, R. G. An Automated Multi-Criteria Cartographic Aid for Point Annotation. *The Cartographic Journal*, 39(1):51–64, 2002. doi:10.1179/caj.2002.39.1.51. 21

Hwang, G. H. and Yoon, W. C. A new approach to requirement development for a common operational picture to support distributed situation awareness. *Safety Science*, 125, 2020. doi:10.1016/j.ssci.2019.104569. 107

Imai, H. and Asano, T. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *Journal of Algorithms*, 4(4):310–323, 1983. doi:10.1016/0196-6774(83)90012-3. 29, 38, 53

Imhof, E. Die Anordnung der Namen in der Karte. *International Yearbook of Cartography*, 2:93–129, 1962. 19

Imhof, E. Positioning Names on Maps. *The American Cartographer*, 2(2):128–144, 1975. doi:10.1559/152304075784313304. 19, 108

Irani, P., Gutwin, C., and Yang, X. D. Improving selection of off-screen targets with hopping. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'06)*, pp. 299–308. ACM, 2006. doi:10.1145/1124772.1124818. 13

Iturriaga, C. and Lubiw, A. Elastic labels around the perimeter of a map. *Journal of Algorithms*, 47(1): 14–39, 2003. doi:10.1016/S0196-6774(03)00004-X. 20

Jabrayilov, A. and Mutzel, P. New Integer Linear Programming Models for the Vertex Coloring Problem. In *Theoretical Informatics: Proc. of the Latin American Symp. on Theoretical Informatics (LATIN'18)*, volume 10807 of *Lecture Notes in Computer Science*, pp. 640–652. Springer, 2018. doi:10.1007/978-3-319-77404-6_47. 29, 30, 53

Jackson, W. and Pao. *SmartWatch Design Fundamentals*. Springer, 2019. doi:10.1007/978-1-4842-4369-5. 42, 57, 70, 80, 93

Jiang, B., Xu, S., and Li, Z. Polyline simplification using a region proposal network integrating raster and vector features. *GIScience & Remote Sensing*, 60(1), 2023. doi:10.1080/15481603.2023.2275427. 18

Jiang, M. A new approximation algorithm for labeling points with circle pairs. *Information Processing Letters*, 99(4):125–129, 2006. doi:10.1016/j.ipl.2006.04.006. 21

Jiang, M., Bereg, S., Qin, Z., and Zhu, B. New Bounds on Map Labeling with Circular Labels. In *Algorithms and Computation: Proc. of the Int. Symp. on Algorithms and Computation (ISAAC'04)*, volume 3341 of *Lecture Notes in Computer Science*, pp. 606–617. Springer, 2005. doi:10.1007/978-3-540-30551-4_53. 21

Jiang, S., Kang, P., Song, X., Lo, B. P., and Shull, P. B. Emerging Wearable Interfaces and Algorithms for Hand Gesture Recognition: A Survey. *IEEE Reviews in Biomedical Engineering*, 15:85–102, 2021. doi:10.1109/RBME.2021.3078190. 10

Johnson, D. S., Aragon, C. R., McGeoch, L. A., and Schevon, C. Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning. *Operations Research*, 39 (3):378–406, 1991. doi:10.1287/opre.39.3.378. 29

Jones, C. E., Haklay, M., Griffiths, S., and Vaughan, L. A less-is-more approach to geovisualization – enhancing knowledge construction across multidisciplinary teams. *International Journal of Geographical Information Science*, 23(8):1077–1093, 2009. doi:10.1080/13658810802705723. 8

Jul, S. and Furnas, G. W. Critical zones in desert fog: aids to multiscale navigation. In *Proc. of the Annual ACM Symp. on User Interface Software and Technology (UIST'98)*, pp. 97–106. ACM, 1998. doi:10.1145/288392.288578. 2, 9, 12

Kaneider, D., Seifried, T., and Haller, M. Automatic annotation placement for interactive maps. In *Proc. of the ACM Int. Conf. on Interactive Tabletops and Surfaces (ITS'13)*, pp. 61–70. ACM, 2013. doi:10.1145/2512349.2512809. 108

Karagiannis, G. M. and Synolakis, C. E. Collaborative Incident Planning and the Common Operational Picture. In *Dynamics of Disasters: Proc. of the Int. Conf. on Dynamics of Disasters (DOD'16)*, Springer Proceedings in Mathematics & Statistics, pp. 91–112. Springer, 2016. doi:10.1007/978-3-319-43709-5_6. 107

Karsznia, I., Wereszczyńska, K., and Weibel, R. Make It Simple: Effective Road Selection for Small-Scale Map Design Using Decision-Tree-Based Models. *ISPRS International Journal of Geo-Information*, 11(8): 457, 2022. doi:10.3390/ijgi11080457. 19

Karsznia, I., Adolf, A., Leyk, S., and Weibel, R. Using machine learning and data enrichment in the selection of roads for small-scale maps. *Cartography and Geographic Information Science*, 51(1):60–78, 2023. doi:10.1080/15230406.2023.2283075. 18

Keates, J. S. *Cartographic Design and Production*. Longman Scientific & Technical, second edition, 1989. 8

Kedia, T., Ratcliff, J., O'Connor, M., Oluic, S., Rose, M., Freeman, J., and Rainwater-Lovett, K. Technologies Enabling Situational Awareness During Disaster Response: A Systematic Review. *Disaster Medicine and Public Health Preparedness*, 16(1):341–359, 2022. doi:10.1017/dmp.2020.196. 108

Keehner, M., Hegarty, M., Cohen, C., Khooshabeh, P., and Montello, D. R. Spatial Reasoning With External Visualizations: What Matters Is What You See, Not Whether You Interact. *Cognitive Science*, 32(7): 1099–1132, 2008. doi:10.1080/03640210801898177. 8

Kindermann, P., Niedermann, B., Rutter, I., Schaefer, M., Schulz, A., and Wolff, A. Multi-sided Boundary Labeling. *Algorithmica*, 76(1):225–258, 2016. doi:10.1007/s00453-015-0028-4. 23

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by Simulated Annealing. *Science*, 220(4598): 671–680, 1983. doi:10.1126/science.220.4598.671. 28, 114

Klau, G. W. and Mutzel, P. Optimal labeling of point features in rectangular labeling models. *Mathematical Programming*, 94:435–458, 2003. doi:10.1007/s10107-002-0327-9. 21

Klee, V. and Minty, G. J. How Good Is the Simplex Algorithm? In *Inequalities III*, pp. 159–175. Academic Press, 1972. 26

Klute, F., Löffler, M., and Nöllenburg, M. Labeling nonograms: Boundary labeling for curve arrangements. *Computational Geometry: Theory and Applications*, 98:101791, 2021. doi:10.1016/j.comgeo.2021.101791. 23

Knura, M. Deep Learning for Map Generalization: Towards a new Approach using Vector Data. *Abstracts of the International Cartographic Association (ICA)*, 3:1–2, 2021. doi:10.5194/ica-abs-3-152-2021. 18

Knura, M. Learning from vector data: enhancing vector-based shape encoding and shape classification for map generalization purposes. *Cartography and Geographic Information Science*, 51(1):146–167, 2023. doi:10.1080/15230406.2023.2273397. 9, 18

Knuth, D. E. Postscript about NP-hard problems. *ACM SIGACT News*, 6(2):15–16, 1974. doi:10.1145/1008304.1008305. 8

Komosko, L., Batsyn, M., Segundo, P. S., and Pardalos, P. M. A fast greedy sequential heuristic for the vertex colouring problem based on bitwise operations. *Journal of Combinatorial Optimization*, 31:1665–1677, 2016. doi:10.1007/s10878-015-9862-1. 29

Korpi, J. and Ahonen-Rainio, P. Clutter Reduction Methods for Point Symbols in Map Mashups. *The Cartographic Journal*, 50(3):257–265, 2013. doi:10.1179/1743277413Y.0000000065. 11, 12

Korte, B. H., Vygen, J., Korte, B., and Vygen, J. *Combinatorial Optimization: Theory and Algorithms*, volume 21. Springer, sixth edition, 2011. 24

Kostelnick, J. C., Dobson, J. E., Egbert, S. L., and Dunbar, M. D. Cartographic Symbols for Humanitarian Demining. *The Cartographic Journal*, 45(1):18–31, 2008. doi:10.1179/000870408X276585. 107

Kraak, M.-J. Geovisualization illustrated. *ISPRS Journal of Photogrammetry and Remote Sensing*, 57(5–6): 390–399, 2003. doi:10.1016/S0924-2716(02)00167-3. 8

Kralev, V. and Kraleva, R. A comparative analysis between two heuristic algorithms for the graph vertex coloring problem. *International Journal of Electrical and Computer Engineering (IJECE)*, 13(3):2981–2989, 2023. doi:10.11591/ijece.v13i3.pp2981-2989. 29

Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 (1-2):83–97, 1955. doi:10.1002/nav.3800020109. 75

Kuvedžić Divjak, A. and Lapaine, M. The Role of the Map in a Crisis Management Environment: Applying the Theory of Cartographic Communication and Visualization. *Collegium Antropologicum*, 38(1):187–193, 2014. URL https://hrcak.srce.hr/121036. 107, 108

Kuveždić Divjak, A., Dapo, A., and Pribičević, B. Cartographic Symbology for Crisis Mapping: A Comparative Study. *ISPRS International Journal of Geo-Information*, 9(3):142, 2020. doi:10.3390/ijgi9030142. 107

Lai, J., Zhang, D., and Wang, S. ContextZoom: A Single-Handed Partial Zooming Technique for Touch-Screen Mobile Devices. *International Journal of Human–Computer Interaction*, 33(6):475–485, 2017. doi:10.1080/10447318.2016.1275433. 10

Lakehal, A., Lepreux, S., Efstratiou, C., Kolski, C., and Nicolaou, P. Spatial Knowledge Acquisition for Pedestrian Navigation: A Comparative Study between Smartphones and AR Glasses. *Information*, 14 (7):353, 2023. doi:10.3390/info14070353. 9

Lamers, C. and Denker, A. Methoden der Visualisierung in Führungsstäben der Feuerwehr. In *Handbuch Stabsarbeit*, pp. 319–325. Springer, 2022. doi:10.1007/978-3-662-63035-8_39. 106, 107, 108, 115

Laugwitz, B., Held, T., and Schrepp, M. Construction and Evaluation of a User Experience Questionnaire. In *HCI and Usability for Education and Work: Proc. of the Symp. of the Austrian HCI and Usability Engineering Group (USAB'08)*, volume 5298 of *Lecture Notes in Computer Science*, pp. 63–76. Springer, 2008. doi:10.1007/978-3-540-89350-9_6. 11

Lawler, E. A Note on the Complexity of the Chromatic Number Problem. *Information Processing Letters*, 5: 66–67, 1976. doi:10.1016/0020-0190(76)90065-X. 29

Lei, Y., He, S., Khamis, M., and Ye, J. An End-to-End Review of Gaze Estimation and its Interactive Applications on Handheld Mobile Devices. *ACM Computing Surveys*, 56(2):1–38, 2023. doi:10.1145/3606947. 10

Leighton, F. T. A graph coloring algorithm for large scheduling problems. *Journal of Research of the National Bureau of Standards*, 84(6):489, 1979. doi:10.6028/jres.084.024. 29

Li, J. and Mao, K. The morphing for continuous generalization of linear features based on Dynamic Time Warping distance. *Geocarto International*, 38(1):2197516, 2023. doi:10.1080/10106049.2023.2197516. 2, 18

Li, J., Ai, T., Liu, P., and Yang, M. Continuous Scale Transformations of Linear Features Using Simulated Annealing-Based Morphing. *ISPRS International Journal of Geo-Information*, 6(8):242, 2017a. doi:10.3390/ijgi6080242. 18

Li, J., Li, X., and Xie, T. Morphing of Building Footprints Using a Turning Angle Function. *ISPRS International Journal of Geo-Information*, 6(6):173, 2017b. doi:10.3390/ijgi6060173. 18

Li, L., Hang, Z., Zhu, H., Kuai, X., and Hu, W. A Labeling Model Based on the Region of Movability for Point-Feature Label Placement. *ISPRS International Journal of Geo-Information*, 5(9):159, 2016. doi:10.3390/ijgi5090159. 20, 21, 109

Li, P., Yan, H., and Lu, X. MultiLineStringNet: a deep neural network for linear feature set recognition. *Cartography and Geographic Information Science*, 51(1):114–129, 2023a. doi:10.1080/15230406.2023.2264756. 9

Li, Q. Variable-scale representation of road networks on small mobile devices. *Computers & Geosciences*, 35(11):2185–2190, 2009. doi:10.1016/j.cageo.2008.12.009. 2, 13

Li, R. Spatial Learning in Smart Applications: Enhancing Spatial Awareness through Visualized Off-Screen Landmarks on Mobile Devices. *Annals of the American Association of Geographers*, 110(2):421–433, 2020. doi:10.1080/24694452.2019.1670611. 13

Li, R. Augmented reality landmarks on windshield and their effects on the acquisition of spatial knowledge in autonomous vehicles. *Journal of Location Based Services*, 17(4):412–425, 2023. doi:10.1080/17489725.2023.2238661. 13

Li, R., Korda, A., Radtke, M., and Schwering, A. Visualising distant off-screen landmarks on mobile devices to support spatial orientation. *Journal of Location Based Services*, 8(3):166–178, 2014. doi:10.1080/17489725.2014.978825. 13

Li, W., Yan, H., Lu, X., and Shen, Y. A Heuristic Approach for Resolving Spatial Conflicts of Buildings in Urban Villages. *ISPRS International Journal of Geo-Information*, 12(10):392, 2023b. doi:10.3390/ijgi12100392. 19

Liao, C.-S., Liang, C.-W., and Poon, S. H. Approximation algorithms on consistent dynamic map labeling. *Theoretical Computer Science*, 640:84–93, 2016. doi:10.1016/j.tcs.2016.06.006. 2, 22

Liao, H., Wang, X., Dong, W., and Meng, L. Measuring the influence of map label density on perceived complexity: a user study using eye tracking. *Cartography and Geographic Information Science*, 46(3): 210–227, 2019. doi:10.1080/15230406.2018.1434016. 11, 108

Liao, S., Bai, Z., and Bai, Y. Errors prediction for vector-to-raster conversion based on map load and cell size. *Chinese Geographical Science*, 22:695–704, 2012. doi:10.1007/s11769-012-0544-y. 18

Lin, C.-C. Crossing-free many-to-one boundary labeling with hyperleaders. In *Proc. of the IEEE Pacific Visualization Symp. (PacificVis'10)*, pp. 185–192. IEEE, 2010. doi:10.1109/PACIFICVIS.2010.5429592. 23

Lin, C.-C., Kao, H.-J., and Yen, H.-C. Many-to-One Boundary Labeling. *Journal of Graph Algorithms and Applications*, 12(3):319–356, 2008. doi:10.7155/jgaa.00169. 23

Liu, B., Ding, L., and Meng, L. Spatial knowledge acquisition with virtual semantic landmarks in mixed reality-based indoor navigation. *Cartography and Geographic Information Science*, 48(4):305–319, 2021. doi:10.1080/15230406.2021.1908171. 9

Löffler, M., Nöllenburg, M., and Staals, F. Mixed Map Labeling. In *Algorithms and Complexity: Proc. of the Int. Conf. on Algorithms and Complexity (CIAC'15)*, volume 9079 of *Lecture Notes in Computer Science*, pp. 339–351. Springer, 2015. doi:10.1007/978-3-319-18173-8_25. 109

Luboschik, M., Schumann, H., and Cords, H. Particle-based labeling: Fast point-feature labeling without obscuring other visual features. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1237–1244, 2008. doi:10.1109/TVCG.2008.152. 21

Lyu, Z., Sun, Q., Ma, J., Xu, Q., Li, Y., and Zhang, F. Road Network Generalization Method Constrained by Residential Areas. *ISPRS International Journal of Geo-Information*, 11(3):159, 2022. doi:10.3390/ijgi11030159. 19

Ma, J., Sun, Q., Xu, L., Sun, S., and Wen, B. A Displacement Algorithm Considering Geometry Constraints to Resolve Spatial Conflicts between Roads and Buildings. *The Cartographic Journal*, 60(1):43–55, 2023. doi:10.1080/00087041.2023.2182372. 19

MacEachren, A. M. and Kraak, M.-J. Exploratory cartographic visualization: advancing the agenda. *Computers & Geosciences*, 23(4):335–343, 1997. doi:10.1016/S0098-3004(97)00018-6. 7

Mackaness, W. A., Fisher, P., and Wilkinson, G. Towards a cartographic expert system. In *Proc. of the Int. Conf. on the Acquisition, Management and Presentation of Spatial Data*, volume 1, pp. 578–587. distributed Royal Institution of Chartered Surveyors, 1986. 7

Malaguti, E., Monaci, M., and Toth, P. An exact approach for the Vertex Coloring Problem. *Discrete Optimization*, 8(2):174–190, 2011. doi:10.1016/j.disopt.2010.07.005. 29

Malaguti, E. and Toth, P. A survey on vertex coloring problems. *International Transactions in Operational Research*, 17(1):1–34, 2010. doi:10.1111/j.1475-3995.2009.00696.x. 29

Malaguti, E., Monaci, M., and Toth, P. A Metaheuristic Approach for the Vertex Coloring Problem. *INFORMS Journal on Computing*, 20(2):302–316, 2008. doi:10.1287/ijoc.1070.0245. 29

Manivannan, C., Krukar, J., and Schwering, A. Spatial generalization in sketch maps: A systematic classification. *Journal of Environmental Psychology*, 83:101851, 2022. doi:10.1016/j.jenvp.2022.101851. 18

Marín, A. and Pelegrín, M. Towards unambiguous map labeling – Integer programming approach and heuristic algorithm. *Expert Systems with Applications*, 98:221–241, 2018. doi:10.1016/j.eswa.2017.11.014. 21

Marinova, S. T. New Map Symbol System for Disaster Management. In *Proc. of the International Cartographic Association (ICA)*, volume 1, pp. 74. Copernicus, 2018. doi:10.5194/ica-proc-1-74-2018. 107

Marks, J. and Shieber, S. M. The Computational Complexity of Cartographic Label Placement. Technical Report TR-05-91, Harvard Computer Science Group, 1991. 8

Mauri, G. R., Ribeiro, G. M., and Lorena, L. A. A new mathematical model and a Lagrangean decomposition for the point-feature cartographic label placement problem. *Computers & Operations Research*, 37(12): 2164–2172, 2010. doi:10.1016/j.cor.2010.03.005. 21, 109

McMaster, R. B. and Monmonier, M. A Conceptual Framework for Quantitative and Qualitative Raster-Mode Generalization. In *Proc. of the Annual GIS/LIS Conference*, pp. 390–403, 1989. 14

McMaster, R. B. Automated Line Generalization. *Cartographica*, 24(2):74–111, 1987. doi:10.3138/3535-7609-781G-4L20. 15

McMaster, R. B. and Shea, K. S. *Generalization in Digital Cartography*. 1992. 14, 15

McNeese, M. D., Pfaff, M. S., Connors, E. S., Obieta, J. F., Terrell, I. S., and Friedenberg, M. A. Multiple Vantage Points of the Common Operational Picture: Supporting International Teamwork. *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, 50(3):467–471, 2006. doi:10.1177/154193120605000354. 107

Mehlhorn, K., Näher, S., and Uhrig, C. The LEDA platform for combinatorial and geometric computing. In *Automata, Languages and Programming: Proc. of the Int. Colloquium on Automata, Languages, and Programming (ICALP'97)*, volume 1256 of *Lecture Notes in Computer Science*, pp. 7–16. Springer, 1997. doi:10.1007/3-540-63165-8_161. 113

Mehrotra, A. and Trick, M. A Column Generation Approach for Graph Coloring. *INFORMS Journal On Computing*, 8(4):344–354, 1996. doi:10.1287/ijoc.8.4.344. 29

Mehrotra, A. and Trick, M. A. A Branch-And-Price Approach for Graph Multi-Coloring. In *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, pp. 15–29. Springer, 2007. doi:10.1007/978-0-387-48793-9_2. 30

Méndez-Díaz, I. and Zabala, P. A Branch-and-Cut algorithm for graph coloring. *Discrete Applied Mathematics*, 154(5):826–847, 2006. doi:10.1016/j.dam.2005.05.022. 29, 30

Méndez-Díaz, I. and Zabala, P. A cutting plane algorithm for graph coloring. *Discrete Applied Mathematics*, 156(2):159 – 179, 2008. doi:10.1016/j.dam.2006.07.010. 29

Meng, L. Egocentric Design of Map-Based Mobile Services. *The Cartographic Journal*, 42(1):5–13, 2005. doi:10.1179/000870405X57275. 10

Meng, Y., Zhang, H., Liu, M., and Liu, S. Clutter-aware label layout. In *Proc. of the IEEE Pacific Visualization Symp. (PacificVis'15)*, pp. 207–214. IEEE, 2015. doi:10.1109/PACIFICVIS.2015.7156379. 20

Mitchell, J. S. Approximating Maximum Independent Set for Rectangles in the Plane. In *IEEE Annual Symp. on Foundations of Computer Science (FOCS'21)*, pp. 339–350. IEEE, 2021. doi:10.1109/FOCS52979.2021.00042. 20

Mohammed-Amin, R. K., Levy, R. M., and Boyd, J. E. Mobile augmented reality for interpretation of archaeological sites. In *Proc. of the Int. ACM Workshop on Personalized Access to Cultural Heritage (PATCH'12)*, pp. 11–14, 2012. doi:10.1145/2390867.2390871. 9

Morgenstern, C. Distributed coloration neighborhood search. In *Cliques, Coloring, and Satisfiability: 2nd DIMACS Implementation Challenge, 1993*, volume 26 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 335–358. American Mathematical Society, 1996. 29

Morrison, J. L. Computer Technology and Cartographic Change. *The Computer in Contemporary Cartography. J. Hopkins Univ. Press, New York*, 1980. 19, 108

Mote, K. Fast Point-Feature Label Placement for Dynamic Visualizations. *Information Visualization*, 6(4): 249–260, 2007. doi:10.1057/palgrave.ivs.9500163. 22

Muehlenhaus, I. *Web Cartography: Map Design for Interactive and Mobile Devices*. CRC Press, first edition, 2013. doi:10.1201/b16229. 1, 10, 11

Müller, J., Weibel, R., Lagrange, J., and Salgé, F. Generalization: State of the Art and Issues. In *GIS and Generalisation: Methodology and Practice*, pp. 3–17. CRC Press, 1995. doi:10.1201/9781003062646-2. 16

Munkres, J. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957. doi:10.1137/0105003. 75

Neshati, A., Rey, B., Mohommed Faleel, A. S., Bardot, S., Latulipe, C., and Irani, P. BezelGlide: Interacting with Graphs on Smartwatches with Minimal Screen Occlusion. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'21)*, pp. 1–13. ACM, 2021. doi:10.1145/3411764.3445201. 10

Nickerson, B. G. Automated Cartographic Generalization For Linear Features. *Cartographica*, 25(3):15–66, 1988. doi:10.3138/4144-3U7G-MW01-1Q72. 8

Niedermann, B. and Haunert, J. Focus+context map labeling with optimized clutter reduction. *International Journal of Cartography*, 5(2–3):158–177, 2019. doi:10.1080/23729333.2019.1613072. 19, 24

Niedermann, B., Nöllenburg, M., and Rutter, I. Radial contour labeling with straight leaders. In *Proc. of the Pacific Visualization Symp. (PacificVis'17)*, pp. 295–304. IEEE, 2017. doi:10.1109/PACIFICVIS.2017.8031608. 23, 109

Nielsen, J. *Usability Engineering*. Morgan Kaufmann, 1993. doi:10.1016/C2009-0-21512-1. 8, 11, 46, 86, 120

Nivala, A.-M., Sarjakoski, L., and Sarjakoski, T. User-Centred Design and Development of a Mobile Map Service. In *Proc. of the Scandinavian Research Conf. on Geographical Information Science (ScanGIS'05)*, pp. 109–123, 2005. 11

Nouraniyand, Y. and Andresenz, B. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General*, 31(41):8373–8385, 1998. doi:10.1088/0305-4470/31/41/011. 28

Nurminen, A. O. and Sirvio, K. M. Bus Stop Spotting: a Field Experiment Comparing 2D Maps, Augmented Reality and 3D Maps. In *Proc. of the Int. Conf. on Mobile Human-Computer Interaction (MobileHCI'21)*, pp. 1–14. ACM, 2021. doi:10.1145/3447526.3472051. 9

Nöllenburg, M., Merrick, D., Wolff, A., and Benkert, M. Morphing polylines: A step towards continuous generalization. *Computers, Environment, and Urban Systems*, 32(4):248–260, 2008. doi:10.1016/j.compenvurbsys.2008.06.004. 2, 18

Nöllenburg, M., Polishchuk, V., and Sysikaski, M. Dynamic one-sided boundary labeling. In *Proc. of the SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (GIS'10)*, pp. 310–319. ACM, 2010. doi:10.1145/1869790.1869834. 23

Opach, T. and Rød, J. K. A user-centric optimization of emergency map symbols to facilitate common operational picture. *Cartography and Geographic Information Science*, 49(2):134–153, 2022. doi:10.1080/15230406.2021.1994469. 107

Opach, T., Korycka-Skorupa, J., Karsznia, I., Nowacki, T., Gołębiowska, I., and Rød, J. Visual clutter reduction in zoomable proportional point symbol maps. *Cartography and Geographic Information Science*, 46(4):347–367, 2019. doi:10.1080/15230406.2018.1490202. 12

Opach, T., Rød, J. K., and Munkvold, B. E. Map functions to facilitate situational awareness during emergency events. *Cartography and Geographic Information Science*, 50(6):546–561, 2023. doi:10.1080/15230406.2023.2264759. 107

Osserman, R. The isoperimetric inequality. *Bulletin of the American Mathematical Society*, 84(6):1182–1238, 1978. doi:10.1090/S0002-9904-1978-14553-4. 31

Oucheikh, R. and Harrie, L. A feasibility study of applying generative deep learning models for map labeling. *Cartography and Geographic Information Science*, 51(1):168–191, 2024. doi:10.1080/15230406.2023.2291051. 9

Papadimitriou, C. H. and Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation, 1998. 8, 24, 26, 27

Pardalos, P. M. and Xue, J. The maximum clique problem. *Journal of Global Optimization*, 4(3):301–328, 1994. doi:10.1007/BF01098364. 20

Pavlovec, V. and Cmolik, L. Rapid Labels: Point-Feature Labeling on GPU. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):604–613, 2021. doi:10.1109/TVCG.2021.3114854. 22

Peng, D., Wolff, A., and Haunert, J.-H. Using the A* Algorithm to Find Optimal Sequences for Area Aggregation. In *Advances in Cartography and GIScience: Selections of the Int. Cartographic Conf. (ICACI'17)*, Lecture Notes in Geoinformation and Cartography, pp. 389–404. Springer, 2017. doi:10.1007/978-3-319-57336-6_27. 2, 18

Peng, D. and Touya, G. Continuously Generalizing Buildings to Built-up Areas by Aggregating and Growing. In *Proc. of the SIGSPATIAL Workshop on Smart Cities and Urban Analytics (UrbanGIS'17)*, pp. 1–8. ACM, 2017. doi:10.1145/3152178.3152188. 18

Peng, D., Deng, M., and Zhao, B. Multi-scale transformation of river networks based on morphing technology. *National Remote Sensing Bulletin*, 16(5):953–968, 2012. doi:10.11834/jrs.20121272. 18

Peng, D., Haunert, J.-H., Wolff, A., and Hurter, C. Morphing Polylines Based on Least Squares Adjustment. In *Proc. of the ICA Workshop on Generalisation and Multiple Representation*, 2013. 18

Peng, D., Wolff, A., and Haunert, J.-H. Continuous Generalization of Administrative Boundaries Based on Compatible Triangulations. In *Geospatial Data in a Changing World: Selected Papers of the AGILE Conf. on Geographic Information Science (AGILE'16)*, Lecture Notes in Geoinformation and Cartography, pp. 399–415. Springer, 2016. doi:10.1007/978-3-319-33783-8_23. 18

Peng, D., Wolff, A., and Haunert, J.-H. Finding Optimal Sequences for Area Aggregation – A* vs. Integer Linear Programming. *Transactions on Spatial Algorithms and Systems (TSAS)*, 7(1):1–40, 2020. doi:10.1145/3409290. 2, 18

Peng, D., Meijers, M., and van Oosterom, P. Generalizing Simultaneously to Support Smooth Zooming: Case Study of Merging Area Objects. *Journal of Geovisualization and Spatial Analysis*, 7(12):1–19, 2023. doi:10.1007/s41651-022-00109-x. 2, 18

Perea, P., Morand, D., and Nigay, L. Halo3D: a technique for visualizing off-screen points of interest in mobile augmented reality. In *Proc. of the Conf. on l'Interaction Homme-Machine (IHM'17)*, pp. 43–51. ACM, 2017. doi:10.1145/3132129.3132144. 13

Perea, P., Morand, D., and Nigay, L. Spotlight on Off-Screen Points of Interest in Handheld Augmented Reality: Halo-based techniques. In *Proc. of the Int. Conf. on Interactive Surfaces and Spaces (ISS'19)*, pp. 43–54. ACM, 2019. doi:10.1145/3343055.3359719. 13

Petzold, I., Gröger, G., and Plümer, L. Fast screen map labeling – data structures and algorithms. In *Proc. of the Int. Cartographic Conf. (ICC'03)*, pp. 288–298, 2003. 22

Phillips, R. J. and Noyes, L. An Investigation of Visual Clutter in the Topographic Base of a Geological Map. *The Cartographic Journal*, 19(2):122–132, 1982. doi:10.1179/caj.1982.19.2.122. 12

Pietriga, E., Appert, C., and Beaudouin-Lafon, M. Pointing and beyond: an operationalization and preliminary evaluation of multi-scale searching. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'07)*, pp. 1215–1224. ACM, 2007. doi:10.1145/1240624.1240808. 12

Pilehforooshha, P., Karimi, M., and Mansourian, A. A new model combining building block displacement and building block area reduction for resolving spatial conflicts. *Transactions in GIS*, 25(3):1366–1395, 2021. doi:10.1111/tgis.12731. 19

Plaisant, C., Carr, D., and Shneiderman, B. Image-browser taxonomy and guidelines for designers. *IEEE Software*, 12(2):21–32, 1995. doi:10.1109/52.368260. 12

Pombinho, P., Carmo, M. B., and Afonso, A. P. Adaptive Mobile Visualization – The Chameleon Framework. *Computer Science and Information Systems*, 12(2):445–464, 2015. doi:10.2298/CSIS140607004P. 10

Poon, S.-H. and Shin, C.-S. Adaptive Zooming in Point Set Labeling. In *Fundamentals of Computation Theory: Proc. of the Int. Symp. on Fundamentals of Computation Theory (FCT'05)*, volume 3623 of *Lecture Notes in Computer Science*, pp. 233–244. Springer, 2005. doi:10.1007/11537311_21. 22

Poon, S.-H., Shin, C.-S., Strijk, T., Uno, T., and Wolff, A. Labeling Points with Weights. *Algorithmica*, 38 (2):341–362, 2004. doi:10.1007/s00453-003-1063-0. 21

Purchase, H. C. *Experimental Human-Computer Interaction: A Practical Guide with Visual Examples*. Cambridge University Press, 2012. doi:10.1017/CBO9780511844522. 95

Qian, H., Yang, L., Zuo, Z., Wang, Y., Yang, N., Zhou, S., and Yang, E. Method for continuous multiscale representation of terrain contours using T-spline technique. *Transactions in GIS*, 26(7):2818–2849, 2022. doi:10.1111/tgis.12983. 18

Qin, Z. and Zhu, B. A Factor-2 Approximation for Labeling Points with Maximum Sliding Labels. In *Algorithm Theory: Proc. of the Scandinavian Workshop on Algorithm Theory (SWAT'02)*, volume 2368 of *Lecture Notes in Computer Science*, pp. 100–109. Springer, 2002. doi:10.1007/3-540-45471-3_11. 21

Qin, Z., Wolff, A., Xu, Y., and Zhu, B. New Algorithms for Two-Label Point Labeling. In *Algorithms: Proc. of the Annual European Symp. on Algorithms (ESA'00)*, volume 1879 of *Lecture Notes in Computer Science*, pp. 368–380. Springer, 2000. doi:10.1007/3-540-45253-2_34. 21

Raidl, G. A Genetic Algorithm for Labeling Point Features. In *Proc. of the Int. Conf. on Imaging Science, Systems, and Technology*, pp. 189–196. CSREA Press, 1998. 21

Ramchurn, S. D., Huynh, T. D., Wu, F., Ikuno, Y., Flann, J., Moreau, L., Fischer, J. E., Jiang, W., Rodden, T., Simpson, E., et al. A Disaster Response System based on Human-Agent Collectives. *Journal of Artificial Intelligence Research*, 57(1):661–708, 2016. doi:10.1613/jair.5098. 108

Raposo, P., Touya, G., and Bereuter, P. A Change of Theme: The Role of Generalization in Thematic Mapping. *ISPRS international Journal of Geo-Information*, 9(6):371, 2020. doi:10.3390/ijgi9060371. 18

Raubal, M. and Panov, I. A Formal Model for Mobile Map Adaptation. *Location based services and telecartography II*, pp. 11–34, 2009. doi:10.1007/978-3-540-87393-8_2. 10

Regnauld, N. and McMaster, R. B. A Synoptic View of Generalisation Operators. In *Generalisation of Geographic Information: Cartographic Modelling and Applications*, pp. 37–66. Elsevier, 2007. doi:10.1016/B978-008045374-3/50005-3. 14, 16

Reichenbacher, T. Adaptive Methods for Mobile Cartography. In *Proc. of the Int. Cartographic Conf. (ICC'03)*, pp. 1311–1322, 2003. 10

Reichenbacher, T. *Mobile Cartography – Adaptive Visualisation of Geographic Information on Mobile Devices*. PhD thesis, Technischen Universität München, 2004. 10

Revell, P., Regnauld, N., and Bulbrooke, G. OS VectorMap District: Automated Generalisation, Text Placement and Conflation in Support of Making Pubic Data Public. In *Proc. of the Int. Cartographic Conf. (ICC'11)*, pp. 3–8, 2011. 17

Ribeiro, G. M., Mauri, G. R., and Lorena, L. A. N. A lagrangean decomposition for the maximum independent set problem applied to map labeling. *Operational Research*, 11:229–243, 2011. doi:10.1007/s12351-009-0075-1. 21

Ribeiro, G. M. and Lorena, L. A. N. Lagrangean relaxation with clusters for point-feature cartographic label placement problems. *Computers & Operations Research*, 35(7):2129–2140, 2008. doi:10.1016/j.cor.2006.09.024. 21

Rieger, M. K. and Coulson, M. R. Consensus or confusion: cartographers' knowledge of generalization. *Cartographica*, 30(2-3):69–80, 1993. doi:10.3138/M6H4-1006-6422-H744. 7, 15

Robinson, A. H., Sale, R. D., and Morrison, J. L. *Elements of Cartography*. John Wiley & Sons, 1978. 14

Robinson, A. C., Chen, J., Lengerich, E. J., Meyer, H. G., and MacEachren, A. M. Combining Usability Techniques to Design Geovisualization Tools for Epidemiology. *Cartography and Geographic Information Science*, 32(4):243–255, 2005. doi:10.1559/152304005775194700. 5, 11, 89

Robinson, A. C., Roth, R. E., and MacEachren, A. M. Understanding User Needs for Map Symbol Standards in Emergency Management. *Journal of Homeland Security and Emergency Management*, 8(1), 2011. doi:10.2202/1547-7355.1811. 107

Robinson, A. C., Pezanowski, S., Troedson, S., Bianchetti, R., Blanford, J., Stevens, J., Guidero, E., Roth, R. E., and MacEachren, A. M. Symbol Store: sharing map symbols for emergency management. *Cartography and Geographic Information Science*, 40(5):415–426, 2013. doi:10.1080/15230406.2013.803833. 107

Roth, R., Young, S., Nestel, C., Sack, C., Davidson, B., Janicki, J., Knoppke-Wetzel, V., Ma, F., Mead, R., Rose, C., et al. Global Landscapes: Teaching Globalization through Responsive Mobile Map Design. *The Professional Geographer*, 70(3):395–411, 2018. doi:10.1080/00330124.2017.1416297. 11

Roth, R. E. Interactive maps: What we know and what we need to know. *Journal of Spatial Information Science (JOSIS)*, (6):59–115, 2013. doi:10.5311/JOSIS.2013.6.105. 7, 8

Roth, R. E. User Interface and User Experience (UI/UX) Design. In *The Geographic Information Science & Technology Body of Knowledge*. John P. Wilson (ed.), 2nd quarter 2017 edition, 2017. doi:10.22224/gistbok/2017.2.5. 11, 125

Roth, R. E. and Harrower, M. Addressing Map Interface Usability: Learning from the Lakeshore Nature Preserve Interactive Map. *Cartographic Perspectives*, (60):46–66, 2008. doi:10.14714/CP60.231. 8

Roth, R. E. and MacEachren, A. M. Geovisual analytics and the science of interaction: an empirical interaction study. *Cartography and Geographic Information Science*, 43(1):30–54, 2016. doi:10.1080/15230406.2015.1021714. 8

Roth, R. E., Brewer, C. A., and Stryker, M. S. A typology of operators for maintaining legible map designs at multiple scales. *Cartographic Perspectives*, (68):29–64, 2011. doi:10.14714/CP68.7. 14

Roth, R. E., Ross, K. S., and MacEachren, A. M. User-Centered Design for Interactive Maps: A Case Study in Crime Analysis. *ISPRS International Journal of Geo-Information*, 4(1):262–301, 2015. doi:10.3390/ijgi4010262. 11

Roth, R. E., Çöltekin, A., Delazari, L., Filho, H. F., Griffin, A., Hall, A., Korpi, J., Lokka, I., Mendonça, A., Ooms, K., et al. User studies in cartography: opportunities for empirical research on interactive maps and visualizations. *International Journal of Cartography*, 3(sup1):61–89, 2017. doi:10.1080/23729333.2017.1288534. 11

Rottmann, P., Driemel, A., Haverkort, H., Röglin, H., and Haunert, J.-H. Bicriteria Aggregation of Polygons via Graph Cuts. In *Proc. of the Int. Conf. on Geographic Information Science (GIScience'21) – Part II*, volume 208 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 6:1–6:16. Leibniz Center for Informatics, 2021. doi:10.4230/LIPIcs.GIScience.2021.II.6. 18

Ruas, A. and Plazanet, C. Strategies for Automated Generalization. In *Proc. of the Int. Symp. on Spatial Data Handling (SDH'96)*, pp. 6.1–6.18, 1996. 16

Rudi, A. G. Maximizing the Number of Visible Labels on a Rotating Map. *Scientific Annals of Computer Science*, 31(2):293–313, 2021. doi:10.7561/SACS.2021.2.293. 22

Rylov, M. A. and Reimer, A. W. A Comprehensive Multi-criteria Model for High Cartographic Quality Point-Feature Label Placement. *Cartographica*, 49(1):52–68, 2014. doi:10.3138/carto.49.1.2137. 21

Rylov, M. A. and Reimer, A. W. Improving label placement quality by considering basemap detail with a raster-based approach. *GeoInformatica*, 19(3):463–486, 2015. doi:10.1007/s10707-014-0214-6. 21

Samsonov, T. and Krivosheina, A. Joint generalization of city points and road network for small-scale mapping. In *Proc. of the Int. Conf. on Geographic Information Science (GIScience'12)*, pp. 18–21, 2012. 19

Sayidov, A., Aliakbarian, M., and Weibel, R. Geological Map Generalization Driven by Size Constraints. *ISPRS International Journal of Geo-Information*, 9(4):284, 2020. doi:10.3390/ijgi9040284. 18

Schinke, T., Henze, N., and Boll, S. Visualization of off-screen objects in mobile augmented reality. In *Proc. of the Int. Conf. on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'10)*, pp. 313–316. ACM, 2010. doi:10.1145/1851600.1851655. 13

Schmid, F., Kuntzsch, C., Winter, S., Kazerani, A., and Preisig, B. Situated local and global orientation in mobile you-are-here maps. In *Proc. of the Int. Conf. on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'10)*, pp. 83–92. ACM, 2010. doi:10.1145/1851600.1851617. 2, 12

Schwartges, N., Haunert, J.-H., Wolff, A., and Zwiebler, D. Point Labeling with Sliding Labels in Interactive Maps. In *Connecting a Digital Europe Through Location and Place*, Lecture Notes in Geoinformation and Cartography, pp. 295–310. Springer, 2014. doi:10.1007/978-3-319-03611-3_17. 22

Segundo, P. S. A new DSATUR-based algorithm for exact vertex coloring. *Computers & Operations Research*, 39(7):1724 – 1733, 2012. doi:10.1016/j.cor.2011.10.008. 29

Seppänen, H., Mäkelä, J., Luokkala, P., and Virrantaus, K. Developing shared situational awareness for emergency management. *Safety Science*, 55:1–9, 2013. doi:10.1016/j.ssci.2012.12.009. 107

Sester, M. and Brenner, C. Continuous Generalization for Visualization on Small Mobile Devices. In *Developments in Spatial Data Handling*, pp. 355–368, 2004. doi:10.1007/3-540-26772-7_27. 2, 18

Sester, M. Optimization approaches for generalization and data abstraction. *International Journal of Geographical Information Science*, 19(8-9):871–897, 2005. doi:10.1080/13658810500161179. 17

Sester, M., Feng, Y., and Thiemann, F. Building Generalization Using Deep Learning. *ISPRS Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4:565–572, 2018. doi:10.5194/isprs-archives-XLII-4-565-2018. 18

Sewell, E. An improved algorithm for exact graph coloring. In *Discrete Mathematics and Theoretical Computer Science: Cliques, Coloring, and Satisfiability*, volume 26. American Mathematical Society, 1993. doi:10.1090/dimacs/026/17. 29

Shinde, P. P. and Shah, S. A Review of Machine Learning and Deep Learning Applications. In *Proc. of the Int. Conf. on Computing Communication Control and Automation (ICCUBEA'18)*, pp. 1–6. IEEE, 2018. doi:10.1109/ICCUBEA.2018.8697857. 9

Singh, G., Delamare, W., and Irani, P. D-SWIME: A Design Space for Smartwatch Interaction Techniques Supporting Mobility and Encumbrance. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'18)*, pp. 1–13. ACM, 2018. doi:10.1145/3173574.3174208. 10

Slocum, T. A. *Thematic Cartography and Visualization*. Pearson, first edition, 1998. 14

Smirnoff, A., Huot-Vézina, G., Paradis, S. J., and Boivin, R. Generalizing geological maps with the GeoScaler software: The case study approach. *Computers & Geosciences*, 40:66–86, 2012. doi:10.1016/j.cageo.2011.07.013. 18

Sophronides, P., Papadopoulou, C.-A., Giaoutzi, M., and Scholten, H. J. A Common Operational Picture in Support of Situational Awareness for Efficient Emergency Response Operations. *Journal of Future Internet*, 2(1):10–35, 2017. doi:10.18488/journal.102.2017.21.10.35. 107

Southard, R. Automation in Cartography – Revolution or Evolution? *The Cartographic Journal*, 24(1): 59–63, 1987. doi:10.1179/caj.1987.24.1.59. 7

Štampach, R. and Mulíčková, E. Automated generation of tactile maps. *Journal of Maps*, 12(sup1):532–540, 2016. doi:10.1080/17445647.2016.1196622. 18

Stanke, D., Schroth, P., and Rohs, M. TrackballWatch: Trackball and Rotary Knob as a Non-Occluding Input Method for Smartwatches in Map Navigation Scenarios. *Proc. of the ACM on Human-Computer Interaction*, 6(MHCI):1–14, 2022. doi:10.1145/3546734. 10

Steen-Tveit, K. Identifying Information Requirements for Improving the Common Operational Picture in Multi-Agency Operations. In *Proc. of the Int. Conf. on Information Systems for Crisis Response and Management (ISCRAM'20)*, pp. 252–263, 2020. 107

Steen-Tveit, K. and Munkvold, B. E. From common operational picture to common situational understanding: An analysis based on practitioner perspectives. *Safety Science*, 142, 2021. doi:10.1016/j.ssci.2021.105381. 107

Stefanidis, A., Crooks, A., and Radzikowski, J. Harvesting ambient geospatial information from social media feeds. *GeoJournal*, 78:319–338, 2013. doi:10.1007/s10708-011-9438-2. 9

Šterk, M. and Praprotnik, M. Improving emergency response logistics through advanced GIS. *Open Geospatial Data, Software and Standards*, 2(1):1–6, 2017. doi:10.1186/s40965-017-0014-7. 107, 108

Stoter, J., Baella, B., Blok, C., Burghardt, D., Duchêne, C., Pla, M., Regnauld, N., and Touya, G. State-of-the-art of automated generalisation in commercial software. Technical report, EuroSDR, 2010. 18

Stoter, J., Post, M., van Altena, V., Nijhuis, R., and Bruns, B. Fully automated generalization of a 1:50k map from 1:10k data. *Cartography and Geographic Information Science*, 41(1):1–13, 2014. doi:10.1080/15230406.2013.824637. 17

Strijk, T. and Van Kreveld, M. Practical Extensions of Point Labeling in the Slider Model. *GeoInformatica*, 6(2):181–197, 2002. doi:10.1023/A:1015202410664. 20

Suba, R., Meijers, M., and van Oosterom, P. Continuous Road Network Generalization throughout All Scales. *ISPRS International Journal of Geo-Information*, 5(8), 2016. doi:10.3390/ijgi5080145. 2, 18

Tauscher, S. and Neumann, K. A Displacement Method for Maps Showing Dense Sets of Points of Interest. *Progress in Cartography: EuroCarto 2015*, pp. 3–16, 2016. doi:10.1007/978-3-319-19602-2_1. 12

Tavra, M., Racetin, I., and Peroš, J. The role of crowdsourcing and social media in crisis mapping: a case study of a wildfire reaching Croatian City of Split. *Geoenvironmental Disasters*, 8:1–16, 2021. doi:10.1186/s40677-021-00181-3. 9

Thiemann, F. and Sester, M. An Automatic Approach for Generalization of Land-Cover Data from Topographic Data. In *Trends in Spatial Analysis and Modelling*, Geotechnologies and the Environment, pp. 193–207. Springer, 2018. doi:10.1007/978-3-319-52522-8_10. 18

Thomson, R. and Richardson, D. The 'Good Continuation' Principle of Perceptual Organization applied to the Generalization of Road Networks. In *Proc. of the Int. Cartographic Conf. (ICC'99)*, pp. 1215–1223, 1999. 31

Tomaszewski, B., Judex, M., Szarzynski, J., Radestock, C., and Wirkus, L. Geographic Information Systems for Disaster Response: A Review. *Journal of Homeland Security and Emergency Management*, 12(3):571–602, 2015. doi:10.1515/jhsem-2014-0082. 107

Tomlin, C. D. Cartographic Modeling. In *International Encyclopedia of Geography: People, the Earth, Environment and Technology*, pp. 1–6. Wiley Online Library, 2016. doi:10.1002/9781118786352.wbieg0128. 7

Touya, G. A Road Network Selection Process Based on Data Enrichment and Structure Detection. *Transactions in GIS*, 14(5):595–614, 2010. doi:10.1111/j.1467-9671.2010.01215.x. 19

Touya, G. and Dumont, M. Progressive Block Graying and Landmarks Enhancing as Intermediate Representations between Buildings and Urban Areas. In *Proc. of the ICA Workshop on Generalisation and Multiple Representation*, 2017. 18

Touya, G. and Lokhat, I. Deep Learning for Enrichment of Vector Spatial Databases: Application to Highway Interchange. *Transactions on Spatial Algorithms and Systems (TSAS)*, 6(3):1–21, 2020. doi:10.1145/3382080. 18

Touya, G., Duchêne, C., and Ruas, A. Collaborative Generalisation: Formalisation of Generalisation Knowledge to Orchestrate Different Cartographic Generalisation Processes. In *Geographic Information Science: Proc. of the Int. Conf. on Geographic Information Science (GIScience'10)*, volume 6292 of *Lecture Notes in Computer Science*, pp. 264–278. Springer, 2010. doi:10.1007/978-3-642-15300-6_19. 9, 15, 19, 30

Touya, G., Zhang, X., and Lokhat, I. Is deep learning the new agent for map generalization? *International Journal of Cartography*, 5(2-3):142–157, 2019. doi:10.1080/23729333.2019.1613071. 9, 18

Touya, G., Lobo, M.-J., Mackaness, W., and Muehlenhaus, I. Please, Help Me! I Am Lost in Zoom. In *Proc. of the Int. Cartographic Conf. (ICC'21)*, volume 4, pp. 107, 2021. doi:10.5194/ica-proc-4-107-2021. 2, 12

Touya, G., Gruget, M., and Muehlenhaus, I. Where Am I Now? Modelling Disorientation in Pan-Scalar Maps. *ISPRS International Journal of Geo-Information*, 12(2):62, 2023a. doi:10.3390/ijgi12020062. 2, 9, 12

Touya, G., Potié, Q., and Mackaness, W. A. Incorporating ideas of structure and meaning in interactive multi scale mapping environments. *International Journal of Cartography*, 9(2):342–372, 2023b. doi:10.1080/23729333.2023.2215960. 7

Tsou, M.-H. Revisiting Web Cartography in the United States: the Rise of User-Centered Design. *Cartography and Geographic Information Science*, 38(3):250–257, 2011. doi:10.1559/15230406382250. 11

Tversky, B. Visualizing Thought. In *Handbook of Human Centric Visualization*, pp. 3–40. Springer, 2014. doi:10.1007/978-1-4614-7485-2_1. 18

Ulugtekin, N. and Dogru, A. O. Current Scopes of Cartography: Small Display Map Design. In *Proc. of the Middle East Spatial Technology Conference & Exhibition (MEST'07)*, pp. 1–11, 2007. 2

van Dijk, S., van Kreveld, M., Strijk, T., and Wolff, A. Towards an evaluation of quality for names placement methods. *International Journal of Geographical Information Science*, 16(7):641–661, 2002. doi:10.1080/13658810210138742. 21

van Dijk, T. C. and Haunert, J.-H. Interactive focus maps using least-squares optimization. *International Journal of Geographical Information Science*, 28(10):2052–2075, 2014. doi:10.1080/13658816.2014.887718. 14

van Kreveld, M., Strijk, T., and Wolff, A. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13(1):21–47, 1999. doi:10.1016/S0925-7721(99)00005-X. 20

van Oosterom, P. and Meijers, M. Vario-scale data structures supporting smooth zoom and progressive transfer of 2D and 3D data. *International Journal of Geographical Information Science*, 28(3):455–478, 2014. doi:10.1080/13658816.2013.809724. 2, 18

van Oosterom, P. The GAP-tree, an approach to 'on-the-fly' map generalization of an area partitioning. In *GIS and Generalization: Methodology and Practice*, pp. 120–132. CRC Press, 1995. doi:10.1201/9781003062646-12. 32

Vansteenwegen, P., Souffriau, W., and Van Oudheusden, D. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011. doi:10.1016/j.ejor.2010.03.045. 76

Verma, P., Agrawal, K., and Sarasvathi, V. Indoor Navigation Using Augmented Reality. In *Proc. of the Int. Conf. on Virtual and Augmented Reality Simulations (ICVARS'20)*, pp. 58–63. ACM, 2020. doi:10.1145/3385378.3385387. 9

Verner, O. V., Wainwright, R. L., and Schoenefeld, D. A. Placing Text Labels on Maps and Diagrams using Genetic Algorithms with Masking. *INFORMS Journal on Computing*, 9(3):266–275, 1997. doi:10.1287/ijoc.9.3.266. 21

Vincent, K., Roth, R. E., Moore, S. A., Huang, Q., Lally, N., Sack, C. M., Nost, E., and Rosenfeld, H. Improving spatial decision making using interactive maps: An empirical study on interface complexity and decision complexity in the North American hazardous waste trade. *Environment and Planning B: Urban Analytics and City Science*, 46(9):1706–1723, 2019. doi:10.1177/2399808318764122. 8, 124

Wabiński, J. and Mościcka, A. Automatic (Tactile) Map Generation – A Systematic Literature Review. *ISPRS International Journal of Geo-Information*, 8(7):293, 2019. doi:10.3390/ijgi8070293. 18

Wagner, F. and Wolff, A. A practical map labeling algorithm. *Computational Geometry: Theory and Applications*, 7(5-6):387–404, 1997. doi:10.1016/S0925-7721(96)00007-7. 21

Wagner, F. and Wolff, A. A Combinatorial Framework for Map Labeling. In *Graph Drawing: Proc. of the Int. Symp. on Graph Drawing (GD'98)*, volume 1547 of *Lecture Notes in Computer Science*, pp. 316–331. 1998. doi:10.1007/3-540-37623-2_24. 20

Waidyanatha, N. and Frommberger, L. Comprehension and appropriateness of complex mobile pictographs for crisis communication. *Natural Hazards*, 114:583–604, 2022. doi:10.1007/s11069-022-05402-y. 107

Wang, C., Fang, T., and Miao, R. Learning performance and cognitive load in mobile learning: Impact of interaction complexity. *Journal of Computer Assisted Learning*, 34(6):917–927, 2018. doi:10.1111/jcal.12300. 8, 124

Wang, F., Wen, R., and Zhong, S. Key Issues in Mapping Technologies for Disaster Management. In *Proc. of the Int. Conf. on Information Engineering and Computer Science (ICIECS'10)*, pp. 1–4. IEEE, 2010. doi:10.1109/ICIECS.2010.5677703. 109

Webb, N. and Renshaw, T. Eyetracking in HCI. In *Research Methods for Human-Computer Interaction*, pp. 35–69. Cambridge University Press, 2008. doi:10.1017/CBO9780511814570.004. 11

Wei, Z., He, J., Wang, L., Wang, Y., and Guo, Q. A Collaborative Displacement Approach for Spatial Conflicts in Urban Building Map Generalization. *IEEE Access*, 6:26918–26929, 2018. doi:10.1109/ACCESS.2018.2836188. 19

Weibel, R. Amplified Intelligence and Rule-Based Systems. *Map Generalization: Making Rules for Knowledge Representation*, pp. 172–186, 1991. 16

Weibel, R. Three essential building blocks for automated generalization. In *GIS and Generalisation: Methodology and Practice*, pp. 56–69. CRC Press, 1995. doi:10.1201/9781003062646-7. 7, 9, 17

Weibel, R. and Dutton, G. Generalising spatial data and dealing with multiple representations. *Geographical Information Systems: Principles, Techniques, Management and Applications*, pp. 125–155, 1999. 14

Wertheimer, M. Laws of organization in perceptual forms. In *A Source Book of Gestalt Psychology*, pp. 71–88. Routledge and Kegan Paul, London, 1938. doi:10.1037/11496-005. 109

White, T. Symbolization and the Visual Variables. In *The Geographic Information Science & Technology Body of Knowledge*. John P. Wilson (ed.), 2nd quarter 2017 edition, 2017. doi:10.22224/gistbok/2017.2.3. 11, 125

Wieland, J., Garcia, R. C. H., Reiterer, H., and Feuchtner, T. Arrow, Bézier Curve, or Halos? – Comparing 3D Out-of-View Object Visualization Techniques for Handheld Augmented Reality. In *Proc of the IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR'22)*, pp. 797–806. IEEE, 2022. doi:10.1109/ISMAR55827.2022.00098. 13

Wolbers, J. and Boersma, K. The Common Operational Picture as Collective Sensemaking. *Journal of Contingencies and Crisis Management*, 21(4):186–199, 2013. doi:10.1111/1468-5973.12027. 107

Wolfe, J. M. Visual Search: How Do We Find What We Are Looking For? *Annual Review of Vision Science*, 6(1):539–562, 2020. doi:10.1146/annurev-vision-091718-015048. 3

Wolsey, L. A. and Nemhauser, G. L. *Integer and Combinatorial Optimization*, volume 55. John Wiley & Sons, 1999. doi:10.1002/9781118627372. 27

Wu, H.-Y., Takahashi, S., Poon, S.-H., and Arikawa, M. Introducing Leader Lines into Scale-Aware Consistent Labeling. In *Advances in Cartography and GIScience: Selections of the Int. Cartographic Conf. 2017 (ICACI'17)*, Lecture Notes in Geoinformation and Cartography, pp. 117–130. Springer, 2017. doi:10.1007/978-3-319-57336-6_9. 20, 22

Wu, H.-Y., Takahashi, S., Lin, C.-C., and Yen, H.-C. A Zone-Based Approach for Placing Annotation Labels on Metro Maps. In *Smart Graphics: Proc. of the Int. Symp. on Smart Graphics (SG'11)*, volume 6815 of *Lecture Notes in Computer Science*, pp. 91–102. Springer, 2011. doi:10.1007/978-3-642-22571-0_8. 19

Xi, D., Hu, X., Yang, L., Yang, N., Liu, Y., and Jiang, H. Research on map emotional semantics using deep learning approach. *Cartography and Geographic Information Science*, 50(5):465–480, 2023. doi:10.1080/15230406.2023.2172081. 9

Xin, Y. and MacEachren, A. M. Characterizing traveling fans: a workflow for event-oriented travel pattern analysis using Twitter data. *International Journal of Geographical Information Science*, 34(12):2497–2516, 2020. doi:10.1080/13658816.2020.1770259. 9

Yamamoto, D., Ozeki, S., and Takahashi, N. Focus+Glue+Context: An Improved Fisheye Approach for Web Map Services. In *Proc. of the SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (GIS'09)*, pp. 101–110. ACM, 2009. doi:10.1145/1653771.1653788. 13

Yamamoto, M. and Lorena, L. A. A Constructive Genetic Approach to Point-Feature Cartographic Label Placement. volume 32 of *Operations Research/Computer Science Interfaces Series*, pp. 287–302. Springer, 2005. doi:10.1007/0-387-25383-1_13. 21

Yan, X. and Yang, M. A deep learning approach for polyline and building simplification based on graph autoencoder with flexible constraints. *Cartography and Geographic Information Science*, 51(1):79–96, 2023. doi:10.1080/15230406.2023.2218106. 18

Yan, X., Ai, T., Yang, M., and Tong, X. Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. *International Journal of Geographical Information Science*, 35(3): 490–512, 2021. doi:10.1080/13658816.2020.1768260. 18

Yannakakis, M. Expressing combinatorial optimization problems by Linear Programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991. doi:10.1016/0022-0000(91)90024-Y. 25

Yaolin, L., Molenaar, M., Tinghua, A., and Yanfang, L. Frameworks for generalization constraints and operations based on object-oriented data structure in database generalization. *Geo-spatial Information Science*, 4(3):42–49, 2001. doi:10.1007/BF02826923. 14

Yoeli, P. The Logic of Automated Map Lettering. *The Cartographic Journal*, 9(2):99–108, 1972. doi:10.1179/000870472787352505. 19

Yokosuka, Y. and Imai, K. Polynomial Time Algorithms for Label Size Maximization on Rotating Maps. *Journal of Information Processing*, 25:572–579, 2017. doi:10.2197/ipsjjip.25.572. 22

Zadeh, N. A bad network problem for the simplex method and other minimum cost flow algorithms. *Mathematical Programming*, 5(1):255–266, 1973. doi:10.1007/BF01580132. 26

Zhang, J., Xia, X., Liu, R., and Li, N. Enhancing human indoor cognitive map development and wayfinding performance with immersive augmented reality-based navigation systems. *Advanced Engineering Informatics*, 50:101432, 2021. doi:10.1016/j.aei.2021.101432. 9

Zhang, Q. and Harrie, L. Placing Text and Icon Labels Simultaneously: A Real-Time Method. *Cartography and Geographic Information Science*, 33(1):53–64, 2006. doi:10.1559/152304006777323127. 19, 21

Zhang, X., Poon, S.-H., Li, M., and Lee, V. On Maxmin Active Range Problem for Weighted Consistent Dynamic Map Labeling. In *Proc. of the Int. Conf. on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing'15)*, pp. 32–37. IARIA XPS Press, 2015. 22

Zhang, X., Poon, S.-H., Liu, S., Li, M., and Lee, V. C. Consistent dynamic map labeling with fairness and importance. *Computer Aided Geometric Design*, 81:101892, 2020. doi:10.1016/j.cagd.2020.101892. 2, 22

Zhao, R., Ai, T., and Wen, C. A Method for Generating Variable-Scale Maps for Small Displays. *ISPRS International Journal of Geo-Information*, 9(4):250, 2020. doi:10.3390/ijgi9040250. 14

Zhou, Q. and Li, Z. Empirical determination of geometric parameters for selective omission in a road network. *International Journal of Geographical Information Science*, 30(2):263–299, 2016. doi:10.1080/13658816.2015.1085538. 9

Zhou, Q. and Li, Z. A Comparative Study of Various Supervised Learning Approaches to Selective Omission in a Road Network. *The Cartographic Journal*, 54(3):254–264, 2017. doi:10.1179/1743277414Y.0000000083. 18

Zhu, B. and Poon, C. K. Efficient Approximation Algorithms for Multi-label Map Labeling. In *Algorithms and Computation: Proc. of the Int. Symp. on Algorithms and Computation (ISAAC'99)*, volume 1741 of *Lecture Notes in Computer Science*, pp. 143–152. Springer, 1999. doi:10.1007/3-540-46632-0_15. 21

Zhu, R., Lin, D., Jendryke, M., Zuo, C., Ding, L., and Meng, L. Geo-Tagged Social Media Data-Based Analytical Approach for Perceiving Impacts of Social Events. *ISPRS International Journal of Geo-Information*, 8(1):15, 2018. doi:10.3390/ijgi8010015. 9

Zoraster, S. Integer Programming Applied To The Map Label Placement Problem. *Cartographica*, 23(3):16–27, 1986. doi:10.3138/9258-63QL-3988-110H. 21

Zoraster, S. The Solution of Large 0–1 Integer Programming Problems Encountered in Automated Cartography. *INFORMS Operations Research*, 38(5):752–759, 1990. doi:10.1287/opre.38.5.752. 21

Zoraster, S. Practical Results Using Simulated Annealing for Point Feature Label Placement. *Cartography and Geographic Information Systems*, 24(4):228–238, 1997. doi:10.1559/152304097782439259. 21