International Cartographic Association
Association Cartographique Internationale

Multi-page Labeling of Small-screen Maps with a Graph-coloring Approach

Sven Gedicke*, **Benjamin Niedermann**, **Jan-Henrik Haunert**

Institute of Geodesy and Geoinformation, University of Bonn, Germany, [lastname]@igg.uni-bonn.de

* Corresponding Author

Abstract: Annotating small-screen maps with additional content such as labels for points of interest is a highly challenging problem that requires new algorithmic solutions. A common labeling approach is to select a maximum-size subset of all labels such that no two labels constitute a graphical conflict and to display only the selected labels in the map. A disadvantage of this approach is that a user often has to zoom in and out repeatedly to access all points of interest in a certain region. Since this can be very cumbersome, we suggest an alternative approach that allows the scale of the map to be kept fixed. Our approach is to distribute all labels on multiple pages through which the user can navigate, for example, by swiping the pages from right to left. We in particular optimize the assignment of the labels to pages such that no page contains two conflicting labels, more important labels appear on the first pages, and sparsely labeled pages are avoided. Algorithmically, we reduce this problem to a weighted and constrained graph coloring problem based on a graph representing conflicts between labels such that an optimal coloring of the graph corresponds to a multi-page labeling. We propose a simple greedy heuristic that is fast enough to be deployed in web-applications. We evaluate the quality of the obtained labelings by comparing them with optimal solutions, which we obtain by means of integer linear programming formulations. In our evaluation on real-world data we particularly show that the proposed heuristic achieves near-optimal solutions with respect to the chosen objective function and that it substantially improves the legibility of the labels in comparison to the simple strategy of assigning the labels to pages solely based on the labels' weights.

Keywords: interactive maps, map labeling, graph coloring

1. Introduction

Mobile devices with small screens such as smart watches and smart phones have essentially changed the usage of maps in daily life. With those devices maps became an easily accessible medium that helps users explore their surroundings, for example, when searching for amenities or ongoing events. A widespread approach to present the information is to annotate the currently displayed map section with labels such as text or small icons. However, depending on the amount of information to be presented this might yield a map with overlapping labels. This effect can be avoided by presenting only a selection of the labels. Still, this might be unsatisfactory for the user when the selection does not reflect their interests adequately. Alternatively, the possibility of zooming in can be used to reveal information of large scale maps such that the density of information is small enough to present all labels without overlaps. The major drawback of this approach, however, is that the user needs to zoom in and out repetitively to explore the currently displayed map section entirely. As zooming in leads to a smaller area that is displayed, the user easily loses the overall context.

Several strategies have been proposed to reduce the necessity of zooming to explore the map and its annotations.

For example focus+context maps present the focus region of the user in a large-scale map, while the context region around the focus region is shown as a small-scale map. Yamamoto et al. (2009) introduced an intermediate, distorted region in between the focus and context region to obtain a smooth transition between both maps. However, due to the different scales of focus and context region, the overall map content is strongly distorted. Another approach is to leave the map undistorted and to move the labels outside of the focus region (Bertini et al., 2009; Fink et al., 2012; Haunert and Hermes, 2014). The visual connection between labels and their point feature is then established by connecting lines. While this technique helps to uncover the underlying map within the focus region, it does not solve the problem that labels might overlap or only a selection of the labels is shown.

In this paper we consider a different approach by distributing the information on multiple pages. Each page shows the current section of the map and a selection of the labels. While the map layer is fixed, the user can navigate through the labels, e.g., by swiping the pages from right to left; see Figure 1 for an illustration. The computational problem is then to find an appropriate distribution of the labels over a certain number of pages. We call this problem *multi-page labeling*. Related approaches are used in web applications of online services like Google[1] or Airbnb[2], e.g., for displaying hotel search results. However, these approaches do not avoid overlapping labels.

A simple strategy might optimize the label placement by creating the pages subsequently, 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 goal 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

[1] www.google.com [2] www.airbnb.com

Advances in Cartography and GIScience of the International Cartographic Association, 2, 2019.
15th International Conference on Location Based Services, 11–13 November 2019, Vienna, Austria. This contribution underwent double-blind peer review based on the full paper | https://doi.org/10.5194/ica-adv-2-4-2019 |

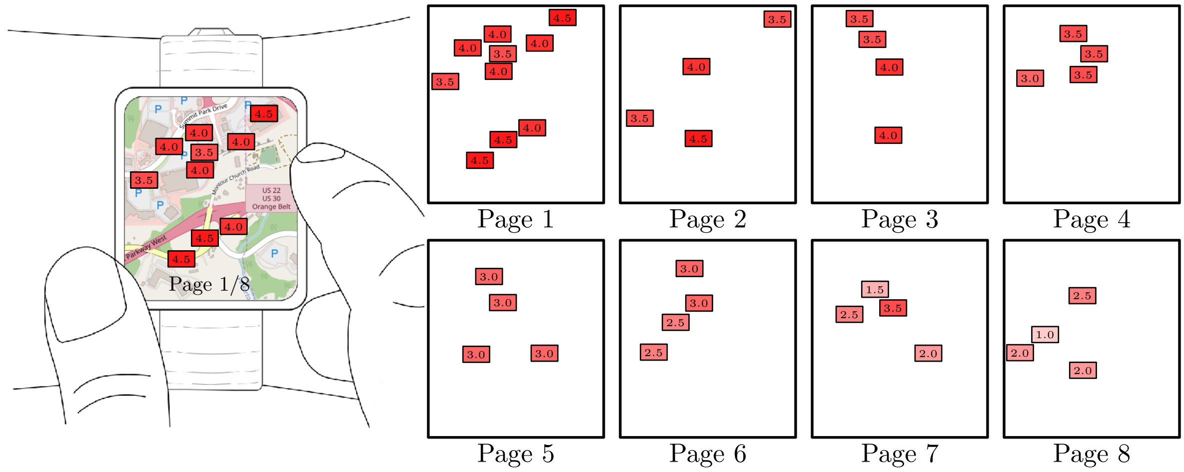

Figure 1. Optimal multi-page labeling schematically presented on a smart watch and produced with our approach. The user navigates through the pages by swiping them from right to left. Only for Page 1 we have added a background map for illustration. The saturation of the color and the inscriptions of the labels represent the weight of the labels. Map data ©OpenStreetMap contributors 2019.

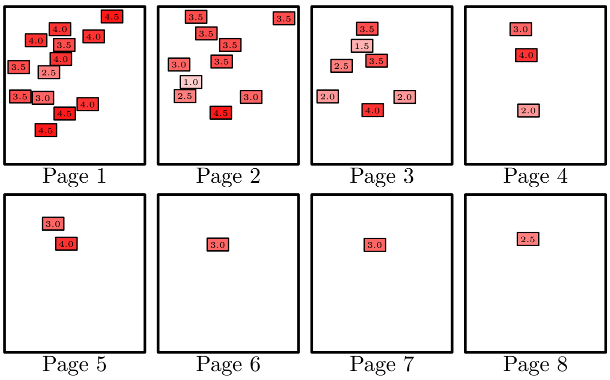

Figure 2. Example of a multi-page labeling of the same instance as in Figure 1 for the case that the pages are created sequentially. While the first page is densely packed with labels, the last three pages only contain one label each.

finding a maximum weight independent set within the according conflict graph of the remaining labels, i.e., a set of conflict-free 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 2 shows an example that we have obtained in such a way using a simple integer linear programming formulation for maximum weight independent set (Haunert and Wolff, 2017). We observe that the last three pages only contain one label each, while the first two pages are densely packed. Swiping through such 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 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 swipes necessary to navigate through the pages, but also yields a packing of the labels exploiting the available space on each page. On the other hand, merely optimizing the number of pages does not satisfactorily map the semantic information of the labels onto the pages. 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.

Hence, we assign to each label a weight and to each page a factor representing the percentage of the label's weight that is taken into account when the label is placed on this page. We call the thus obtained weight the *effective weight* of the label. The labels should then be distributed such that the mean effective weight of the labels is maximized, but at the same time sparsely labeled pages should be avoided. 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.

C1 A small number of pages is preferable in order to reduce the interaction necessary by the user.
C2 Labels of more important features should appear on the first pages of the multi-page labeling.
C3 Sparsely labeled pages should be avoided.

After discussing related work in Section 2, we formalize the problem of computing multi-page labelings as an optimization problem based on graph coloring in Section 3. We particularly introduce three problem settings with increasing generality. While the first problem setting only considers Criterion C1, the second one additionally considers Criterion C2 and the last one considers all three criteria. In Section 4 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 real-time applications. As we are interested in interactive scenarios that require the computation of labelings in real-time, we further present a simple but fast greedy heuristic in Section 5. In Section 6 we then present our evaluation of the algorithms on real-world data.

2. Related Work

As label placement is a highly time consuming task in map making (Morrison, 1980), much research has been invested into the development of efficient algorithms for automatic label placement. A common way to model the problem is to build a conflict graph $G = (V, E)$ based on a set of label candidates for each feature, 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 reduces to identifying a maximum (weight) independent set in G (Wagner and Wolff, 1998). 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). Hence, heuristics (e.g., Christensen et al. (1994)), approximation algorithms (e.g., van Kreveld et al. (1999)) and exact algorithms based on integer linear programming (e.g., Haunert and Wolff (2017)) have been developed.

With the rise of the digital age maps became dynamic requiring adaptive label placement. Basic operations such as *zooming*, *panning* and *rotation* have been considered to provide the user with the possibility of interactively adjusting the displayed map section concerning their requirements. However, the newly gained degree of flexibility requires that the map content and especially the placement of labels is modified during user interaction. Been et al. (2006) modeled this by assigning to each label an activity interval in which the label is displayed, e.g., for zooming each label has an interval of scales during which it is presented to the user. This work was the starting point for further considerations on zooming (Been et al., 2010), rotating maps (Gemsa et al., 2016a,b) and more generally maps that change over time (Barth et al., 2016; Gemsa et al., 2013) considering activity intervals. They all have in common that the core of the problem can still be seen as finding an independent set in a (more sophisticated) conflict graph. In contrast, in this paper we reduce the considered labeling problem more generally to finding a coloring in the given conflict graph, which implies that the existing techniques for label placement are not directly applicable.

As internally placed labels easily clutter the map, also techniques for placing labels in the margins of the map have been extensively considered in research. To ensure a clear visual association, features and labels are typically connected via thin lines, so-called *leaders*. For static maps the labels are placed along the boundary of the map, which became known as *boundary labeling* (Bekos et al., 2004). For dynamic maps research has focused on labeling features within a small dynamic focus region such that the labels are placed outside of the focus region. This problem has become known as *excentric labeling*. If the labels are required to touch the boundary of the focus region (Fink et al., 2012; Haunert and Hermes, 2014), this technique is mostly useful for uncovering the underlying map in the focus region, but it does not provide more space for placing labels. In contrast, when labels might also be placed in the entire region outside of the focus region (Fekete and Plaisant, 1999; Bertini et al., 2009; Heinsohn et al., 2014; Balata et al., 2014), substantially more labels can be placed. Still, in both settings excentric labeling is hardly applicable in small-screen devices such as smart watches because the screen size usually restricts the displayed map section to a small focus region without the possibility of displaying a context region. We therefore feel that multi-page labeling using internally placed labels is a reasonable alternative for existing labeling techniques when it comes to small-screen devices.

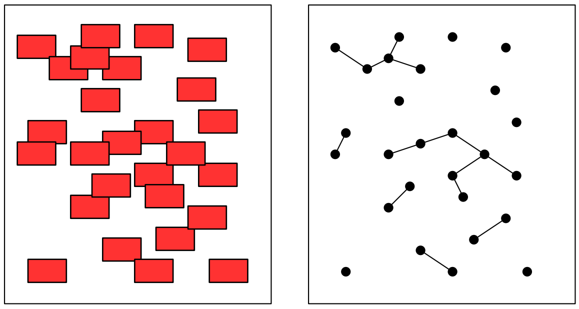

Figure 3. Example of a map with labels (left) and the corresponding conflict graph (right): each label corresponds to one vertex in the graph, two vertices are adjacent in the conflict graph if and only if the according labels intersect.

3. Model

In this section we introduce a formal model that we use throughout this paper. We assume that we are given a set P of point features within a pre-defined region $R \subset \mathbb{R}^2$. Each point feature $p \in P$ has a label that we represent by an axis-parallel rectangle with center at p; 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. Further, 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. 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 rectangles of 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 coloring problem is NP-hard (Imai and Asano, 1983).

We are now ready to introduce the considered optimization problems. We start with the certainly most basic variant minimizing the number of pages.

Problem 1 (MINIMUMNUMBEROFPAGES) *Given a set of labels L, find the smallest multi-page labeling $\mathcal{L} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ of L, i.e., there is no multi-page labeling of L that has fewer than k pages.*

By guaranteeing a minimum number of swipes 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. 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 necessary to view the page containing the label. In our experiments we choose h to be an exponentially decreasing function assuming that the interest of the user substantially decreases with the number of performed swipes. Altogether, this directly leads to the following optimization problem.

Problem 2 (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 $\mathcal{L} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ that maximizes the mean effective label weight of $\mathcal{L}$, i.e., that maximizes $\frac{1}{n}\sum_{i=1}^{k}\sum_{\ell\in\mathcal{P}_i} h(i)\cdot w(\ell)$.*

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

Problem 3 (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 $\mathcal{L} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ that maximizes*

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

where $n_{\mathcal{L}} = \min_{\forall i\in\{1,\ldots,k\}}\{\mid \mathcal{P}_i \mid\}$.

4. Mathematical Programming

In order to compare our algorithms with respect to mathematically optimal solutions, we use integer linear programming formulations (ILPs) for solving Problems 1–3 optimally. These are rather straightforward formulations, but for completeness and the convenience of the reader we present them here.

We first present 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 $x_{\ell,i} \in \{0,1\}$. We interpret $x_{\ell,i}$ such that $x_{\ell,i} = 1$ if the label ℓ is placed on the i-th page.

In order to obtain a valid multi-page labeling we introduce for each page i with $1 \le i \le k$ and each pair of labels ℓ and ℓ' with $\{\ell,\ell'\} \in E$ the constraint

$$x_{\ell,i} + x_{\ell',i} \le 1. \qquad (2)$$

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

$$\sum_{i=1}^{k} x_{\ell,i} = 1 \qquad (3)$$

Hence, a valid multi-page labeling is defined as $\mathcal{L} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ with $\mathcal{P}_i = \{\ell \in L \mid x_{\ell,i} = 1\}$. Subject to Constraint (2) and Constraint (3) we solve Problem 1 by maximizing the objective

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

and we solve Problem 2 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 Problem 3 we introduce an additional integer variable z, which we interpret as the minimum number of labels on any non-empty page. We further introduce a binary variable $y_i \in \{0,1\}$ for each page i with $1 \le i \le k$. With the following constraint we enforce that $y_i = 1$ if page i is actually used, i.e., is not empty.

$$\sum_{\ell\in L} x_{\ell,i} \le n\cdot y_i \qquad (4)$$

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

$$z \le \sum_{\ell\in L} x_{\ell,i} + (1-y_i)\cdot n. \qquad (5)$$

Subject to Constraints (2)–(5) we maximize

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

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

In our experiments we use specialized solvers 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 is too slow to be deployed in practice for real-time scenarios.

5. Greedy Heuristic

In this section we describe a simple but fast heuristic that computes a multi-page labeling for a given label set. The heuristic consists of two phases; Figure 4 illustrates both phases using an exemplary instance. In the first phase the heuristic creates a multi-page labeling optimizing the average effective label weight taking the objective of Problem 2 into account. The second phase is optional and improves that labeling with respect to the minimum number of labels per page without decreasing the value of the composed objective (1) of Problem 3. Hence, the result is a multi-page labeling that is geared towards optimizing Problem 3.

15th International Conference on Location Based Services, 11–13 November 2019, Vienna, Austria. This contribution underwent double-blind peer review based on the full paper | https://doi.org/10.5194/ica-adv-2-4-2019 |

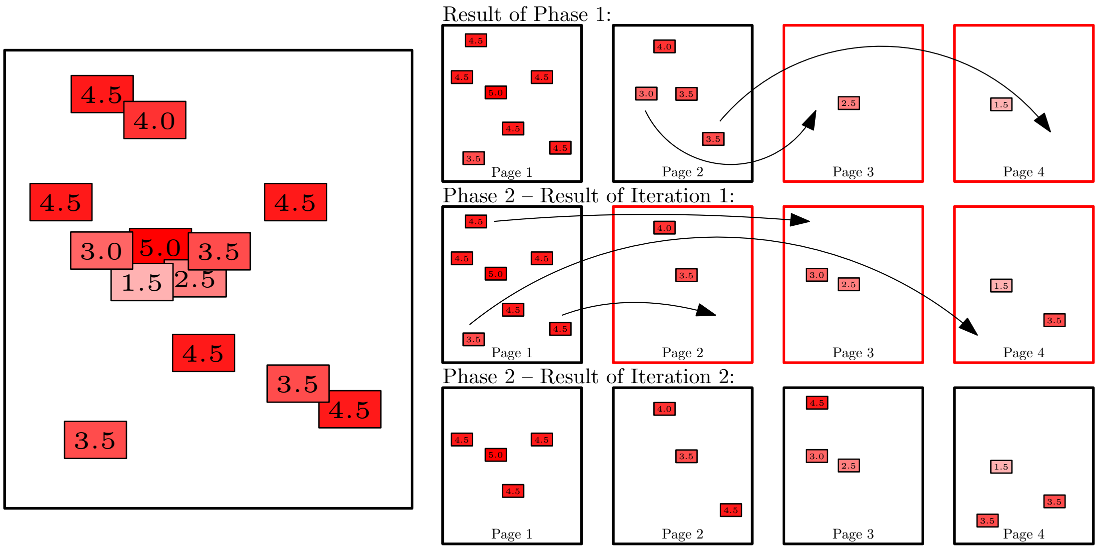

Figure 4. Illustration of the greedy heuristic using an exemplary instance. The example instance is shown on the left. On the right side, the resulting multi-page labelings are presented. The first row shows the result of Phase 1, the second row shows the result of the first iteration of Phase 2, and the last row shows the result of the second and final iteration of Phase 2. Sparsest pages are framed red and arrows indicate which labels are moved in the following step.

Algorithm 1: Iteration of Phase 2

Data: Multi-page labeling $\mathcal{L}$
Result: Improved multi-page labeling $\mathcal{L}'$
foreach *sparsest page* $R \in \mathcal{L}$ *in reverse order* **do**
  foreach *surplus page* $Q \in \mathcal{L}$ *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 (1) is not improved* **then**
  restore the input labeling

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. To that end, 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 consisiting 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. If such a page does not exist, a new page is appended to the sequence and the label is placed on this page. By construction 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 function (1) of Problem 3. In each iteration we first identify all pages with the minimum number of labels, which we call *sparsest pages*; Algorithm 1 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 preserving a 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 function (1) of Problem 3. 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 4 two iterations of Phase 2 led to improvements of the objective function (1) of Problem 3.

6. Experiments

The following section presents the experiments we have carried out. First the experimental setup is presented and afterwards the results of the experiments are discussed.

6.1 Experimental Setup

A data set of point features provided by the recommendation portal Yelp[3] 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. For each point feature we introduce an axis-parallel rectangular label. All labels have the same height and width. In order to ensure a balanced ratio between the size of the labels and the displayed map area, we have chosen the size based on existing digital maps; the ratio between label width and screen width is 7:1 and the ratio between label height and screen height is 12:1. Each label is placed with its center at its point feature.

Assuming that higher-rated businesses are more relevant to the 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, the interest of a user substantially decreases.

[3] www.yelp.com

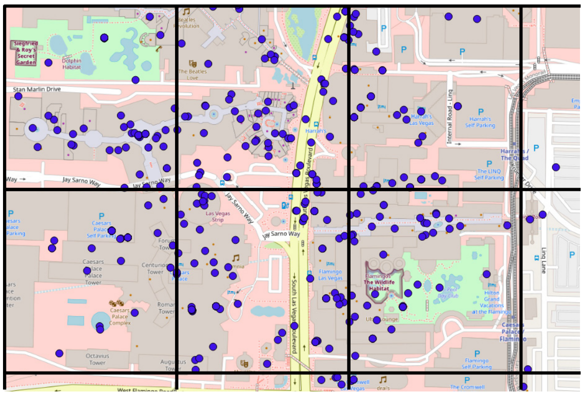

Figure 5. Map showing an example of the data used in the evaluation. The underlying grid is used for sampling the data. Map data ©OpenStreetMap contributors 2019.

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.

Keeping the use-case of small smart devices in mind, we determined the scale and size of the displayed map section based on initial experiments. The scale is 1:15000 and the displayed section covers an area of 1000 by 1100 meters. Using this setting we created 248 instances for multi-page labeling. To that end, we moved the displayed section grid-wise over the complete data set; see Figure 5 for an illustration. For an instance we took all labels entirely contained in the according map section. We only consider instances with at least 20 labels to obtain reasonably large labelings. In total we have obtained instances whose sizes range from 20 to 50 labels. In our experiments we only considered instance sizes for which we could create at least eight instances; for each size we took the first eight of those.

We optimally solve Problems 1–3 using the ILP formulations of Section 4. In our evaluation we denote Problem 1 by *MinPage*. Further, we denote Problem 3 by *BiCriteriaX* with $X \in \{0, 25, 100\}$ where we set the balance factor $\alpha = \frac{X}{100}$. We note that *BiCriteria0* corresponds to Problem 2 and *BiCriteria100* only takes the minimum number of labels per page into account. In preliminary experiments, considering 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*.

Our implementation of the ILP formulations is written in Java and solved by Gurobi[4] 8.1.0. The greedy heuristic is implemented in JavaScript in order to provide a realistic setup simulating a digital map in a web-interface.

6.2 Evaluation of Objectives

In the following we discuss the evaluation of the optimization problems of Section 3. For this purpose, Figure 6 shows the comparison of optimal solutions of different objective functions regarding three different criteria.

Figure 6a) shows the average number of pages. We observe that BiCriteria100 exceeds the minimum number of pages by 1.3 on average and 3.1 in maximum, which means that 9.9% more pages are necessary for hosting all labels on average and 18.9% in maximum. As BiCriteria100 only maximizes the minimum number of labels per page, an optimal solution for BiCriteria100 might contain unnecessary pages whose labels could be redistributed on other pages. In comparison, both BiCriteria0 and BiCriteria25 achieve near optimal results, which we do not depict in Figure 6a) as they mostly coincide with the results of MinPage. In particular BiCriteria25 leads to an average number of pages that is only 0.2% above the minimum number of pages on average and 2.3% in maximum. Concerning Problem 3, this shows that an almost minimal number of pages can be achieved with a suitable choice of α, i.e., optimizing both criteria C2 and C3 indirectly minimizes the number of pages as well.

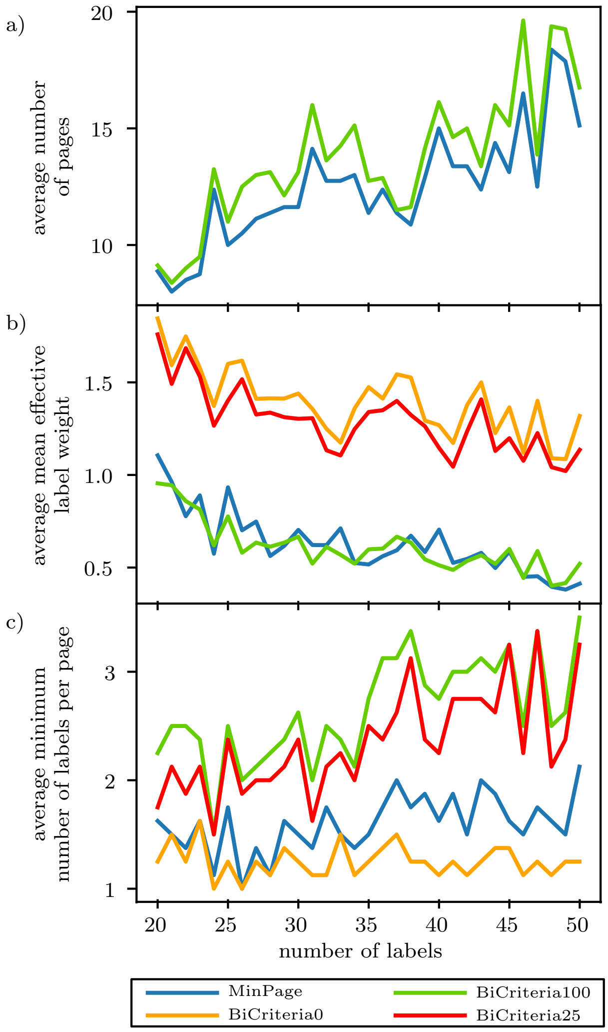

Figure 6. Evaluated parameters. X-axis: Number of labels of the considered instances. For each size eight instances have been considered. Y-axis: The average value of the eight instances with respect to the considered parameter. In a): BiCriteria0 and BiCriteria25 are not displayed, because they mostly coincide the results of MinPage.

Next, we consider the mean effective label weight; see Figure 6b). We observe that MinPage and BiCriteria100 achieve 45% and 44% of the maximum mean effective label weight on average and 44% and 36% in minimum, which was to be expected as both objectives do not take the label weights into account. In contrast, BiCriteria25 yields

[4] www.gurobi.com

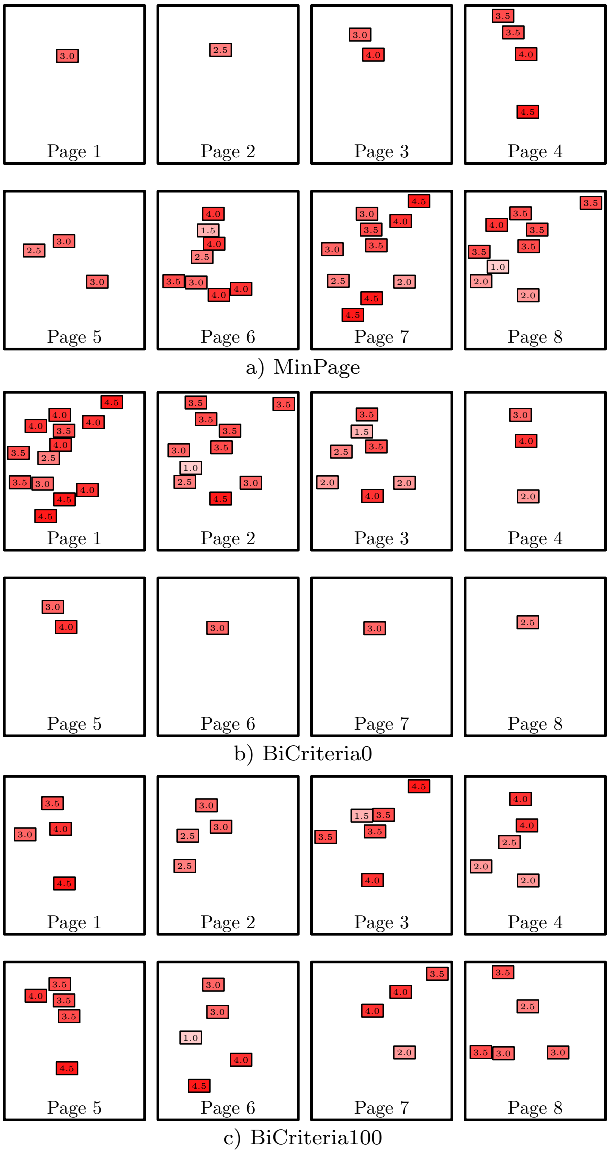

Figure 7. Optimal multi-page labelings of the instance shown in Figure 1 with respect to the objectives a) MinPage b) BiCriteria0 and c) BiCriteria100.

solutions that come close to the maximum mean effective label weight with 92% on average and 87% in minimum.

As a last criterion, we consider the average minimum number of labels per page in Figure 6c). On average MinPage and BiCriteria0 lead to average minimum numbers of labels per page of 1.6 and 1.3 respectively, which is only 61% and 49% of the optimal solution of the minimum number of labels per page. MinPage and BiCriteria0 only reach 52% and 36% of the optimal solution of the minimum number of labels per page in minimum, which corresponds to a label difference of 1.6 and 2.3, respectively. A better result is achieved by BiCriteria25, which reaches 89% of the optimal solution of the minimum number of labels per page on average and 76% in minimum.

Figure 7 shows exemplary labelings of the same instance for MinPage, BiCriteria0 and BiCriteria100. The corresponding labeling of BiCriteria25 is shown in Figure 1. 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 7a). BiCriteria0, which optimizes the average effective label weight, results in pages that are sparsely labeled; see Figure 7b). For BiCriteria100, which maximizes the minimum number of labels per page, the labels are distributed without any priority; see Figure 7c).

The labeling provided by BiCriteria25 (see Figure 1) also achieves eight pages, but in contrast to the other objectives it yields a good compromise between Criteria C1–C3. It provides the minimum number of pages, while more important labels are distributed on front pages. Moreover, sparsely labeled pages are avoided. Altogether, the evaluation shows that for an appropriate choice of α Problem 3 is a suitable optimization for finding multi-page labelings.

6.3 Evaluation of Greedy Heuristic

In the following we evaluate the quality of the greedy heuristic from Section 5. Note that in the first phase the heuristic optimizes the average effective label weight. To that end, we first compare its results after Phase 1 with the ones of BiCriteria0 regarding the objective of Problem 2; see Figure 8a). The first phase achieves 94% of the optimal solution on average and 89% in minimum. After additionally performing Phase 2, we compare our approach with BiCriteria25 regarding the obective function of Problem 3; see Figure 8b). The heuristic achieves 96% of the optimal solution on average and 93% in minimum.

In summary our greedy heuristic provides results that are close to optimal for both Problem 2 and Problem 3. In particular for Problem 3 this is to be emphasized, since all criteria for multi-page labeling are taken into account here.

In order to investigate the applicability of the greedy heuristic in web-applications, the running time is also considered. We implemented the heuristic in JavaScript[5] and ran our experiments on an Intel Core i7-3770K CPU clocked at 3.5 GHz. For all instances, only running Phase 1 took 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 application on a smart watch, on which it operates smoothly. As these results show, our approach is fast enough to be applied to realistic instance sizes in real time.

7. Conclusion

In this paper we have considered the problem of label placement in maps for small-screen devices. In contrast to previous work, we resolved label overlaps neither by presenting only a subset of labels nor by moving them apart but followed the simple idea to distribute them on multiple pages. Altogether, we can draw the following scientific conclusions from our experiments:

- 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-criterial model.
- When additionally optimizing Criteria C2 and C3, which deal with label weights and balancing of pages, the number of pages is still close to the minimal possible number.

[5] http://www2.geoinfo.uni-bonn.de/html/MultiPageLabeling/

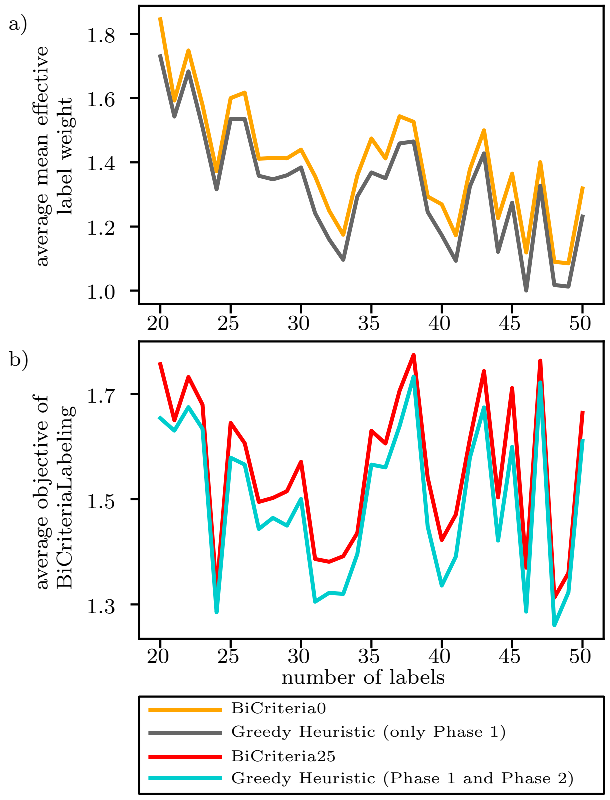

Figure 8. Evaluated parameters. X-axis: Number of labels of the considered instances. For each size eight instances have been considered. Y-axis: The average value of the eight instances with respect to the considered parameter.

- Our simple and fast greedy heuristic (with both phases) yields solutions that are close to optimal with respect to our mathematical model, namely 96% on average and 93% in minimum.

In future work, small label shifts or slight overlaps might be allowed to further improve all three criteria. We also plan a user study to evaluate the usability of our model.

8. Acknowledgements

This research was supported by *Zoomless Maps: Models and Algorithms for the Exploration of Dense Maps with a Fixed Scale* of the German Research Foundation (DFG) [grant number 5451/6-1].

References

Agarwal, P. K., van Kreveld, M. and Suri, S., 1998. Label placement by maximum independent set in rectangles. *Comput. Geom. Theory Appl.* 11(3), pp. 209–218.

Balata, J., Čmolík, L. and Mikovec, Z., 2014. On the selection of 2d objects using external labeling. In: *Human Factors in Computing Systems (CHI'14)*, ACM, pp. 2255–2258.

Barth, L., Niedermann, B., Nöllenburg, M. and Strash, D., 2016. Temporal map labeling: a new unified framework with experiments. In: *Proc. 24th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (ACM-GIS'16)*, p. 23.

Been, K., Daiches, E. and Yap, C., 2006. Dynamic map labeling. *IEEE Trans. Visual Comput. Graphics* 12(5), pp. 773–780.

Been, K., Nöllenburg, M., Poon, S.-H. and Wolff, A., 2010. Optimizing active ranges for consistent dynamic map labeling. *Comput. Geom. Theory Appl.* 43(3), pp. 312–328.

Bekos, M. A., Kaufmann, M., Symvonis, A. and Wolff, A., 2004. Boundary labeling: Models and efficient algorithms for rectangular maps. In: *12th Int. Symp. on Graph Drawing (GD'04)*, LNCS, Vol. 3383, Springer, pp. 49–59.

Bertini, E., Rigamonti, M. and Lalanne, D., 2009. Extended excentric labeling. *Comput. Graphics Forum* 28(3), pp. 927–934.

Christensen, J., Marks, J. and Shieber, S., 1994. Placing text labels on maps and diagrams. In: *Graphics Gems IV*, Academic Press Professional, Inc., pp. 497–504.

Fekete, J.-D. and Plaisant, C., 1999. Excentric labeling: Dynamic neighborhood labeling for data visualizaition. In: *Human Factors in Computing Systems (CHI'99)*, ACM Press, pp. 512–519.

Fink, M., Haunert, J., Schulz, A., Spoerhase, J. and Wolff, A., 2012. Algorithms for labeling focus regions. *IEEE Trans. Visual Comput. Graphics* 18(12), pp. 2583–2592.

Fowler, R. J., Paterson, M. S. and Tanimoto, S. L., 1981. Optimal packing and covering in the plane are np-complete. *Inf. Process. Lett.* 12(3), pp. 133 – 137.

Gemsa, A., Niedermann, B. and Nöllenburg, M., 2013. Trajectory-based dynamic map labeling. In: *Algorithms and Computation (ISAAC'13)*, LNCS, Vol. 8283, Springer, pp. 413–423.

Gemsa, A., Nöllenburg, M. and Rutter, I., 2016a. Consistent labeling of rotating maps. *J. Comput. Geom.* 7(1), pp. 308–331.

Gemsa, A., Nöllenburg, M. and Rutter, I., 2016b. Evaluation of labeling strategies for rotating maps. *ACM J. Exp. Algorithmics* 21(1), pp. 1–21.

Guan, D. and Xuding, Z., 1997. A coloring problem for weighted graphs. *Inf. Process. Lett.* 61(2), pp. 77–81.

Gyárfás, A. and Lehel, J., 1988. On-line and first fit colorings of graphs. *J. Graph Theory* 12(2), pp. 217–227.

Haunert, J.-H. and Hermes, T., 2014. Labeling circular focus regions based on a tractable case of maximum weight independent set of rectangles. In: *Proc. 2nd ACM SIGSPATIAL Int. Workshop on Interacting with Maps (MapInteract'14)*, pp. 15–21.

Haunert, J.-H. and Wolff, A., 2017. Beyond maximum independent set: An extended integer programming formulation for point labeling. *ISPRS Int. J. Geo-Inf.* 6(11), pp. 342.

Heinsohn, N., Gerasch, A. and Kaufmann, M., 2014. Boundary labeling methods for dynamic focus regions. In: *IEEE Pac. Vis. Symp. (PacificVis'14)*, pp. 243–247.

Imai, H. and Asano, T., 1983. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *J. Algo.* 4(4), pp. 310–323.

Morrison, J. L., 1980. Computer technology and cartographic change. In: *The Computer in Contemporary Cartography*, Johns Hopkins University Press, pp. 5–24.

van Kreveld, M., Strijk, T. and Wolff, A., 1999. Point labeling with sliding labels. *Comput. Geom. Theory Appl.* 13(1), pp. 21–47.

Wagner, F. and Wolff, A., 1998. A combinatorial framework for map labeling. In: *6th Int. Symp. on Graph Drawing (GD'98)*, LNCS, Vol. 1547, Springer, pp. 316–331.

Yamamoto, D., Ozeki, S. and Takahashi, N., 2009. Focus+glue+context: An improved fisheye approach for web map services. In: *Proc. 17th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (ACM-GIS'09)*, pp. 101–110.

