JOURNAL OF SPATIAL INFORMATION SCIENCE
Number 13 (2016), pp. 3–32 doi:10.5311/JOSIS.2016.13.264

RESEARCH ARTICLE

Mixed map labeling*

Maarten Löffler[1], Martin Nöllenburg[2], and Frank Staals[3]

[1]Department of Information and Computing Sciences, Utrecht University, the Netherlands
[2]Algorithms and Complexity Group, TU Wien, Vienna, Austria
[3]MADALGO, Aarhus University, Denmark

Received: October 30, 2015; returned: May 18, 2016; revised: August 23, 2016; accepted: September 14, 2016.

Abstract: Point feature map labeling is a geometric visualization problem, in which a set of input points must be labeled with a set of disjoint rectangles (the bounding boxes of the label texts). It is predominantly motivated by label placement in maps but it also has other visualization applications. Typically, labeling models either use internal labels, which must touch their feature point, or external (boundary) labels, which are placed outside the input image and which are connected to their feature points by crossing-free leader lines. In this paper we study polynomial-time algorithms for maximizing the number of internal labels in a mixed labeling model that combines internal and external labels. The model requires that all leaders are parallel to a given orientation $\theta \in [0, 2\pi)$, the value of which influences the geometric properties and hence the running times of our algorithms.

Keywords: algorithms, map labeling, computational geometry, geovisualization

1 Introduction

Annotating features of interest in maps and other images of geographic information with textual labels or graphical icons is an important task in geovisualization. The traditional design principles and quality criteria used in cartographic label placement easily generalize to label placement in arbitrary information graphics and illustrations. In a map, labels are mostly placed internally, that is, touching their respective feature points. Common cartographic placement guidelines demand that each label is placed in the immediate neighborhood of its feature and that the association between labels and features is unambiguous, while no two labels may overlap each other [16, 27]. Different styles of label placement

*A preliminary version of this paper appeared in Proc. 9th International Conference on Algorithms and Complexity (CIAC 2015) [22].

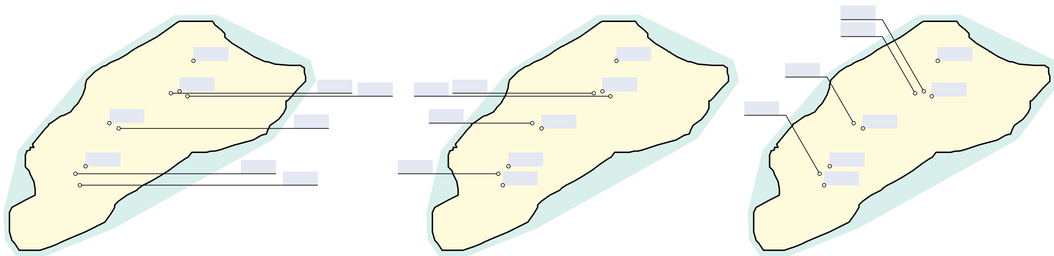
© by the author(s) Licensed under Creative Commons Attribution 3.0 License

and their merits have been studied extensively in cartography and geographic information science. However, even when an ideal set of placement rules is agreed upon, it is often not trivial to compute an optimal placement. In the computational geometry literature, there has been extensive work investigating the tractability of different labeling styles. It is known that maximizing the number of non-overlapping labels for a given set of input points is NP-hard, even for very restricted labeling models [10, 23]. In terms of labeling algorithms, several approximations, polynomial-time approximation schemes (PTAS), and exact approaches are known [1, 8, 20, 28], as well as many practically effective heuristics, see the bibliography of Wolff and Strijk [29]. If, however, feature points lie too dense in the map or if their labels are relatively large, often only small fractions of the features obtain a label, even in an optimal solution.

An alternative labeling approach uses external instead of internal labels, which are sometimes known as *callouts*. External labels are remotely placed, often outside the image itself, and connected by leaders to their respective feature points. This is known as *boundary labeling* [6] in the algorithmic literature. The boundary labeling style is most frequently used when annotating anatomical drawings [13, 26] and technical illustrations, where different and often small parts are identified using labels and longer descriptive texts outside the actual picture. Other popular applications of boundary labeling are annotation placement in correction mode of word processing software and commenting in document viewers [18]. But also maps and cartographic visualizations can carry external labels, e.g., the Maplex Label Engine in ArcGIS 10.3 offers labels with callouts and leaders.

While the association between points and external labels may be more difficult to see, the big advantages of boundary labeling are that even dense feature sets can be labeled and that larger labels can be accommodated on the margins of the illustration. Many efficient boundary labeling algorithms are known. They can be classified by the leader shapes that are used and by the sides of the picture's bounding box that are used for placing the labels [3, 6, 7, 11, 15, 19, 25]. A recent user study evaluates the readability and aesthetic preference for differently shaped leader types and concludes that, especially for dense feature sets, straight leaders or leaders with at most one bend (horizontal/vertical or horizontal/diagonal) perform best [2]. Orthogonal leaders with two bends were clearly outperformed in most cases.

The combination of internal and external labeling models using internal labels where possible and external labels where necessary seems natural and has been proposed as an open problem by Kaufmann [17]; however, only few results are known in such *hybrid* or *mixed* settings. In a mixed labeling, the final image will be an overlay of the original map or drawing, a collection of internal text labels, and a collection of leaders leading out of the image towards their external labels. Depending on the application, we may wish to forbid intersections between some or all of these layers. When no additional intersection constraints are imposed, the problem reduces to classical internal map labeling. Under the very natural restriction that leaders cannot intersect internal labels, the internal labels need to be placed carefully, as not every set of disjoint internal labels creates sufficient gaps for routing leaders of the prescribed shape from all remaining feature points to the image boundary. Löffler and Nöllenburg [21] studied a restricted case of hybrid labeling, where a partition of the feature points into points with internal fixed position labels and points with external labels to be connected by one-bend orthogonal leaders is given as input. They presented efficient algorithms and hardness results, depending on three different problem parameters. Bekos et al. [4] studied a mixed labeling model with fixed-position internal



(a) $\theta = \pi$ (b) $\theta = 0$ (c) $\theta = \pi/3$

Figure 1: A sample point set with mixed labelings of three different slopes. In (a) five external labels are necessary, whereas (b) and (c) require only four external labels. The slope in (c) yields aesthetically pleasing results.

labels and external labels on one or two opposite sides of the bounding box, connected by two-bend orthogonal leaders. Their goal is to maximize the number of internally labeled points, while labeling all remaining points externally. Polynomial and quasi-polynomial-time algorithms, as well as an approximation algorithm and an ILP formulation were presented.

Contribution In this paper, we extend the known theoretical results on mixed map labeling as follows. We present a mixed labeling model, in which each point is assigned either an axis-aligned fixed-position internal label (e.g., to the top right of the point) or an external label connected with a leader of slope θ, where $\theta \in [0, 2\pi)$ is an input parameter defining the unique leader direction for all external labels, measured clockwise from the negative x-axis (see Figure 1). In this model, we present a new dynamic-programming algorithm to maximize the number of internally labeled points for any given slope θ, including the left- and right-sided case ($\theta = 0$ or $\theta = \pi$), which was studied by Bekos et al. [4]. While for the right-sided case Bekos et al. provided a faster $O(n \log^2 n)$-time algorithm, where n is the number of input points, our algorithm improves upon their pseudo-polynomial $O(n^{\log n+3})$-time algorithm for the left-sided case. We solve this problem in $O(n^3(\log n + \delta))$ time, where $\delta = \min\{n, 1/d_{\min}\}$ is the inverse of the distance $d_{\min}$ of the closest pair of points in $\mathcal{P}$ and expresses the maximum density of $\mathcal{P}$ (Section 2). In the general case it turns out that the set of slopes can be partitioned into twelve intervals, in each of which the geometric properties of the possible leader-label intersections are similar for all slopes. Depending on the particular slope interval, the amount $\iota(n, \delta, \theta)$ of "interference" between sub-problems varies. This significantly affects the algorithm's performance and leads to running times between $O(n^3 \log n)$ and $O(n^3(\log n + \iota(n, \delta, \theta))) \subseteq O(n^7)$ (Section 3). From a theoretical point of view, this shows that mixed map labeling can be solved optimally in polynomial time for any leader slope. From a practical point of view, near-cubic running times up to $O(n^7)$ (depending on the leader slope) seem too slow at first sight. However, in many applications, e.g., maps on small, mobile devices or thematic maps for newspapers and other public media, the number of labels to show is moderate, say 10–50. In such a setting, our algorithms may well prove themselves to be sufficiently fast in practice depending on the input parameters θ and δ. Moreover, we can use our algorithm to optimize the number of internal labels over all slopes θ at an increase in running time by a factor of $O(n^2)$, as is shown in Section 4.2.

Problem statement We are given a map (or any other illustration) $\mathcal{M}$, which we model for simplicity as a convex polygon (this is easy to relax to larger classes of well-behaved domains), and a set $\mathcal{P}$ of n points in $\mathcal{M}$ that must be labeled by rectangular labels (the bounding boxes of the label texts). In addition, we are given a leader slope $\theta \in [0, 2\pi)$. For simplicity we assume that θ is none of the slopes defined by two points in $\mathcal{P}$. We discuss in Section 4.6 how to remove this restriction. There are two choices for assigning a label to a point $p \in \mathcal{P}$: either we assign an *internal label* λ_p on $\mathcal{M}$ in a one-position model, or an *external label* outside of $\mathcal{M}$ that is connected to p with a leader γ_p. An internal label λ_p is a rectangle that is anchored at p by its lower left corner. A leader γ_p is a line segment of slope θ inside $\mathcal{M}$; it may bend to the horizontal direction outside of $\mathcal{M}$ in order to connect to its horizontally aligned label, see Figure 1c. So in this model, the labeling is fixed once the choice for an internal or external label has been made for each point $p \in \mathcal{P}$. For a *valid* label assignment we require that (i) the labels do not overlap each other or the leaders, and that (ii) the leaders themselves do not intersect each other. As we will see in Section 4.4 it is easy to realize constraint (i) for the external labels. Figure 1 shows valid mixed labelings for three different slopes.

Given a set of points $P \subseteq \mathcal{P}$, let $\Lambda(P) = \{\lambda_p \mid p \in P\}$ denote the set of (candidate) labels corresponding to the points in P and let $\Gamma(P) = \{\gamma_p \mid p \in P\}$ denote the set of (candidate) leaders corresponding to the points in P. A *labeling* of $\mathcal{P}$ is a partition of $\mathcal{P}$ into sets $\mathcal{I}$ and $\mathcal{E}$, the points in $\mathcal{I}$ labeled internally, the points in $\mathcal{E}$ labeled externally, such that no two labels in $\Lambda(\mathcal{I})$ intersect, no two leaders in $\Gamma(\mathcal{E})$ intersect, and no label from $\Lambda(\mathcal{I})$ intersects a leader from $\Gamma(\mathcal{E})$.

For ease of presentation we first assume that all labels have the same size, which, without loss of generality, we assume to be 1×1, but the problem can be solved with the same algorithms for rectangular labels of any other fixed size and shape. Hence, an internal label λ_p is a unit square with its bottom left corner on p. This may be a realistic model in some settings (e.g., unit-size icons as labels [14]), but generally not all labels have the same size. We will sketch how to relax this restriction in Section 4.5.

Each leader γ_p can be split into an *inner* part (or *inner leader*), which is a line segment of slope θ from p to the intersection point with the boundary of $\mathcal{M}$, and an *outer* part (or *outer leader*) from the boundary of $\mathcal{M}$ to the actual label. In this paper we focus our attention on the inner leaders as they determine how $\mathcal{P}$ is separated into different subinstances. Hence we can basically think of the leaders as half-lines with slope θ. Again, this assumption is not realistic for mixed map labeling in practice, where the placement of labels and the routing of leaders outside of $\mathcal{M}$ is an important sub-problem. We explain a simple method of routing the outer leaders in Section 4.4 and discuss possible consequences and approaches if the space for external labels is bounded. Studying this external labeling problem in its entirety is beyond the scope of this paper and gives rise to computational problems of independent interest.

It is well known that in general not all points in $\mathcal{P}$ can be assigned an internal label. The corresponding label number maximization problem is NP-hard [10, 23], even if each label has just one candidate position [21]. If, however, all labels have the same position (e.g., to the top left of the anchor points) and no input point may be covered by any other label, the one-position case can be solved efficiently by first discarding all labels containing an anchor point and then applying a simple greedy algorithm on the resulting staircase patterns [21]. On the other hand, it is also known that any instance can be labeled with external labels using efficient algorithms [5, 6]. Mixed labelings combine both label types and sit between the two extremes of purely internal and purely external labeling [4, 21]. Here we are interested in the *internal label number maximization problem*, which was first studied for $\theta \in \{0, \pi\}$ by Bekos et al. [4]: Given a map $\mathcal{M}$, a set of points $\mathcal{P}$ in $\mathcal{M}$ and a slope $\theta \in [0, 2\pi)$, we wish to find a valid mixed labeling that maximizes the number $|\mathcal{I}|$ of internally labeled points.

2 Leaders from the left

We start with the case that $\theta = 0$, i.e., all leaders are horizontal half-lines leading from the points towards the left of $\mathcal{M}$. Our approach for maximizing the number of internal labels is to process the points in $\mathcal{P}$ from right to left and to recursively determine the optimal rightmost unprocessed point p to be assigned an external label. Since no leader may cross any internal label, the leader γ_p decomposes the current instance left of p into two (almost) independent parts, one above γ_p and one below. As it turns out, a generic subinstance can be defined by an upper and a lower leader shielding it from the outside and additional information about at most one point outside the subinstance. The problem is then solved using dynamic programming.

2.1 Geometric properties

In this section we prove that the problem has a certain geometric structure that we can use to obtain an efficient algorithm. In particular, we show that if we consider a horizontal slab defined by two points ℓ and u as shown in Figure 2, the points in the purple region are shielded from the labels and leaders of the points outside of the horizontal slab by the leaders of the points ℓ and u. In such a configuration, there can be at most one point r outside the slab, i.e., outside the purple and yellow region, whose label or leader may interfere with the labeling of the points in the purple region. Such a point r must lie in the unit square whose top-left corner is ℓ (the blue region in Figure 2).

Let $p = (p_x, p_y)$ be a point in the plane and let $L_p = \{q \mid q_x < p_x\}$ and $R_p = \{q \mid q_x > p_x\}$ denote the half-planes containing all points strictly to the left and to the right of p, respectively. Analogously, we define the half-planes T_p and B_p above and below p, respectively. Let $\overline{S(\ell, u)} = T_\ell \cap B_u$ denote the horizontal slab defined by points ℓ and u (with $\ell_y < u_y$), and let $S(\ell, u) = \overline{S(\ell, u)} \cap L_\ell \cap L_u$ denote the set of points in this slab that lie to the left of both ℓ and u, see Figure 2(a). We define $P_{\ell,u}$ as the subset of $\mathcal{P}$ in $S(\ell, u)$ including ℓ and u, i.e., $P_{\ell,u} = \mathcal{P} \cap (S(\ell, u) \cup \{\ell, u\})$. With some abuse of notation we will sometimes also use L_p, R_p, T_p, and B_p to mean the subset of $\mathcal{P}$ that lies in the respective half-plane rather than the entire half-plane.

Recall that $\delta = \min\{n, 1/d_{\min}\}$ is a parameter that captures the maximum density of $\mathcal{P}$ as the inverse of the smallest distance $d_{\min}$ between any two points in $\mathcal{P}$. We can use δ to bound the number of candidate points in a unit square that may be labeled internally.[1]

[1] Since all internal labels are unit squares, any unit square can have only one point that is labeled internally. We use "the number of points that can, or may, be labeled internally" to refer to the number of candidates for such a point.

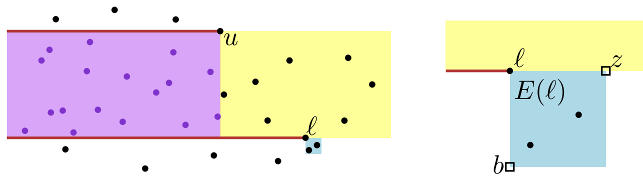


Figure 2: (a) The slab $\overline{S(\ell, u)}$ in yellow, the region $S(\ell, u)$ and its points from $\mathcal{P}$ in purple, and the region $E(\ell)$ in blue. (b) An enlarged view of the region $E(\ell)$.

Lemma 1. *At most $O(\delta)$ points in any unit square have a label λ that does not contain another point in $\mathcal{P}$.*

Proof. Let E be a unit square and let $E_{\mathcal{P}} = E \cap \mathcal{P}$ be the input points in E. Since the label λ_p of every point $p \in E_{\mathcal{P}}$ is a unit square anchored by its lower left corner at p, no other point $q \in \mathcal{P}$ may lie to the top-right of p—otherwise p must be labeled externally. Hence any set of points in $E_{\mathcal{P}}$ whose labels do not contain another point of $\mathcal{P}$ must form a sequence $p^{(1)}, p^{(2)}, \ldots, p^{(k)}$ such that $p_x^{(i)} < p_x^{(j)}$ and $p_y^{(i)} > p_y^{(j)}$ for any $i < j$. Since the minimum distance of any two points is $d_{\min}$ we immediately obtain that $E_{\mathcal{P}}$ contains at most $O(\delta)$ points whose label does not contain another point in $\mathcal{P}$. □

Next, we characterize which leaders or labels outside of $\overline{S(\ell, u)}$ can interfere with a potential labeling of $P_{\ell,u}$ assuming that ℓ and u are labeled externally.

Lemma 2. *Let $\ell, u \in \mathcal{P}$, let $(\mathcal{I}', \mathcal{E}')$, with $\ell, u \in \mathcal{E}'$ be a labeling of $P_{\ell,u}$. There is no point in $T_u \cup B_\ell$ whose leader intersects a label from $\Lambda(\mathcal{I}')$ and there is no point in T_u whose label intersects a label from $\Lambda(\mathcal{I}')$.*

Proof. In any labeling of $P_{\ell,u}$ there are no intersections between labels and leaders. In particular, no label λ_p for $p \in \mathcal{I}'$ intersects γ_ℓ or γ_u. It follows that all labels for $\mathcal{I}'$ lie inside the slab $\overline{S(\ell, u)}$. By definition, all leaders for $\mathcal{E}'$ also lie inside $\overline{S(\ell, u)}$. Since leaders of points in $T_u \cup B_\ell$ do not intersect $\overline{S(\ell, u)}$, no such leader can intersect a label from $\mathcal{I}'$. Moreover, all labels are anchored by their bottom left corner, and hence all labels of points in T_u lie above u and do not intersect $\overline{S(\ell, u)}$. Thus, no label for a point in T_u can intersect a label in $\Lambda(\mathcal{I}')$. □

It is not true, however, that labels for points in B_ℓ cannot intersect labels for $P_{\ell,u}$. Still, the influence of B_ℓ is very limited as the next lemma shows. Let $E(\ell)$ denote the open unit square with top-left corner ℓ, i.e., $E(\ell) = R_\ell \cap B_\ell \cap L_z \cap T_b$, where $z = (\ell_x + 1, \ell_y)$ and $b = (\ell_x, \ell_y - 1)$. See Figure 2(b).

Lemma 3. *Let $\ell, u \in \mathcal{P}$, let $(\mathcal{I}', \mathcal{E}')$, with $\ell, u \in \mathcal{E}'$ be a labeling of $P_{\ell,u}$, and let $(\mathcal{I}'', \mathcal{E}'')$ denote a labeling of $\mathcal{P} \cap B_\ell \cup \{\ell\}$ with $\ell \in \mathcal{E}''$. There is at most one point $p \in \mathcal{I}''$ whose label may intersect a label of $\mathcal{I}'$, and $p \in E(\ell)$.*

Proof. Since $\ell \in \mathcal{E}' \cap \mathcal{E}''$ we know that no label in $\Lambda(\mathcal{I}')$ or $\Lambda(\mathcal{I}'')$ intersects γ_ℓ. Thus γ_ℓ serves as a separation line between the labels $\Lambda(\mathcal{I}')$ and $\Lambda(\mathcal{I}'')$. Let $\mathcal{P}'' \subseteq \mathcal{I}''$ denote the set of points whose labels intersect a label of $\mathcal{I}'$, and let $\Lambda(\mathcal{P}'')$ denote the corresponding set of labels. We first argue that $\mathcal{P}'' \subseteq E(\ell)$. Then we argue that $\mathcal{P}''$ can contain at most one point.

The labels in $\Lambda(\mathcal{P}'')$ do not intersect γ_ℓ, hence they lie strictly right of ℓ. Thus, $\mathcal{P}'' \subseteq R_\ell$. All points in $\mathcal{I}'$ lie to the left of ℓ, and their labels have width one. All labels in $\Lambda(\mathcal{P}'')$ intersect such a label, thus all points in $\mathcal{P}''$ must lie in L_z, where $z = (\ell_x + 1, \ell_y)$. All labels in $\Lambda(\mathcal{P}'')$ intersect a label from a point in $\mathcal{I}'$. Thus, all labels in $\Lambda(\mathcal{P}'')$ intersect the horizontal line containing γ_ℓ. Since all labels have height one, it follows that all points in $\mathcal{P}''$ lie in T_b, where $b = (\ell_x, \ell_y - 1)$. By definition the points in $\mathcal{P}''$ lie in B_ℓ. So we have $\mathcal{P}'' \subseteq R_\ell \cap L_z \cap T_b \cap B_\ell = E(\ell)$.

The labels in $\Lambda(\mathcal{P}'')$ are pairwise disjoint and all intersect the top side s of $E(\ell)$. Since the length of s is smaller than one, each label in $\Lambda(\mathcal{P}'')$ has width exactly one, and all labels lie in R_ℓ it follows that there can be at most one label in $\Lambda(\mathcal{P}'')$. Thus, there is also at most one point in $\mathcal{P}''$. □

From Lemma 2 and Lemma 3 it follows that if ℓ and u are labeled externally, there is at most one point r below ℓ that can influence the labeling of the points in $S(\ell, u)$.

2.2 Computing an optimal labeling

Using the result from the previous section, we now show that the rightmost point that is labeled externally decomposes the problem into two almost independent smaller sub-problems. We can use this to compute an optimal labeling efficiently. Consider again a horizontal slab defined by points ℓ and u, as shown in Figure 3, and assume that point r in the blue region is labeled internally. The key idea is then that if we guess the rightmost point p from the slab labeled internally, then we can compute an optimal solution for the points in the slab by combining optimal solutions for two smaller sub-problems. Namely, the sub-problem defined by ℓ, p, and r (the blue points from Figure 3(b) in the slab defined by ℓ and p), and the sub-problem defined by p and u and an appropriate point r' (the orange points from Figure 3(b) in the slab defined by p and u). This means that we can express the number of points that can be labeled internally as a recursive function Φ, parameterized by three points: ℓ, u, and r.

We compute the values $\Phi(\ell, u, r)$ by dynamic programming. When we combine the solutions $\Phi(\ell, p, r)$ and $\Phi(p, u, r')$ of smaller sub-problems into a solution for the larger sub-problem $\Phi(\ell, u, r)$ (i.e., a solution for our slab as defined above) we have to test if all labels of points right of p are free of intersections. We show that we can precompute this information using a sweepline algorithm.

We define $\Phi(\ell, u, r)$, with $\ell, u \in \mathcal{P}$, and $r \in E(\ell) \cup \{\perp\}$ as the maximum number of points in $S(\ell, u)$ that can be labeled internally, given that

(i) the points ℓ and u are labeled externally,
(ii) all remaining points in $\overline{S(\ell, u)} \setminus S(\ell, u)$ have been labeled internally, and
(iii) point r is labeled internally. If $r = \perp$ then no point in $E(\ell)$ is labeled internally.

See Figure 3(a) for an illustration. Given points ℓ, r, and $p \in T_\ell$ we define $\varrho(p, \ell, r)$ to be the topmost point in $E(p) \cap (T_\ell \cup \{r\})$ if such a point exists. Otherwise we define $\varrho(p, \ell, r) = \perp$. See Figure 3(c).

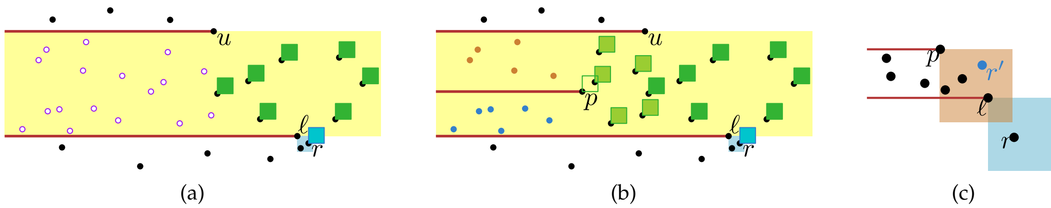


Figure 3: (a) $\Phi(\ell, u, r)$ expresses the maximum number of points in $S(\ell, u)$ (white marks with purple outline), that can be labeled internally in the depicted situation. (b) The rightmost point p that is labeled with an external label decomposes the problem into two subproblems (the orange and blue points). (c) Point $r' = \varrho(p, \ell, r)$ (blue) is the topmost point from $T_\ell \cup \{r\}$ that lies in the region $E(p)$ (the orange unit square).

Lemma 4. *For any $\ell, u \in \mathcal{P}$, and $r \in E(\ell) \cup \{\bot\}$, we have that $\Phi(\ell, u, r) = |S(\ell, u)|$, or $\Phi(\ell, u, r) = |R_p \cap S(\ell, u)| + \Phi(\ell, p, r) + \Phi(p, u, r')$, where p is the rightmost point in $S(\ell, u)$ with an external label and $r' = \varrho(p, \ell, r)$.*

Proof. Let $(\mathcal{I}^*, \mathcal{E}^*)$ be an optimal labeling of $S(\ell, u)$ that satisfies the constraints *(i)–(iii)* on $\Phi(\ell, u, r)$, i.e., $\Phi(\ell, u, r) = |\mathcal{I}^*|$. In case $\mathcal{E}^* = \emptyset$, we have $\Phi(\ell, u, r) = |S(\ell, u)|$ and the lemma trivially holds. Otherwise, there must be a rightmost point $p \in \mathcal{E}^*$ with an external label. Consider the partition of $\mathcal{I}^*$ at point p into the lower left part $B^* = B_p \cap L_p \cap \mathcal{I}^*$, the upper left part $T^* = T_p \cap L_p \cap \mathcal{I}^*$, and the right part $R^* = R_p \cap \mathcal{I}^*$, see Figure 3b. We show that $|R^*| = |R_p \cap S(\ell, u)|$, $|B^*| = \Phi(\ell, p, r)$, and $|T^*| = \Phi(p, u, r')$, which proves the lemma.

Since p is the rightmost point with an external label it follows that all points in $S(\ell, u)$ right of p are labeled internally. Hence, $R^* = R_p \cap S(\ell, u)$.

Next, we observe that $\mathcal{L}_B = (B^*, S(\ell, p) \setminus B^*)$ as a sub-labeling of $(\mathcal{I}^*, \mathcal{E}^*)$ forms a valid labeling of $S(\ell, p)$, so by Lemma 3 there is at most one point $\hat{r}$ below ℓ that can influence the labeling of $S(\ell, p)$. This point $\hat{r}$, if it exists, lies in $E(\ell)$. By constraint *(iii)* point r lies in $E(\ell)$ or $r = \bot$ and no point in $E(\ell)$ is labeled internally, and thus r can be the only point in $E(\ell)$ labeled internally, i.e., $\hat{r} = r$. So, we have that *(i)* ℓ and p are labeled externally, *(ii)* all points in $\overline{S(\ell, p)} \setminus S(\ell, p)$ are labeled internally, and *(iii)* point r is the only internally labeled point in $E(\ell)$. Thus the definition of Φ applies and we obtain $|B^*| \leq \Phi(\ell, p, r)$.

Lemmas 2 and 3 together imply that any labeling of $S(\ell, p)$ is independent from any labeling of $S(p, u)$. Thus, it follows that $\mathcal{L}_B$ is an optimal labeling of $S(\ell, p)$ (given the constraints), since otherwise $(\mathcal{I}^*, \mathcal{E}^*)$ could also be improved. Thus $|B^*| \geq \Phi(\ell, p, r)$ and we obtain $|B^*| = \Phi(\ell, p, r)$.

Finally, we consider the upper left part T^*. By Lemma 3 there is at most one point r' in B_p with an internal label that can influence the labeling of $S(p, u)$ and we have $r' \in E(p)$. We need to show that $r' = \varrho(p, \ell, r)$. Then the rest of the argument is analogous to the argument for B^*.

We claim that r' is the topmost point in $E(p) \cap (T_\ell \cup \{r\})$. Assume that $r' \notin T_\ell$, which means $r' \in B_\ell$. We know that γ_ℓ does not intersect $\lambda_{r'}$ and hence $r' \in R_\ell$. This means that $r' \in E(p) \cap B_\ell \cap R_\ell =: X$ and since p lies to the top-left of ℓ we have $X \subseteq E(\ell)$. By definition r is the only point with an internal label in $E(\ell)$ and hence $r' = r$. So if $r' \neq r$ we have $r' \in E(p) \cap T_\ell$. Now assume that $r' \neq \varrho(p, \ell, r)$. Then there is another point $q \in E(p) \cap T_\ell$ above r'. This point q must be labeled externally since no two points in

$E(p)$ can be labeled internally. This is a contradiction since by definition p is the rightmost externally labeled point in $S(\ell, u)$ and by constraint *(ii)* all points in $\overline{S(\ell, u)} \setminus S(\ell, u)$ are labeled internally. So indeed $r' = \varrho(p, \ell, r)$ and the same arguments as for B^* can be used to obtain $|T^*| = \Phi(p, u, r')$. $\square$

Let ℓ, $u \in \mathcal{P}$, and $p \in S(\ell, u)$. We observe that $|S(\ell, p)|$ and $|S(p, u)|$ are strictly smaller than $|S(\ell, u)|$. Thus, Lemma 4 gives us a proper recursive definition for Φ:

$$\Phi(\ell, u, r) = \max \big\{ \Psi(S(\ell, u)), \\ \max_{p \in S(\ell, u)} \{ \Psi(R_p \cap S(\ell, u)) + \Phi(\ell, p, r) + \Phi(p, u, \varrho(p, \ell, r)) \} \big\},$$

where

$$\Psi(P) = \begin{cases} |P| & \text{if all labels in } \Lambda(P \cup \{r\} \cup (\overline{S(\ell, u)} \setminus S(\ell, u))) \text{ are pairwise disjoint,} \\ & \text{and their intersection with } \gamma_\ell \text{ and } \gamma_u \text{ is empty,} \\ -\infty & \text{otherwise.} \end{cases}$$

We can now express the maximum number of points in $\mathcal{P}$ that can be labeled internally using Φ. We add two dummy points to $\mathcal{P}$ that we assume are labeled externally: a point p_∞ that lies sufficiently far above and to the right of all points in $\mathcal{P}$, and a point $p_{-\infty}$ below and to the right of all points in $\mathcal{P}$. The maximum number of points labeled internally is then $\Phi(p_{-\infty}, p_\infty, \perp)$.

Computing $\Phi(\ell, u, r)$. We use dynamic programming to compute $\Phi(\ell, u, r)$ for all $\ell, u \in \mathcal{P} \cup \{p_\infty, p_{-\infty}\}$ with $\ell_y < u_y$ and $r \in E(\ell) \cup \{\perp\}$. By finding the maximum in a set of size $O(n)$, each value $\Phi(\ell, u, r)$ can be computed in $O(n)$ time, given that the values $\Phi(\ell', u', r')$ for all sub-problems have already been computed and stored in a table and the relevant values for the functions ϱ and Ψ have been precomputed. There are $O(n)$ choices for each of ℓ and u; further there are $O(\delta)$ choices for the point r given ℓ since r is labeled internally and we know from Lemma 1 that there are at most $O(\delta)$ points in $E(\ell)$ as candidates for an internal label. We assume the $O(\delta)$ candidate points in $E(\ell)$ are stored; note that we can easily precompute these sets for all points ℓ in $O(n^2)$ time. This results in an $O(n^3\delta)$ time and $O(n^2\delta)$ space dynamic-programming algorithm. We show next that the preprocessing of ϱ and Ψ can be done in $O(n^3 \log n)$ time.

Computing ϱ and Ψ. To compute $\Phi(\ell, u, r)$ we actually have to compute $\varrho(p, \ell, r)$ and $\Psi'(p, r) := \Psi(R_p \cap S(\ell, u))$ for all points $p \in S(\ell, u)$. We can preprocess all points in $\mathcal{P}$ in $O(n \log n)$ time, such that we can compute each $\varrho(p, \ell, r)$ in $O(1)$ time as follows. First, we compute and store for each point $p \in \mathcal{P}$ the topmost point $q_p \in \mathcal{P}$ in $E(p)$. This requires n standard priority range queries that take $O(n \log n)$ time in total using priority range trees with fractional cascading [9, Chapter 5]. To compute $\varrho(p, \ell, r)$ we then check if q_p lies above ℓ. If it does, we have $\varrho(p, \ell, r) = q_p$. Otherwise, the only candidate point for $\varrho(p, \ell, r)$ is r and we can check in $O(1)$ time if r lies in $E(p)$. This takes $O(1)$ time for each triple (p, ℓ, r) and $O(n^2\delta)$ time in total.

Next, we fix ℓ and u, and compute a representation of Ψ' in $O(n \log n)$ time, such that for each $p \in S(\ell, u)$ and $r \in E(\ell) \cup \{\perp\}$ we can obtain $\Psi'(p, r)$ in constant time.

We start by computing the values $\Psi'(p, \perp)$, for all p. We sweep a vertical line from right to left. That is, we sort all points in $\overline{S(\ell, u)}$ by decreasing x-coordinate, and process the points in that order. The status structure of the sweep line contains the number of points N in $S(\ell, u)$ to the right of the sweep line, and a (semi-)dynamic data structure $\mathcal{T}$, which stores the labels from the points to the right of the sweep line, and can report all labels intersected by an (axis-parallel) rectangular query window. All labels are unit squares, so λ_r intersects a label λ_q if and only if λ_r contains a corner point of λ_q. Furthermore, we only ever insert new labels (points) into $\mathcal{T}$, thus it suffices if $\mathcal{T}$ supports only insert and query operations. It follows that we can implement $\mathcal{T}$ using a semi-dynamic range tree using dynamic fractional cascading [24]. In this data structure insertions and queries take $O(\log n)$ time.

When we encounter a new point p, $p \notin \{\ell, u\}$ we test if the label of p intersects any of the labels encountered so far. We can test this using a range query in the tree $\mathcal{T}$. If $p \in S(\ell, u)$ we also explicitly test if λ_p intersects γ_ℓ or γ_u. If there are no points in the query range λ_p, and λ_p does not intersect γ_ℓ or γ_u we report $\Psi'(p, \perp) = N$, increment N (if applicable), and insert the corner points of λ_p into $\mathcal{T}$. If the query range λ_p is not empty, it follows that $\Psi'(p', \perp) = -\infty$, for $p' = p$ as well as for any point to the left of p. Hence, we report that and stop the sweep. Our algorithm runs in $O(n \log n)$ time: sorting all points takes $O(n \log n)$ time, and handling each of the $O(n)$ events takes $O(\log n)$ time.

Now consider a point $r \in E(\ell)$. We observe that for all points p right of r, we have that $\Psi'(p, r) = \Psi'(p, \perp) = 0$ since r is right of all points in $S(\ell, u)$. Consider the points left of r ordered by decreasing x-coordinate. There are two options, depending on whether or not λ_r intersects the label λ_p of the current point p. If λ_r intersects λ_p, we have $\Psi'(p, r) = -\infty$ as well as $\Psi'(p', r) = -\infty$ for all points p' left of p. If λ_r does not intersect λ_p we still have $\Psi'(p, r) = \Psi'(p, \perp)$. We can test if λ_r intersects any other label using a range priority query with λ_r in (the final version of) the range tree $\mathcal{T}$. We need $O(\delta)$ such queries, which take $O(\log n)$ time each. This gives a total running time of $O(n \log n)$. We then conclude:

Lemma 5. *Given points $\ell, u \in \mathcal{P}$, we can compute $\Psi'(p, r)$ for all points $p \in \overline{S(\ell, u)}$ and all $r \in E(\ell)$ in $O(n \log n)$ time.*

The above algorithm can also be used when the leaders have a slope $\theta \neq 0$. However, the data structure $\mathcal{T}$ that we use is fairly complicated. In this specific case where $\theta = 0$, we can also use a much easier data structure, and still get a total running time of $O(n \log n)$. Instead of using the semi-dynamic range tree as status structure, we use a simple balanced binary search tree that stores (the end-points of) a set of vertical *forbidden intervals*. When we encounter a new point p, we check if p_y lies in a forbidden interval. If this is the case then λ_p intersects another label. Otherwise we can label p internally. This set of forbidden intervals is easily maintained in $O(\log n)$ time.

We use this algorithm for every pair (ℓ, u). Hence, after a total of $O(n^3 \log n)$ preprocessing time, we can answer $\Psi'(p, r)$ queries for any p and r in constant time. This yields the following result, which improves the previously best known pseudo-polynomial $O(n^{\log n + 3})$-time algorithm of Bekos et al. [4] for the left-sided case $\theta = 0$.

Theorem 6. *Given a set $\mathcal{P}$ of n points, we can compute a labeling of $\mathcal{P}$ that maximizes the number of internal labels for $\theta = 0$ in $O(n^3 \log n + n^3 \delta)$ time and $O(n^2 \delta)$ space, where $\delta = \min\{n, 1/d_{\min}\}$ for the minimum distance $d_{\min}$ in $\mathcal{P}$.*

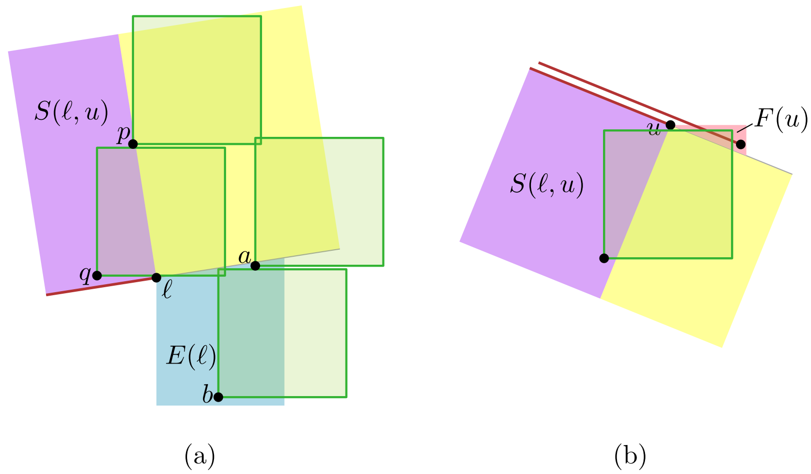


Figure 4: (a) There may be more than one point "below" ℓ with an internal label if the leaders arrive from the bottom left. (b) For other directions there is also a region $F(u)$ "above" the sub-problem that can influence the labeling of $S(\ell, u)$.

3 Other leader directions

For other leader slopes $\theta \neq 0$ we use a similar approach as before. We consider a sub-problem $S(\ell, u)$ defined by two externally labeled points ℓ and u. We again find the "right-most" point in the slab labeled externally. This gives us two sub-problems, which we solve recursively using dynamic programming. However, there are three complications:

- The region $E(\ell)$ containing the points "below" the slab $\overline{S(\ell, u)}$ that can influence the labeling of $S(\ell, u)$ is no longer a unit square. Depending on the orientation, it can contain more than one point with an internal label. See Figure 4(a).
- In addition to the region $E(\ell)$, which contains points that can interfere with a sub-problem from below, we now also need to consider a second region, which we call $F(u)$, containing points whose labels can interfere with a sub-problem from above. See Figure 4(b).
- The labels of points in $S(\ell, u)$ are no longer fully contained in the slab $\overline{S(\ell, u)}$. Hence, we have to check that they do not intersect with leaders of points outside $\overline{S(\ell, u)}$. See Figure 4(b).

We start by explicitly finding the points in $\mathcal{P}$ whose internal labels contain other points. We are forced to label these points externally. It is easy to find those points in $O(n^2)$ time in total. Let $\mathcal{P}_X$ denote this set of points. Additionally, we spend $O(n^2)$ time to mark each point if its label intersects $\Gamma(\mathcal{P}_X)$. Hence, for each point we can determine in constant time if it intersects $\Gamma(\mathcal{P}_X)$. For ease of notation we will write $\mathcal{P}$ to mean the set $\mathcal{P} \setminus \mathcal{P}_X$ in the remainder of this section.

Let θ be the given orientation for the leaders. Consider conceptually rotating the coordinate system such that orientation θ corresponds to the negative x-axis as in the previous section, and let $\tilde{B}_p, \tilde{T}_p, \tilde{L}_p$, and $\tilde{R}_p$ denote the points in $\mathcal{P}$ in the bottom, top, left, and right half-planes bounded by p with respect to this coordinate system. Analogously, we define $\overline{S(\ell,u)} = \tilde{T}_\ell \cap \tilde{B}_u$, and $S(\ell,u) = \overline{S(\ell,u)} \cap \tilde{L}_\ell \cap \tilde{L}_u$.

Our goal is again to bound the number of different labelings of the points in $\tilde{B}_\ell$ and $\tilde{T}_u$ that can influence the labeling of $S(\ell,u)$. We will show that whether a labeling of $\tilde{B}_\ell$ influences the labeling of $S(\ell,u)$ depends only on a small subset of the points in $\tilde{B}_\ell$. We refer to a labeling of $\tilde{B}_\ell$ restricted to those points as a *configuration* of $\tilde{B}_\ell$. The same holds for the points in $\tilde{T}_u$.

3.1 Bounding the number of configurations

Before we solve the problem for arbitrary orientations, we first investigate what structural changes we encounter as we rotate through the full circle of possible orientations. Since we assume that labels are always placed to the top right of the points and are square shaped, it follows that the behavior of the problem is symmetric in leader directions to the top left and directions to the bottom right of the main diagonal $x = y$. (Remember that the square labels are not a restriction; see Section 4.5.) Hence, the problem for leaders leading down is symmetric to the problem for leaders leading to the left, and can be solved using the same algorithm as described in the previous section.

As we rotate the leader direction, we encounter structural changes whenever the leader angle is a multiple of $45°$. This is illustrated in the "wheel of direction" in Lemma 7 and in Figure 5. Specifically, we now define two regions E and F analogous to E in the previous section; these regions contain all points whose placement influences the solution of a sub-problem. The number of points that can be labeled internally in E and F directly influences the running time, since we need to try all possibilities. Figure 5 illustrates the shape of E (the shape of F is symmetric by reflection in the line $x = y$, see Figure 7).

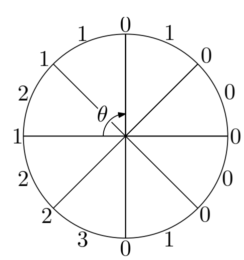
An additional complication is that the labels of points in $S(\ell,u)$ are no longer fully contained in the slab defined by ℓ and u. This means that they could potentially intersect with leaders of points outside the slab, whose state we do not yet know when solving the sub-problem. We prove that the number of ways the solution can influence the situation outside the slab in this way is only linear.

We start by bounding the number of configurations of $\tilde{B}_\ell$. Let $E(\ell)$ denote the *bottom influence region* of ℓ. That is, the points in $\tilde{B}_\ell$ "below" the slab $\overline{S(\ell,u)}$ whose labels can intersect a label of a point in $S(\ell,u)$. Figure 5 shows the regions $E(\ell)$ for various orientations of the leaders.

Similarly, we can define a region $E'(\ell) \subset \tilde{B}_\ell$ such that the *leaders* of points in $E'(\ell)$ can intersect a label of a point in $S(\ell,u)$.

Lemma 7. *For a sub-problem $S(\ell,u)$ the size of the bottom influence region $E(\ell)$ is at most $1 \times e(\theta)$ or $e(\theta) \times 1$, where*

$$e(\theta) \leq \begin{cases} 1 & \text{if } \theta = 0 \\ 2 & \text{if } \theta \in (0, \pi/4) \\ 1 & \text{if } \theta \in [\pi/4, \pi/2) \\ 0 & \text{if } \theta = \pi/2 \\ 1 & \text{if } \theta \in (\pi/2, 3\pi/4) \end{cases} \qquad e(\theta) \leq \begin{cases} 0 & \text{if } \theta \in [3\pi/4, 5\pi/4] \\ 1 & \text{if } \theta \in (5\pi/4, 3\pi/2) \\ 0 & \text{if } \theta = 3\pi/2 \\ 3 & \text{if } \theta \in (3\pi/2, 7\pi/4) \\ 2 & \text{if } \theta \in [7\pi/4, 2\pi). \end{cases}$$



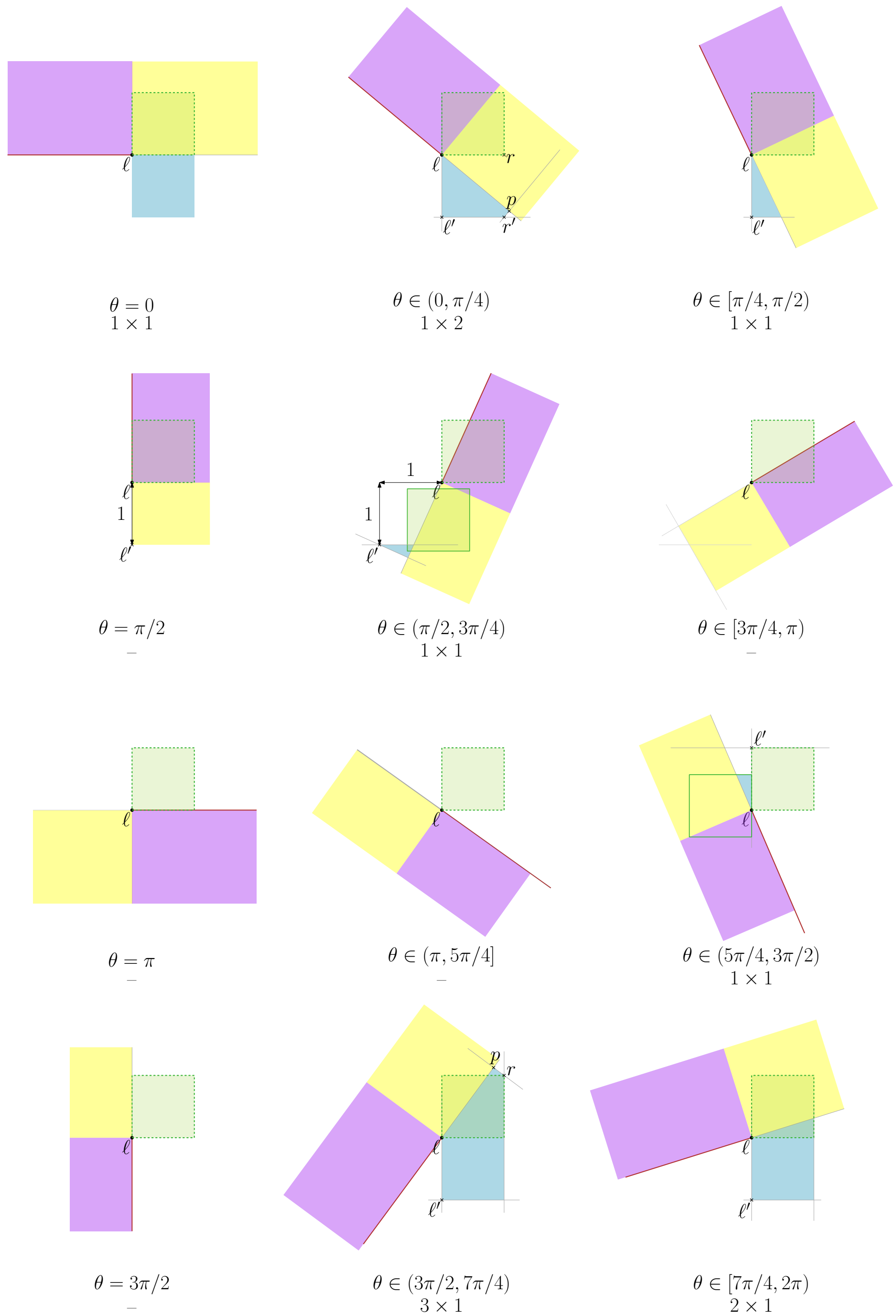


Figure 5: The region $E(\ell)$ (blue) with respect to sub-problem $S(\ell, u)$ (purple), depending on the orientation θ of the leaders. For each orientation we give an upper bound on the size of $E(\ell)$.

Proof. We prove this by case distinction on θ. (See also Figure 5)

case $\theta = 0$**.** See Lemma 3.

case $\theta \in (0, \pi/4)$**.** All labels in $\Lambda(S(\ell, u))$ lie in T_ℓ and in $\tilde{L_r}$, where $r = (\ell_x+1, \ell_y)$. It follows that $E(\ell) \subseteq T_{\ell'} \cap \tilde{L_{r'}}$, where $\ell' = (\ell_x, \ell_y - 1)$ and $r' = (r_x, r_y - 1)$. Furthermore, it is easy to see that $E(\ell) \subset R_\ell$, and that $E(\ell) \subseteq \tilde{B}_\ell$. Hence, $E(\ell) \subset T_{\ell'} \cap R_\ell \cap \tilde{L_{r'}} \cap \tilde{B}_\ell$.
Since the leaders are sloped downwards it follows that the height of $E(\ell)$ is at most one. The maximum width of $E(\ell)$ is realized by ℓ and the intersection point p of the lines bounding $\tilde{B}_\ell$ and $\tilde{L_{r'}}$. Using that $\theta \in (0, \pi/4)$ basic trigonometry shows that the width is at most two.

case $\theta \in [\pi/4, \pi/2)$**.** Similar to the previous case. However, now the width is determined by the intersection between $T_{\ell'}$ and $\tilde{B}_\ell$. From basic trigonometry it then follows that the width is at most one.

case $\theta = \pi/2$**.** For labels from $E(\ell)$ to intersect $\Lambda(S(\ell, u))$ we need $E(\ell) \subseteq T_{\ell'}$, where $\ell' = (\ell_x, \ell_y - 1)$. However, to avoid intersecting γ_ℓ we need $E(\ell) \subseteq B_{\ell'}$. It follows that $E(\ell) = \emptyset$.

case $\theta \in (\pi/2, 3\pi/4)$**.** Using similar arguments as before it follows that $E(\ell) \subset B_{\ell'} \cap \tilde{L_{\ell'}} \cap \tilde{B}_\ell \cap L_\ell \cap R_{\ell'}$, where $\ell' = (\ell_x - 1, \ell_y - 1)$. Since $E(\ell) \subset L_\ell \cap R_{\ell'}$ the width is at most one. Basic trigonometry again shows that the height is at most one.

case $\theta \in [3\pi/4, 5\pi/4]$**.** For the sub-case $\theta \in [3\pi/4, \pi)$ the lines bounding $\tilde{L_\ell}$ and $R_{\ell'}$, with $\ell' = (\ell_x - 1, \ell_y - 1)$ intersect below the line containing γ_ℓ. We then obtain $E(\ell) \subset R_{\ell'} \cap B_{\ell'} \cap \tilde{B}_\ell = \emptyset$. In the remaining sub-case $\theta \in [\pi, 5\pi/4]$ the regions $\Lambda(S(\ell, u))$ and $\Lambda(\tilde{B}_\ell)$ are disjoint. It follows that $E(\ell)$ is empty.

case $\theta \in (5\pi/4, 3\pi/2)$**.** Point ℓ is now the point with the maximum y-coordinate. It then follows that all labels of $S(\ell, u)$ lie in $B_{\ell'}$, where $\ell' = (\ell_x, \ell_y + 1)$. Hence, we also get $E(\ell) \subset B_{\ell'}$. The labels of $S(\ell, u)$ do not intersect γ_ℓ, hence they are contained in $L_\ell \cup \tilde{T}_\ell$. We then have $E(\ell) \subset \tilde{B}_\ell \cap (L_\ell \cup \tilde{T}_\ell) = \tilde{B}_\ell \cap L_\ell$. Since $\theta \in (5\pi/4, 3\pi/2)$ it now follows that the height and width are both at most one.

case $\theta = 3\pi/2$**.** All labels from $S(\ell, u)$ lie in L_ℓ, all labels from $\tilde{B}_\ell = R_\ell$ lie in R_ℓ. Hence, $E(\ell) = \emptyset$.

case $\theta \in (3\pi/2, 7\pi/4)$**.** It is again easy to show that $E(\ell) \subset R_\ell \cap L_r$, where $r = (\ell_x+1, \ell_y+1)$, and thus has width (at most) one. All labels from points in $S(\ell, u)$ have width and height one, and are thus contained in $\tilde{L_r}$. Furthermore, they do not intersect γ_ℓ, from which we obtain that they are contained in $\tilde{T}_\ell \cup T_\ell$. From the former we get that $E(\ell) \subset \tilde{L_r}$. From the latter we get that $E(\ell) \subset \tilde{T_{\ell'}} \cup T_{\ell'}$, where $\ell' = (\ell_x, \ell_y - 1)$. Hence, we obtain $E(\ell) \subset R_\ell \cap L_r \cap \tilde{L_r} \cap (\tilde{T_{\ell'}} \cup T_{\ell'}) \cap \tilde{B}_\ell$.
Since $\theta \in (3\pi/2, 7\pi/4)$ the height of $E(\ell)$ is determined by ℓ' and the intersection point p between $\tilde{B}_\ell$ and $\tilde{L_r}$. Trigonometry now shows that the height is at most three.

case $\theta \in [7\pi/4, 2\pi)$ Similar to the previous case we get a width of at most one. The height is now determined by ℓ' and the intersection of $\tilde{B}_\ell$ and R_r. Since $\theta \in [7\pi/4, 2\pi)$ the height is at most two. □

Corollary 8. *There can be at most $e(\theta)$ points in $E(\ell)$ labeled internally such that their labels are disjoint.*

Next, we turn our attention to the points in $E'(\ell)$ whose leader can intersect a label of a point in $S(\ell, u)$. A leader of a point in $\tilde{B}_\ell$, and thus in $E'(\ell)$, can intersect a label of $S(\ell, u)$ only if the labels intersect $\tilde{B}_\ell$. This happens only if $\theta \in (5\pi/4, 3\pi/2)$ or $\theta \in$

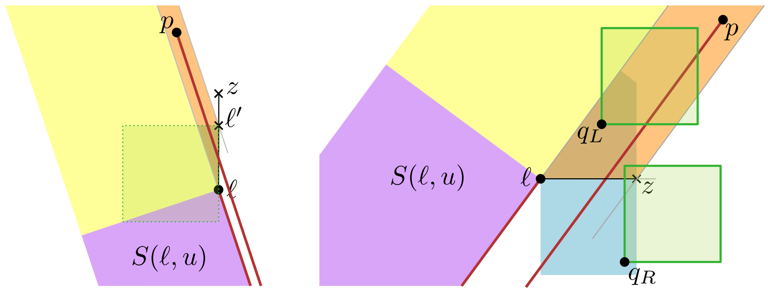


Figure 6: The region $E'(\ell)$ (in orange) for the cases $\theta \in (5\pi/4, 3\pi/2)$ (a) and $\theta \in (3\pi/2, 2\pi)$ (b). In both cases the leader γ_p of a point $p \in E''(\ell)$ intersects the line segment $\overline{\ell z}$ and thus subdivides $E(\ell)$ into a left region L and a right region R.

$(3\pi/2, 2\pi)$, see Figure 5. In the former case we thus have $E'(\ell) = \tilde{B}_\ell \cap \tilde{T}_{\ell'} \cap L_\ell$, where $\ell' = (\ell_x, \ell_y + 1 - \tan(\theta - 5\pi/4))$, and in the latter case we have $E'(\ell) = \tilde{B}_\ell \cap \tilde{T}_z \cap T_z$, where $z = (\ell_x + 1, \ell_y)$, see Figure 6.

We now note that if we label the set $Q \subseteq E(\ell)$ internally, then all other points in $E(\ell)$ are labeled externally. Hence, if the leaders of the remaining points (e.g., those in $E(\ell) \setminus Q$) intersect with labels of points in $S(\ell, u)$, this is already captured by the configuration involving Q. Therefore, the points in $E(\ell) \setminus Q$ themselves do not define new configurations. Similarly, the points that we were forced to label externally, the set $\mathcal{P}_X$, do not define any new configurations.

The points in $E'(\ell)$ that lie outside of $E(\ell)$ can still be labeled both internally or externally. Let $E''(\ell) = E'(\ell) \setminus E(\ell)$ denote the region containing these points. We now observe:

Lemma 9. *Let Q be the set of points in $E(\ell)$ labeled internally, and let $p \in E''(\ell)$ be labeled externally. For all points $q \in Q$ we have: λ_q intersects the leader γ_p, or if there is a label λ_a, $a \in S(\ell, u)$, that intersects λ_q, then it also intersects the leader γ_p.*

Proof. It is easy to see that any leader γ_p intersects the line segment $\overline{\ell z}$, with $z = (\ell_x, \ell_y + 1)$ if $\theta \in (5\pi/4, 3\pi/2)$, and $z = (\ell_x + 1, \ell_y)$ if $\theta \in (3\pi/2, 2\pi)$, see Figure 6. In both cases γ_p subdivides $E(\ell)$ into a left region L and a right region R. At any height (y-coordinate), this left region L has width at most one. Hence, if there is a point $q \in L$ labeled internally, then its label intersects γ_p. Any point $q \in R$ is separated from $S(\ell, u)$ by γ_p. So, if there is a label $a \in S(\ell, u)$ that intersects λ_q, then it also intersects γ_p. $\square$

Observation 10. *Let $p \in \mathcal{P} \cap E''(\ell)$ be the point closest to the slab $\overline{S(\ell, u)}$, and let q be any point in $\mathcal{P} \cap E''(\ell)$. If there is a point $a \in S(\ell, u)$ whose label λ_a intersects γ_q, then λ_a also intersects γ_p.*

Observation 10 gives us that there is only one relevant point in $E''(\ell)$, namely the point p closest to $\overline{S(\ell, u)}$. Furthermore, from Lemma 9 it follows that if p exists, then there are no relevant points in $E(\ell)$ labeled internally. Hence, p by itself determines a configuration. We can then define the universe $\mathcal{U}_{\ell E}$ of possible configurations of $\tilde{B}_\ell$ as follows:

$$\begin{aligned}\mathcal{U}_{\ell E} = \{(\mathcal{I}, \emptyset) \mid \mathcal{I} \subseteq E(\ell) \text{ and all labels in } \Lambda(\mathcal{I}) \text{ are pairwise disjoint}\} \cup \\ \{(\emptyset, \{p\}) \mid p \in E''(\ell)\}\end{aligned}$$

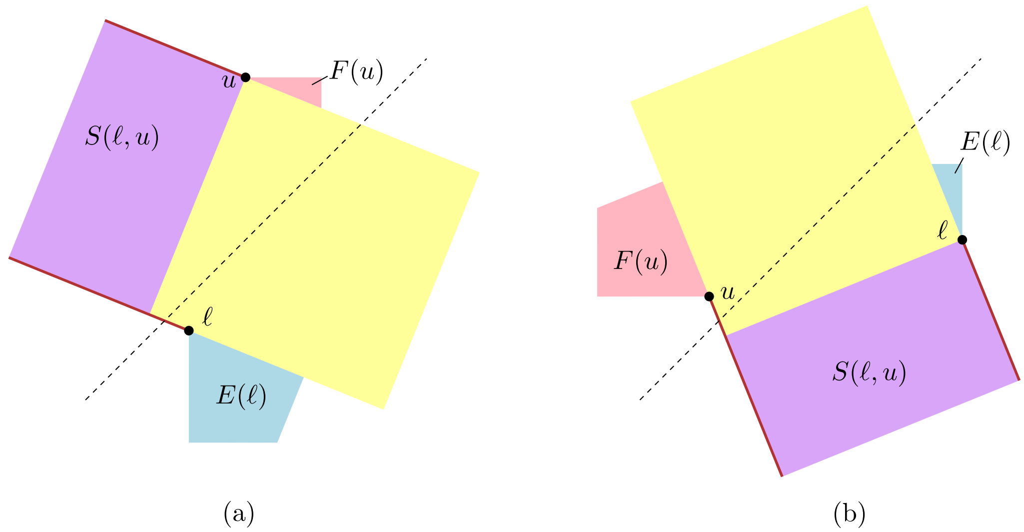


Figure 7: Because internal labels are square-shaped and to the top right of points, the problem is symmetric by reflection in the line $x = y$. The roles of u and ℓ and E and F are swapped.

Let $e'(\theta) = 1$ if there can a point in $E''(\ell)$ labeled externally, and $e'(\theta) = 0$ otherwise (this includes the case in which $E'(\ell) = \emptyset$). We then have that $e'(\theta) \leq 1$ if $\theta \in (5\pi/4, 3\pi/2) \cup (3\pi/2, 2\pi)$ and $e'(\theta) = 0$ otherwise. Using that the labels of points in $E(\ell)$ do not contain any other points, together with Lemma 1 we then obtain:

Lemma 11. *The number of configurations of $\tilde{B}_\ell$ is at most $O(|\mathcal{U}_{\ell E}|) = O(\delta^{e(\theta)} + n^{e'(\theta)})$.*

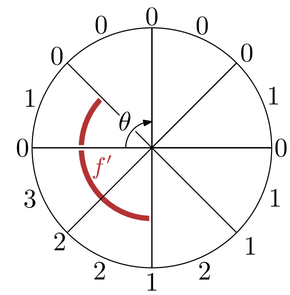
Bounding the number of configurations of $\tilde{T}_u$ Analogously to the bottom influence region $E(\ell)$ in $\tilde{B}_\ell$ we define a *top influence region* $F(u)$ containing the points from $\tilde{T}_u$ whose label can intersect a label of $S(\ell, u)$, and a region $F'(u)$ containing the points whose leader can intersect a label of the points in $S(\ell, u)$. Figure 4 illustrates this. We observe that $F(u)$ and $F'(u)$ are symmetric to $E(\ell)$ and $E'(\ell)$ by mirroring in a line with slope 1, see Figure 7. We thus get similar results for F and F' as those stated in Lemmas, Corollaries, and Observations 7–10. So, similarly we define the universe of configurations $\mathcal{U}_{uF}$ of labelings of $\tilde{T}_\ell$. We can then summarize our results in the following lemma:

Lemma 12. *The number of configurations of $\tilde{T}_u$ is at most $O(|\mathcal{U}_{uF}|) = O(\delta^{f(\theta)} + n^{f'(\theta)})$, where*

$$f(\theta) \leq \begin{cases} 0 & \text{if } \theta = 0 \\ 1 & \text{if } \theta \in (0, \pi/4) \\ 0 & \text{if } \theta \in [\pi/4, 3\pi/4] \\ 1 & \text{if } \theta \in (3\pi/4, \pi) \\ 0 & \text{if } \theta = \pi \end{cases} \qquad f(\theta) \leq \begin{cases} 1 & \text{if } \theta \in (\pi, 5\pi/4] \\ 2 & \text{if } \theta \in (5\pi/4, 3\pi/2) \\ 1 & \text{if } \theta = 3\pi/2 \\ 2 & \text{if } \theta \in (3\pi/2, 7\pi/4] \\ 3 & \text{if } \theta \in (7\pi/4, 2\pi) \end{cases}$$

and

$$f'(\theta) \leq \begin{cases} 1 & \textit{if } \theta \in (0, \pi/4) \cup (3\pi/2, 2\pi) \\ 0 & \textit{otherwise.} \end{cases}$$



3.2 Computing an optimal labeling

In this section, we show how to compute an optimal labeling for a given choice of θ. Using the observations in the previous section, we know the shape of E and F, and the maximum number of internally labeled points in them. The main idea is to create a dynamic program analogous to that in Section 2, where we recursively compute the optimal solutions to subproblems $\Phi(\ell, u, \mathcal{C}_{\ell E}, \mathcal{C}_{uF})$, where $\mathcal{C}_{\ell E}$ and $\mathcal{C}_{uF}$ encode the labeling of points in E and F respectively.

Let $\Phi(\ell, u, \mathcal{C}_{\ell E}, \mathcal{C}_{uF})$ denote the maximum number of points in $S(\ell, u)$ that can be labeled internally, given configurations $\mathcal{C}_{\ell E} = (\mathcal{I}_\ell, \mathcal{E}_\ell)$ and $\mathcal{C}_{uF} = (\mathcal{I}_u, \mathcal{E}_u)$. That is, the maximum number of points in $S(\ell, u)$ that can be labeled internally assuming that (i) the points in $\mathcal{I}_\ell \subseteq E(\ell)$ and $\mathcal{I}_u \subseteq F(u)$ are labeled internally, and (ii) the points in $\mathcal{E}_\ell$, $\mathcal{E}_u$, and the remaining points in $E(\ell)$ and $F(u)$ are labeled externally.

We now proceed completely analogous to Section 2.2. We define functions ϱ, and Ψ that have the same goal as before, and a function ς symmetric to ϱ. With these functions, and an argument analogous to Lemma 4 we can then give a recursive definition for Φ.

We define a function $\varrho(p, \ell, u, \mathcal{C}_{\ell E})$ that restricts the universe of configurations $\mathcal{U}_{pE}$ to the sets *compatible* with the labeling so far. More formally, we have

$$\varrho(p, \ell, u, \mathcal{C}_{\ell E}) = \left\{ (\mathcal{I}, \mathcal{E}) \;\middle|\; \begin{array}{l} (\mathcal{I}, \mathcal{E}) \in \mathcal{U}_{pE}, \\ \mathcal{I} \supseteq E(p) \cap ((\overline{S(\ell, u)} \cap \tilde{R}_p) \cup \mathcal{I}_\ell), \text{ and} \\ \mathcal{E} = \begin{cases} \{\ell\} & \text{if } \ell \in E''(p) \\ \mathcal{E}_\ell & \text{if } \ell \notin E''(p) \wedge \mathcal{E}_\ell \subset E''(p) \\ \emptyset & \text{otherwise.} \end{cases} \end{array} \right\}$$

Symmetrically, we define $\varsigma(p, \ell, u, \mathcal{C}_{uF})$ as universe $\mathcal{U}_{pF}$ restricted to the sets compatible with the labeling so far:

$$\varsigma(p, \ell, u, \mathcal{C}_{uF}) = \left\{ (\mathcal{I}, \mathcal{E}) \;\middle|\; \begin{array}{l} (\mathcal{I}, \mathcal{E}) \in \mathcal{U}_{pF}, \\ \mathcal{I} \supseteq F(p) \cap ((\overline{S(\ell, u)} \cap \tilde{R}_p) \cup \mathcal{I}_u), \text{ and} \\ \mathcal{E} = \begin{cases} \{u\} & \text{if } u \in F''(p) \\ \mathcal{E}_u & \text{if } u \notin F''(p) \wedge \mathcal{E}_u \subset F''(p) \\ \emptyset & \text{otherwise.} \end{cases} \end{array} \right\}$$

An argument similar to that of Lemma 4 then gives us the following recurrence for $\Phi(\ell, u, \mathcal{C}_{\ell E}, \mathcal{C}_{uF})$:

$$\Phi(\ell, u, \mathcal{C}_{\ell E}, \mathcal{C}_{uF}) = \max \left\{ \Psi(S(\ell,u)), \quad \max_{\substack{p \in S(\ell,u), \\ \mathcal{C}_{pE} \in \varrho(p,\ell,u,\mathcal{C}_{\ell E}), \\ \mathcal{C}_{pF} \in \varsigma(p,\ell,u,\mathcal{C}_{uF})}} \left\{ \begin{array}{c} \Psi(\tilde{R}_p \cap S(\ell,u)) \\ + \ \Phi(\ell, p, \mathcal{C}_{\ell E}, \mathcal{C}_{pF}) \\ + \ \Phi(p, u, \mathcal{C}_{pE}, \mathcal{C}_{uF}) \end{array} \right\} \right\},$$

where, similar to Section 2.2, $\Psi(P)$ is defined as the number of points in P, provided that they can be labeled internally without intersecting each other or the existing part of the labeling:

$$\Psi(P) = \begin{cases} |P| & \text{if all labels in } \Lambda(P \cup \mathcal{I}_\ell \cup \mathcal{I}_u \cup (\overline{S(\ell,u)} \setminus S(\ell,u))) \text{ are pairwise disjoint, and} \\ & \text{their intersection with the leaders in } \Gamma(\mathcal{P}_X \cup \{\ell, u\} \cup \mathcal{E}_\ell \cup \mathcal{E}_u \cup (E(\ell) \setminus \mathcal{I}_\ell) \cup \\ & (F(u) \setminus \mathcal{I}_u)) \text{ is empty,} \\ -\infty & \text{otherwise.} \end{cases}$$

Computing $\Phi(\ell, u, \mathcal{C}_{\ell E}, \mathcal{C}_{uF})$ We again use dynamic programming. The size of our table is now $O(n^2|\mathcal{U}_{\ell E}||\mathcal{U}_{uF}|)$. To compute the value of an entry $\Phi(\ell, u, \mathcal{C}_{\ell E}, \mathcal{C}_{uF})$, we maximize over $O(\sum_{p \in S(\ell,u)} |\mathcal{U}_{pE}||\mathcal{U}_{pF}|)$ other entries. For each such entry, we need to compute the value of $\Psi(P)$, for some set of points P. We can do this in $O(n \log n)$ time using the algorithm from Section 2. In total this yields an $O(n^3|\mathcal{U}_{\ell E}||\mathcal{U}_{uF}|(\sum_{p \in S(\ell,u)} |\mathcal{U}_{pE}||\mathcal{U}_{pF}|) \log n)$ time algorithm. Next, we describe how to improve this to $O(n^2(n + |\mathcal{U}_{\ell E}| + |\mathcal{U}_{uF}|) \log n + n^2(|\mathcal{U}_{\ell E}||\mathcal{U}_{uF}| \sum_{p \in S(\ell,u)} |\mathcal{U}_{pE}||\mathcal{U}_{pF}|))$ time by precomputing Ψ.

Fix two points ℓ and u, and let $\Psi'(p, \mathcal{C}_{\ell E}, \mathcal{C}_{uF}) := \Psi(\tilde{R}_p \cap S(\ell, u))$, given configurations $\mathcal{C}_{\ell E}$ and $\mathcal{C}_{uF}$. We use a similar approach as in Section 2. We first compute Ψ' for all points p, assuming that no other points above or below $\overline{S(\ell,u)}$ interfere with $S(\ell,u)$. That is, we compute all values $\Psi'(p, (\emptyset, \emptyset), (\emptyset, \emptyset))$. This takes $O(n \log n)$ time using the same algorithm as before. For each of the remaining pairs of configurations $(C_{\ell E}, C_{uF})$, we find the rightmost (with respect to the rotated coordinate system) point p in $\overline{S(\ell,u)}$ such that the label of p conflicts with $C_{\ell E}$ or C_{uF}. It then follows that $\Psi'(p', C_{\ell E}, C_{uF}) = -\infty$ for all $p' \in \tilde{L_p} \cup \{p\}$.

Next, we describe how we can find the rightmost point that conflicts with $\mathcal{C}_{\ell E}$ in $O(\log n)$ time, after $O(n \log n)$ time preprocessing. We find the rightmost point that conflicts with $\mathcal{C}_{uF}$ analogously. It then follows that we can compute $\Psi'(p, \mathcal{C}_{\ell E}, \mathcal{C}_{uF})$ for all configurations and all points $p \in \overline{S(\ell,u)}$ in $O(n \log n + |\mathcal{U}_{\ell E}||\mathcal{U}_{uF}| + (|\mathcal{U}_{\ell E}| + |\mathcal{U}_{uF}|) \log n)$ time in total.

The rightmost point that conflicts with a configuration $\mathcal{C}_{\ell E} = (\mathcal{I}, \mathcal{E})$ conflicts with the set of internally labeled points $\mathcal{I}$, or the set of externally labeled points in $\mathcal{E} \cup E(\ell) \setminus \mathcal{I}$. For both these sets we find the rightmost point conflicting with it, and return the rightmost point of those two.

To find the rightmost point q conflicting with $\mathcal{I}$, we use the same procedure as in the previous section. We build a range tree on the corner points of $\Lambda(\overline{S(\ell,u)})$, and use a priority range query to find $q_r \in \overline{S(\ell,u)}$ whose label intersects a query label λ_r, $r \in \mathcal{I}$. We can thus find q in $O(|\mathcal{I}| \log n) = O(\log n)$ time.

If the labels of $S(\ell,u)$ are not contained in $\overline{S(\ell,u)}$ then we also need to find the rightmost point whose label intersects a leader in $Y := \Gamma(\mathcal{E} \cup E(\ell) \setminus \mathcal{I})$. To do this we need two more

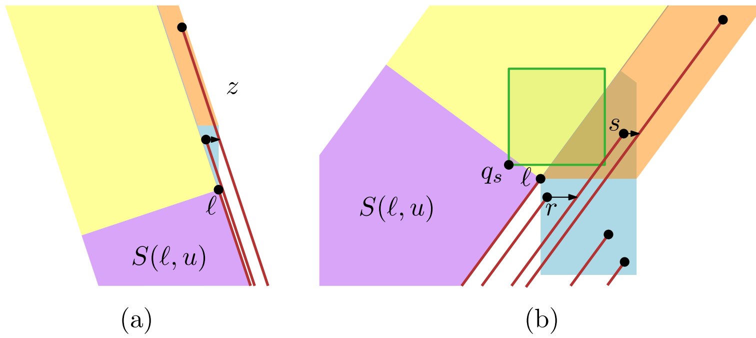


Figure 8: We can find the leaders that can intersect a label from a point in $S(\ell, u)$ by a series of horizontal ray shooting queries.

data structures. We preprocess the edges of $Z := \Lambda(\overline{S(\ell, u)})$ to allow for ray shooting queries with rays of orientation θ. Since all edges of Z are either horizontal or vertical, and all query rays have the same orientation, this can be done with a shear transformation and two standard one-dimensional interval or segment trees $\mathcal{T}'$ (one for the horizontal edges of Z and one for the vertical edges of Z). We can build these trees in $O(n \log n)$ time [9]. This allows us to find the rightmost point q_r whose label intersects a given leader γ_r in $O(\log n)$ time. To find the rightmost point whose label intersects a leader among *all* leaders in Y we use the following approach. We query $\mathcal{T}'$ with the leader $\gamma_r \in Y$ closest to $\overline{S(\ell, u)}$, and find q_r (if it exists). We then find the first leader γ_s that is hit by a horizontal rightward ray starting in r (see Figure 8), and recursively process γ_s. For any subsequent pair of such leaders (γ_r, γ_s) we have that point s lies outside of the label λ_r. Since all but one of these points lie in $E(\ell)$, there are only a constant number of such pairs. Hence, we also need only a constant number of queries in $\mathcal{T}'$, each of which takes $O(\log n)$ time.

The only question remaining is how to find the next leader γ_s given point r. To this end we maintain a second data structure. We use a shear transformation such that the leaders are all vertical. We then build a dynamic data structure $\mathcal{D}$ for horizontal ray shooting queries among vertical half-lines. Such a data structure can be build in $O(n \log n)$ time, and allows for $O(\log n)$ updates and queries [12]. [2] We can update $\mathcal{D}$ for the next configuration $\mathcal{C}'_{\ell E}$ in $O(\log n)$ time, since only a constant number of points change from being labeled internally to labeled externally and vice versa.

Hence, after $O(n \log n)$ time preprocessing, we can compute the rightmost point conflicting with each configuration $\mathcal{C}_{\ell E}$ in $O(\log n)$ time. The total time required to compute $\Psi'(p, \mathcal{C}_{\ell E}, \mathcal{C}_{uF})$, for all points $p \in S(\ell, u)$, is thus $O(|\mathcal{U}_{\ell E}||\mathcal{U}_{uF}| + (n + |\mathcal{U}_{\ell E}| + |\mathcal{U}_{uF}|) \log n)$.

So, after a total of $O(n^2|\mathcal{U}_{\ell E}||\mathcal{U}_{uF}| + n^2(n + |\mathcal{U}_{\ell E}| + |\mathcal{U}_{uF}|) \log n))$ time preprocessing, the dynamic programming algorithm runs in $O(n^2|\mathcal{U}_{\ell E}||\mathcal{U}_{uF}| \sum_{p \in S(\ell,u)} |\mathcal{U}_{pE}||\mathcal{U}_{pF}|)$ time.

[2] Note that there is a simpler implementation that works in this case: maintain the half-lines (leaders) in a fully persistent balanced binary search tree, ordered on x-coordinate. In addition, augment the tree to maintain the maximum y-coordinate of the starting points of the half-lines in the subtree. When querying for the next half-line hit by a rightward horizontal ray starting in point p, split T at p_x, and use the subtree maxima in the right tree to find the first half-line whose starting point is at least p_y.

Combining these results, gives us a running time of

$$O(n^2(n + |\mathcal{U}_{\ell E}| + |\mathcal{U}_{uF}|) \log n + n^2(|\mathcal{U}_{\ell E}||\mathcal{U}_{uF}| \sum_{p \in S(\ell,u)} |\mathcal{U}_{pE}||\mathcal{U}_{pF}|)) \tag{1}$$

as claimed. Finally, we use that $|\mathcal{U}_{\ell E}|$ and $|\mathcal{U}_{pE}|$, for all p, are all at most $O(n^{e'(\theta)} + \delta^{e(\theta)})$ (Lemma 11), and that $|\mathcal{U}_{uF}|$ and $|\mathcal{U}_{pF}|$, for all p, are all at most $O(n^{f'(\theta)} + \delta^{f(\theta)})$ (Lemma 12). We define the term $\iota(n, \delta, \theta)$ to bound the products of the form $|\mathcal{U}_{\ell E}| \cdot |\mathcal{U}_{uF}| \cdot |\mathcal{U}_{pE}| \cdot |\mathcal{U}_{pF}|$ appearing in (1) for $p \in S(\ell, u)$ in terms of n, δ and θ as follows

$$\begin{aligned}
\iota(n,\delta,\theta) = \; & n^{2e'(\theta)+2f'(\theta)} && + n^{2e'(\theta)+f'(\theta)}\delta^{f(\theta)} && + n^{e'(\theta)+2f'(\theta)}\delta^{e(\theta)} \\
& + n^{e'(\theta)+f'(\theta)} \; \delta^{e(\theta)+f(\theta)} && + n^{2e'(\theta)} \; \delta^{2f(\theta)} && + n^{2f'(\theta)} \; \delta^{2e(\theta)} \\
& + n^{e'(\theta)} \; \delta^{e(\theta)+2f(\theta)} && + n^{f'(\theta)} \; \delta^{2e(\theta)+f(\theta)} && + \; \delta^{2e(\theta)+2f(\theta)}.
\end{aligned}$$

This allows us to rewrite expression (1) to $O(n^3(\log n + \iota(n, \delta, \theta)))$, where the summation over $p \in S(\ell, u)$ yields a factor n and $\iota(n, \delta, \theta)$ basically models how much the sub-problems can influence each other. For $\delta = O(n)$, this gives us a worst case running time depending on the choice of θ that varies between $O(n^3 \log n)$ if $\theta \in \{\pi/2, 3\pi/4, \pi\}$ and $O(n^{13})$ if $\theta \in (3\pi/2, 7\pi/4) \cup (7\pi/4, 2\pi)$. We have that (for any p) $|\mathcal{U}_{\ell E}| \cdot |\mathcal{U}_{uF}| \cdot |\mathcal{U}_{pE}| \cdot |\mathcal{U}_{pF}| = O(\iota(n, \delta, \theta))$. Thus, our dynamic programming table has size at most $O(n^2 |\mathcal{U}_{\ell E}||\mathcal{U}_{uF}|) = O(n^2 \sqrt{\iota(n, \delta, \theta)})$. We conclude:

Proposition 13. *Given a set $\mathcal{P}$ of n points and an angle θ, we can compute a labeling of $\mathcal{P}$ that maximizes the number of internal labels in $O(n^3(\log n + \iota(n, \delta, \theta)))$ time and $O(n^2\sqrt{\iota(n, \delta, \theta)})$ space, where $\delta = \min\{n, 1/d_{\min}\}$ for the minimum distance $d_{\min}$ in $\mathcal{P}$, and $\iota(n, \delta, \theta)$ models how much sub-problems can influence each other.*

3.3 An improved bound on the number of configurations

The analysis above, together with the fact that $e(\theta)$ and $f(\theta)$ are both at most three, yet not simultaneously, gives us a worst case running time of $O(n^{13})$. We now study the situation a bit more carefully, and show that the number of interesting configurations is much smaller. More specifically, that we can replace $e(\theta)$ and $f(\theta)$ by quantities $e^*(\theta)$ and $f^*(\theta)$ that are both at most one. This significantly improves the running time of our algorithm.

We start by observing that for some points q in $E(\ell)$, the sub-problem $S(\ell, u)$ does not have a labeling compatible with q, irrespective of whether q is labeled internally or externally. Let $Q(\ell)$ denote the set of such points. It follows that we can restrict ourselves to labelings, and thus configurations, that do not contain points from $Q(\ell)$. Let $\mathcal{U}^*_{\ell E} = \{(\mathcal{I}, \mathcal{E}) \mid (\mathcal{I}, \mathcal{E}) \in \mathcal{U}_{\ell E} \wedge \mathcal{I} \subseteq E(\ell) \setminus Q(\ell)\}$ denote this subset of configurations.

Lemma 14. *For every configuration $(\mathcal{I}, \mathcal{E}) \in \mathcal{U}^*_{\ell E}$, the set $\mathcal{I}$ has size at most two.*

Proof. By Corollary 8 there are at most $e(\theta)$ points in $E(\ell)$ that can be labeled internally simultaneously. Hence $|\mathcal{I}| \leq e(\ell)$. For $\theta \in [0, 3\pi/2] \cup [7\pi/4, 2\pi]$ we have $e(\theta) \leq 2$, and thus the lemma follows immediately. For $\theta \in (3\pi/2, 7\pi/4)$ we have $e(\theta) \leq 3$. We prove this remaining case by contradiction.

Assume that $\theta \in (3\pi/2, 7\pi/4)$, and that $\mathcal{I} = \{a, b, c\}$. Since the region $E(\ell)$ has size 1×3, it follows that one of these points, say point a, lies in the triangular region $\tilde{B}_\ell \cap \tilde{L}_r \cap T_r$,

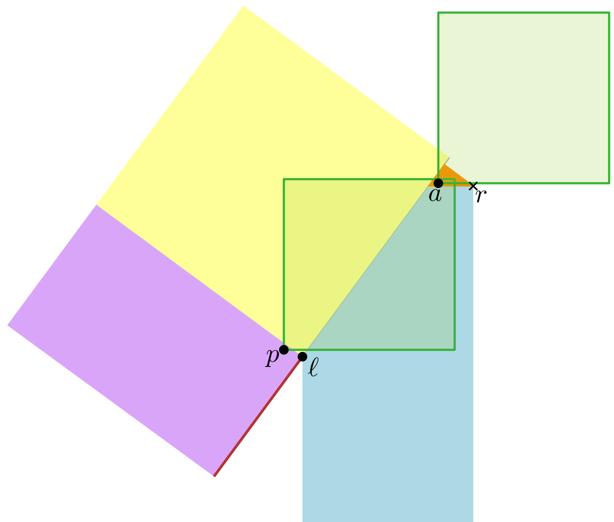


Figure 9: $S(\ell, u)$ is incompatible with any point $a \in \tilde{B}_\ell \cap \tilde{L}_r \cap T_r$ (the orange region).

with $r = (\ell_x + 1, \ell_y + 1)$, see Figure 9. Let p be a point in $S(\ell, u)$ whose label intersects λ_a. Using that $\theta \in (3\pi/2, 7\pi/4)$ it follows that p is to the bottom-left of a. Since the labels are to the top-right of a point, we then obtain that $a \in \lambda_p$. Therefore, the leader γ_a also intersects λ_p. So, both the label and the leader of a interfere with $S(\ell, u)$, and thus $a \in Q(\ell)$. It follows that $a \notin \mathcal{I}$. Contradiction. $\square$

Lemma 15. *Let p be a point in $S(\ell, u)$, and let $(\mathcal{I}, \mathcal{E}) \in \mathcal{U}^*_{\ell E}$. The label of p intersects at most one label λ_q of a point $q \in \mathcal{I}$.*

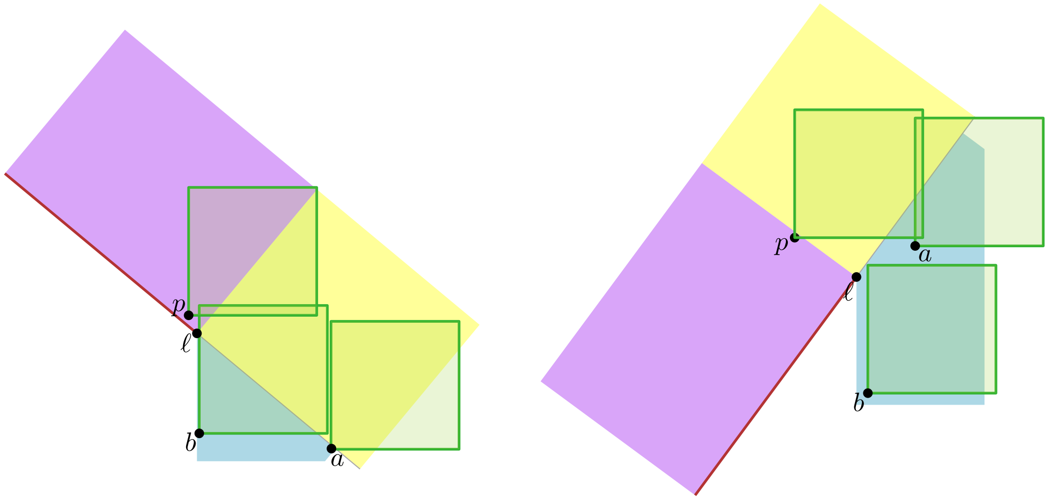
Proof. When $|\mathcal{I}| \leq 1$, the lemma is trivially true. By Lemma 14, we otherwise have $|\mathcal{I}| \leq 2$ and $\theta \in (0, \pi/4) \cup (3\pi/2, 2\pi)$. Let $\mathcal{I} = \{a, b\}$. For the case $\theta \in (0, \pi/4)$, assume w.l.o.g. that b is the leftmost point. Since a and b are both in $\mathcal{I}$, their labels are disjoint, and thus we have $b_x + 1 < a_x$ (see Figure 10(a)). Furthermore, we have $p_y \geq \ell_y \geq b_y$. Since λ_p intersects λ_b, but $p \notin \lambda_b$ this means $p_x < b_x$. Since λ_p has width one, it thus cannot intersect λ_a.

For the case $\theta \in (3\pi/2, 2\pi)$ we use a similar argument. Assume w.l.o.g. that a is the topmost point, and thus $a_y > b_y + 1$ (see Figure 10(b)). Since $p \in S(\ell, u)$, λ_p intersects λ_a, and $a \notin \lambda_p$ we have that p is to the top-left of a. And thus, $p_y > a_y > b_y + 1$. The label of b has height at most one, and thus cannot intersect λ_p. $\square$

Lemma 16. *Let $\mathcal{C} = (\{a, b\}, \mathcal{E}) \in \mathcal{U}^*_{\ell E}$, where a is to the right of b if $\theta \in (0, \pi/4)$, and to the top of b if $\theta \in (3\pi/2, 2\pi)$. Point b uniquely determines a.*

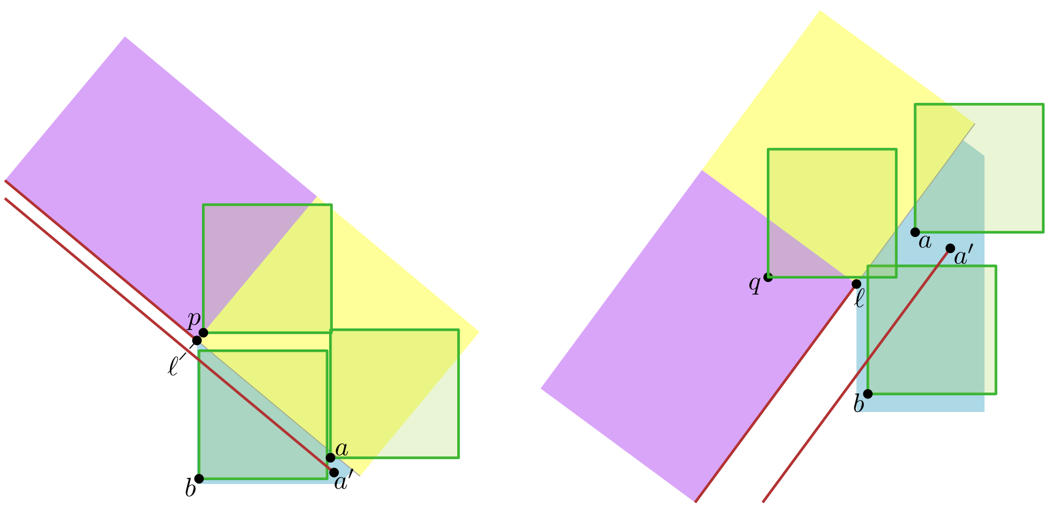
Proof. We prove this by contradiction. Assume that there is another configuration $\mathcal{C}' = (\{a', b\}, \mathcal{E}') \in \mathcal{U}^*_{\ell E}$, such that $a' \in E(\ell) \setminus Q(\ell)$ is to the right of b in case $\theta \in (0, \pi/4)$, or above b in case $\theta \in (3\pi/2, 2\pi)$.

The points a and b can influence the labeling of $S(\ell, u)$. Hence, their labels intersect with labels of points in $S(\ell, u)$. By Lemma 15 λ_a and λ_b cannot both intersect the same label of a point in $S(\ell, u)$. So, let p be the point whose label intersects λ_a, and let q be the point whose label intersects λ_b. See Figure 11.



(a) (b)

Figure 10: Point $p \in S(\ell, u)$ can intersect only one label of a point in $E(\ell) \setminus Q(\ell)$. (a) The case $\theta \in (0, \pi/4)$. (b) The case $\theta \in (3\pi/2, 2\pi)$.



(a) (b)

Figure 11: The points a, a', and b in $E(\ell)$ and the label(s) in $S(\ell, u)$ they intersect. (a) The case $\theta \in (0, \pi/4)$, note that this figure is not on scale. (b) The case $\theta \in (3\pi/2, 2\pi)$.

We start with the case $\theta \in (0, \pi/4)$. Point a is to the right of b, and λ_a and λ_b are disjoint. Hence, $a_x > b_x + 1$. Since λ_p intersects λ_a, and $p \notin \lambda_b$, it follows that p lies above λ_p. Hence $p_y > b_y + 1$. Again using that λ_p intersects λ_a, it then also follows that $a_y > b_y$. Using the same argument we get $a'_y > b_y$. Now in the configuration $\mathcal{C}$, point a' is labeled externally. However, this means its leader intersects λ_b. Hence, $\mathcal{C} \notin \mathcal{U}^*_{\ell E}$. Contradiction.

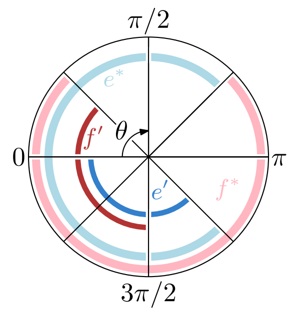


Figure 12: A depiction of the upper bounds on e^*, e', f^*, and f' as a function of θ. The marked circular arcs indicate the ranges of θ, where $e^*(\theta), e'(\theta), f^*(\theta)$, and $f'(\theta)$ take a value of 1; outside these ranges the value is 0.

For the case $\theta \in (3\pi/2, 2\pi)$ we have that $b_y + 1 < a_y$. As in the proof of Lemma 14 we have that $a_y < \ell_y + 1$, and hence $b_y < \ell_y$. Using that λ_q does not intersect γ_ℓ, we have $q_y > \ell_y$. Since $q \notin \lambda_b$ it follows that $q_x < b_x$. In configuration $\mathcal{C}'$, $\lambda_{a'}$ and λ_b are also pairwise disjoint. It then follows that a' is to the top-right of b. In the configuration $\mathcal{C} = (\{a, b\}, \mathcal{E})$, point a' is labeled externally. However, this means its leader intersects λ_b. Hence, $\mathcal{C} \notin \mathcal{U}^*_{\ell E}$. Contradiction. □

Corollary 17. *The number of configurations in $\mathcal{U}^*_{\ell E}$ is at most $O(n^{e'(\theta)} + \delta^{e^*(\theta)})$, where $e^*(\theta) = \min\{1, e(\theta)\}$.*

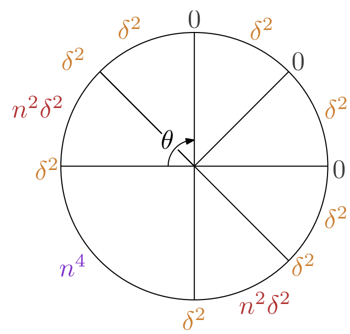
Note that Lemmas 15 and 16 also allow us to generate only the relevant configurations. It is easy to generate all configurations with only one internally labeled point in $E(p)$, to generate the configurations with two points labeled internally we consider the points in $E(\ell)$ from left to right when $\theta \in (0, \pi/4)$ and bottom to top if $\theta \in (3\pi/2, 2\pi)$. For each such point b, the other point $a \in E(\ell)$ labeled internally is the topmost point to the right (top) of λ_b. We can precompute all such pairs (a, b) using a priority range query.

We can again use a symmetric argument for the number of configurations in $\tilde{T}_u$. Figure 12 gives a graphical summary of these results. Similarly to $\iota(n, \delta, \theta)$ in Section 3.2 we now define $\iota^*(n, \delta, \theta)$ as

$$\begin{aligned}\iota^*(n,\delta,\theta) = \; & n^{2e'(\theta)+2f'(\theta)} && + n^{2e'(\theta)+f'(\theta)}\delta^{f^*(\theta)} && + n^{e'(\theta)+2f'(\theta)}\delta^{e^*(\theta)} \\ & + n^{e'(\theta)+f'(\theta)}\delta^{e^*(\theta)+f^*(\theta)} && + n^{2e'(\theta)}\delta^{2f^*(\theta)} && + n^{2f'(\theta)}\delta^{2e^*(\theta)} \\ & + n^{e'(\theta)}\delta^{e^*(\theta)+2f^*(\theta)} && + n^{f'(\theta)}\delta^{2e^*(\theta)+f^*(\theta)} && + \delta^{2e^*(\theta)+2f^*(\theta)}.\end{aligned}$$

which, by case distinction on the intervals of θ marked in Figure 12, solves to:

$$\iota^*(n,\delta,\theta) = \begin{cases} \delta^2 & \text{if } \theta = 0 \\ n^2\delta^2 & \text{if } \theta \in (0, \pi/4) \\ \delta^2 & \text{if } \theta \in [\pi/4, \pi/2) \\ 0 & \text{if } \theta = \pi/2 \\ \delta^2 & \text{if } \theta \in (\pi/2, 3\pi/4) \\ 0 & \text{if } \theta = 3\pi/4 \\ \delta^2 & \text{if } \theta \in (3\pi/4, \pi) \\ 0 & \text{if } \theta = \pi \\ \delta^2 & \text{if } \theta \in (\pi, 5\pi/4] \\ n^2\delta^2 & \text{if } \theta \in (5\pi/4, 3\pi/2) \\ \delta^2 & \text{if } \theta = 3\pi/2 \\ n^4 & \text{if } \theta \in (3\pi/2, 2\pi). \end{cases}$$



For $\delta = O(n)$, this improves the worst case running time to values that vary between $O(n^3 \log n)$ and $O(n^7)$, depending on the choice of θ. We thus obtain the following result:

Theorem 18. *Given a set $\mathcal{P}$ of n points and an angle θ, we can compute a labeling of $\mathcal{P}$ that maximizes the number of internal labels in $O(n^3(\log n + \iota^*(n,\delta,\theta)))$ time and $O(n^2\sqrt{\iota^*(n,\delta,\theta)})$ space, where $\delta = \min\{n, 1/d_{\min}\}$ for the minimum distance $d_{\min}$ in $\mathcal{P}$.*

Recall that for $\theta \in \{0, 3\pi/2\}$ Theorem 6 provides a better bound of $O(n^3(\log n + \delta))$ compared to the $O(n^3(\log n + \delta^2))$ bound of Theorem 18.

4 Extensions and limitations

So far, we have considered a stylized version of the question we set out to solve. In this section we discuss how our solution may be adapted and extended, depending on the exact requirements of the application.

4.1 Weighted points

Our approach directly extends to the situation in which the points have a weight, and we wish to maximize the total weight of the points labeled internally. Indeed, the leader of the "rightmost" point labeled externally still partitions the problem into two almost independent sub-problems as before, and hence we can solve the problem using dynamic programming. The only difference is that the recurrence Φ now represents the sum of the weights rather than the number of points labeled internally. The running time of our algorithm remains unaffected.

4.2 Optimizing the direction

Rather than fixing the direction for the leaders in advance, we may be willing to let the algorithm specify the optimal orientation that maximizes the number of points that can be labeled internally. Or, perhaps we wish to compute a chart that plots the maximum number of internally labeled points as a function of the leader orientation θ, leaving the final decision to the judgement of the user.

In both scenarios, we need to efficiently iterate over all possible orientations. We adapt our method straightforwardly. Let Q be the set of all $4n$ corner points of all potential labels. For every pair $p, q \in Q$ consider the slope $\theta_{p,q}$ of the line through p and q. All values $\theta_{p,q}$ partition all possible angles into $O(n^2)$ intervals. For all values θ in the same interval J, any leader γ_p intersects the same set of potential labels, so the optimal set of internal labels is constant throughout J. We compute it separately for each interval.

By applying Theorem 18, we achieve a total of $O(n^2 \cdot n^3(\log n + \iota(n, \delta, \theta))) = O(n^5(\log n + \iota(n, \delta, \theta)))$ time to compute the optimal labelings for all orientations, or to optimize the orientation by performing a simple linear scan.

4.3 Using multiple directions and placements

In this paper, we studied a labeling model where each label has two possible locations: internal or external. However, all internal labels are required to be in the same location (the top right of their points) and all external leaders are required to leave in the same direction. While using a restricted set of options may be justified from an aesthetic point of view, in practice, it would be more reasonable to allow multiple possible internal placements and/or external leader directions.

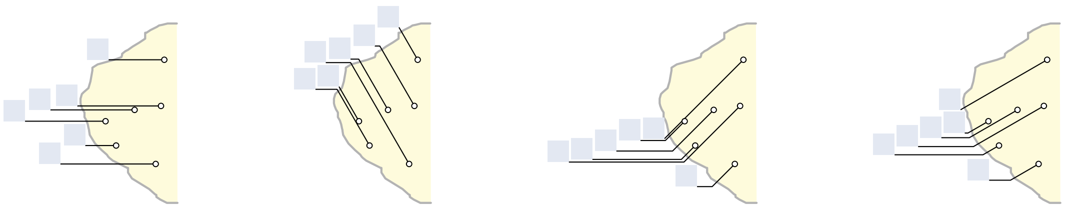
It is well-known that optimally placing internal labels when multiple locations for each label are available is NP-hard [10,23]. More specifically, it is NP-hard to optimize the number of internally placed labels, when we do not label the remaining points at all. This does not necessarily imply that the same problem is still NP-hard when the remaining points must be labeled externally, since the leaders this would require subdivide the problem. However, we do conjecture the problem is NP-hard in this case.

When allowing only one internal placement location, but multiple leader directions, the situation looks more promising. If we require leader directions to be clustered (when traversing the outer boundary of the map), as would be necessary for infinite leaders or for leader directions that are separated by more than $90°$, we believe our approach extends naturally: one could compute sub-problems for multiple directions, although care must be taken when considering the shapes of the combined E and F regions. When we do not require clustering, it is less clear how to extend the approach, but for a constant number of leader directions a dynamic programming approach still seems feasible.

For the most general version of the problem, with multiple possible internal placements and external leader directions, we believe a different approach is necessary. It would be very interesting to know if an approximation algorithm exists for this problem.

4.4 Routing the outer leaders

Once the core combinatorial problem of deciding which points have to be labeled internally is solved, it remains to route the outer leaders and place the external labels. Since our goal in this paper is to maximize the number of internally labeled points, we are only interested in finding a valid labeling, in which neither labels nor leaders intersect each other. Let us assume that $\theta \in [0, \pi/2] \cup [3\pi/2, 2\pi]$, i.e., all external labels are oriented to the left. The case of labels oriented to the right is symmetric. We consider the leaders in counterclockwise order around the boundary of $\mathcal{M}$ and place them one by one starting with the topmost leader. The first label is placed with its lower right corner anchored at the endpoint of its inner leader. For all subsequent labels we test if the label anchored at the endpoint of



(a) $\theta = 0$ (b) $\theta = \pi/3$ (c) $\theta = 7\pi/4$ (d) $\theta = 11\pi/6$

Figure 13: Examples for routing the outer leaders and placing the external labels for different slopes.

the inner leader intersects the previously placed label. If there is no intersection, we use that label position. Otherwise, we draw an outer leader extending horizontally to the left starting from the endpoint of the inner leader until the label can be placed without overlap. Obviously this algorithm takes only linear time. Figure 13 shows the resulting labelings for four different slopes. We note that depending on the slope θ of the inner leaders other methods for routing the outer leaders might yield more pleasing external labelings. This is, however, beyond the scope of this paper.

Moreover, the approach described above assumes that there is always sufficient space in the external part of the map to place all external labels and their leaders. This may lead to long chains of labels, e.g., as shown in Figure 13(c). In reality, there is a trade-off between the internal map space, i.e., the actual map $\mathcal{M}$ containing the feature points, and the external map space outside $\mathcal{M}$. The larger the bounding box of the union of $\mathcal{M}$ and all external labels, the smaller the scale of $\mathcal{M}$ (assuming that $\mathcal{M}$ and all labels are shown on a fixed-size display). It is clear, however, that if we bound the available external map space, an instance may become infeasible to solve, for some or even for all leader slopes θ. Thus, it is interesting to consider the optimization problem to place as many external labels as possible within a bounded external map space and for a given slope θ. In addition to selecting which labels to display, the task is to determine for each leader where to place its bend, which directly and indirectly influences the space usage. This is an interesting open algorithmic question in its own. Recall that the algorithm described above always places the bend on the boundary of $\mathcal{M}$. Optimizing over all $\theta \in [0, 2\pi)$ is an obvious extension, either as a secondary step after maximizing the number of internal labels (the topic of this paper) or in combination with it. One may also try to determine as a first step a promising slope or range of slopes for which leaders generally point towards larger areas of free space around $\mathcal{M}$.

4.5 Non-square labels

Square labels are not very realistic in most map-labeling applications. Their use is justified by the observation that if all labels are homothetic rectangles, we can scale the plane in one dimension to obtain square labels without otherwise changing the problem. Nonetheless, reality is not quite that simple, for two reasons: firstly, the scaling does alter inter-point distances, so if we wish to parameterize our solution by $d_{\min}$ we need to take this into account. Secondly, in real-world applications, labels may arguably have the same height, but not usually the same width.

If all labels are homothetic rectangles with a height of one and a width of w, scaling the plane by a factor $1/w$ in the horizontal direction potentially decreases the closest interpoint distance by the same factor. Now, the number of points in a unit-area region that do not contain each other's potential labels is bounded by δw, immediately yielding a result of $O(n^3(\log n + \iota(n, \delta w, \theta)))$ using exactly the same approach.

When all labels have equal heights but may have arbitrary widths, we conjecture that a variation of our approach will still work, but a careful analysis of the intricacies involved is required. If the labels may also have arbitrary heights the problem is open. It is unclear if there is a polynomial time solution in this case.

4.6 Obstacles

In this paper, we have considered only abstract point sets to be labeled, using leaders that are allowed to go anywhere, as long as they do not intersect any internal labels. While this is justified in some applications (e.g., in anatomical drawings, it is common practice to ignore the drawing when placing the leaders, as they are very thin and do not occlude any part of the drawing [13, 26]), in others this may be undesirable (in certain map styles, leaders may be confused for region boundaries or linear features). As a solution, we may identify a set of polygonal *obstacles* in the map, that cannot be intersected by leaders or internal labels.

In this setting, obviously not every input has a valid labeling: a point that lies inside an obstacle can never be labeled, or obstacles may surround points or force points into impossible configurations in more complex ways. Nonetheless, we can test whether an input has a valid labeling and if so, compute the labeling that maximizes the number of internal labels in polynomial time with our approach.

The main idea is to preprocess the input points in a similar way as in the beginning of Section 3. Whenever a point has a potential leader that intersects an obstacle, it must be labeled internally; similarly, whenever a point has a potential internal label that intersects an obstacle, it must be labeled externally. If we include such "forced" leaders or labels into our set of obstacles and apply this approach recursively, we will either find a contradiction or be left with a set of points whose potential leaders and potential internal labels do not intersect any obstacle, and we can apply our existing algorithm on this point set.

The same approach may be used for point sets that are not in general position: if we disallow leaders that pass through other points, they are forced to be labeled internally. Note that this again may result in situations where no valid labeling exists.

5 Concluding remarks

We studied algorithms for map labeling in a recently proposed mixed map labeling model. In this model, we have a map with a set of feature points, each of which should be labeled using either an internal label or an external label. Internal labels are placed directly on the map, near their feature points, and external labels are placed along the boundary of the map and connected to their feature points by a leader line. The labels are not allowed to intersect each other, or other leaders. We then wish to maximize the number of points labeled internally.

We presented an efficient algorithm for when the internal labels have the same size and a fixed position with respect to their feature points, and the leaders are all parallel to a given orientation $\theta \in [0, 2\pi)$. The running time of our algorithm depends on the number of points n, the "density" of the point set δ, the width of the labels w, and, perhaps surprisingly, the orientation θ. Furthermore, we showed how to find an orientation that maximizes the number of internal labels, and we investigated how the aspect ratio of the labels influences our algorithm.

Perhaps the most interesting question is whether our algorithms are applicable in practice. This would involve validating the model, , how many additional points can be labeled internally compared to an "internal labels only" model, checking whether or not the produced maps are visually pleasing, and testing whether or not the running time of our algorithm is acceptable. The running time of our algorithm is $O(n^3(\log n + \iota(n, \delta w, \theta)))$, where $\iota(n, \delta w, \theta)$ is a term measuring the "interference" between sub-problems. Expressed as a function of only n, the running time varies between $O(n^3 \log n)$ and $O(n^7)$. At first glance, this may seem prohibitively large. However, we expect that in most cases the interference between the sub-problems is small (as it seems likely that in most cases the number of points in the regions $E(\cdot)$. $F(\cdot)$, etc. is small). Therefore we expect the running times to be closer to $O(n^3 \log n)$ than to $O(n^7)$. Furthermore, in many applications, e.g., maps on small, mobile devices or thematic maps for newspapers and other public media, the number of labels to show is also relatively small, say 10–50. In such a setting, our algorithms may well prove themselves to be sufficiently fast in practice. It would be interesting to verify this. Additionally, our theoretic results show that the amount of interference heavily depends on the orientation θ. Another interesting question would be to check if this dependence also actually shows up in practice, or if the interference is more or less the same for every orientation.

There are also some theoretical questions remaining. For example, can we still compute an optimal mixed map labeling efficiently if the labels have different heights? It would also be interesting to see if we can reduce the running time for our algorithm that computes an optimal orientation θ. We currently compute a new labeling from scratch after each event. It might be possible to reuse the labeling computed before, as at any event only two points change order.

Acknowledgments

We would like to thank one of the reviewers for the observation that our algorithm extends to weighted points. Work on this paper was partially supported by the Netherlands Organisation for Scientific Research (NWO) under grant 639.021.123 and 612.001.022, respectively, and by the Danish National Research Foundation under grant nr. DNRF84.

References

[1] AGARWAL, P. K., VAN KREVELD, M., AND SURI, S. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications 11*, 3–4 (1998), 209–218. doi:10.1016/S0925-7721(98)00028-5.

[2] BARTH, L., GEMSA, A., NIEDERMANN, B., AND NÖLLENBURG, M. On the readability of boundary labeling. In *Graph Drawing (GD)* (2015), E. Di Giacomo and A. Lubiw,

Eds., Lecture Notes in Computer Science, Springer, pp. 515–527. doi:10.1007/978-3-319-27261-0_42.

[3] BEKOS, M., KAUFMANN, M., NÖLLENBURG, M., AND SYMVONIS, A. Boundary labeling with octilinear leaders. *Algorithmica 57* (2010), 436–461. doi:10.1007/s00453-009-9283-6.

[4] BEKOS, M., KAUFMANN, M., PAPADOPOULOS, D., AND SYMVONIS, A. Combining traditional map labeling with boundary labeling. In *Proc. 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)* (2011), vol. 6543 of *Lecture Notes in Computer Science*, pp. 111–122. doi:10.1007/978-3-642-18381-2_9.

[5] BEKOS, M. A., KAUFMANN, M., AND SYMVONIS, A. Efficient labeling of collinear sites. *Journal of Graph Algorithms and Applications 12*, 3 (2008), 357–380. doi:10.7155/jgaa.00170.

[6] BEKOS, M. A., KAUFMANN, M., SYMVONIS, A., AND WOLFF, A. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry: Theory and Applications. 36*, 3 (2007), 215–236. doi:10.1016/j.comgeo.2006.05.003.

[7] BENKERT, M., HAVERKORT, H., KROLL, M., AND NÖLLENBURG, M. Algorithms for multi-criteria boundary labeling. *Journal of Graph Algorithms and Applications 13*, 3 (2009), 289–317. doi:http://dx.doi.org/10.7155/jgaa.00189.

[8] CHALERMSOOK, P., AND CHUZHOY, J. Maximum independent set of rectangles. In *Discrete Algorithms (SODA)* (2009), pp. 892–901. doi:10.1137/1.9781611973068.97.

[9] DE BERG, M., VAN KREVELD, M., OVERMARS, M., AND SCHWARZKOPF, O. *Computational Geometry: Algorithms and Applications*, 2nd ed. Springer, Berlin, Germany, 2000.

[10] FORMANN, M., AND WAGNER, F. A packing problem with applications to lettering of maps. In *Computational Geometry (SoCG)* (1991), ACM, pp. 281–288. doi:10.1145/109648.109680.

[11] GEMSA, A., HAUNERT, J.-H., AND NÖLLENBURG, M. Boundary-labeling algorithms for panorama images. In *Advances in Geographic Information Systems (SIGSPATIAL)* (2011), ACM, pp. 289–298. doi:10.1145/2093973.2094012.

[12] GIORA, Y., AND KAPLAN, H. Optimal dynamic vertical ray shooting in rectilinear planar subdivisions. *ACM Transactions on Algorithms 5*, 3 (July 2009), 28:1–28:51. doi:10.1145/1541885.1541889.

[13] GRAY, H. *Anatomy of the human body*. Lea & Febiger, 1918.

[14] HARRIE, L., STIGMAR, H., KOIVULA, T., AND LEHTO, L. An algorithm for icon labelling on a real-time map. In *Spatial Data Handling (SDH)* (2005), Springer Berlin Heidelberg, pp. 493–507. doi:10.1007/3-540-26772-7_38.

[15] HUANG, Z.-D., POON, S.-H., AND LIN, C.-C. Boundary labeling with flexible label positions. In *Algorithms and Computation (WALCOM)* (2014), vol. 8344 of *Lecture Notes in Computer Science*, Springer, pp. 44–55. doi:10.1007/978-3-319-04657-0_7.

[16] IMHOF, E. Positioning names on maps. *The American Cartographer 2*, 2 (1975), 128–144.

[17] KAUFMANN, M. On map labeling with leaders. In *Festschrift Mehlhorn*, S. Albers, H. Alt, and S. Näher, Eds., vol. 5760 of *Lecture Notes in Computer Science*. Springer, 2009, pp. 290–304. doi:10.1007/978-3-642-03456-5_20.

[18] KINDERMANN, P., LIPP, F., AND WOLFF, A. Boundary labeling for annotations in texts. In *Graph Drawing (GD)* (2014), C. A. Duncan and A. Symvonis, Eds., vol. 8871 of *Lecture Notes in Computer Science*, Springer, pp. 76–88. doi:10.1007/978-3-662-45803-7_7.

[19] KINDERMANN, P., NIEDERMANN, B., RUTTER, I., SCHAEFER, M., SCHULZ, A., AND WOLFF, A. Two-sided boundary labeling with adjacent sides. In *Algorithms and Data Structures (WADS)* (2013), vol. 8037 of *Lecture Notes in Computer Science*, pp. 463–474. doi:10.1007/978-3-642-40104-6_40.

[20] KLAU, G. W., AND MUTZEL, P. Optimal labeling of point features in rectangular labeling models. *Mathematical Programming 94*, 2 (2003), 435–458. doi:10.1007/s10107-002-0327-9.

[21] LÖFFLER, M., AND NÖLLENBURG, M. Shooting bricks with orthogonal laser beams: A first step towards internal/external map labeling. In *Proc. Canadian Conference Computational Geometry (CCCG)* (2010), University of Manitoba, pp. 203–206.

[22] LÖFFLER, M., NÖLLENBURG, M., AND STAALS, F. Mixed map labeling. In *Proc. 9th International Conference on Algorithms and Complexity*, Lecture Notes in Computer Science. Springer, 2015, pp. 339–351. doi:10.1007/978-3-319-18173-8_25.

[23] MARKS, J., AND SHIEBER, S. The computational complexity of cartographic label placement. Tech. rep., Harvard University, 1991.

[24] MEHLHORN, K., AND NÄHER, S. Dynamic fractional cascading. *Algorithmica 5*, 1–4 (1990), 215–241. doi:10.1007/BF01840386.

[25] NÖLLENBURG, M., POLISHCHUK, V., AND SYSIKASKI, M. Dynamic one-sided boundary labeling. In *Advances in Geographic Information Systems (SIGSPATIAL)* (2010), pp. 310–319. doi:10.1145/1869790.1869834.

[26] PAULSEN, F., AND WASCHKE, J. *Sobotta, Atlas of Human Anatomy*, 15th ed. Urban & Fischer, 2011.

[27] RYLOV, M. A., AND REIMER, A. W. A comprehensive multi-criteria model for high cartographic quality point-feature label placement. *Cartographica 49*, 1 (2014), 52–68. doi:10.3138/carto.49.1.2137.

[28] VAN KREVELD, M., STRIJK, T., AND WOLFF, A. Point labeling with sliding labels. *Computational Geometry: Theory and Applications 13*, 1 (1999), 21–47. doi:10.1016/S0925-7721(99)00005-X.

[29] WOLFF, A., AND STRIJK, T. The map labeling bibliography. http://i11www.iti.kit.edu/~awolff/map-labeling/bibliography/.