Two-Sided Boundary Labeling with Adjacent Sides

Philipp Kindermann* Benjamin Niedermann† Ignaz Rutter‡ Marcus Schaefer‡ André Schulz§ Alexander Wolff†

Abstract

In the *Boundary Labeling* problem, we are given a set of n points, referred to as *sites*, inside an axis-parallel rectangle R, and a set of n pairwise disjoint rectangular labels that are attached to R from the outside. The task is to connect the sites to the labels by non-intersecting polygonal paths, so-called *leaders*.

In this paper, we study the *Two-Sided Boundary Labeling with Adjacent Sides* problem, with labels lying on two adjacent sides of the enclosing rectangle. We restrict ourselves to rectilinear leaders with at most one bend. We present a polynomial-time algorithm that computes a crossing-free leader layout if one exists. So far, such an algorithm has only been known for the simpler cases that labels lie on one side or on two opposite sides of R (where a crossing-free solution always exists).

1 Introduction

Label placement is an important problem in cartography and, more generally, information visualization. Features such as points, lines, and regions in maps, diagrams, and technical drawings often have to be labeled so that users understand better what they see. The general label-placement problem is NP-hard [6], which explains why labeling a map manually is a tedious task that has been estimated to take 50% of total map production time [5].

Boundary labeling can be seen as a graph-drawing problem where the class of graphs to be drawn is restricted to matchings.

Problem statement. Following Bekos et al. [2], we define the BOUNDARY LABELING problem as follows. We are given an axis-parallel rectangle $R = [0,0] \times [W,H]$, which is called the *enclosing rectangle*, a set $P = \{p_1, \ldots, p_n\} \subset R$ of n points, called *sites*, within the rectangle R, and a set L of $m \le n$ axis-parallel rectangles $\ell_1, \ldots, \ell_m$, called *labels*, that lie in the complement of R and touch the boundary of R. No two labels overlap. We denote an instance of the problem by the triplet (R, L, P). A *solution* to the problem is a set of m curves $c_1, \ldots, c_m$, called *leaders*, that connect sites to labels such that the leaders a) produce a matching between the labels and (a subset of) the sites, b) are contained inside R, and c) touch the associated labels on the boundary of R.

A solution is *planar* if the leaders do not intersect. Note that we do not prescribe which site connects to which label. The endpoint of a leader at a label is called a *port*. We distinguish two incarnations of the BOUNDARY LABELING problem: either the position of the ports on the boundary of R is fixed and part of the input, or the ports *slide*, i.e., their exact location is not prescribed.

We restrict our solutions to *po-leaders*, that is, starting at a site, the first line segment of a leader is parallel (p) to the side of R containing the label it leads to, and the second line segment is orthogonal (o) to that side. Bekos et al. [1, Fig. 12] observed that not every instance (with $m = n$) admits a planar solution with *po*-leaders where all sites are labeled.

Previous work. For *po*-labeling, Bekos et al. [2] gave a simple quadratic-time algorithm for the one-sided case that, in a first pass, produces a labeling of minimum total leader length by matching sites and ports from bottom to top. In a second pass, their algorithm removes all intersections without increasing the total leader length. This result was improved by Benkert et al. [3] who gave an $O(n \log n)$-time algorithm for the same objective function and an $O(n^3)$-time algorithm for a very general class of objective functions, including, for example, bend minimization. They extend the latter result to the two-sided case (with labels on opposite sides of R), resulting in an $O(n^8)$-time algorithm. For the special case of leader-length minimization, Bekos et al. [2] gave a simple dynamic program running in $O(n^2)$ time. All these algorithms work both for fixed and sliding ports.

Our contribution. We investigate the problem TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES where all labels lie on two *adjacent* sides of R,

*Lehrstuhl für Informatik I, Universität Würzburg, Germany. WWW: www1.informatik.uni-wuerzburg.de/en/staff

†Fakultät für Informatik, Karlsruher Institut für Technologie (KIT), Germany. Email: {rutter, benjamin.niedermann} @kit.edu

‡College of Computing and Digital Media, DePaul University, Chicago, IL, USA. Email: mschaefer@cs.depaul.edu

§Institut für Mathematische Logik und Grundlagenforschung, Universität Münster, Germany. Email: andre.schulz@uni-muenster.de

for example, on the top and right side. Note that point data often comes in a coordinate system; then it is natural to have labels on adjacent sides (for example, opposite the coordinate axes). We argue that this problem is more difficult than the case where labels lie on opposite sides, which has been studied before: with labels on opposite sides, (a) there is always a solution where all sites are labeled (if $m = n$) and (b) a feasible solution can be obtained by considering two instances of the one-sided case.

Our result is an algorithm that, given an instance with n labels and n sites, decides whether a planar solution exists where all sites are labeled and, if yes, computes a layout of the leaders (see Section 3). We use dynamic programming to "guess" a partition of the sites into the two sets that are connected to the leaders on the top side and on the right side. The algorithm runs in $O(n^2)$ time and uses $O(n)$ space.

Notation. We call the labels that lie on the right (top) side of R *right (top) labels*. The *type* of a label refers to the side of R on which it is located. The *type* of a leader (or a site) is simply the type of its label. We assume that no two sites lie on the same horizontal or vertical line, and no site lies on a horizontal or vertical line through a port or an edge of a label.

For a solution $\mathcal{L}$ of a boundary labeling problem, we define the total length of all leaders in $\mathcal{L}$ by $\text{length}(\mathcal{L})$.

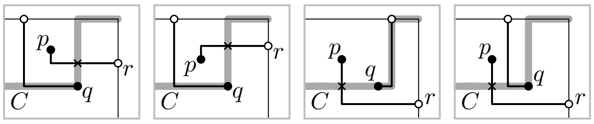
2 Structure of Planar Solutions

In this section, we attack our problem presenting a series of structural results of increasing strength. For simplicity, we assume fixed ports. For sliding ports, we can simply fix all ports to the bottom-left corner of their corresponding labels. First we show that we can split a planar two-sided solution into two one-sided solutions by constructing an xy-monotone, rectilinear curve from the top-right to the bottom-left corner of R. Afterwards, we provide a necessary and sufficient criterion to decide whether for a given separation there exists a planar solution. This will form the basis of our dynamic programming algorithm, which we present in the next section.

Definition 1 *We call an xy-monotone, rectilinear curve connecting the top-right to the bottom-left corner of R an* xy-separating curve. *We say that a planar solution to* Two-Sided Boundary Labeling with Adjacent Sides *is* xy-separated *if and only if there exists an xy-separating curve C such that*
a) the top sites and all their leaders lie on or above C
b) the right sites and all their leaders lie below C.

It is not hard to see that a planar solution is not xy-separated if there exists a site p that is labeled to the right side and a site q that is labeled to the top side with $x(p) < x(q)$ and $y(p) > y(q)$. There are exactly four patterns in a possible planar solution that satisfy this condition. Moreover, these patterns are the only ones that can violate xy-separability.

Lemma 1 *A planar solution is xy-separated iff it does not contain any of the following patterns P1–P4*



(P1) (P2) (P3) (P4)

Next, we claim that any planar solution can be transformed into an xy-separated planar solution. Our proof shows that each of the four patterns of Lemma 1 can be resolved by rerouting leaders such that no crossings arise and the leader length decreases. We cannot present the proof due to space constraints.

Proposition 1 *If there exists a planar solution $\mathcal{L}$ to* Two-Sided Boundary Labeling with Adjacent Sides, *then there exists an xy-separated planar solution $\mathcal{L}'$ with* $\text{length}(\mathcal{L}') \leq \text{length}(\mathcal{L})$.

Since every solvable instance of Two-Sided Boundary Labeling with Adjacent Sides admits an xy-separated planar solution, it suffices to search for such a solution. Moreover, an xy-separated planar solution that minimizes the total leader length is a solution of minimum length. In Lemma 2 we provide a necessary and sufficient criterion to decide whether, for a given xy-monotone curve C, there is a planar solution that is separated by C. We denote the region of R above C by R_T and the region of R below C by R_R. For each horizontal segment of C consider the horizontal line through the segment. We denote the parts of these lines within R by $h_1, \ldots, h_k$, respectively. Further let h_0 be the top edge of R. The line segments $h_1, \ldots, h_k$ partition R_T into k strips, which we denote by $S_1, \ldots, S_k$ from top to bottom, such that each strip S_i is bounded by h_i from below for $i = 1, \ldots, k$; see Fig. 1a. Additionally, we define S_0 to be the empty strip that coincides with h_0. Note that this strip cannot contain any site of P. For any point p on one of the horizontal lines h_i inside R, we define the rectangle R_p, spanned by the top-right corner of R and p. We define R_p such that it does *not* contain its top-left corner. In particular, we consider the port of a top label as contained in R_p, only if it is not the upper left corner.

A rectangle R_p is *valid* if the number of sites of P above C that belong to R_p is at least as large as the number of ports on the top side of R_p. The central idea is that the sites of P inside a valid rectangle R_p can be connected to labels on the top side of the valid

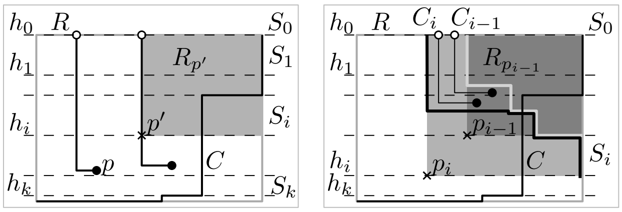


Fig. 1: The 1-sided strip condition. a) The horizontal segments of C induce the strips $S_0, S_1, \ldots, S_k$. b) Constructing a planar labeling from a sequence of valid rectangles.

rectangle by leaders that are completely contained inside the rectangle. We are now ready to present the *1-sided strip condition.*

Condition 1 *The* 1-sided strip condition *of strip S_i is satisfied if there exists a point $p_i \in h_i \cap R_\mathrm{T}$, such that R_{p_i} is valid.*

We now prove that, for a given xy-montone curve C going from the top-right corner to the bottom-left corner of R, there exists a planar solution in R_T for the top labels if and only if C satisfies the 1-sided strip condition for all strips $S_0, \ldots, S_k$ in R_T.

Lemma 2 *Let C be an xy-monotone curve from the top-right corner of R to the bottom-left corner of R. Let $P' \subseteq P$ be the sites that are in R_T. There is a planar solution that uses all top labels of R to label sites in P' in such a way that all leaders are in R_T if and only if each horizontal strip S_i, as defined above, satisfies the 1-sided strip condition.*

Proof. To show that the conditions are necessary, let $\mathcal{L}$ be a planar solution for which all top leaders are above C. Consider strip S_i, which is bounded from below by line $h_i, 0 \leq i \leq k$. If there is no site of P' below h_i, rectangle R_p is clearly valid, where p is the intersection of h_i with the left side of R , and thus the 1-sided strip condition is satisfied. Hence, assume that there is a site $p \in P'$ that is labeled by a top label, and is in strip S_j with $j > i$; see Fig. 1a. Then, the vertical segment of this leader crosses h_i in R_T. Let p' denote the rightmost such crossing of a leader of a point in P' with h_i. We claim that $R_{p'}$ is valid. To see this, observe that all sites of P' top-right of p' are contained in $R_{p'}$. Since no leader may cross the vertical segments defining p', the region $R_{p'} \cap R_\mathrm{T}$ contains the same number of sites as $R_{p'}$ contains ports on its top side, i.e., $R_{p'}$ is valid.

Conversely, we show that if the conditions are satisfied, then a corresponding planar solution exists. Let S_k be the last strip that contains sites of P'. For $i = 0, \ldots, k-1$, let p'_i denote the rightmost point of $h_i \cap R_\mathrm{T}$, such that $R_{p'_i}$ is valid. We define p_i to be the point on $h_i \cap R_\mathrm{T}$, whose x-coordinate is $\min_{j \leq i}\{x(p'_j)\}$. Note that R_{p_i} is a valid rectangle, as, by definition, it completely contains some valid rectangle $R_{p'_j}$ with $x(p'_j) = x(p_i)$. Also by definition the sequence formed by the points p_i has decreasing x-coordinates, i.e., the R_{p_i} grow to the left; see Fig. 1b.

We can prove inductively that, for each $i = 0, \ldots, k$, there is a planar labeling $\mathcal{L}_i$ that matches the labels on the top side of R_{p_i} to points contained in R_{p_i}, in such a way that there exists an xy-monotone curve C_i from the upper-left corner of R_{p_i} to its lower right corner that separates the labeled sites from the unlabeled sites without intersecting any leaders. Then $\mathcal{L}_k$ is the claimed labeling. ☐

A similar 1-sided strip condition (with vertical strips) can be obtained for the right region R_R of a partitioned instance. The characterization is completely symmetric.

3 The Algorithm

Now we describe how to find an xy-monotone chain C that satisfies the 1-sided strip conditions. For that purpose we only consider xy-monotone chains that lie on the dual of the grid induced by the sites and ports of the given instance. When traversing this grid from grid point to grid point, we either pass a site (*site event*) or a port (*port event*). By passing a site, we decide if the site is connected to the top or to the right side. In the following, we describe a dynamic program that finds an xy-separating chain in $O(n^3)$ time.

Let there be m_R ports on the right side of R and m_T ports on the top side of R, then the grid has size $[n + m_\mathrm{T} + 2] \times [n + m_\mathrm{R} + 2]$. We define the grid points as $G(x, y)$, with $G(0, 0)$ being the bottom-left and $r := G(n+m_\mathrm{T}+2, n+m_\mathrm{R}+2)$ being the top-right corner of R. Further, we define $G_x(s) := x(G(s, 0))$ and $G_y(t) := y(G(0, t))$.

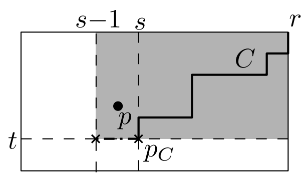


Fig. 2: Step of the dynamic program where p enters the rectangle spanned by r and $G(s-1, t)$.

An entry in the table of our dynamic program is described by three values. The first two values are s and t, which give the position of the current search for the curve C. The interpretation is that the entry encodes the possible xy-monotone curves from r to $p_C := G(s, t)$; see Fig. 2. The remaining value u denotes the number of sites above C in the rectangle spanned by r and p_C. Note that it suffices to store u, as the number of sites below the curve C can directly be derived from u and all sites that are contained in the rectangle spanned by r and p_C. We denote the first values describing the positions of the curves by the vector $\mathbf{c} = (s, t)$. Our goal is to compute a table $T[\mathbf{c}, u]$, such that $T[\mathbf{c}, u] = \texttt{true}$ if and only if there exists an xy-monotone chain C, such that the following conditions hold. (i) Curve C starts at r and ends at p_C. (ii) Inside the rectangle spanned by r and p_C, there are u sites of P above C. (iii) For each strip in the two regions R_T and R_R defined by C the 1-sided strip condition holds.

It follows from these conditions, Proposition 1 and Lemma 2 that the instance admits a planar solution if and only if $T[(0, 0), u] = \texttt{true}$ for some u.

Let us now proceed to describe how to compute the table. Initially, we set $\mathbf{c} = (n + m_\mathrm{T} + 2, n + m_\mathrm{R} + 2)$. We initialize the first entry $T[\mathbf{c}, 0]$ with `true`. The remaining entries are initialized with `false`.

Let $\mathbf{c} := (s, t)$ be the current grid point we checked as endpoint for C. Based on the table $T[\mathbf{c}, \cdot]$ we then compute the entries $T[\mathbf{c} - \mathbf{\Delta c}, \cdot]$ where the vector $\mathbf{\Delta c} = (\Delta s, \Delta t)$ is chosen in such a way that exactly one of both entries $\Delta s, \Delta t \in \{0, 1\}$ has value 1. We classify such steps, depending on whether we cross a site or a port. We give a full description for $\mathbf{\Delta c} = (1, 0)$, i.e, we decrease s by 1. The other case is completely symmetric. Assume $T[\mathbf{c}, u] = \texttt{true}$. We distinguish two cases, based on whether we cross a site or a port.

Case 1: Going from s to $s - 1$ is a site event, i.e., there is a site p with $G_x(s) > x(p) > G_x(s - 1)$. Note that, by our general position assumption and by the definition of the coordinates, the point p is unique. If $y(p) > G_y(t)$, then p enters the rectangle spanned by $G(s-1, t)$ and r, and it is located above C. We thus set $T[\mathbf{c} - \mathbf{\Delta c}, u + 1] = \texttt{true}$. Otherwise we set $T[\mathbf{c} - \mathbf{\Delta c}, u] = \texttt{true}$. Note that the one-sided strip conditions remain satisfied since we do not decrease the number of sites in any region.

Case 2: Going from s to $s - 1$ is a port event, i.e., there is a label ℓ on the top side, whose port is between $G_x(s - 1)$ and $G_x(s)$. Thus, the region above C contains one more label. We therefore check the 1-sided strip condition for the strip above the horizontal line through $G(s - 1, t)$. If it is satisfied, we set $T[\mathbf{c} - \mathbf{\Delta c}, u] = \texttt{true}$.

This immediately gives us a polynomial-time algorithm for Two-Sided Boundary Labeling with Adjacent Sides. The running time crucially relies on the number of 1-sided strip conditions that need to be checked. We show that after a $O(n^2)$ preprocessing phase, such queries can be answered in $O(1)$ time.

To implement the test of the 1-sided strip conditions, we use a table B_T, which stores in the position $B_\mathrm{T}[s, t]$ how large a deficit of top sites to the right can be compensated by sites above and to the left of $G(s, t)$. To compute this matrix, we use a simple dynamic program, which calculates the entries of B_T by going from the left to the right side. Once we have computed this matrix, it is possible to query the 1-sided strip condition in the dynamic program that computes T in $O(1)$ time. The table can be clearly filled out in $O(n^2)$ time. A similar matrix B_R can be computed for the vertical strips. Altogether, this yields an algorithm for Two-Sided Boundary Labeling with Adjacent Sides that runs in $O(n^3)$ time and uses $O(n^3)$ space. However, the entries of each row and column of T depend only on the previous row and column, which allows us to reduce the storage requirement to $O(n^2)$. Using Hirschberg's algorithm [4], we can still backtrack the dynamic program and find a solution corresponding to an entry in the last cell in the same running time.

Theorem 3 Two-Sided Boundary Labeling with Adjacent Sides *can be solved in $O(n^3)$ time using $O(n^2)$ space.*

In order to increase the performance of our algorithm, we can reduce the dimension of the table T by 1. For any search position $\mathbf{c}$, the possible values of u, for which $T[\mathbf{c}, u] = \texttt{true}$ form an interval. Thus, we only need to store the boundaries of the u-interval. Further, we can compute the tables B_T and B_R backwards, i.e., in the direction of the dynamic program, by precomputing the entries of B_T and B_R on the top and right side. Using Hirschberg's algorithm, this reduces the running time to $O(n^2)$ and the space to $O(n)$.

Theorem 4 Two-Sided Boundary Labeling with Adjacent Sides *can be solved in $O(n^2)$ time using $O(n)$ space.*

References

[1] Michael A. Bekos, Michael Kaufmann, Katerina Potika, and Antonios Symvonis. Area-feature boundary labeling. *Comput. J.*, 53(6):827–841, 2010. 1

[2] Michael A. Bekos, Michael Kaufmann, Antonios Symvonis, and Alexander Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Comput. Geom. Theory Appl.*, 36(3):215–236, 2007. 1

[3] Marc Benkert, Herman J. Haverkort, Moritz Kroll, and Martin Nöllenburg. Algorithms for multi-criteria boundary labeling. *J. Graph Algorithms Appl.*, 13(3):289–317, 2009. 1

[4] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Comm. ACM*, 18(6):341–343, 1975. 4

[5] Joel L. Morrison. Computer technology and cartographic change. In D.R.F. Taylor, editor, *The Computer in Contemporary Cartography.* Johns Hopkins University Press, 1980. 1

[6] Marc van Kreveld, Tycho Strijk, and Alexander Wolff. Point labeling with sliding labels. *Comput. Geom. Theory Appl.*, 13:21–47, 1999. 1