

Multi-Sided Boundary Labeling*

Philipp Kindermann[†] Benjamin Niedermann[‡] Ignaz Rutter[‡] Marcus Schaefer[§]
André Schulz[¶] Alexander Wolff[†]

Submitted to Algorithmica in May 2014, revised in April 2015

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 *Multi-Sided Boundary Labeling* problem, with labels lying on at least two 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 in which labels lie on one side or on two opposite sides of R (here a crossing-free solution always exists). The case where labels may lie on adjacent sides is more difficult. We present efficient algorithms for testing the existence of a crossing-free leader layout that labels all sites and also for maximizing the number of labeled sites in a crossing-free leader layout. For two-sided boundary labeling with adjacent sides, we further show how to minimize the total leader length in a crossing-free layout.

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 [22], which explains why labeling a map manually is a tedious task that has been estimated to take 50% of total map production time [19]. The ACM Computational Geometry Impact Task Force report [9] 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).

The traditional label-placement models for point labeling require 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

*A preliminary version of this paper has appeared in *Proc. 13th Int. Algorithms Data Struct. Symp. (WADS'13)*, volume 8037 of *Lect. Notes Comput. Sci.*, pages 463–474, Springer-Verlag. This research was initiated during the GraDr Midterm meeting at the TU Berlin in October 2012. The meeting was supported by the ESF EuroGIGA networking grant. Ph. Kindermann acknowledges support by the ESF EuroGIGA project GradR (DFG grant Wo 758/5-1).

[†]Lehrstuhl für Informatik I, Universität Würzburg, Germany. Homepage: <http://www1.informatik.uni-wuerzburg.de/en/staff>

[‡]Fakultät für Informatik, Karlsruher Institut für Technologie (KIT), Germany. Email: {rutter, 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

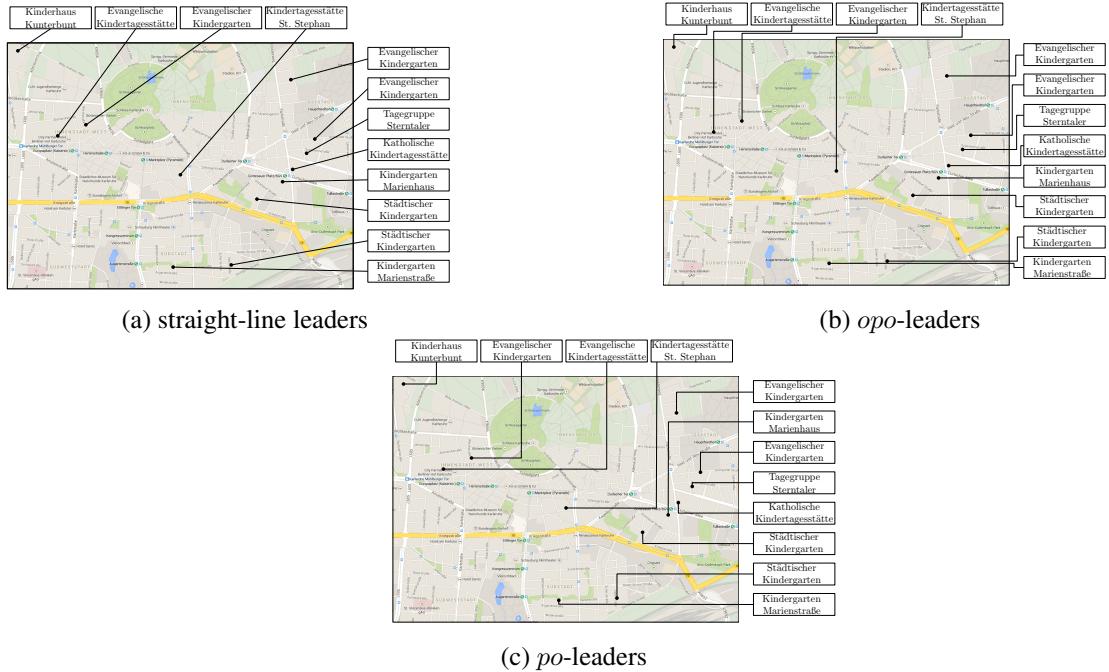


Fig. 1: Labeling of kindergartens in Karlsruhe, Germany. The pictures show different types of leaders with labels on adjacent sides of the map. For better readability, we have simplified the label texts.

impossible to label all sites with labels of sufficient size if some sites are very close together. For this reason, Freeman et al. [11] and Zoraster [23] advocated the use of *leaders*, (usually short) line segments that connect sites to labels. In order to ensure that the background image or map remains visible even in the presence of large labels, Bekos et al. [6] 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. [6] 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 of equal size, called *labels*. The labels 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* of a problem instance is a set of m curves c_1, \dots, c_m in the interior of R , called *leaders*, that connect sites to labels such that the leaders a) induce a matching between the labels and (a subset of) the sites, b) touch the associated labels on the boundary of R . Following previous work, we do not define labels as the text associated with the sites, but as the empty rectangles into which that text will be placed (during a post-processing step). This approach is justified by our assumption that all label rectangles have the same size.

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.

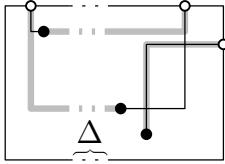


Fig. 2: Length-minimal solutions may have crossings. By increasing Δ we can make the ratio between the length-minimal matching and the length-minimal crossing-free matching arbitrarily small.

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 touching 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. [6]). Bekos et al. [5, Fig. 16] observed that not every instance admits a planar solution with *po-leaders* in which all sites are labeled (even if $m = n$).

Previous and related work. For *po*-labeling, Bekos et al. [6] 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. [7] 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. [6] 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. [7] and by Bekos et al. [4]. Recently, Nöllenburg et al. [20] have investigated a dynamic scenario for the one-sided case, Gemsa et al. [12] have used multi-layer boundary labeling to label panorama images, and Fink et al. [10] have boundary labeled focus regions, for example, in interactive on-line maps. Lin et al. [18] consider boundary labeling where more than one site may be labeled by the same label. Lin [17] and Bekos et al. [3] study hyperleaders that connect each label to a set of sites.

At its core, the boundary labeling 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].

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 [13]. For polygonal edges, Bastert and Fekete [2] proved that PSE with minimum number of bends or minimum total edge length is NP-hard, even when the graph is a matching. For minimizing the total edge length and the same graph class, Liebling et al. [16] introduced heuristics and Chan et al. [8] presented approximation algorithms. Chan et al. also considered paths and general planar graphs. PSE has also been combined with the ortho-geodesic drawing convention [15], which generalizes *po*-labeling by allowing edges to have more than one bend. The case where the mapping between ports and sites is given has been studied in VLSI layout [21].

Our contribution. In the first part of the paper, we investigate the problem TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES where all labels lie on two *adjacent* sides of R , without loss of generality, 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.

We present an algorithm that, given an instance with n labels with fixed ports 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.

We study several extensions of our main result (see Section 4). First, we show that our approach for fixed ports can also be made to work for sliding ports. Second, we optimally solve the label-number maximization problem (in $O(n^3 \log n)$ time using $O(n)$ space). This is interesting if the position of the sites and labels does not allow for a perfect matching or if there are more sites than labels. Finally, we present a modification of our algorithm that minimizes the leader length (in $O(n^8 \log n)$ time and $O(n^6)$ space).

In the second part of the paper, we investigate the problems THREE-SIDED BOUNDARY LABELING and FOUR-SIDED BOUNDARY LABELING where the labels may lie on three or even all four sides of R , respectively. To that end we generalize the concept of partitioning the sites labeled by leaders of different sides. In this way we obtain subinstances that we can solve using the algorithm for the two-sided case. We obtain an algorithm solving the four-sided case in $O(n^9)$ time and $O(n)$ space and an algorithm solving the three-sided case in $O(n^4)$ time and $O(n)$ space. Except for the leader-length minimization, all extensions presented previously extend to the three- and four-sided case, of course with a corresponding impact on the running time and space requirements.

Notation. We call the labels that lie on the right (left/top/bottom) side of R *right (left/top/bottom) 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 left or right label to a site. Similarly, we denote by $|\mathcal{L}|_y$ the total length of the vertical segments of leaders that connect top or bottom labels to sites. Note that generally, $|\mathcal{L}|_x + |\mathcal{L}|_y \neq \text{length}(\mathcal{L})$.

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 $\text{bend}(p, \ell)$, respectively.

2 Structure of Two-Sided Planar Solutions

In this section, we tackle the two-sided boundary labeling problem with adjacent sides by presenting a series of structural results of increasing strength. We assume that the labels are located on the top and right sides of R . For simplicity, we assume that we have fixed ports. By identifying the ports with their labels, we use L to denote the set of ports of all labels. For sliding ports, we can simply fix all ports to the bottom-left corner of their corresponding labels. First we show that

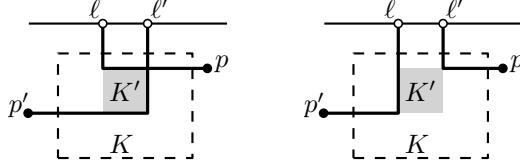


Fig. 3: Illustration of the proof of Lemma 1. Rerouting $\lambda(p, \ell)$ and $\lambda(p', \ell')$ to $\lambda(p, \ell')$ and $\lambda(p', \ell)$ changes leaders only on the boundary of K'

a planar two-sided solution admits a transformation sustaining planarity such that the result of the transformation can be split 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 there exists a planar solution for a given separation. 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 rematch 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 . This enables us to construct the required solution.

Claim 1. 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. 3 for an example. Thus, any changes concerning the leaders occur only in K' . The statement of the claim follows.

Since any rerouting can be seen as a sequence of pairwise reroutings, the above claim shows that we can rematch L' and P' arbitrarily without running the risk of creating new conflicts outside of K . To resolve the conflicts inside K , we use the length-minimization algorithm for one-sided boundary labeling by Benkert et al. [7], with the sites and ports outside K projected onto the boundary of K . Thus, we obtain a solution \mathcal{L}' satisfying properties (i)–(iv). \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. 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 sites that are connected to the top side and all their leaders lie on or above C
- b) the sites that are connected to the right side 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; see Fig. 5. In Lemma 2, we show that these patterns are the only ones that can violate xy -separability.

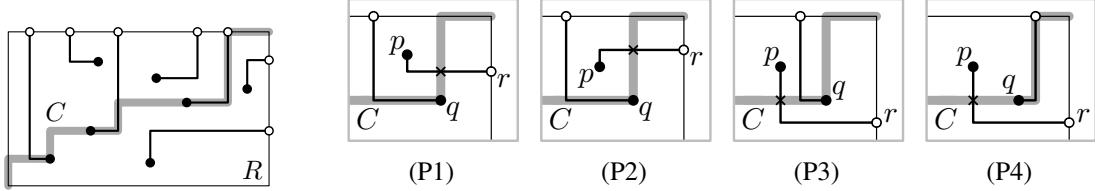


Fig. 4: An xy -separating curve of a planar solution.

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

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.*

Proof. Obviously, the planar solution is not xy -separated if one of these patterns occurs. Let us assume that none of these patterns exists. We construct an xy -monotone curve C from the top-right corner of R to its bottom-left corner. We move to the left whenever possible, and down only when we reach the x -coordinate of a site p that is connected to the top, or when we reach the x -coordinate of a port of a top label, labeling a site p . If we have to move down, we move down as far as necessary to avoid the corresponding leader, namely down to the y -coordinate of p . Finally, when we reach the left boundary of R , we move down to the bottom-left corner of R . If C is free of crossings, then we have found an xy -separating curve. (For an example, see curve C in Fig. 4.)

Assume for a contradiction, that a crossing arises during the construction, and consider the topmost such crossing. Note that, by the construction of C , crossings can only occur with leaders that connect a site p to a right port r . We distinguish two cases, based on whether the crossing occurs on a horizontal or a vertical segment of the curve C .

If C is crossed on a vertical segment, then this segment belongs to a leader connecting a site q to a top port t , and we have reached the x -coordinate of either the port or the site. Had we, however, reached the x -coordinate of the port, this would imply a crossing between $\lambda(p, r)$ and $\lambda(q, t)$. Thus, we have reached the x -coordinate of q . This means that p lies to the left of and above q , and we have found one of the patterns P1 and P2; see Fig. 5.

If C is crossed on a horizontal segment, then p must lie above r . Otherwise, there would be another crossing of C with the same leader, which is above the current one. This would contradict the choice of the topmost crossing. Consider the previous segment of C , which is responsible for reaching the y -coordinate of the current segment. This vertical segment belongs to a leader connecting a site q to a top port t . Since leaders do not cross, q is to the right of p , and the crossing on C implies that q is below p . We have found one of the patterns P3 and P4; see Fig. 5. \square

Observe that patterns P1 and P2 can be transformed into patterns P3 and P4, 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.

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.

Consider all patterns (p, q) in \mathcal{L} of type P3 or P4 such that p is a right site (with port r) and q is a top site (with port t). Among all such patterns, consider the ones where p is rightmost and



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

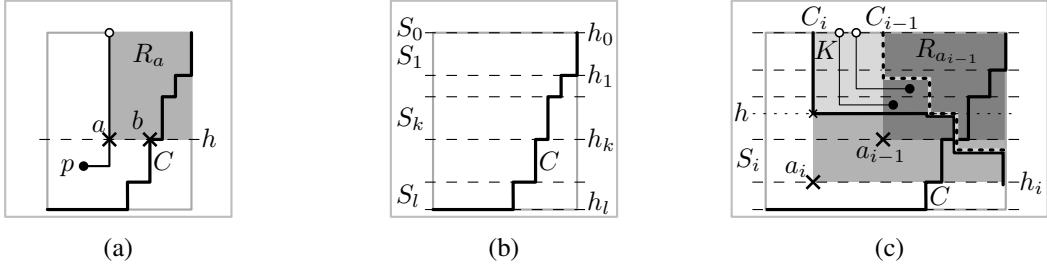


Fig. 7: The strip condition. a) The horizontal strip condition of b is satisfied by a . b) The horizontal segments of C partition the strips S_0, S_1, \dots, S_k . c) Constructing a planar labeling from a sequence of valid rectangles.

among these 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 length is a solution of minimum length. 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 . We do not include C in either R_T or R_R , so these regions are open at C .

For any point $a \in R$, we define the rectangle R_a , spanned by the top-right corner of R and a .

We define R_a 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_a , only if it is not the upper left corner.

A rectangle R_a is *valid* if the number of sites of P above C that belong to R_a is at least as large as the number of ports on the top side of R_a . The central idea is that the labels on the top side of a valid rectangle R_a can be connected to the sites in R_a by leaders that are completely contained inside that rectangle. We are now ready to present the *strip condition*.

Condition 1. *The horizontal strip condition of the point $b \in C$ is satisfied if there exists a point $a \in R_T$ with $y(a) = y(b)$ and $x(a) \leq x(b)$ such that R_a is valid.*

Without loss of generality we may assume that the curve C is rectilinear. The condition is named after the horizontal segments through points in C .

We now prove that, for a given xy -monotone curve C connecting 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 all points of C satisfy the strip condition.

Lemma 3. *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 point of C satisfies the strip condition.*

Proof. For the proof we call a region $S \subseteq R$ *balanced* if it contains the same number of sites as it contains ports. To show that the conditions are necessary, let \mathcal{L} be a planar solution for which all top leaders are above C . Consider a point $b \in C$. If $y(p) \geq y(b)$ for all sites $p \in P'$, rectangle R_a with $a = (0, y(b))$ is clearly valid, and thus the strip condition for b is satisfied. Hence, assume that there is a site $p \in P'$ with $y(p) < y(b)$ that is labeled by a top label; see Fig. 7a. Then, the vertical segment of this leader crosses the horizontal line h through b . Let a denote the rightmost such crossing of a leader of a site in P' with h . We claim that R_a is valid. To see this, observe that all sites of P' top-right of a are contained in R_a . Since no leader may cross the vertical segment defining a , the number of sites in $R_a \cap R_T$ is balanced, i.e., R_a is valid.

Conversely, we show that if the conditions are satisfied, then a corresponding planar solution exists. 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_l , respectively, and let h_0 be the top side of R . The line segments h_1, \dots, h_l partition R_T into l strips, which we denote by S_1, \dots, S_l from top to bottom, such that strip S_i is bounded by h_i from below for $i = 1, \dots, l$; see Fig. 7b. Additionally, we define S_0 to be the empty strip that coincides with h_0 . Let S_k be the last strip that contains sites of P' . For $i = 0, \dots, k-1$, let a'_i denote the rightmost point of $h_i \cap R_T$ such that $R_{a'_i}$ is valid. Such a point exists since the leftmost point of $h_i \cap C$ satisfies the strip condition. We define a_i to be the point on $h_i \cap R_T$, whose x -coordinate is $\min_{j \leq i} \{x(a'_j)\}$. Note that R_{a_i} is a valid rectangle, as, by definition, it completely contains some valid rectangle $R_{a'_j}$ with $x(a'_j) = x(a_i)$. Also by definition the sequence formed by the points a_i has decreasing x -coordinates, i.e., the R_{a_i} grow to the left; see Fig. 7c.

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

For $i = 0$, $\mathcal{L}_0 = \emptyset$ is a planar solution. Consider a strip S_i with $0 < i \leq k$; see Fig. 7c. 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 $R_{a_{i-1}}$ to the sites in $R_{a_{i-1}}$ above C_{i-1} . To extend it to a planar solution \mathcal{L}_i , we additionally need to match the remaining labels on the top side of R_{a_i} and construct a corresponding curve C_i . Let P_i denote the set of unlabeled sites in R_{a_i} . By the validity of R_{a_i} ,

this number is at least as large as the number of unused ports at the top side of R_{a_i} . We arbitrarily match these ports to the topmost sites of P_i that are not labeled in \mathcal{L}_{i-1} . We denote the resulting labeling by \mathcal{L}'_i . We observe that no leader of \mathcal{L}'_i crosses the curve C_{i-1} , and hence such leaders cannot cross leaders in \mathcal{L}_{i-1} . Let h be the topmost horizontal line such that all labeled sites of \mathcal{L}'_i lie above h . Further, let K be the rectangle that is spanned by the top-left corner of $R_{a_{i-1}}$ and the intersection of h with the left side of R_{a_i} . Since the ports of \mathcal{L}'_i lie on the top side of K , any leader's bend of \mathcal{L}'_i lies in K . We apply Lemma 1 on \mathcal{L}'_i to obtain a planar labeling \mathcal{L}''_i , which has no crossings with \mathcal{L}_{i-1} . Hence, the set $\mathcal{L}_i = \mathcal{L}''_i \cup \mathcal{L}_{i-1}$ is the required labeling.

It remains to construct the curve C_i . For this, we start at the top-left corner of R_{a_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_{a_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.

In the following we observe two properties of the strip condition. The first observation states that the horizontal strip condition at (x, y) is independent of the exact shape of the curve between the top-right corner r of R and (x, y) , as long as the number of sites above the curve remains the same. This is crucial for using dynamic programming to test the existence of a suitable curve. The second observation states that the horizontal strip condition can only be violated when the curve passes the x -coordinate of a top site. This enables us to discretize the problem.

Observation 1. *The horizontal strip condition for a point $a \in C$ depends only on the number of sites in R_a above C , in the following sense: Let C and C' be two xy -monotone curves from r to a with u sites in R_a above C and C' , respectively. Then, a satisfies the strip condition for C if and only if it satisfies the strip condition for C' .*

Observation 2. *Let $a, b \in C$, $x(a) \leq x(b)$ such that there is no top site ℓ with $x(a) < x(\ell) \leq x(b)$. Then, a satisfies the horizontal strip condition for C if and only if b satisfies the horizontal strip condition for C .*

Symmetric statements hold for the vertical strip condition. In the following, we say that a point (x, y) on a curve C satisfies the *strip condition* if it satisfies both the horizontal and the vertical strip condition.

3 Algorithm for the Two-Sided Case

How can we find an xy -monotone curve C that satisfies the strip conditions? For that purpose we only consider xy -monotone curves contained in some graph G that is dual to the rectangular grid induced by the sites and ports of the given instance. Note that this is not a restriction since all leaders are contained in the grid induced by the sites and ports. Thus, every xy -monotone curve that does not intersect the leaders can be transformed into an equivalent xy -monotone curve that lies on G .

When traversing an edge e of G , we pass the x - or y -coordinate of exactly one entity of our instance; either a site (*site event*) or a port (*port event*). When passing a site, the position of the site relative to e (above/below e or right/left of e) decides whether 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.

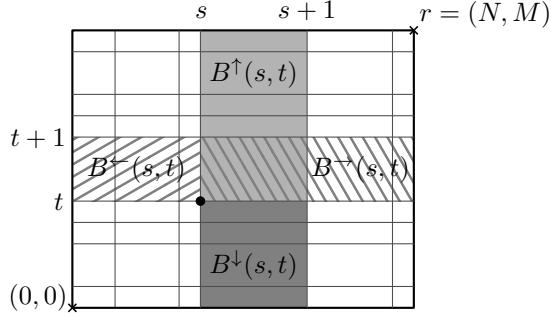


Fig. 8: The four boxes $B^\uparrow(s, t)$, $B^\downarrow(s, t)$, $B^\leftarrow(s, t)$ and $B^\rightarrow(s, t)$ defined by grid point (s, t) .

Let m_R and m_T be the numbers of ports on the right and top side of R , respectively. Also, let $N = n + m_T + 2$ and $M = n + m_R + 2$, then the grid G has size $N \times M$. We define the grid points as $G(s, t)$, $0 \leq s \leq N$, $0 \leq t \leq M$ with $G(0, 0)$ being the bottom-left and $r := G(N, M)$ being the top-right corner of R . Finally, let $G_x(s) := x(G(s, 0))$ and $G_y(t) := y(G(0, t))$.

For each grid point (s, t) that is neither on the topmost row nor on the rightmost column, we define four boxes $B^\uparrow(s, t)$, $B^\downarrow(s, t)$, $B^\leftarrow(s, t)$ and $B^\rightarrow(s, t)$ as follows; see Fig. 8 for an illustration.

1. $B^\uparrow(s, t) = \{(x, y) \in R \mid G_x(s) \leq x \leq G_x(s+1) \wedge y \geq G_y(t)\}$
2. $B^\downarrow(s, t) = \{(x, y) \in R \mid G_x(s) \leq x \leq G_x(s+1) \wedge y \leq G_y(t)\}$
3. $B^\leftarrow(s, t) = \{(x, y) \in R \mid G_y(t) \leq y \leq G_y(t+1) \wedge x \leq G_x(s)\}$
4. $B^\rightarrow(s, t) = \{(x, y) \in R \mid G_y(t) \leq y \leq G_y(t+1) \wedge x \geq G_x(s)\}$

We define a table $T[(s, t), u, b]$ that assigns to each grid position (s, t) and number of points u and b a Boolean value. We define $T[(s, t), u, b]$ to be `true` if and only if there exists an xy -monotone curve C satisfying the following conditions.

- (i) Curve C starts at r and ends at $G(s, t)$.
- (ii) Inside the rectangle spanned by r and $G(s, t)$, there are u sites of P above C and b sites of P below C .
- (iii) For each grid point on C , the strip condition holds.

These conditions together with Proposition 1 and Lemma 3 imply that the instance admits a planar solution if and only if $T[(0, 0), u, b] = \text{true}$ for some u and b .

We define a Boolean function $S[(s, t), u, b]$ that is true if and only if the strip condition at (s, t) is satisfied for some xy -monotone curve C (and thus by Observation 1 for all such curves) from r to $G(s, t)$ with u sites above and b sites below C . The following lemma gives a recurrence for T , which is essentially a disjunction of two values, each of which is determined by distinguishing three cases.

Lemma 4. *For $s = N$ and $t = M$, it holds that $T[(s, t), 0, 0] = \text{true}$. For $s \in [0, N - 1]$ and $t \in [0, M - 1]$, it holds that*

$$T[(s, t), u, b] = \begin{cases} T[(s+1, t), u, b] \wedge S[(s, t), u, b] & \text{if } L \cap B^\uparrow(s, t) \neq \emptyset \\ T[(s+1, t), u-1, b] & \text{if } P \cap B^\uparrow(s, t) \neq \emptyset \\ T[(s+1, t), u, b] & \text{if } P \cap B^\downarrow(s, t) \neq \emptyset \end{cases} \vee \\ \begin{cases} T[(s, t+1), u, b] \wedge S[(s, t), u, b] & \text{if } L \cap B^\rightarrow(s, t) \neq \emptyset \\ T[(s, t+1), u, b-1] & \text{if } P \cap B^\rightarrow(s, t) \neq \emptyset \\ T[(s, t+1), u, b] & \text{if } P \cap B^\leftarrow(s, t) \neq \emptyset \end{cases}.$$

Proof. We show equivalence of the two terms. Let C be an xy -monotone curve from r to (s, t) . Let e be the last segment of C and let $C' = C - e$. Since C is xy -monotone, C' ends either

at the grid point $(s + 1, t)$ or at $(s, t + 1)$. Without loss of generality, we assume that C' ends at $(s + 1, t)$. We show that $T[(s, t), u, b] = \text{true}$ if and only if the first term of the right hand side is true. Analogous arguments apply for C' ending at $(s, t + 1)$ and the second term. Note that, by construction, property (i) is satisfied for C and C' .

We distinguish cases based on whether the traversal along the segment e from $(s+1, t)$ to (s, t) is a port event or a site event.

Case 1: Traversal of e is a port event. Since e passes a port, all sites that lie in the rectangle spanned by r and $G(s, t)$ also lie in the rectangle spanned by r and $G(s+1, t)$. Thus, the numbers u and b of such sites above and below C is the same as the numbers of sites above and below C' , respectively. Hence, property (ii) holds for C if and only if it holds for C' .

Because C' is a subset of C , the strip condition holds for every point of C if and only if it holds for every point of C' and for (s, t) . Thus, property (iii) is satisfied for C if and only if it is satisfied for C' and $S[(s, t), u, b] = \text{true}$.

Case 2: Traversal of e passes a site p . For property (iii), observe that, since the traversal of e is a site event, the strip conditions for (s, t) and $(s + 1, t)$ are equivalent by Observation 2.

For property (ii), note that, except for p , the sites that lie in the rectangle spanned by r and $G(s, t)$ also lie in the rectangle spanned by r and $G(s + 1, t)$. If p lies above e , there are u sites above and b sites below C if and only if there are $u - 1$ sites above and b sites below C' , respectively. Symmetrically, if p lies below e , there are u sites above and b sites below C if and only if there are u sites above and $b - 1$ sites below C' , respectively. In either case, C satisfies condition (ii) if and only if C' does. \square

Clearly, the recurrence from Lemma 4 can be used to compute T in polynomial time via dynamic programming. 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 $G(s, t)$. Thus, in the following we work with $T[(s, t), u]$. 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 $B_T[s, t]$ how large a deficit of 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. 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 as follows: Assume we have an entry $T[(s, t), u]$, and we wish to check its strip condition. Consider a curve C from r to $G(s, t)$ such that u sites are above C . The strip condition is satisfied if and only if $u + B_T[s, t]$ is at least as large as the number of top ports to the right of $G(s, t)$. This is true if the rectangle spanned by the lower left corner of $K_{B_T[s, t]}$ and r contains at least $u + B_T[s, t]$ sites, which is an upper bound on the number of ports on the top side of that rectangle.

We now show how to compute B_T in $O(n^2)$ time. We compute each row separately, starting from the left side. We initialize $B_T[0, t] = 0$ for $t = 0, \dots, M$, since in the final column, no deficit can be compensated. The matrix B can be filled by a horizontal sweep. The entry $B_T[s, t]$ can be derived from the already computed entry $B_T[s - 1, t]$. If the step from $s - 1$ to s is a site event, the amount of the deficit we can compensate increases by 1. If it is a port event the amount of the deficit we can compensate decreases by 1. Moreover, the compensation potential never goes below 0. We obtain

$$B_T[s, t] = \begin{cases} B_T[s - 1, t] + 1 & \text{if step is site event,} \\ \max\{B_T[s - 1, t] - 1, 0\} & \text{if step is port event.} \end{cases}$$

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 [14], we can still backtrack the dynamic program and find a solution corresponding to an entry in the last cell in the same running time. We have the following theorem.

Theorem 1. TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES *can be solved in $O(n^3)$ time using $O(n^2)$ space.*

Our next goal is to improve the performance of our algorithm by reducing the number of dimensions of the table T by 1. As a first step, we show that for any search position $\mathbf{c} = (s, t)$, the set of all u with $T[\mathbf{c}, u] = \text{true}$ is an interval.

Lemma 5. *Let $T[\mathbf{c}, u] = T[\mathbf{c}, u'] = \text{true}$ with $u < u'$. Then $T[\mathbf{c}, u''] = \text{true}$ for $u \leq u'' \leq u'$.*

Proof. Let C be a curve corresponding to the entry $T[\mathbf{c}, u]$. That is C connects r to \mathbf{c} such that any point on C satisfies the strip condition. Similarly, let C' be a curve corresponding to $T[\mathbf{c}, u']$; see Fig. 9.

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 curve C'' that starts at r and follows C until v , then moves down vertically to v' , and from there follows C' to p . Obviously C'' is an xy -monotone 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 points of C'' satisfy the strip condition, then this implies $T[\mathbf{c}, u''] = \text{true}$.

We show that indeed the strip condition is satisfied for any point on C'' . Let C_1 be the subcurve of C'' that connects r to v , let C_2 be the segment vv' and let C_3 be the subcurve of C'' that connects v' to \mathbf{c} . Since C_1 is also a subcurve of C and it starts at r , it directly follows that any point of C_1 satisfies the strip condition. For the points on C_2 we can argue as follows. Since C_2 lies below C and any point of C satisfies the horizontal strip condition, any point of C_2 must satisfy the horizontal strip condition. Analogously, because C_2 lies above C' and any point of C' satisfies the vertical strip condition, each point of C_2 must satisfy the vertical strip condition. Finally, since C_3 is a subcurve of C' , any point of C' satisfies the strip condition and any point of C_1 and C_2 satisfies the strip condition, it directly follows that any point of C_3 satisfies the strip condition. \square

Using Lemma 5, we can reduce the dimension of the table T by 1. It suffices to store at each entry $T[\mathbf{c}]$ the boundaries of the u -interval. This reduces the amount of storage to $O(n^2)$ without increasing the running time. Using Hirschberg's algorithm, the storage for T even decreases to $O(n)$. Tables B_T and B_R still have size $O(n^2)$, however.

Our next goal is to reduce the running time to $O(n^2)$. An entry in $B_T[s, t]$ tells us which deficits can be compensated. This can also be interpreted as a lower bound on the number of sites a curve from r to $G(s, t)$ must have above it, in order to satisfy the horizontal strip condition. Namely, let $\tau_{s,t}$ denote the number of ports on the top side of the rectangle spanned by $G(s, t)$ and r . Then $u \geq \tau_{s,t} - B_T[s, t]$ is equivalent to satisfying the horizontal strip condition for the strip directly above $G(s, t)$. Similarly, the corresponding entry $B_R[s, t]$ gives a lower bound on the number of sites below such a curve, which in turn, together with the number of sites contained in the rectangle spanned by $G(s, t)$ and r implies an *upper bound* on the number of sites above the curve. Thus, B_T , B_R , and the information on how many sites, top ports and right ports are in the

rectangle spanned by $G(s, t)$ and r together imply a lower and an upper bound, and thus an interval of u -values, for which the horizontal and vertical strip conditions at $G(s, t)$ is satisfied. Hence the program can simply intersect this interval with the union of the intervals obtained from $T[(s, t) - \Delta c]$, where Δc has exactly one non-zero entry, which is 1. Consequently, the amount of work per entry of T is still $O(1)$. Note that by Lemma 5 the result of this computation is again an interval.

Now we turn to the space consumption. Hirschberg's algorithm [14] immediately reduces the space consumption of T to $O(n)$. We would like to apply the same trick to B_T and to B_R . Recall that B_T is computed from left to right and B_R from bottom to top. Unfortunately, this is opposite to the order we use for computing T , where we proceed from top-right to bottom-left. We can fix this problem by running the dynamic programs for computing B_T and B_R backwards, by precomputing the entries of B_T and B_R on the top and right side, and then running the updates backwards. This allows us to use Hirschberg's algorithm, and the algorithms can run in a synchronized manner such that at any point in time the required data is available, using only $O(n)$ space.

A new issue, however, appears. The update $B_T[s, t] = \max\{B_T[s - 1, t] - 1, 0\}$ is not easily reversible. When running the dynamic program backwards, it is not clear whether $B_T[s, t] = 0$ implies $B_T[s - 1, t] = 0$ or $B_T[s - 1, t] = 1$ at a port step. To remedy this issue, fix a column s of the table corresponding to a port event and consider the circumstances under which $B_T[s - 1, t] - 1 = -1$, i.e., $B_T[s - 1, t] = 0$. This implies that, for any rectangle K with lower right corner $G(s - 1, t)$ whose top side is contained in the top side of R , there are at most as many sites in K as there are ports in the top side of K . Assume that this is the case for some fixed value t_0 , i.e., $B_T[s - 1, t_0]$. Since the possible rectangles for an entry $B_T[s - 1, t]$ with $t \geq t_0$ contain at most as many sites as the ones for $B_T[s - 1, t_0]$, this implies $B_T[s - 1, t_0] = B_T[s - 1, t] = 0$ for all $t \geq t_0$. If on the other hand, t_0 is such that $B_T[s - 1, t_0] > 0$, then the rectangles corresponding to $B_T[s - 1, t]$ for $t < t_0$ contain at least as many sites as the ones for $B_T[s - 1, t_0]$, and we have $B_T[s - 1, t] \geq B_T[s - 1, t_0]$ for $t < t_0$. Thus, there is a single gap t_0 such that, for any $t \geq t_0$, we have $B_T[s - 1, t] = 0$ and, for any $t < t_0$, we have $B_T[s - 1, t] > 0$; see Fig. 10. Storing this gap for each column s that is a port event allows us to efficiently reverse the dynamic program. Note that storing one value per column only incurs $O(n)$ space overhead. Of course, the same approach works for the dynamic program computing B_R . Overall, we have shown the following theorem.

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

4 Extensions

The techniques we used to obtain Theorem 2 can be applied to solve a variety of different extensions of the two-sided labeling problem with adjacent sides. We now show how to a) generalize to sliding ports instead of fixed ports, b) maximize the number of labeled sites, and c) minimize the total leader length in a planar solution.

4.1 Sliding Ports

First, observe that Proposition 1, which guarantees the existence of an xy -separated planar solution, also holds for sliding ports. The same proofs apply by conceptually fixing the ports of a given planar solution when applying the rerouting operations. The following lemma shows that, without loss of generality, we can simply fix all ports at the bottom-left corner of their corresponding labels. This immediately solves the problem.

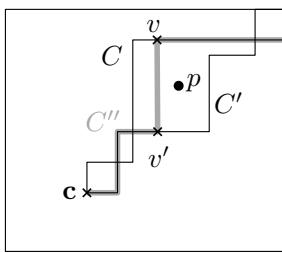


Fig. 9: Sketch for the proof of Lemma 5.

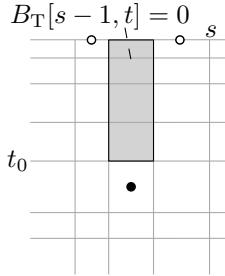


Fig. 10: The gap t_0 is defined such that we have $B_T[s-1, t] = 0$ for any $t \geq t_0$, and $B_T[s-1, t] > 0$ for any $t < t_0$.

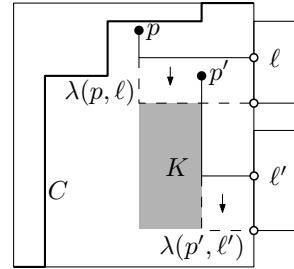


Fig. 11: Sketch for the proof of Lemma 6

Lemma 6. *If there exists an xy -separated planar solution \mathcal{L} for the two-sided boundary labeling problem with adjacent sides and sliding ports, then there also exists an xy -separated planar solution \mathcal{L}' in which the ports are fixed at the bottom-left corners of the labels.*

Proof. We show how to transform \mathcal{L} into \mathcal{L}' . Let C be the xy -monotone curve that separates the top leaders from the right leaders of \mathcal{L} . We move the ports induced by \mathcal{L} to the bottom-left corner of their corresponding labels such that the assignment between labels and sites remains; see Fig. 11. Obviously, the bends of the leaders connected to the right side only move downwards. Thus, the leaders lie entirely below C . Symmetrically, the bends of the leaders connected to the top side only move to the left and thus these leaders lie entirely above C .

Consequently, only conflicts between the same type of leaders can arise. Consider the topmost intersection of two leaders $\lambda(p, \ell)$ and $\lambda(p', \ell')$ connected to the right side and assume that p lies to the left of p' . Let K be the rectangle that is spanned by the bends of $\lambda(p, \ell)$ and $\lambda(p', \ell')$. Due to moving the ports downwards, the leaders lie entirely below C and the bend of $\lambda(p', \ell')$ must lie below $\lambda(p, \ell)$. Hence, K lies completely in R_R . In order to resolve the conflict, we reroute p to ℓ' and p' to ℓ using the bottom-left corners of ℓ and ℓ' as ports. Obviously, the leaders only change on ∂K . Therefore, new conflicts can only arise on the left and bottom sides of K . In particular, only the leader of ℓ' can be involved in new conflicts, while the leader of ℓ is free of any conflict. Thus, after finitely many such steps we have resolved all conflicts, from top to bottom. Symmetric arguments apply for the leaders connected to the top side. \square

Theorem 3. **TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES AND SLIDING PORTS** can be solved in $O(n^2)$ time using $O(n)$ space.

4.2 Maximizing the Number of Labeled Sites

So far our algorithm only returns a leader layout if there is a planar solution that matches each label to a site. As Bekos et al. [5, Fig. 16] observed, this need not always be the case, so it becomes important to be able to maximize the number of labels connected to sites in a planar solution. We achieve this by removing labels from a given instance and using our algorithm to decide whether a crossing-free solution exists.

Lemma 6 shows that we can move top ports to the left and right ports to the bottom without making a solvable instance unsolvable. Thus, it suffices to remove the rightmost top labels and the topmost right labels. Let k be the number of labels we want to use with k_T of them being top labels and k_R right labels, so that $k_T + k_R = k$. For a given k , we can decide whether a crossing-free solution that uses exactly k labels exists by removing the $m_T - k_T$ rightmost top labels and the $m_R - k_R$ topmost right labels for any possible k_T and k_R . We therefore start

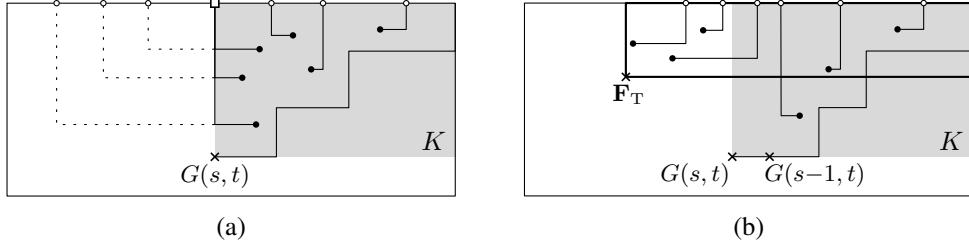


Fig. 12: Illustration of the curve C and the rectangle K spanned by $G(s, t)$ and the top-right corner of R . a) There are more sites than ports in K above C . The unlabeled sites are connected to a dummy port located at the top-left corner of K . The dummy port is illustrated as a square. b) There are more ports than sites in K above C . The unlabeled ports are labeled to sites that lie to the left of K , which induce the front with bottom-left point F_T .

with $k_T = \min\{k, m_T\}$ and $k_R = k - k_T$. We keep decreasing k_T and increasing k_R by 1, until a crossing-free solution is found or $k_R = \min\{k, m_R\}$. In the latter case, no crossing-free solution that uses exactly k labels exists. With this approach we can use binary search to find the maximum k , using our algorithm up to k times per step. Since $k \leq n$, this yields an algorithm for TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES that maximizes the number of labeled sites that runs in $O(n^3 \log n)$ time and uses $O(n)$ space.

Theorem 4. TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES can be solved in $O(n^3 \log n)$ time using $O(n)$ space such that the number of labeled sites is maximized.

Assume that t sites cannot be labeled. Then TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES can be solved in $O(n^2 t \log t)$ time using $O(n)$ space and such that the number of labeled sites is maximized. To that end we use the following approach. We check for $h = 2^i$ with $i = 0, 1, 2, \dots$ whether there is a planar solution with h unlabeled sites. We stop this procedure when we have found such a solution, which takes place after $\lceil \log(t) \rceil$ steps. Using the approach described above, we need $O(n^2 t)$ time for each test. We then know that $\frac{h}{2} < t \leq h$. We apply a binary search to determine t . Overall, this approach needs $O(n^2 t \log t)$ time.

4.3 Minimizing the total leader length

Recall that, by Proposition 1, there always exists a length-minimal planar solution that is xy -separated. To obtain a length-minimal planar solution, we mainly change the table T used by the dynamic program given in Section 3. Let C be an xy -monotone curve C that starts at r and ends at $G(s, t)$. We assign to every table entry the length of the leaders that are connected to the ports in the rectangle K spanned by r and $G(s, t)$.

If there are more sites than top ports in K above C , we have to connect some of these sites to ports that lie to the left of K ; see Fig. 12a. The vertical lengths of their leaders, however, are fixed. We imagine a dummy top port at the left border of K and connect all unlabeled sites to this port. When traversing the grid horizontally, this dummy port moves to the left. In order to update the total length of the leaders in K , we only have to keep track of the number of unlabeled sites and increase the horizontal length of their leaders. The sites in K below C are handled analogously.

If there are more top ports than sites in K above C , we have to connect these ports to sites that lie to the left of K ; see Fig. 12b. In order to remember which sites are already labeled, we store the *top front* as the rectangle with top-right corner r that includes all sites that are already connected to a top port inside K , and the *right front* as the rectangle with top-right corner r that includes all sites that are already connected to a right port inside K .

Let $\mathbf{F}_T = (x_T, y_T)$ be the bottom-left point of the top front for a given xy -monotone curve C that starts at r and ends at $G(s, t)$. Similarly, let $\mathbf{F}_R = (x_R, y_R)$ be the right front for C . We define $T[\mathbf{c} = (s, t), u, \mathbf{F}_T, \mathbf{F}_R] = (l, g_T, g_R)$ if there exists an xy -monotone curve C and leaders inside $K \cup F_T \cup F_R$ such that the following conditions hold, otherwise it contains $(-1, 0, 0)$.

- (i) Curve C starts at the top-right corner r of R and ends at $G(s, t)$.
- (ii) Inside the rectangle K spanned by r and $G(s, t)$, 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.
- (iv) The sites in $K \cup \mathbf{F}_T \cup \mathbf{F}_R$ are connected to the ports on the border of $K \cup \mathbf{F}_T \cup \mathbf{F}_R$ such that the induced solution is planar, length-minimal, the sites above C or in \mathbf{F}_T are only connected to top ports, and the sites below C or in \mathbf{F}_R are only connected to right ports.
- (v) There are g_T unlabeled top sites and g_R unlabeled right sites in K .

Note that g_T and g_R depend on s, t, u and can be precomputed, but to make the algorithm more intuitive, we update these values on-line and store them in T . We first describe how to handle the top front while traversing the grid. Initially, $\mathbf{F}_T = G(s, t)$. As long as we have more top sites than top ports in K , we can connect all ports to sites and thus can maintain $\mathbf{F}_T = G(s, t)$. Once we have exactly the same number of top ports and top sites in K and we encounter a port event for a top port, we have to check the strip condition and find the rightmost point F_T with $y(F_T) = G_y(t)$ such that the rectangle $R_{\mathbf{F}_T}$ spanned by \mathbf{F}_T and r is valid. By storing \mathbf{F}_T , we know that all ports to the right of $x(\mathbf{F}_T)$ are already connected to a site, all sites to the top-right of \mathbf{F}_T are already connected to a port, and all top sites to the bottom-left of \mathbf{F}_T have to be connected to a port that lies to the left of $x(\mathbf{F}_T)$. Thus, we do not have to check new strip conditions until $s < x(\mathbf{F}_T)$. We handle \mathbf{F}_R similarly.

We now look at the length of the top leaders, the length of the right leaders can be handled similarly. Note that by moving from t to $t - 1$, the length of the top leaders does not change. If $g_T > 0$, we imagine an additional port at $x(\mathbf{F}_T)$ that can be connected to g_T top sites. When moving from s to $s - 1$, we add $g_T \cdot (G_x(s) - G_x(s - 1))$ to l . When we calculate a new value \mathbf{F}_T by checking the strip condition, we can immediately connect all top sites inside the top front to top ports, and add the corresponding leader length to l . Thus, we only encounter site events for sites that are a) inside $\mathbf{F}_T \setminus K$ or b) have to be connected to a top port that lies to the left of $x(\mathbf{F}_T)$. In case a) we do not change l , in case b) we connect the site to the imaginary port, add the length of the corresponding leader to l and increase g_T by 1. When we encounter a port event, if the port lies inside $\mathbf{F}_T \setminus K$, we do not change l , otherwise we can connect any of the unlabeled sites to this port. We add the horizontal distance between $G_x(s)$ and the port to l and decrease g_T by 1. Note that by choosing any unlabeled site, the resulting solution may not be planar. However, because the bends of all unconnected sites will be above C , we can use Lemma 1 to remove the crossings without changing the total leader length.

Since the matrix now has four additional fields, the running time and storage is increased by a factor of n^4 over the algorithm from Theorem 1. Additionally, we need $O(n \log n)$ time to check the strip condition and to compute a length-minimal solution for the sites and ports inside $\mathbf{F}_T \setminus K$ and $\mathbf{F}_R \setminus K$.

Theorem 5. TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES *can be solved in $O(n^8 \log n)$ time using $O(n^6)$ space such that the total leader length is minimized.*

Using an appropriate data structure to precompute the fronts, it may be possible to decrease the running time slightly.

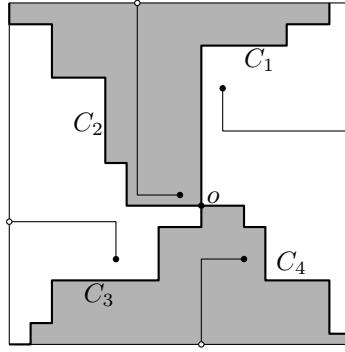


Fig. 13: The curves C_1 , C_2 , C_3 and C_4 meeting at the point o partition the rectangle into four regions.

5 The Three- and Four-Sided Cases

In this section, we also allow labels on the bottom and the left side of R . In order to solve an instance of the three- and four-sided case, we adapt the techniques we developed for the two-sided case. We assume that the ports are fixed and the number of labels and sites is equal. In Section 5.1 we first analyze the structure of planar solutions obtaining a result similar to Proposition 1. In Sections 5.2 and 5.3, we present algorithms for the three- and four-sided cases.

5.1 Structure of Three- and Four-Sided Planar Solutions

Similar to our approach to two-sided boundary labeling, we pursue the idea that if there exists a planar solution, then we can also find a planar solution such that there are four xy -monotone curves connecting the four corners of R to a common point o , and such that these curves separate the leaders of the different label types from each other; see Fig. 13. To that end, we first show that leaders of left and right labels can be separated vertically and leaders of top and bottom labels can be separated horizontally. Afterwards, we apply the result of Lemma 2 in order to resolve the remaining overlaps, e.g., between top and right leaders. We first introduce some notions.

Definition 2. A planar solution for the four-sided boundary labeling problem is

- (i) *x-separated* if there exists a vertical line ℓ such that the sites that are labeled to the left side are to the left of ℓ and the sites that are labeled to the right side are to the right of ℓ , and
- (ii) *y-separated* if there exists a horizontal line ℓ such that the sites that are labeled to the top side are above ℓ and the sites that are labeled to the bottom side are below ℓ .

A left leader λ and a right leader λ' overlap if $x(\text{bend}(\lambda)) > x(\text{bend}(\lambda'))$. Analogously, a bottom leader λ and a top leader λ' overlap if $y(\text{bend}(\lambda)) > y(\text{bend}(\lambda'))$. Hence, a planar solution \mathcal{L} is both *x-separated* and *y-separated* if and only if no left and right leaders overlap, and no bottom and top leaders overlap. We are now ready to prove that we can always find a planar solution that is both *x-separated* and *y-separated*, if a solution exists.

Lemma 7. If there exists a planar solution for the four-sided boundary labeling problem, then there exists a planar solution \mathcal{L} that is both *x-separated* and *y-separated*.

Proof. Among all planar solutions let \mathcal{L} be one that minimizes $|\mathcal{L}|_x + |\mathcal{L}|_y$. We prove that then \mathcal{L} is *x*- and *y*-separated by showing that otherwise we could reroute some leaders and obtain a planar solution \mathcal{L}' with $|\mathcal{L}'|_x + |\mathcal{L}'|_y < |\mathcal{L}|_x + |\mathcal{L}|_y$.

Assume that \mathcal{L} is not *x-separated*. Symmetric arguments hold for the case that \mathcal{L} is not *y-separated*. Then there exist sites p_R and p_L with $x(p_R) < x(p_L)$, such that p_R is labeled by a right

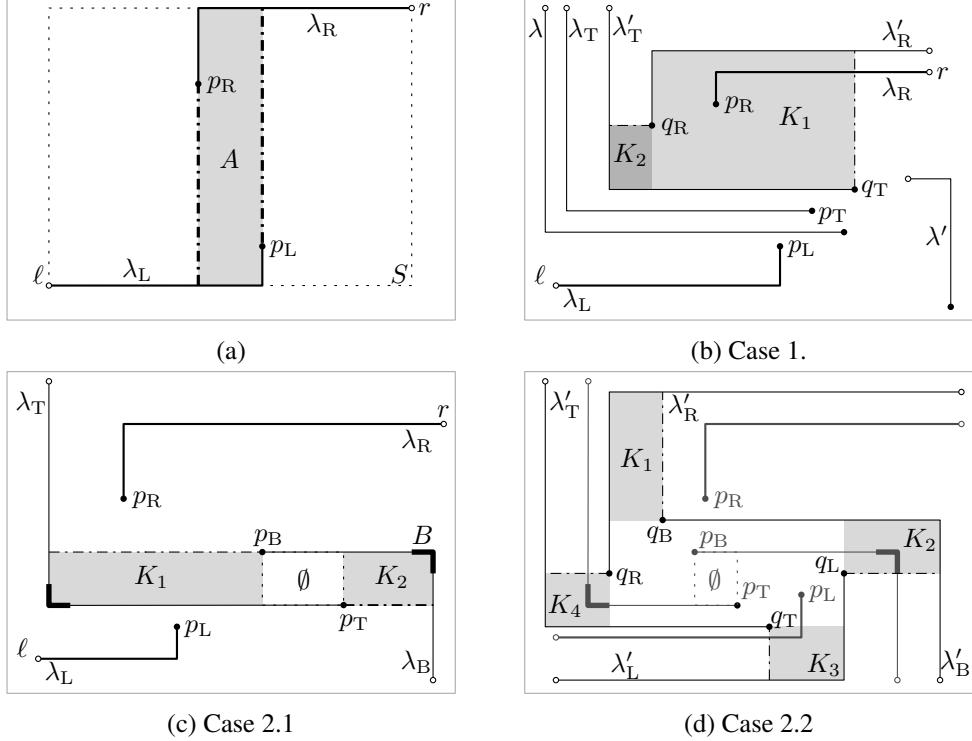


Fig. 14: Different constellations of leaders intersecting the rectangle A . a) The rectangle A is empty. b)–d) Different cases where A is intersected by a top leader. A is not explicitly illustrated, but spanned by $\text{bend}(\lambda_R)$ and $\text{bend}(\lambda_L)$.

port r , and p_L is labeled by a left port ℓ ; see Fig. 14a. Without loss of generality, assume that the horizontal segment of $\lambda_R = \lambda(p_R, r)$ is above the horizontal segment of $\lambda_L = \lambda(p_L, \ell)$, otherwise we mirror the instance vertically.

We choose p_L and p_R as a closest pair in the sense that the horizontal segments of their leaders have minimum vertical distance among all such pairs. Let A be the rectangle spanned by $\text{bend}(\lambda_L)$ and $\text{bend}(\lambda_R)$. By the minimality of p_L and p_R , that rectangle can only be intersected by top and bottom leader, but not by left or right leaders. If no such leader intersects A , we reroute p_R to the port of λ_L and p_L to the port of λ_R , which decreases $|\mathcal{L}|_x$ without increasing $|\mathcal{L}|_y$; see Fig. 14a. It does not introduce any crossings.

In the following we assume that some leaders intersect A . Without loss of generality we assume that there is a top leader λ_T that intersects A ; otherwise we rotate the instance by 180° . We denote its site by p_T . Let S be the rectangle spanned by the ports ℓ and r ; see Fig. 14a. Depending on the leaders intersecting S , we distinguish two cases. Note that in particular λ_T intersects S .

Case 1: For any top leader λ intersecting S and for any bottom leader λ' intersecting S such that λ and λ' overlap, the site of λ lies to the left of the site of λ' ; see Fig. 14b. Let q_R denote the bottommost site that is connected by a right leader, and that lies in the rectangle spanned by $\text{bend}(\lambda_T)$ and p_R . Since p_R lies in that rectangle, the site q_R exists. We denote the leader of q_R by λ'_R . Further, let q_T be the topmost site that is connected by a top leader and that lies in the rectangle spanned by $\text{bend}(\lambda'_R)$ and the bottom-right corner of R . Since p_T lies in that rectangle, the site q_T exists. We denote its leader by λ'_T .

We now define two rectangles that we use to reroute leaders such that $|\mathcal{L}|_x + |\mathcal{L}|_y$ is decreased and arising crossings can be resolved. The rectangle K_1 is spanned by $\text{bend}(\lambda'_R)$ and q_T , and the

rectangle K_2 is spanned by $\text{bend}(\lambda'_T)$ and q_R .

Claim 2.

- (1) K_1 is only intersected by right leaders whose bends are contained in K_1 ,
- (2) K_2 is only intersected by top leaders whose bends are contained in K_2 , and
- (3) K_1 and K_2 are internally disjoint.

Assuming that the claim holds, we can reroute the sites as follows; we illustrate this rerouting by dash-dotted lines in Fig. 14b. The site q_T is rerouted to the port of λ'_R creating crossings only on the right side of K_1 . The site q_R is rerouted to the port of λ'_T creating crossings only on the top side of K_2 . Each rerouting decreases either $|\mathcal{L}|_x$ or $|\mathcal{L}|_y$ increasing the other one. Further, only crossings between leaders of the same type are created. We apply Lemma 1 to resolve the conflicts without increasing $|\mathcal{L}|_x$ or $|\mathcal{L}|_y$. In the remainder of this case we show that the stated claim holds.

First, we show that K_1 is only intersected by right leaders whose bends lie in K_1 . It is not intersected by any bottom leader, because such a leader would overlap λ'_T , and its site would lie to the left of q_T —a contradiction to the assumption of this case. It is not intersected by any left leader, because such a leader would intersect λ'_T . It is not intersected by any top leader, because such a leader would either intersect λ'_R or contradict the choice of λ'_T . Hence, K_1 can only be intersected by right leaders. Further, all those leaders have their bend in K_1 , because the bottom-right corner is a site connected by a top leader. That leader would be intersected if a right leader intersecting K_1 had its bend outside of K_1 .

Next, we show that K_2 is only intersected by top leaders whose bends lie in K_2 . It is not intersected by any right leader, because such a leader would contradict the choice of λ'_R or intersect λ'_T . It is not intersected by any bottom leader, because such a leader would overlap λ'_T , and its site would lie to the left of q_T —a contradiction to the assumption of this case. It is not intersected by any left leader, because such a leader would intersect λ'_T . Hence, K_2 can only be intersected by top leaders. Further, all those leaders have their bend in K_2 , because the top-right corner is a site connected by a right leader.

Finally, the rectangles K_1 and K_2 are internally disjoint, because K_1 lies to the right of the vertical line through q_R , while K_2 lies to the left of that line.

Case 2: There exist a top leader λ_T intersecting S and a bottom leader λ_B intersecting S such that they overlap and the site of λ_T lies to the right of the site of λ_B ; see Fig. 14c. Among all such pairs we choose λ_T and λ_B such that their horizontal segments have minimal vertical distance. We denote the site of λ_T by p_T and the site of λ_B by p_B . Due to the choice of λ_T and λ_B , the open rectangle that is spanned by p_B and p_T is intersected by no leader. The open rectangle spanned by $\text{bend}(\lambda_T)$ and $\text{bend}(\lambda_B)$ is denoted by B . Depending on the sites that are contained in B , we distinguish four cases.

Case 2.1: The rectangle B contains no sites that are connected by left or right leaders; see Fig. 14c. Let K_1 be the rectangle spanned by $\text{bend}(\lambda_T)$ and p_B , and let K_2 be the rectangle spanned by $\text{bend}(\lambda_B)$ and p_T . While K_1 is only intersected by left leaders, K_2 is only intersected by right leaders. Further, both rectangles are disjoint. We reroute p_B to the port of λ_T and p_T to the port of λ_B . Obviously, this decreases $|\mathcal{L}|_y$ without increasing $|\mathcal{L}|_x$. By applying Lemma 1, we resolve the arising conflicts.

Case 2.2: The rectangle B contains sites that are connected by left leaders as well as sites that are connected by right leaders; see Fig. 14d. Let q_R be the bottommost site in B that is connected to the right. We denote the leader of q_R by λ'_R . Let q_B be the leftmost site with $y(q_B) \geq y(p_B)$ and $x(q_B) \leq x(p_B)$ that is connected to the bottom. Since p_B also satisfies these requirements, the site q_B exists. We denote the leader of q_B by λ'_B . Let q_L be the topmost site in B that is connected to the left. We denote the leader of q_L by λ'_L . Finally, let q_T be the rightmost site with $y(q_T) \leq y(p_T)$ and $x(q_T) \geq x(p_T)$ that is connected to the top. Since p_T also satisfies these

requirements, the site q_T exists. We denote the leader of q_T by λ'_T .

We now define four rectangles that we use to reroute leaders such that $|\mathcal{L}|_x + |\mathcal{L}|_y$ is decreased and arising crossings can be resolved. The rectangle K_1 is spanned by $\text{bend}(\lambda'_R)$ and q_B , the rectangle K_2 is spanned by $\text{bend}(\lambda'_B)$ and q_L , the rectangle K_3 is spanned by $\text{bend}(\lambda'_L)$ and q_T , and the rectangle K_4 is spanned by $\text{bend}(\lambda'_T)$ and q_R . Note that the rectangles K_3 and K_4 are rotationally symmetric to K_1 and K_2 , respectively.

Claim 3.

- (1) K_1 is only intersected by right leaders whose bends are contained in K_1 ,
- (2) K_2 is only intersected by bottom leaders whose bends are contained in K_2 ,
- (3) K_3 is only intersected by left leaders whose bends are contained in K_3 ,
- (4) K_4 is only intersected by top leaders whose bends are contained in K_4 , and
- (5) K_1, K_2, K_3 and K_4 are pairwise internally disjoint.

Assuming that the claim holds, we can reroute the sites in a circular fashion as follows; we illustrate the rerouting as dash-dotted lines in Fig. 14d. The site q_B is rerouted to the port of λ'_R creating crossings only on the right side of K_1 . The site q_L is rerouted to the port of λ'_B creating crossings only on the bottom side of K_2 . The site q_T is rerouted to the port of λ'_L creating crossing only on the left side of K_3 . Finally, the site q_R is rerouted to the port of λ'_T creating crossings only on the top side of K_4 . Each rerouting decreases either $|\mathcal{L}|_x$ or $|\mathcal{L}|_y$ without increasing the other one. Further, only crossings between leaders of the same type are created. We apply Lemma 1 to resolve the conflicts. In the remainder of this case we show that the stated claim holds.

First, we show that K_1 is only intersected by right leaders whose bends lie in K_1 . This rectangle is not intersected by any bottom leader, because q_B is the leftmost site with $y(q_B) \geq y(p_B)$ and $x(q_B) \leq x(p_B)$ that is connected to the bottom. It is not intersected by any top leader, because such a leader would intersect λ'_R whose site lies below q_B . Finally, it is not intersected by any left leader, because such a leader would intersect λ'_T whose site lies to the right of q_B . Hence, only right leaders intersect K_1 . In particular, all those leaders have their bend in K_1 , because the bottom-right corner of K_1 is the site of a bottom leader. That leader would be intersected if a right leader intersecting K_1 had its bend outside of K_1 . Since K_3 is rotationally symmetric to K_1 , we can use symmetric arguments to prove that K_3 is only intersected by left leaders whose bends are contained in K_3 .

Next, we show that K_2 is only intersected by bottom leaders whose bends lie in K_2 . This rectangle is not intersected by any left leader, because such a leader would contradict the choice of q_L . It is also not intersected by any top leader, because such a leader would intersect λ'_R or contradict the choice of λ_T and λ_B . Finally, it cannot be intersected by any right leader, because such a leader would intersect λ'_B . Hence, K_2 is only intersected by bottom leaders. Further, all those leaders have their bend in K_2 , because the bottom-left corner of K_2 is a site connected to a left leader. That leader would be intersected if a bottom leader intersecting K_2 had its bend outside of K_2 . Since K_4 is rotationally symmetric to K_2 , we can use symmetric arguments to prove that K_4 is only intersected by top leaders whose bends are contained in K_4 .

Finally, we show that the rectangles K_1, K_2, K_3 and K_4 are pairwise internally disjoint. For a site p let $v(p)$ denote the vertical line through p and let $h(p)$ denote the horizontal line through p . By construction we have that $h(q_B)$ lies above $h(q_T)$, K_1 lies above $h(q_B)$, and K_3 lies below $h(q_T)$. Hence, the rectangles K_1 and K_3 are internally disjoint. Analogously, we have that $v(q_L)$ lies to the right of $v(q_R)$, K_2 lies to the right of $v(q_L)$, and K_4 lies to the left of $v(q_R)$. Hence, the rectangles K_2 and K_4 are internally disjoint. Further, the sites q_L and q_R lie in between $h(q_B)$ and $h(q_T)$, because both lie in B . Consequently, K_1 and K_3 do not intersect K_2 and K_4 , respectively.

Case 2.3: The rectangle B contains only sites connected by right leaders. We apply the same

procedure as in the previous case. However, we do not need to consider left leaders. Hence, K_3 is removed and K_2 is the rectangle that is spanned by $\text{bend}(\lambda'_B)$ and p_T . By the choice of B , the rectangle K_2 is only intersected by right leaders whose bend is contained in K_2 . Further, the remaining rectangles K_1 , K_2 and K_4 are pairwise internally disjoint. The reroutings are again done in a circular fashion decreasing $|\mathcal{L}|_x + |\mathcal{L}|_y$. Finally, we apply Lemma 1 to resolve crossings.

Case 2.4: The rectangle B contains only sites connected by left leaders. This case can be handled analogously to the previous case by mirroring the instance vertically. \square

This lemma shows that, when searching for a planar solution of the labeling problem, we can restrict ourselves to solutions that are x -separated and y -separated. Let \mathcal{L} denote such a solution, and let ℓ_v and ℓ_h be the lines separating the sites labeled by left and right labels, and the ones labeled by top and bottom labels, respectively. Let $o \in R$ denote the intersection of ℓ_v and ℓ_h , called *center point*. Let r_1, \dots, r_4 denote the corners of R , named in counterclockwise ordering, and such that r_1 is the top-right corner. Consider the rectangles that are spanned by o and r_i for $i = 1, \dots, 4$. Each of them contains only two types of leaders. For example, the top-right rectangle contains only top and right leaders. An x - and y -separated planar solution is *partitioned* if, for each rectangle spanned by o and one of the corners r_i of R , there exists an xy -monotone curve C_i from r_i to o that separates the two different types of leaders contained in that rectangle; see Fig. 13. Our next step is to show that a planar solution can be transformed into a partitioned solution without increasing $|\mathcal{L}|_x$ and $|\mathcal{L}|_y$.

Proposition 2. *If there exists a planar solution \mathcal{L} for FOUR-SIDED BOUNDARY LABELING, then there exists a partitioned solution \mathcal{L}' .*

Proof. By Lemma 7, we can assume that \mathcal{L} is x - and y -separated. Let o be the center point as defined above and let ℓ_v be the vertical line through o . We show how to ensure that the area K of R right of ℓ_v admits an xy -monotone curve from the top-right corner of R to o that separates the top leaders from the right leaders inside K . The remaining cases are symmetric.

Essentially, we proceed as in the proof of Proposition 1 to remove obstructions of types (P1)–(P4); see Fig. 5. We note that in the rerouting, we only shorten vertical segments of top leaders and right segments of right leaders; hence the solution remains x - and y -separated. Moreover, in each step we decrease both $|\mathcal{L}|_x$ and $|\mathcal{L}|_y$. Hence, after finitely many steps all patterns between top and right leaders have been removed without creating new patterns with other types of leaders.

After all patterns have been removed, an xy -monotone curve connecting the top-right corner of R to o , separating the top labels from the right labels, can be found as in the proof of Lemma 2. \square

5.2 Algorithm for the Three-Sided Case

In the three-sided case, we assume that the ports of the given instance I are located on three sides of R ; without loss of generality, on the left, top and right side of R . Basically, we solve a three-sided instance by splitting the instance into two two-sided L -shaped instances that can be solved independently; see Fig. 15a.

Let G be the dual of the grid that is induced by the sites and ports of the given instance. The idea is that each grid point s of G induces two two-sided L -shaped instances with some useful properties. We will show that there is a planar solution for I if and only if there is a grid point s of G such that its induced two-sided instances both have planar solutions. Thus, considering all $O(n^2)$ grid points of G the problem reduces to solve those L -shaped instances of the two-sided case. By means of a simple adaption of the dynamic program presented in Section 3 we solve these instances in $O(n^2)$ time achieving $O(n^4)$ running time in total.

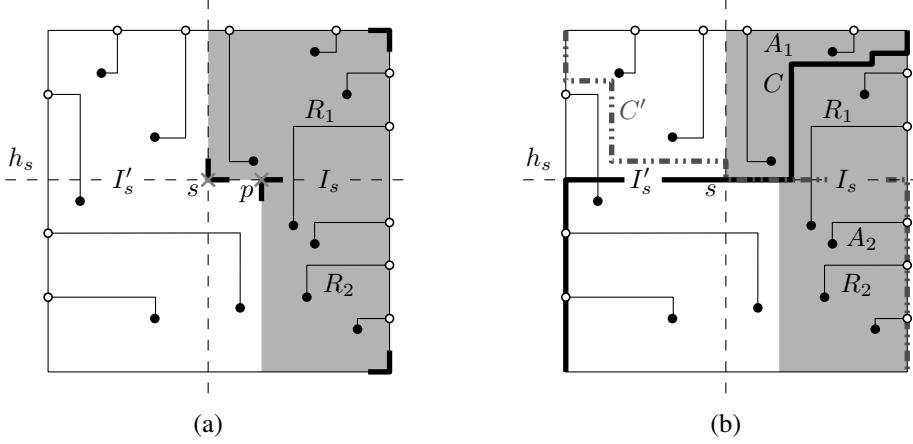


Fig. 15: a) The three-sided instance partitioned into two two-sided L -shaped instances I_s and I'_s . The instances are induced by the grid point s of G and are balanced. b) Illustration of the proof for Lemma 8. Assuming that the grid point s of G , the balanced instances I_s and I'_s , and the curves C and C' are given, a planar solution for the whole instance can be constructed.

In the following we call horizontal and vertical lines through grid points of G *horizontal* and *vertical grid lines*, respectively. We now define the two two-sided L -shaped instances I_s and I'_s of a grid point s of G formally. To that end, let R_1 be the rectangle that is spanned by the top-right corner of R and s , and let $R_2(p)$ be the rectangle that is spanned by a point p on the horizontal grid line h through s and the bottom-right corner of R ; see Fig. 15a. The instance $I_s(p)$ contains all sites and ports in $R_1 \cup R_2(p)$ and $I'_s(p)$ contains all sites and ports in $R \setminus (R_1 \cup R_2(p))$. We say that $I_s(p)$ and $I'_s(p)$ are *balanced* if all right ports lie in $R_1 \cup R_2(p)$, all left ports lie in $R \setminus (R_1 \cup R_2(p))$ and $R_1 \cup R_2(p)$ contains the same number of sites as it contains ports. Since the number of ports and sites in I is equal, this directly implies that $R \setminus (R_1 \cup R_2(p))$ contains the same number of sites as it contains ports. In particular, the choice of balanced instances $I_s(p)$ and $I'_s(p)$ for a grid point s of G is unique with respect to the contained sites and ports; only the location of p might differ. We can therefore write I_s and I'_s for balanced instances and R_1 and R_2 for their defining rectangles. For any solution of I_s and any solution of I'_s , we require that all leaders are completely contained in $R_1 \cup R_2$ and in $R \setminus (R_1 \cup R_2)$, respectively. The next lemma states that a three-sided instance I has a planar solution if and only if it can be partitioned into two two-sided L -shaped instances that have planar solutions. To that end let h_s denote the horizontal grid line through s . Figure 15 illustrates the lemma.

Lemma 8. *There is a planar solution \mathcal{L} for a three-sided instance I if and only if there is a grid point s of G with balanced instances I_s and I'_s over rectangles (R_1, R_2) , an xy -monotone curve C from the top-right corner to the bottom-left corner of R and an xy -monotone curve C' from the top-left corner to the bottom-right corner of R such that*

1. *each point on C satisfies the strip condition with respect to the ports and sites in I_s ,*
2. *C contains the top-left corner of R_2 and the intersection of h_s with the left segment of R ,*
3. *each point on C' satisfies the strip condition with respect to the ports and sites in I'_s ,*
4. *C' contains the top-left corner of R_2 and the intersection of h_s with the right segment of R .*

Proof. First, assume that s , I_s , I'_s , (R_1, R_2) , C and C' exist as required; see Fig. 15b. The curve C partitions R into two regions; we denote the region above C by A_1 and the region below C by A_2 . By Lemma 3, there is a planar solution \mathcal{L}_1 for the sites and ports in A_1 such that all leaders of \mathcal{L}_1 lie in A_1 . Since C contains the top-left corner of R_2 and does not cross h_s until it reaches the intersection point of h_s with the left segment of R , we know that all leaders of \mathcal{L}_1 are contained

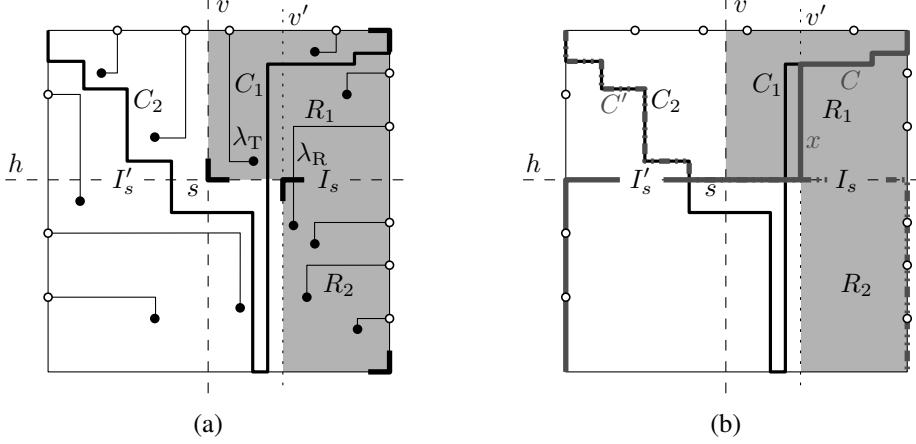


Fig. 16: Illustration of the proof for Lemma 8. It is assumed that the partitioned planar solution \mathcal{L} for the three-sided instance is given. a) By Proposition 2 we can assume that \mathcal{L} is partitioned by the curves C_1 and C_2 . The extremal top leader λ_T induces the site s and the extremal right leader λ_R induces the line v' . b) Based on C_1 , C_2 , h and v' , the curves C and C' can be constructed such that they do not cross any leader of \mathcal{L} .

in $R_1 \cup R_2$. Analogously, there is a planar solution \mathcal{L}_2 for the sites and ports in A_2 such that all leaders of \mathcal{L}_2 lie in A_2 . Consequently, we can combine \mathcal{L}_1 and \mathcal{L}_2 into a planar solution \mathcal{L}_s for the sites and ports in I_s . Using symmetric arguments, we obtain a planar solution \mathcal{L}'_s for I'_s . As I_s and I'_s are defined over complementary areas, the solutions \mathcal{L}_s and \mathcal{L}'_s can be combined into a planar solution of I .

Assume that there is a planar solution \mathcal{L} for a three-sided instance I ; see Fig. 16. First, note that we can imagine an instance of THREE-SIDED BOUNDARY LABELING as a degenerated instance of FOUR-SIDED BOUNDARY LABELING with no bottom ports. Thus, Proposition 2 also holds for the three-sided case, when assuming that the four xy -monotone curves partitioning the solution meet on the bottom segment of R . Hence, without loss of generality, we assume that \mathcal{L} is also partitioned by four xy -monotone curves C_1 , C_2 , C_3 and C_4 . In particular, let C_1 denote the curve that starts at the top-right corner of R and let C_2 denote the curve that starts at the top-left corner of R ; see Fig. 16a. The curves C_3 and C_4 are completely contained in the bottom side of R and can therefore be omitted. We first show how to construct the grid point s and the instances I_s and I'_s such that they are balanced. Afterwards, we explain how to obtain C and C' from C_1 and C_2 , respectively. Finally, we prove that each point on C and C' satisfies the strip condition with respect to I_s and I'_s , respectively.

Let λ_T be the top leader in \mathcal{L} with the longest vertical segment of all top leaders in \mathcal{L} . In case the site of λ_T lies to the right of $\text{bend}(\lambda_T)$, let v be the rightmost vertical grid line that lies to the left of λ_T , and otherwise if the site of λ_T lies to the left of $\text{bend}(\lambda_T)$, let v be the leftmost vertical grid line that lies to the right of λ_T . Furthermore, let h be the topmost horizontal grid line that lies below $\text{bend}(\lambda_T)$; see Fig. 16a. Due to the choice of h and v all top leaders lie above h and none of them intersects h or v . Furthermore, no right or left leader of \mathcal{L} intersects v above h . The desired grid point s is then the intersection point of h and v .

Now, let λ_R be the right leader in \mathcal{L} with longest horizontal segment among all right leaders in \mathcal{L} and let v' be the rightmost vertical grid line that lies to the left of $\text{bend}(\lambda_R)$. Note that v' cannot be intersected by a left or a right leader, because both leader types are x -separated. We define R_1 to be the rectangle that is spanned by the top-right corner of R and s . Also, we define R_2 to be the rectangle spanned by the bottom-right corner of R and the intersection point of v' and h . The instance I_s is defined by $R_1 \cup R_2$ and the instance I'_s by $R \setminus (R_1 \cup R_2)$. We show that I_s and I'_s

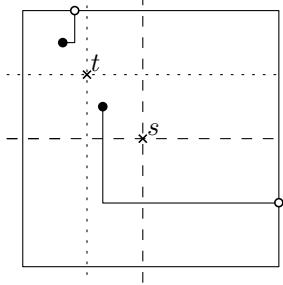


Fig. 17: There are no balanced instances I_s and I'_s for the grid point s . However, by Lemma 8 there must be another grid point t with balanced instances I_t and I'_t if the instance has a planar solution.

are balanced. To that end, we prove that a leader of \mathcal{L} is either completely contained in $R_1 \cup R_2$ or in $R \setminus (R_1 \cup R_2)$, that $R_1 \cup R_2$ contains only right and top leaders, and that $R \setminus (R_1 \cup R_2)$ contains only left and top leaders.

Due to the choice of v' , all right leaders lie to the right of v' . Moreover, all right leaders whose site or port lies above h , must lie to the right of v , because by definition of v no right leader intersects v above h (otherwise it would intersect λ_T), and because otherwise C_1 could not be an xy -monotone curve separating right and top leaders. Thus, all right leaders lie in $R_1 \cup R_2$. For left leaders we can argue similarly. Since left and right leaders of \mathcal{L} are x -separated, all left leaders lie to the left of v' . All left leaders whose site or port lies above h , must lie to the left of v , because by definition of v no left leader intersects v above h , and because otherwise C_2 could not be an xy -monotone curve separating left from top leaders. Thus, all left leaders lie in $R \setminus (R_1 \cup R_2)$. Finally, consider the top leaders in \mathcal{L} . By definition of h and v , none of the top leaders intersects h or v . In particular all top leaders lie above h and cannot intersect R_2 . Consequently, each top leader is either contained in R_1 or in $R \setminus (R_1 \cup R_2)$. This concludes the argument that I_s and I'_s are balanced.

We are left with the construction of the curves C and C' ; see Fig. 16b. The curve C is derived from C_1 as follows. Starting at the top-right corner of R , the curve C coincides with C_1 until C_1 intersects h or v' above h . If C intersects v' above h , it follows v' downwards until it hits h . Then, in both cases, it follows h until h intersects the left segment of R . Finally, C follows the left segment of R to the bottom-left corner of R . The curve C' is constructed symmetrically.

By construction, C contains the top-left corner of R_2 and the intersection point of h with the left segment of R . Symmetrically, C' contains the top-left corner of R_2 and the intersection point of h with the right segment of R . We finally show that each point on C satisfies the strip condition with respect to the sites and ports in I_s . Using symmetric arguments we can prove the analogous statement for C' and I'_s .

By the previous reasoning, we know that each leader of \mathcal{L} either lies completely inside or completely outside of $R_1 \cup R_2$. Each leader that lies in $R_1 \cup R_2$ is either a top or a right leader and does not intersect C . Otherwise, if such a leader intersected C , it would also intersect C_1 or the segment x of v' that is contained in C . In particular, x cannot be intersected by any leader because it lies to the left of all right leaders and below C_1 . Thus, the leaders in $R_1 \cup R_2$ form a planar solution for I_s without intersecting C . Hence, the claim directly follows from Lemma 3. \square

Our approach now works as follows. For each grid point s of G we compute the instances I_s and I'_s such that they are balanced. Then, by Lemma 8, we can apply our algorithm presented in Section 3 in order to solve I_s and I'_s independently. To that end, we slightly adapt the dynamic program such that it only considers curves satisfying the properties required by Lemma 8. If both

instances can be solved, we combine these solutions into one solution and return that solution as the final result. Otherwise, we continue with the next grid point of G . If all grid points of G have been explored without finding a planar solution, the algorithm decides that there is no planar solution.

Note that it may happen that, for a grid point s , there are no balanced instances I_s and I'_s ; for an example see Fig. 17. However, in that case, if I has a solution, we also know by Lemma 8 that there is another grid point t such that for t we find balanced instances. Hence, we can refrain from considering s .

Creating the two instances I_s and I'_s for a grid point s takes linear time, if we assume that the sites are sorted by their x -coordinates. By Theorem 2 we then need $O(n^2)$ time and $O(n)$ space to solve I_s and I'_s . Consequently, we need $O(n^2)$ time and $O(n)$ space to process a single grid point s . Since we consider $O(n^2)$ grid points, the following theorem follows.

Theorem 6. THREE-SIDED BOUNDARY LABELING *can be solved in $O(n^4)$ time using $O(n)$ space.*

Note that, except for the length minimization, our approach for the three-sided case also carries over to the extensions from Section 4, because we only solve subinstances of TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES. In particular with corresponding impact on the running time we can soften the restriction that the number of labels and sites is equal.

5.3 Algorithm for the Four-Sided Case

In this section, we consider the case that the ports lie on all four sides of R . The main idea is to seek a partitioned solution, which exists by Proposition 2. For a given partitioned solution \mathcal{L} , we call a leader *extremal* if all other leaders of the same type in \mathcal{L} have shorter orthogonal segments; recall that the *orthogonal* segment of a *po-leader* is the segment connecting the bend to the port. The algorithm consists of two steps. First, we explore all choices of (non-overlapping) extremal leaders λ_L and λ_R for the left and the right side of R , respectively, plus a horizontal line h that separates the top leaders and the bottom leaders. This information splits the instance into two independent three-sided instances; see Fig. 18a. There are, however, two crucial differences from a usual three-sided instance. First, one side of the instance is not a straight-line segment but an x -monotone orthogonal curve C that is defined by λ_L , λ_R and h . Second, the extremal positions of λ_L and λ_R imply a separation of the points that are labeled from the left and the right side. Let I_1^3 be the three-sided instance above C and let I_2^3 be the three-sided instance below C . The algorithm solves I_1^3 and I_2^3 independently from each other. If for at least one of the two instances there is no solution, the algorithm continues with the next choice of λ_L , λ_R and h . Otherwise, it combines the planar solutions of I_1^3 and I_2^3 into one planar solution and returns this solution. In case that all choices of λ_L , λ_R and h have been explored without finding a solution, the algorithm returns that there is no planar solution.

We next describe how to solve the three-sided instance I_1^3 . A symmetric approach can be applied to I_2^3 . The algorithm explores all choices of the extremal leader λ_T for the top side of R . This extremal leader partitions the instance into two two-sided subinstances I_1^2 and I_2^2 as follows. Let $A_{T,R}$ be the rectangle that is spanned by $\text{bend}(\lambda_T)$ and the top-right corner of R ; see Fig. 18b. Analogously, let $A_{T,L}$ be the rectangle that is spanned by $\text{bend}(\lambda_T)$ and the top-left corner of R . Analogously, for λ_R we define the area $A_{R,T}$ to be the rectangle that is spanned by $\text{bend}(\lambda_R)$ and the top-right corner of R , and $A_{R,B}$ to be the rectangle spanned by $\text{bend}(\lambda_R)$ and the bottom-right corner of R . For the leader λ_L we define $A_{L,B}$ to be the rectangle spanned by $\text{bend}(\lambda_L)$ and the bottom-left corner of R , and $A_{L,T}$ to be the rectangle spanned by $\text{bend}(\lambda_L)$ and the top-left corner of R . We assume that the port p of λ_T is only contained in that area $A \in \{A_{T,R}, A_{T,L}\}$ that also contains the site of λ_T . We make analogous assumptions for λ_L and λ_R .

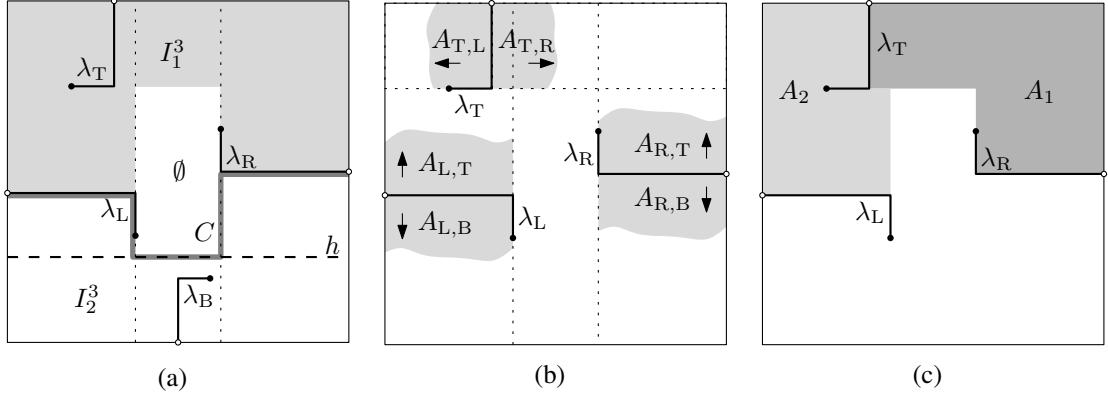


Fig. 18: a) The right leader λ_L , the left leader λ_R and the horizontal line h split the instance into two three-sided instances I_1^3 and I_2^3 . a) Sketch of the areas $A_{T,L}$, $A_{T,R}$, $A_{R,T}$, $A_{R,B}$, $A_{L,T}$ and $A_{L,B}$. c) The leaders λ_L , λ_R and λ_T split the three-sided instance into two two-sided instances.

The instance I_1^2 consists of all ports and sites in $A_1 = (A_{R,T} \cup A_{T,R}) \setminus (A_{T,L} \cup A_{R,B})$, and I_2^2 consists of all ports and sites in $A_2 = (A_{L,T} \cup A_{T,L}) \setminus (A_{T,R} \cup A_{L,B})$; see Fig. 18c. We solve I_1^2 and I_2^2 independently from each other using the dynamic program introduced in Section 3 for each instance. However, we enforce that it only considers xy -monotone curves that exclude top leaders crossing the horizontal line through $\text{bend}(\lambda_T)$, left leaders crossing the vertical line through $\text{bend}(\lambda_L)$ and right leaders crossing the vertical line through $\text{bend}(\lambda_R)$. If for at least one of the two instances there is no solution, the algorithm continues to explore the next choice of λ_T . Otherwise, it combines the solutions of I_1^2 and I_2^2 into one solution and returns the result as the solution of I_1^3 . In case that all choices of λ_T have been explored without finding a solution, the algorithm returns that there is no solution for the given three-sided instance. The following lemma shows that the algorithm is correct.

Lemma 9. *Given an instance I of FOUR-SIDED BOUNDARY LABELING, the following two statements are true.*

1. *If there is no planar solution for I , the algorithm states this.*
2. *If there is a planar solution for I , the algorithm returns such a solution.*

Proof. In case the algorithm returns a solution, it has been constructed from planar solutions of disjoint instances of TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES. As the union of these two-sided instances contains all sites and ports of I , the algorithm returns a planar solution of I , which shows the first statement.

Conversely, assume that I has a planar solution \mathcal{L} . By Proposition 2, we may assume that \mathcal{L} is partitioned. In particular, let λ_T , λ_L , λ_B and λ_R be the extremal leaders in \mathcal{L} of the top, left, bottom and right side of R , respectively, and let h be a horizontal line that separates the top leaders from the bottom leaders.

Obviously, λ_L , λ_R and h split the instance into two three-sided instances I_1^3 and I_2^3 . As the algorithm systematically explores all choices of extremal right leaders, extremal left leaders and horizontal lines partitioning the set of sites, it must find λ_L , λ_R and a horizontal line h' that separates the same sets of sites as h . Thus, I_1^3 and I_2^3 are considered by the algorithm.

Let I_1^3 be the instance above the curve defined by λ_L , λ_R and h' , and let I_2^3 be the instance below that curve. We now show that the algorithm finds a planar solution for I_1^3 . Symmetric arguments hold for I_2^3 . As the algorithm explores all choices of extremal top leaders in I_1^3 , it also considers λ_T to be the extremal top leader. This leader partitions the area of I_1^3 into the two disjoint areas $A_1 = (A_{R,T} \cup A_{T,R}) \setminus (A_{T,L} \cup A_{R,B})$ and $A_2 = (A_{L,T} \cup A_{T,L}) \setminus (A_{T,R} \cup A_{L,B})$;

see Fig. 18a. It directly follows from the extremal choice of λ_R , λ_T and λ_L that there is no leader in \mathcal{L} that intersects both A_1 to A_2 . In particular, no left leader intersects A_1 and no right leader intersects A_2 . Thus, A_1 and A_2 split \mathcal{L} into independent planar solutions \mathcal{L}_1 and \mathcal{L}_2 of two two-sided instances I_1^2 and I_2^2 induced by A_1 and A_2 , respectively. Note that the algorithm considers the same two-sided instances independently from each other. As I_1^2 has a solution, namely \mathcal{L}_1 , we know that the dynamic program finds a solution \mathcal{L}_1^2 for I_1^2 . In particular, all leaders of \mathcal{L}_1^2 lie in A_1 .

Applying symmetric arguments for I_2^2 , the algorithm yields a planar solution \mathcal{L}_2^2 that stays in A_2 . Consequently, combining \mathcal{L}_1^2 and \mathcal{L}_2^2 into one solution yields a planar solution \mathcal{L}_1^3 for I_1^3 . Analogously, we obtain a planar solution \mathcal{L}_2^3 for I_2^3 . Obviously, due to the separation by λ_L , λ_R and h' , the union of \mathcal{L}_1^3 and \mathcal{L}_2^3 is also planar, which is the overall solution returned by the algorithm. This proves the second statement of the lemma. \square

Let us analyze the running time of the algorithm. Obviously, there are $O(n^5)$ possible combinations of left and right extremal leaders and a horizontal line separating the top and bottom-labeled sites. For each combination, we independently solve two three-sided instances. For such a three-sided instance, we consider $O(n^2)$ choices for the extremal leader λ_T and independently solve two independent two-sided instances with Theorem 2 in $O(n^2)$ time. This implies that solving one three-sided instances takes $O(n^4)$ time. Thus, the overall running time is $O(n^9)$. The following theorem summarizes this result.

Theorem 7. FOUR-SIDED BOUNDARY LABELING can be solved in $O(n^9)$ time using $O(n)$ space.

Note that, except for the length minimization, our approach for the four-sided case also carries over to the extensions from Section 4, because we only solve subinstances of TWO-SIDED BOUNDARY LABELING WITH ADJACENT SIDES. In particular with corresponding impact on the running time we can soften the restriction that the number of labels and sites is equal.

6 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.

Our algorithm can also be used to solve a variety of different extensions of the problem. We have shown how to generalize to sliding ports instead of fixed ports without increasing the running time and how to maximize the number of labeled sites such that the solution is planar in $O(n^3 \log n)$ time. We further have given an extension to the algorithm that minimizes the total leader length in $O(n^8 \log n)$ time.

With some additional work, our approach can also be used to solve THREE-SIDED and FOUR-SIDED BOUNDARY LABELING in polynomial time. We have introduced an algorithm solving the three-sided case in $O(n^4)$ time and the four-sided case in $O(n^9)$ time. Also, except for the length minimization, all extensions carry over. It remains open whether a minimum length solution of THREE-SIDED and FOUR-SIDED BOUNDARY LABELING can be computed in polynomial time.

References

- [1] P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29(3):912–953, 1999.

- [2] O. Bastert and S. P. Fekete. Geometrische Verdrahtungsprobleme. Technical Report 96–247, Universität zu Köln, 1996. 3
- [3] M. A. Bekos, S. Cornelsen, M. Fink, S. Hong, M. Kaufmann, M. Nöllenburg, I. Rutter, and A. Symvonis. Many-to-one boundary labeling with backbones. In S. Wismath and A. Wolff, editors, *Proc. 21st Int. Symp. Graph Drawing (GD’13)*, volume 8242 of *Lect. Notes Comput. Sci.*, pages 244–255. Springer-Verlag, 2013. 3
- [4] M. A. Bekos, M. Kaufmann, M. Nöllenburg, and A. Symvonis. Boundary labeling with octilinear leaders. *Algorithmica*, 57(3):436–461, 2010. 3
- [5] M. A. Bekos, M. Kaufmann, K. Potika, and A. Symvonis. Area-feature boundary labeling. *Comput. J.*, 53(6):827–841, 2010. 3, 14
- [6] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Comput. Geom. Theory Appl.*, 36(3):215–236, 2007. 2, 3
- [7] M. Benkert, H. J. Haverkort, M. Kroll, and M. Nöllenburg. Algorithms for multi-criteria boundary labeling. *J. Graph Algorithms Appl.*, 13(3):289–317, 2009. 3, 5
- [8] T. M. Chan, H.-F. Hoffmann, S. Kiazyk, and A. Lubiw. Minimum length embedding of planar graphs at fixed vertex locations. In S. K. Wismath and A. Wolff, editors, *Proc. 21st Int. Symp. Graph Drawing (GD’13)*, volume 8242 of *Lect. Notes Comput. Sci.*, pages 376–387. Springer-Verlag, 2013. 3
- [9] B. Chazelle and 36 co-authors. The computational geometry impact task force report. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223, pages 407–463. American Mathematical Society, Providence, RI, 1999. 1
- [10] M. Fink, J.-H. Haunert, A. Schulz, J. Spoerhase, and A. Wolff. Algorithms for labeling focus regions. *IEEE Trans. Visual. Comput. Graphics*, 18(12):2583–2592, 2012. 3
- [11] H. Freeman, S. Marrinan, and H. Chitalia. Automated labeling of soil survey maps. In *Proc. ASPRS-ACSM Annual Convention, Baltimore*, volume 1, pages 51–59, 1996. 2
- [12] A. Gemsa, J.-H. Haunert, and M. Nöllenburg. Boundary-labeling algorithms for panorama images. In *Proc. 19th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inform. Syst. (ACM-GIS’11)*, pages 289–298, 2011. 3
- [13] P. Gritzmann, B. Mohar, J. Pach, and R. Pollack. Embedding a planar triangulation with vertices at specified positions. *Amer. Math. Mon.*, 98:165–166, 1991. 3
- [14] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Comm. ACM*, 18(6):341–343, 1975. 12, 13
- [15] B. Katz, M. Krug, I. Rutter, and A. Wolff. Manhattan-geodesic embedding of planar graphs. In D. Eppstein and E. R. Gansner, editors, *Proc. 17th Int. Symp. Graph Drawing (GD’09)*, volume 5849 of *Lect. Notes Comput. Sci.*, pages 207–218. Springer-Verlag, 2010. 3
- [16] T. M. Liebling, F. Margot, D. Müller, A. Prodon, and L. Stauffer. Disjoint paths in the plane. *ORSA J. Comput.*, 7(1):84–88, 1995. 3

- [17] C.-C. Lin. Crossing-free many-to-one boundary labeling with hyperleaders. In *Proc. IEEE Pacific Visualization Symp. (PacificVis'10)*, pages 185–192, 2010. 3
- [18] C.-C. Lin, H.-J. Kao, and H.-C. Yen. Many-to-one boundary labeling. *J. Graph Algorithms Appl.*, 12(3):319–356, 2008. 3
- [19] J. L. Morrison. Computer technology and cartographic change. In D. Taylor, editor, *The Computer in Contemporary Cartography*. Johns Hopkins University Press, 1980. 1
- [20] M. Nöllenburg, V. Polishchuk, and M. Sysikaski. Dynamic one-sided boundary labeling. In *Proc. 18th ACM SIGSPATIAL Int. Symp. Adv. Geogr. Inform. Syst. (ACM-GIS'10)*, pages 310–319, 2010. 3
- [21] R. Raghavan, J. Cohoon, and S. Sahni. Single bend wiring. *J. Algorithms*, 7(2):232–257, 1986. 3
- [22] M. van Kreveld, T. Strijk, and A. Wolff. Point labeling with sliding labels. *Comput. Geom. Theory Appl.*, 13:21–47, 1999. 1
- [23] S. Zoraster. Practical results using simulated annealing for point feature label placement. *Cartogr. GIS*, 24(4):228–238, 1997. 2