

Two-Sided Boundary Labeling with Adjacent Sides[★]

Philipp Kindermann¹, Benjamin Niedermann², Ignaz Rutter², Marcus Schaefer³,
André Schulz⁴, and Alexander Wolff¹

¹ Lehrstuhl für Informatik I, Universität Würzburg, Germany.

<http://www1.informatik.uni-wuerzburg.de/en/staff>

² Fakultät für Informatik, Karlsruher Institut für Technologie (KIT), Germany.

{benjamin.niedermann, rutter}@kit.edu

³ College of Computing and Digital Media, DePaul University, Chicago, IL, USA.

mschaefer@cs.depaul.edu

⁴ Institut für Mathematische Logik und Grundlagenforschung, Universität Münster, Germany.

andre.schulz@uni-muenster.de

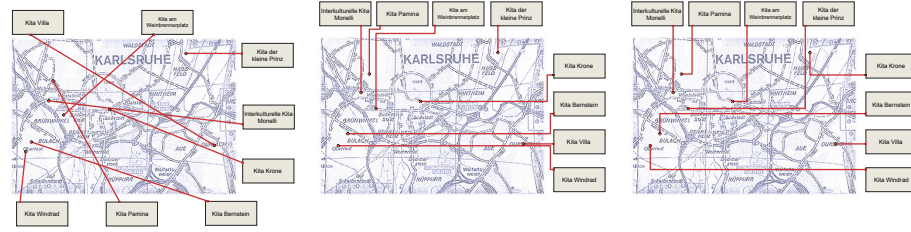
Abstract. In the *Boundary Labeling* problem, we are given a set of n points, referred to as *sites*, inside an axis-parallel rectangle R , and a set of n pairwise disjoint rectangular labels that are attached to R from the outside. The task is to connect the sites to the labels by non-intersecting rectilinear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

In this paper, we study the problem *Two-Sided Boundary Labeling with Adjacent Sides*, where labels lie on two adjacent sides of the enclosing rectangle. 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 cases that labels lie on one side or on two opposite sides of R (where a crossing-free solution always exists). For the more difficult case where labels lie on adjacent sides, we show how to compute crossing-free leader layouts that maximize the number of labeled points or minimize the total leader length.

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. Even very restricted versions of the label-placement problem are NP-hard [14], which explains why labeling a map manually is a tedious task that has been estimated to take 50% of total map production time [15]. The ACM Computational Geometry Impact Task Force report [6] identified label placement as an important research area. The point-labeling problem in particular has received considerable attention, from practitioners and theoreticians alike. The latter have proposed approximation algorithms for various objectives (label number versus label size), label shapes (such as axis-parallel rectangles or disks), and label-placement models (so-called fixed-position models versus slider models).

[★] This research was initiated during the GraDr Midterm meeting at the TU Berlin, which was supported by an ESF networking grant. Ph. Kindermann acknowledges support by the ESF EuroGIGA project GraDR (DFG grant Wo 758/5-1).






(a) original labeling of kindergartens in Karlsruhe, Germany	(b) <i>opo</i> -labeling computed by the algorithm of Bekos et al. [4]	(c) <i>po</i> -labeling using the same ports as (b)
		

Fig. 1: A real-world example of boundary labeling with adjacent sides (taken from [4]). For better readability, we have simplified the label texts.

The traditional label-placement models for point labeling insist that a label is placed such that a point on its boundary coincides with the point to be labeled, the *site*. This can make it impossible to label all sites with labels of sufficient size if some sites are very close together. For this reason, Freeman et al. [8] and Zoraster [19] advocated the use of *leaders*, (usually short) line segments that connect sites to labels. In order to make sure that the background image or map remains visible even in the presence of large labels, Bekos et al. [4] took a more radical approach. They introduced models and algorithms for *boundary labeling*, where all labels are placed beyond the boundary of the map and are connected to the sites by straight-line or rectilinear leaders (see Fig. 1).

Problem statement. Following Bekos et al. [4], we define the BOUNDARY LABELING problem as follows. We are given an axis-parallel rectangle $R = [0, W] \times [0, H]$, which is called the *enclosing rectangle*, a set $P \subset R$ of n points p_1, \dots, p_n , called *sites*, within the rectangle R , and a set L of $m \leq n$ axis-parallel rectangles ℓ_1, \dots, ℓ_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, \dots, 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. We call an instance *solvable* if a planar solution exists. Note that we do not prescribe which site connects to which label. The endpoint of a curve at a label is called a *port*. We distinguish two versions 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; see Fig. 1c. (Fig. 1b shows a labeling with so-called *opo-leaders*, which were investigated by Bekos et al. [4]). Bekos et al. [3, Fig. 12] observed that not every instance (with $m = n$) admits a planar solution with *po-leaders* where all sites are labeled.

Previous and related work. For *po*-labeling, Bekos et al. [4] 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. [5] 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 two-sided case of leader-length minimization, Bekos et al. [4] gave a simple dynamic program running in $O(n^2)$ time. All these algorithms work both for fixed and sliding ports.

Leaders that contain a diagonal part have been studied by Benkert et al. [5] and by Bekos et al. [2]. Recently, Nöllenburg et al. [16] have investigated a dynamic scenario for the one-sided case, Gemsa et al. [9] have used multi-layer boundary labeling to label panorama images, and Fink et al. [7] have boundary labeled focus regions, for example, in interactive on-line maps.

At its core, the boundary label problem asks for a non-intersecting perfect (or maximum) matching on a bipartite graph. Note that an instance may have a planar solution, although all of its leader-length minimal matchings have crossings. In fact, the ratio between a length-minimal solution and a length-minimal crossing-free matching can be arbitrarily bad; see Fig. 2. When connecting points and sites with straight-line segments, the minimum Euclidean matching is necessarily crossing-free. For this case an $O(n^{2+\varepsilon})$ -time $O(n^{1+\varepsilon})$ -space algorithm exists [1]. The minimum-length solution using rectilinear paths with an unbounded number of bends in the presence of obstacles is NP-hard, but there is a 2-approximation [18].

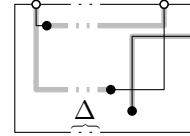


Fig. 2: Length-minimal solutions may have crossings.

Boundary labeling can also be seen as a graph-drawing problem where the class of graphs to be drawn is restricted to matchings. The restriction concerning the positions of the graph vertices (that is, sites and ports) has been studied for less restricted graph classes under the name *point-set embeddability (PSE)*, usually following the straight-line drawing convention for edges [10]. More recently, PSE has also been combined with the ortho-geodesic drawing convention [12], which generalizes *po*-labeling by allowing edges to make more than one bend. The case where the mapping between ports and sites is given has been studied in VLSI layout [17].

Our contribution. We investigate the problem TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES where all labels lie on two *adjacent* sides of R , 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 main 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). Our algorithm uses 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 several measures that will be used to compare different solutions. We denote the total length of all leaders in \mathcal{L} by $\text{length}(\mathcal{L})$. Moreover, we denote by $|\mathcal{L}|_x$ the total length of all horizontal segments of leaders that connect a right label to a site. Similarly, we denote by $|\mathcal{L}|_y$ the total length of the vertical segments of leaders that connect top labels to sites. Note that in general, it is *not* true that $|\mathcal{L}|_x + |\mathcal{L}|_y = \text{length}(\mathcal{L})$.

We denote the (uniquely defined) leader connecting a site p to a port t of a label ℓ by $\lambda(p, t)$. We denote the bend of the leader $\lambda(p, t)$ by $\text{bend}(p, t)$. In the case of fixed ports, we identify ports with labels and simply write $\lambda(p, \ell)$ and $\text{bend}(p, \ell)$, resp.

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 (see the full version of this paper [13]). 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 ; see Fig. 4. 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 Section 3.

Lemma 1. *Consider a solution \mathcal{L} for (R, L, P) and let $P' \subseteq P$ be sites of the same type. Let $L' \subseteq L$ be the set of labels of the sites in P' . Let $K \subseteq R$ be a rectangle that contains all bends of the leaders of P' . If the leaders of $P \setminus P'$ do not intersect K , then we can rewire P' and L' such that the resulting solution \mathcal{L}' has the following properties: (i) all intersections in K are removed, (ii) there are no new intersections of leaders outside of K , (iii) $|\mathcal{L}'|_x = |\mathcal{L}|_x$, $|\mathcal{L}'|_y = |\mathcal{L}|_y$, and (iv) $\text{length}(\mathcal{L}') \leq \text{length}(\mathcal{L})$.*

Proof. Without loss of generality, we assume that P' contains top sites; the other cases are symmetric. We first prove that, no matter how we change the assignment between P' and L' , new intersection points can arise only in K . Then we show how to establish the claimed solution.

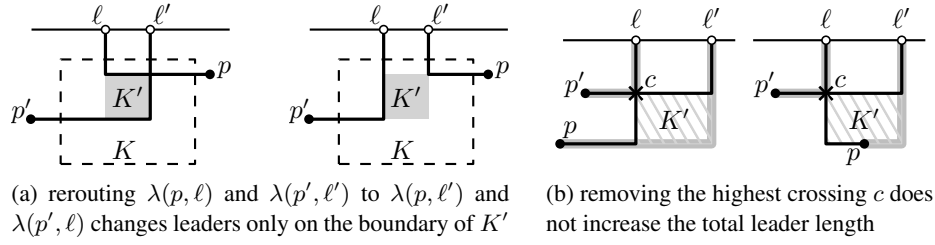


Fig. 3: Illustration of the proof of Lemma 1.

Claim. Let $\ell, \ell' \in L'$ and $p, p' \in P'$ such that ℓ labels p and ℓ' labels p' . Changing the matching by rerouting p to ℓ' and p' to ℓ does not introduce new intersections outside of K .

Let $K' \subseteq K$ be the rectangle spanned by $\text{bend}(p, \ell)$ and $\text{bend}(p', \ell')$. When rerouting, we replace $\lambda(p, \ell) \cup \lambda(p', \ell')$ restricted to the boundary of K' by its complement with respect to the boundary of K' ; see Fig. 3a for an example. Thus, any changes concerning the leaders occur only in K' . The statement of the claim follows.

Since any rewiring can be seen as a sequence of *pairwise* reroutings, the above claim shows that we can rewire L' and P' arbitrarily without running the risk of creating new conflicts outside of K . In order to resolve the conflicts inside K , we use the length-minimization algorithm for one-sided boundary labeling by Benkert et al. [5], with the sites and ports outside K projected onto the boundary of K . Thus, after finitely many such steps, we find a solution \mathcal{L}' that satisfies properties (i)–(iv) in the statement of the lemma. \square

Definition 1. We call an xy -monotone, rectilinear curve connecting the top-right to the bottom-left corner of R an xy -separating curve; see Fig. 4. 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 their leaders lie on or above C , and
- b) the right sites and 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; see Fig. 5. We claim that these patterns are the only ones that violate xy -separability (for the proof, refer to the full version of the paper [13]).

Lemma 2. A planar solution is xy -separated if and only if it does not contain any of the patterns P1–P4 in Fig. 5.

Observe that patterns P1 and P2 can be transformed into patterns P4 and P3, respectively, by mirroring the instance diagonally. Next, we prove constructively that, by rerouting pairs of leaders, any planar solution can be transformed into an xy -separated planar solution.

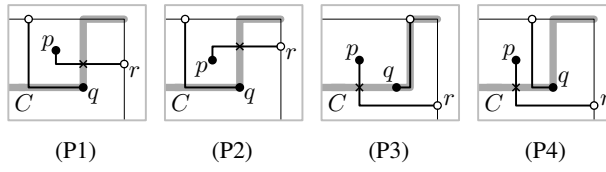


Fig. 5: A planar solution that contains any of the above four patterns P1–P4 is not xy -separated.

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 $\text{length}(\mathcal{L}') \leq \text{length}(\mathcal{L})$, $|\mathcal{L}'|_x \leq |\mathcal{L}|_x$, and $|\mathcal{L}'|_y \leq |\mathcal{L}|_y$.*

Proof. Let \mathcal{L} be a planar solution of minimum total leader length. We show that \mathcal{L} is xy -separated. Assume, for the sake of contradiction, that \mathcal{L} is not xy -separated. Then, by Lemma 2, \mathcal{L} contains one of the patterns P1–P4. Without loss of generality, we can assume that the pattern is of type P3 or P4. Otherwise, we mirror the instance diagonally.

Let p be a right site (with port r) and let q be a top site (with port t) such that (p, q) forms a pattern of type P3 or P4. Among all such patterns, pick one where p is rightmost. Among all these patterns, pick one where q is bottommost. Let A be the rectangle spanned by p and t ; see Fig. 6. Let A' be the rectangle spanned by $\text{bend}(q, t)$ and p . Let B be the rectangle spanned by q and r . Let B' be the rectangle spanned by q and $\text{bend}(p, r)$. Then we claim the following:

- (i) Sites in the interiors of A and A' are connected to the top.
- (ii) Sites in the interiors of B and B' are connected to the right.

Property (i) is due to the choice of p as the rightmost site involved in such a pattern. Similarly, property (ii) is due to the choice of q as the bottommost site that forms a pattern with p . This settles our claim.

Our goal is to change the labeling by rerouting p to t and q to r , which decreases the total leader length, but may introduce crossings. We then use Lemma 1 to remove the crossings without increasing the total leader length. Let \mathcal{L}'' be the labeling obtained from \mathcal{L} by rerouting p to t and q to r . We have $|\mathcal{L}''|_y \leq |\mathcal{L}|_y - (y(p) - y(q))$ and $|\mathcal{L}''|_x = |\mathcal{L}|_x - (x(q) - x(p))$. Moreover, $\text{length}(\mathcal{L}'') \leq \text{length}(\mathcal{L}) - 2(y(p) - y(q))$, as at least twice the vertical distance between p and q is saved; see Fig. 6. Since the original labeling was planar, crossings may only arise on the horizontal segment of $\lambda(p, t)$ and on the vertical segment of $\lambda(q, r)$.

By properties (i) and (ii), all leaders that cross the new leader $\lambda(p, t)$ have their bends inside A' , and all leaders that cross the new leader $\lambda(q, r)$ have their bends inside B' . Thus, we can apply Lemma 1 to the rectangles A' and B' to resolve all new crossings. The resulting solution \mathcal{L}' is planar and has length less than $\text{length}(\mathcal{L})$. This is a contradiction to the choice of \mathcal{L} . \square

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

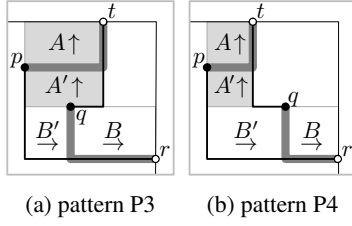


Fig. 6: Types (top = \uparrow / right = \rightarrow) of the sites inside rectangles A , A' , B , and B' . Fat edges: result after rerouting.

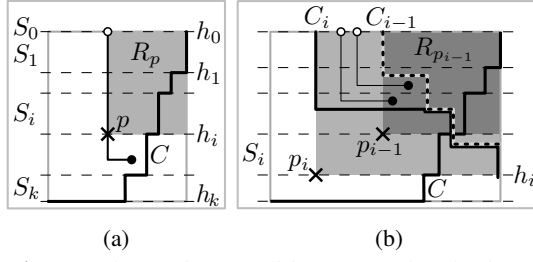


Fig. 7: The strip condition. a) The horizontal segments of C partition R_T into the strips S_0, S_1, \dots, S_k . b) Constructing a planar labeling from a sequence of valid rectangles.

length has minimum leader length among *all* planar solutions. In Lemma 3 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 . These regions are relatively open at C . For each horizontal segment of C consider the horizontal line through the segment. We denote the part of these lines within R by h_1, \dots, h_k , respectively. Further, let h_0 be the top edge of R . The line segments h_1, \dots, h_k partition R_T into k strips, which we denote by S_1, \dots, S_k from top to bottom, such that strip S_i is bounded by h_i from below for $i = 1, \dots, k$; see Fig. 6a. 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 , we define the rectangle R_p , spanned by p and the top-right corner of R . We define R_p such that it is closed but does *not* contain its top-left corner. In particular, we consider the port of a top label as contained in R_p , except if it is the upper left corner of R_p .

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 rectangle by leaders that are completely contained inside the rectangle.

We now prove that, for a given xy -separating curve C , there exists a planar solution in R_T for the top labels if and only if C satisfies the following strip condition for each strip S_0, \dots, S_k in R_T . The *strip condition* of strip S_i is satisfied if there exists a point $p \in h_i \cap R_T$ such that R_p is valid. We call a region $S \subseteq R$ *balanced* if it contains the same numbers of sites and ports.

Lemma 3. *Let C be an xy -separating curve and let $P_T = P \cap R_T$. There is a planar solution that uses all top labels of R to label the sites in P_T such that all leaders are in R_T if and only if S_0, \dots, S_k satisfy the 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_T 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 strip condition is satisfied.

Hence, assume that there is a site $p \in P_T$ that is labeled by a top label, and is in strip S_j with $j > i$; see Fig. 6a. 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 site in P_T with h_i . We claim that $R_{p'}$ is valid. To see this, observe that all sites of P_T top-right of p' are contained in $R_{p'}$. Since no leader may cross the vertical segments defining p' , the number of sites in $R_{p'} \cap R_T$ is balanced, 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_T . For $i = 0, \dots, k$, let p'_i denote the rightmost point of $h_i \cap R_T$ such that $R_{p'_i}$ is valid. We define p_i to be the point on $h_i \cap R_T$ with x -coordinate $\min_{j \leq i} \{x(p'_j)\}$. Note that R_{p_i} is a valid rectangle, as, by definition, R_{p_i} contains some valid rectangle $R_{p'_j}$ with $x(p'_j) = x(p_i)$. Also by definition, the sequence p_0, p_1, \dots, p_k has decreasing x -coordinates, i.e., $R_{p_k} \subseteq \dots \subseteq R_{p_1} \subseteq R_{p_0}$; see Fig. 6b.

We prove inductively that, for $i = 0, \dots, k$, 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} such that there exists an xy -monotone curve C_i from the top-left to the bottom-right corner of R_{p_i} that separates the labeled sites from the unlabeled sites without intersecting any leaders. Then \mathcal{L}_k is the claimed labeling.

For $i = 0$, $\mathcal{L}_0 = \emptyset$ is a planar solution. Consider a strip S_i with $0 < i \leq k$; see Fig. 6b. By the induction hypothesis, we have a curve C_{i-1} and a planar labeling \mathcal{L}_{i-1} , which matches the labels on the top side of $R_{p_{i-1}}$ to the sites in $R_{p_{i-1}}$ above C_{i-1} . In order to extend \mathcal{L}_{i-1} to a planar solution \mathcal{L}_i , we additionally need to match the remaining labels on the top side of R_{p_i} and construct a corresponding curve C_i . Let P_i denote the set of unlabeled sites in R_{p_i} . By the validity of R_{p_i} , this number is at least as large as the number of unused ports at the top side of R_{p_i} . We match these ports from top to bottom to the topmost sites of P_i ; the result is the claimed planar labeling \mathcal{L}_i . The ordering ensures that no two of the new leaders cross. Moreover, no leader crosses the curve C_{i-1} , and hence such leaders cannot cross leaders in \mathcal{L}_{i-1} . It remains to construct the curve C_i . For this, we start at the top-left corner of R_{p_i} and move down vertically, until we have passed all labeled sites. We then move right until we either hit C_{i-1} or the right side of R . In the former case, we follow C_{i-1} until we arrive at the right side of R . Finally, we move down until we arrive at the bottom-right corner of R_{p_i} . Note that all labeled sites are above C_i , unlabeled sites are below C_i , and no leader is crossed by C_i . This is true since we first move below the new leaders and then follow the previous curve C_{i-1} . \square

A symmetric 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 -separating curve C that satisfies the strip conditions. For that purpose we only consider xy -separating curves 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. Clearly, there is an exponential number of possible xy -monotone traversals through the grid. In the following, we describe a dynamic program that finds an xy -separating curve 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 $[n + m_T + 1] \times [n + m_R + 1]$. We define the grid points as $G(x, y)$, $0 \leq x \leq n + m_T + 1$, $0 \leq y \leq n + m_R + 1$ with $G(0, 0)$ being the bottom-left and $r := G(n + m_T + 1, n + m_R + 1)$ being the top-right corner of R . Further, we define $G_x(s) := x(G(s, 0))$ and $G_y(t) := y(G(0, t))$.

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. 8. 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 $T[\mathbf{c}, u] = \text{true}$ if and only if there exists an xy -separating curve 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 strip condition holds.

It follows from these conditions, Proposition 1 and Lemma 3 that the instance admits a planar solution if and only if $T[(0, 0), u] = \text{true}$, for some u . Let us now proceed to describe how to compute the table. Initially, we set $\mathbf{c} = (n + m_T + 1, n + m_R + 1)$. We initialize the first entry $T[\mathbf{c}, 0] = \text{true}$. 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 $T[\mathbf{c} - \Delta\mathbf{c}, \cdot]$ where the vector $\Delta\mathbf{c} = (\Delta s, \Delta t)$ is either $(0, 1)$ or $(1, 0)$. We classify such steps, depending on whether we cross a site or a port. We give a full description for $\Delta\mathbf{c} = (1, 0)$, i.e., we decrease s by 1. The other case is completely symmetric. Assume $T[\mathbf{c}, u] = \text{true}$. 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 assumption of general position and the definition of the coordinates, the site 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 ; see Fig. 8. We thus set $T[\mathbf{c} - \Delta\mathbf{c}, u + 1] = \text{true}$. Otherwise we set $T[\mathbf{c} - \Delta\mathbf{c}, u] = \text{true}$. Note that the 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

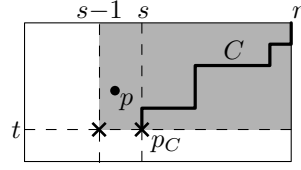


Fig. 8: Possible step of the dynamic program, where p enters the rectangle spanned by r and $G(s - 1, t)$.

more label. We therefore check the strip condition for the strip above the horizontal line through $G(s-1, t)$. If it is satisfied, we set $T[\mathbf{c} - \Delta\mathbf{c}, u] = \text{true}$.

If $T[\mathbf{c}, u] = \text{false}$, there is no xy -separating curve that satisfies the conditions given above, so it suffices to only look at the `true` table entries. 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 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 strip conditions, we use a table B_T , which stores in the position $B_T[s, t]$ how large a deficit of top sites to the right can be compensated by sites above and to the left of $G(s, t)$. That is, $B_T[s, t]$ is the maximum value k such that there exists a rectangle $K_{B_T[s, t]}$ with lower right corner $G(s, t)$ whose top side is bounded by the top side of R , and that contains k more sites in its interior, than it has ports on its top side. 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 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 [11], we can still backtrack the dynamic program and find a solution corresponding to an entry in the last cell in the same running time. The detailed approach on how to calculate and use the tables B_T and B_R is given in the full version of the paper [13].

Theorem 1. 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 number of dimensions of the table T by 1. As a first step, we show that for any search position \mathbf{c} , the possible values of u , for which $T[\mathbf{c}, u] = \text{true}$ form an interval.

Lemma 4. *Let $T[\mathbf{c}, u] = T[\mathbf{c}, u'] = \text{true}$ with $u \leq u'$. Then $T[\mathbf{c}, u''] = \text{true}$ for $u \leq u'' \leq u'$.*

Proof. Let C be the curve corresponding to the entry $T[\mathbf{c}, u]$. That is C connects r to p_C such that u sites in the rectangle spanned by p_C and r are above C , and the strip conditions (both above and below C) are satisfied. Similarly, let C' be the curve corresponding to $T[\mathbf{c}, u']$.

Since u and u' differ, there is a rightmost site p , such that p is below C and above C' . Let v and v' be the grid points of C and C' that are immediately to the left of p . Note that v is above v' since C is above p and C' is below it. Consider the C'' , which starts at r and follows C until v , then it moves down vertically to v' , and from there follows C' to p . Obviously C'' is an xy -separating curve, and it has above it the same sites as C' , except for p , which is below it. Thus there are $u'' = u' - 1$ sites above C'' in the

rectangle spanned by p and r . If all strips defined by C'' satisfy the strip condition, then this implies $T[c, u''] = \text{true}$.

To see that the strip conditions are indeed satisfied, consider a horizontal strip S'' defined by C'' . Let S be the lowest horizontal strip defined by C that is not below the lower boundary of S'' . We know that S fulfills the strip condition, which is witnessed by some valid rectangle K . We can enlarge this rectangle vertically such that it touches the lower boundary of S'' . The enlarged rectangle contains at least as many sites above C'' as there were above C in K . Hence it is a valid rectangle and the strip condition for S'' holds. An analogous statement holds for the vertical strips since C'' is above C' at every x -coordinate. \square

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, respectively. Using Hirschberg's algorithm, this reduces the running time to $O(n^2)$ and the space to $O(n)$. The detailed description is given in the full version of the paper [13].

Theorem 2. TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES *can be solved in $O(n^2)$ time using $O(n)$ space.*

4 Conclusion

In this paper, we have studied the problem of testing whether an instance of TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES admits a planar solution. We have given the first efficient algorithm for this problem, running in $O(n^2)$ time.

The presented algorithm can also be used to solve a variety of different extensions of the problem. In the full version of the paper [13], we show how to generalize from fixed to sliding ports without increasing the asymptotic running time. Further, we show how to maximize the number of labeled sites such that the solution is planar in $O(n^3 \log n)$ time and we give an extension to the algorithm that minimizes the total leader length in $O(n^8 \log n)$ time.

With some additional work, the presented approach can also be used to solve THREE-SIDED and FOUR-SIDED BOUNDARY LABELING in polynomial time. Namely, it can be shown that if a solution to the four-sided problem exists, there exists one that has a central point z such that xy -monotone curves from z to the four corners of the rectangle R partition the solution without intersecting any leaders. To compute such a partitioned solution, assume we are given, for each side s of the rectangle R , the leader whose segment orthogonal to s is maximum among all leaders of side s . These *extremal leaders* essentially partition the instance into four smaller instances of ADJACENT TWO-SIDED BOUNDARY LABELING, one for each corner. These instances can be processed independently. There are $O(n^8)$ choices for these extremal leaders, trying all of them thus yields a running time of $O(n^{10})$ and space consumption $O(n)$. For THREE-SIDED BOUNDARY LABELING, the running time is $O(n^8)$, but can be improved to $O(n^4)$ by guessing only the extremal leader of the middle side of the rectangle. Also, except for the length minimization, all presented extensions carry over. A proof is given in the full version of the paper [13]. It remains open whether a minimum

length solution of THREE- and FOUR-SIDED BOUNDARY LABELING can be computed in polynomial time.

References

1. Agarwal, P.K., Efrat, A., Sharir, M.: Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.* 29(3), 912–953 (1999)
2. Bekos, M.A., Kaufmann, M., Nöllenburg, M., Symvonis, A.: Boundary labeling with octilinear leaders. *Algorithmica* 57(3), 436–461 (2010)
3. Bekos, M.A., Kaufmann, M., Potika, K., Symvonis, A.: Area-feature boundary labeling. *Comput. J.* 53(6), 827–841 (2010)
4. Bekos, M.A., Kaufmann, M., Symvonis, A., Wolff, A.: Boundary labeling: Models and efficient algorithms for rectangular maps. *Comput. Geom. Theory Appl.* 36(3), 215–236 (2007)
5. Benkert, M., Haverkort, H.J., Kroll, M., Nöllenburg, M.: Algorithms for multi-criteria boundary labeling. *J. Graph Algorithms Appl.* 13(3), 289–317 (2009)
6. Chazelle, B., 36 co-authors: The computational geometry impact task force report. In: Chazelle, B., Goodman, J.E., Pollack, R. (eds.) *Advances in Discrete and Computational Geometry*, vol. 223, pp. 407–463. American Mathematical Society, Providence, RI (1999)
7. Fink, M., Haunert, J.H., Schulz, A., Spoerhase, J., Wolff, A.: Algorithms for labeling focus regions. *IEEE Trans. Visual. Comput. Graphics* 18(12), 2583–2592 (2012)
8. Freeman, H., Marrinan, S., Chitalia, H.: Automated labeling of soil survey maps. In: *ASPRS-ACSM Annual Convention*, Baltimore. vol. 1, pp. 51–59 (1996)
9. Gamsa, A., Haunert, J.H., Nöllenburg, M.: Boundary-labeling algorithms for panorama images. In: *19th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inform. Syst.* pp. 289–298 (2011)
10. Gritzmann, P., Mohar, B., Pach, J., Pollack, R.: Embedding a planar triangulation with vertices at specified positions. *Amer. Math. Mon.* 98, 165–166 (1991)
11. Hirschberg, D.S.: A linear space algorithm for computing maximal common subsequences. *Comm. ACM* 18(6), 341–343 (1975)
12. Katz, B., Krug, M., Rutter, I., Wolff, A.: Manhattan-geodesic embedding of planar graphs. In: Eppstein, D., Gansner, E.R. (eds.) *GD 2009, LNCS*, vol. 5849, pp. 207–218. Springer Heidelberg (2010)
13. Kindermann, P., Niedermann, B., Rutter, I., Schaefer, M., Schulz, A., Wolff, A.: Two-sided boundary labeling with adjacent sides. Arxiv report (May 2013), available at <http://arxiv.org/abs/1305.0750>
14. van Kreveld, M., Strijk, T., Wolff, A.: Point labeling with sliding labels. *Comput. Geom. Theory Appl.* 13, 21–47 (1999)
15. Morrison, J.L.: Computer technology and cartographic change. In: Taylor, D. (ed.) *The Computer in Contemporary Cartography*. Johns Hopkins University Press (1980)
16. Nöllenburg, M., Polishchuk, V., Sysikaski, M.: Dynamic one-sided boundary labeling. In: *18th ACM SIGSPATIAL Int. Symp. Adv. Geogr. Inform. Syst.* pp. 310–319 (2010)
17. Raghavan, R., Cohoon, J., Sahni, S.: Single bend wiring. *J. Algorithms* 7(2), 232–257 (1986)
18. Speckmann, B., Verbeek, K.: Homotopic rectilinear routing with few links and thick edges. In: López-Ortiz, A. (ed.) *LATIN 2010, LNCS*, vol. 6034, pp. 468–479. Springer Heidelberg (2010)
19. Zoraster, S.: Practical results using simulated annealing for point feature label placement. *Cartography and GIS* 24(4), 228–238 (1997)