Area-Feature Boundary Labeling

MICHAEL A. BEKOS¹, MICHAEL KAUFMANN², KATERINA POTIKÀ¹, ANTONIOS SYMVONIS¹

¹National Technical University of Athens, School of Applied Mathematical & Physical Sciences, 15780 Zografou, Athens, Greece
Email: mikebekos@math.ntua.gr, epotik@cs.ntua.gr, symvonis@math.ntua.gr

²University of Tübingen, Institute for Informatics, Sand 13, 72076 Tübingen, Germany
Email: mk@informatik.uni-tuebingen.de

Boundary labeling is a relatively new labeling method. It can be useful in automating the production of technical drawings and medical maps, where it is common to explain certain parts of the drawing with text labels, arranged on its boundary so that other parts of the drawing are not obscured.

In boundary labeling, we are given a rectangle $R$ which encloses a set of $n$ sites. Each site $s_i$ is associated with an axis-parallel rectangular label $l_i$. The labels must be placed in distinct positions on the boundary of $R$ and to be connected to their corresponding sites with polygonal lines, called leaders, so that the labels are pairwise disjoint and the leaders do not intersect each other.

In this paper, we study a version of the boundary labeling problem where the sites can “float” within a polygonal region. We present a polynomial time algorithm that produces a labeling of minimum total leader length for labels of uniform size placed in fixed positions on the boundary of $R$.

Received January 2008; revised ...
of $R$. We conclude in Section 5 with open problems and future work.

2. PROBLEM DEFINITION

In boundary labeling, we are given a set $P$ of $n$ sites $s_i, i = 1, 2, \ldots, n$, each associated with a rectangular label $l_i$ of dimensions $w_i \times h_i$. The site set $P$ and the underlying drawing (map) are enclosed in an axis-parallel rectangle $R$ of sufficient size, which is called enclosing rectangle. The labels should be placed on distinct positions on the boundary of $R$ so that they do not overlap, and should be connected to their corresponding sites with non-intersecting polygonal lines, called leaders. Such labelings are referred to as legal or crossing free boundary labelings.

Given that several parameters (sites, labels, leaders, enclosing rectangle) are involved in boundary labeling, there exist several variations of boundary labeling, each giving rise to a different labeling model.

The sites model features of the drawing. In the simplest form of the problem, they model point locations on a map, e.g. a city center, or the capital of a prefecture. In this case, each site $s_i$ is associated with a point $p_i = (x_i, y_i)$ of the plane (see Figure 1 or Figures 3a and 3b). To avoid leader overlaps, which would result in confusing situations, we make an additional assumption regarding the location of the sites: We assume that the sites are placed in general position, i.e. no three sites are collinear and no two sites share the same $x$- or $y$-coordinate.

As mentioned in Section 1, in practice we often want to label area features, e.g. a region of a map. In “area-feature boundary labeling” we are given as part of the input a region in which a point-site has to be selected to represent the region and to be connected to its corresponding label. In order to be consistent with the terminology used in map labeling, we refer to these regions as area-sites, or simply as sites when the context is clear. We study the cases where the area-sites are either generalized canonical polygons or rectangles or line segments. We call generalized canonical polygon or GC-POLYGON, a simple closed polygon whose edges are vertical, horizontal or diagonal (at angles which are multiples of 45 degrees with respect to the axes). Figure 3c shows an example where the area-sites are represented by GC-POLYGONS.

Sites are connected with their corresponding labels with non-intersecting polygonal lines, which are called leaders. We denote by $c_i$ the leader of site $s_i$ (see Figure 4). Since we aim at simple and easy to visualize labelings, we only consider leaders that consist of either a single straight line segment or a sequence of axis-parallel segments either parallel (p) or orthogonal (o) to the side of $R$ containing the label it leads to. The type of a leader is defined by an alternating string over the alphabet $\{p, o\}$. In our approach, we use leaders of type-po and opo:

Type-po leaders: Leaders of type po consist of two line segments. The first one is parallel (p) to the side of $R$ containing the label it leads to, whereas the second one is orthogonal (o) to that side (see Figure 3b). Degenerated case of a po-leader is a leader of type o, which consists of only one line segment orthogonal to side of $R$ containing the label it leads to.

Type-o leadership: Following the same notation scheme, leaders of type opo consist of three line segments (see Figure 3a). For each type opo leader, we further assume that it has its parallel $p$-segment outside the enclosing rectangle $R$, routed in the so-called track routing area. Again, leaders of type o are trivially considered to be of type opo, as well.
Each leader that connects a site to a label, touches the label on a point on its side that faces the enclosing rectangle. The point where the leader touches the label is called label-port. We can assume either fixed ports where the leader is only allowed to use a fixed set of ports on the label side (a typical case is where the leader uses the middle point of the label side; see Figures 3a and 3b) or sliding ports where the leader can touch any point of the label’s side (see Figure 3c).

The labels are placed on the boundary of an axis-parallel rectangle $R = [l_R, r_R] \times [b_R, t_R]$ of height $H = t_R - b_R$ and width $W = r_R - l_R$. In general, we assume that the labels are of arbitrary size (non-uniform labels), i.e. label $l_i$ which corresponds to site $s_i$ has height $h_i$ and width $w_i$. However, it is reasonable to separately consider the restricted cases where the labels are of uniform size, or of maximum uniform size. Figures 3a and 3c display labelings with labels of uniform size, whereas Figure 3b displays a labeling with non-uniform labels.

Several versions of the boundary labeling problem are hard to solve. To realize that, assume that labels of variable height must be placed either to the left or right side of the enclosing rectangle $R$, and that the label heights sum up to twice the height $H$ of $R$. It is clear that the task of assigning the labels to the two sides corresponds to the PARTITION problem, which is a well known NP-complete problem.

Our aim is to obtain legal labelings that optimize some criterion. Keeping in mind that we want to obtain simple and easy to visualize labelings, the criterion of minimizing the total leader length can be adopted from the areas of VLSI and Graph Drawing.

2.1. Previous Work

The first results on boundary labeling were presented by Bekos, Kaufmann, Symvonis and Wolff in [4]. A variety of models based on the type of the leaders, the location of the labels and the size of the labels were studied. The focus of their work was on efficient algorithms for minimizing the total leader length and for minimizing the total number of leader bends. In subsequent work Bekos et al. study variations of boundary labeling, where the labels are arranged in multiple stacks on one side of the rectangle [1] or where the sites to be labeled are collinear [3].

Benkert and Nöllenburg [6] presented algorithms for minimizing the total leader length with type-po or type-do leaders, when uniform labels are allowed to be placed on one side of the enclosing rectangle. Recently, Benkert, Haverkort, Kroll and Nöllenburg [5] studied boundary labeling along a new line of research. They formulated the problem as an optimization problem, where the objective function is a general quality function which evaluates the niceness of the resulting labeling. Then, using dynamic programming presented several results for the case where the labels are of uniform size, placed on one side of the enclosing rectangle and the leaders are either of type-po or of type-do.

Kao, Lin and Yen [9] introduced the Many-to-One boundary labeling to describe a variation of boundary labeling, where several sites are associated with a common label. In the case of Many-to-One boundary labeling, the presence of crossings among leaders often becomes inevitable. Therefore, they presented several algorithms, approximations and heuristics for minimizing the total number of crossings.

A preliminary version of this paper has appeared in [2].

2.2. Notation and Terminology

In this section, we present some necessary notation and definitions that are heavily used in the remaining of the paper. We denote the number of area-sites (and consequently the number of labels) by $n$ and the maximum number of corners of each area-site by $k$, where $k$ is a constant.

GC-POLYGON representation: A GC-POLYGON is a set of at most $k$ points (of integer coordinates), referred to as corners, indexed $\{1, \ldots, k\}$ in clockwise order.
direction that define its boundary. In Figure 4, GC-POLYGON $s_i$ has 4 corners while GC-POLYGON $s_j$ has 5 corners.

**GC-POLYGENS in general position**: We extend the notion of general position for sites represented as points to GC-POLYGENS as follows: We say that GC-POLYGENS $s_1, s_2, \ldots, s_n$ are in general position if for each pair of indices $i, j$ with $i \neq j$,

i) there do not exist two corners belonging to GC-POLYGEN $s_i$ and $s_j$ with the same $x$- or $y$-coordinate

ii) there do not exist two corners belonging to GC-POLYGEN $s_i$ and label $l_j$ with the same $x$- or $y$-coordinate

**Site-ports**: Each leader that connects an area-site to a label, touches the boundary of $s_i$ on a point. This point is referred to as *port of* area-site $s_i$, or simply as *site-port of* $s_i$, and is denoted by $p_{si}$. In Figure 4, the ports of both area-sites $s_i$ and $s_j$ coincide with some corner of their corresponding GC-POLYGENs.

**Leader orientation**: Consider a type-*opo* leader $c_i$, which originates from site-port $p_{si}$ of a GC-POLYGEN $s_i$ and is connected to a label on the right side $AB$ of the enclosing rectangle $R$. The line which contains the first $o$-segment of the leader $c_i$ (i.e. the one which is incident to $p_{si}$) divides the plane into two half-planes (see the dashed line $l$ of Figure 4). We say that leader $c_i$ is oriented towards corner $A$ of the rectangle $R$ if both corner $A$ and the label-port of $s_i$ are on the same half-plane, otherwise, we say that leader $c_i$ is oriented away from corner $A$ (see Figure 4). In the case of type-*$o$ leader, we consider the leader to be oriented towards corner $A$ (and also towards corner $B$).

3. **FOUR-SIDE LABELING OF GC-POLYGENS WITH TYPE-OPO LEADERS**

We consider the more general case of boundary labeling with area-sites represented as GC-POLYGENS. We assume that we have fixed labels of uniform size, placed on all four sides of rectangle $R$, and type-opo leaders. We present a polynomial time algorithm, that returns a legal labeling of minimum total leader length.

Let $P = \{s_1, s_2, \ldots, s_n\}$ be the set of GC-POLYGENS and $L = \{l_1, l_2, \ldots, l_n\}$ be the set of labels. Since the labels have uniform size, each area-site $s_i$ can be connected to any label $l_j$. We seek to connect each area-site $s_i$ to a label $l_j$ and to specify two points one on the periphery of $s_i$ (site-port of $s_i$) and one on the periphery of $l_j$ (label-port of $l_j$), so that the total leader length is minimized. Algorithm 1 outlines our approach.

**Algorithm 1**: 4SIDE-AREA-OPO

**input**: A set $P = \{s_1, \ldots, s_n\}$ of $n$ GC-POLYGENS on the plane and a set $L = \{l_1, \ldots, l_n\}$ of $n$ uniform labels placed on the boundary of $R$.

**output**: A crossing free four-side type-*opo* labeling of minimum total leader length.

**Step A. Shortest Leader Computation**:

Construct a complete weighted bipartite graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$. The weight $w(e_{ij})$ of an edge $e_{ij} = (s_i, l_j) \in E$ is the Manhattan length of the shortest (under the Manhattan metric) leader, say $d_{ij}$, which connects $s_i$ with $l_j$.

**Step B. Compute Minimum Cost Bipartite Matching**:

Proceed by computing a minimum-cost perfect bipartite matching $\mathcal{M}$ on $G$, i.e. compute a matching between area-sites and labels that minimizes the total Manhattan distance of the matched pairs.

**Step C. Obtain a labeling $M$ as follows**:

If an edge $e_{ij} = (s_i, l_j) \in E$ is selected in $\mathcal{M}$ then connect area-site $s_i$ to label $l_j$ with a leader of weight $w(e_{ij})$.

**Step D. Eliminate crossings**:

Eliminate all crossings of leaders and obtain a crossing free labeling $M'$, keeping the total leader length unchanged, i.e. equal to that of $\mathcal{M}$.

Initially, we construct a complete weighted bipartite graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$, where $P = \{s_1, s_2, \ldots, s_n\}$, $L = \{l_1, l_2, \ldots, l_n\}$, $E = \{(s_i, l_j); s_i \in P, l_j \in L\}$ and $w: E \rightarrow \mathbb{N}$ is a cost function (see step A of Algorithm 1). Each edge $e_{ij} = (s_i, l_j) \in E$ of $G$ is assigned a weight $w(e_{ij}) = d_{ij}$, where $d_{ij}$ is equal to the Manhattan length of the shortest (under the Manhattan metric) leader which connects area-site $s_i$ with label $l_j$. We proceed by computing a minimum-cost bipartite matching on $G$, i.e. we compute a matching between area-sites and...
3.1. Shortest Leader Computation

In Step A of Algorithm 1, we have to compute the minimum Manhattan distance between every area-site \( s_i \) in \( P \) and every label \( l_j \) in \( L \). This is equivalent to computing the shortest type-\( \text{o-p} \) leader which connects area-site \( s \) with label \( l \), for all pairs \((s,l)\) where \( s \in P \) and \( l \in L \). As illustrated in Figure 5, the definition of the shortest leader which connects area-site \( s \) with label \( l \) is not unique. This is because of the definition of \( \text{GC-POLYGONS} \). Therefore, there might exist several leaders of minimum leader length connecting \( s \) with \( l \).

**Lemma 1.** Let \( d \) be the length of the shortest (under the Manhattan metric) leader which connects area-site \( s \) with label \( l \). Then, there is a leader \( c \) of length \( d \) which connects \( s \) with \( l \) for which one of the following statements hold:

i) Leader \( c \) is of type-\( \text{o-p} \) originating from some corner of area-site \( s \) and leading to some corner of label \( l \).

ii) Leader \( c \) is of type-\( \text{o-p} \) and either originates from some corner of area-site \( s \) or leads to some corner of label \( l \).

**Proof.** Without loss of generality, we assume that label \( l \) is placed at the right side of rectangle \( R \). To prove Lemma 1, we have to consider all cases regarding the relative positions of area-site \( s \) and label \( l \).

i) In order to prove statement (i), we consider the case where all corners of area-site \( s \) are above the bottom-left corner of label \( l \) (see Figure 6). The case where all corners of \( s \) are below the bottom-left corner of \( l \) can be treated similarly. In this case, the leader which connects \( s \) with \( l \) must be of type-\( \text{o-p} \). So, it remains to show that it originates from some corner of area-site \( s \) and leads to some corner of label \( l \). The latter case is implied immediately since the leader which will be used should touch label \( l \) to its top left corner. Consider now the case where the bottommost corner of area-site \( s \) is unique (see Figure 6a). Then, because of the definition of \( \text{GC-POLYGONS} \), the leader which connects area-site \( s \) with label \( l \) should originate from that corner. To complete the proof of statement (i) it remains to consider the case where the bottommost corner of area-site \( s \) is not unique, i.e. there are several corners of \( s \) with the same \( x \)-coordinate (see Figure 6b). Similarly, in this case, the leader which connects area-site \( s \) with label \( l \) should originate from the rightmost such corner.

ii) In order to prove statement (ii), we distinguish the following cases:
The Computer Journal Vol. 00 No. 0, 2008

M.A. Bekos, M. Kaufmann, K. Potika, A. Symvonis

Therefore, since we assumed that the area-sites are in general position, a computation of such leaders yields into a solution containing no overlapping leaders.

Algorithm 2 outlines our method for computing the shortest type-opo leader which connects s with l, for all pairs (s, l) where s ∈ P and l ∈ L. Initially, for each area-site sᵢ we construct a set sᵢ婆 which contains all candidate site-ports of sᵢ. Similarly, for each label lⱼ we construct a set lⱼ婆 which contains all candidate label-ports of lⱼ (see Step A of Algorithm 2). We proceed by computing the Voronoi diagram Hᵢ (see [7, pp.158], [12]) under the Manhattan distance for each set sᵢ婆. To compute the shortest (under the Manhattan metric) leader which connects area-site sᵢ with label lⱼ we utilize the notion of the nearest neighbor in Hᵢ. More precisely, for each point in lⱼ婆 we determine its nearest neighbor in Hᵢ and we compute their Manhattan distance. Then, the shortest leader which connects area-site sᵢ with label lⱼ corresponds to the one which minimizes the distances computed above and we set w(eᵢⱼ) to be this distance.

THEOREM 3.1. Algorithm 2 computes the minimum distance under the Manhattan metric between any label and any polygon, when the labels are placed in fixed positions on all four sides of rectangle R in $O(n^2 \log n)$ time.

Proof. Each label lⱼ contributes at most 2k elements to each set sᵢ婆, where k is the maximum number of corners of each area-site. From those we only keep the two “closest” ones. Therefore, the number of elements in each set sᵢ婆 is $O(k + n)$. Similarly, each area-site sᵢ contributes at most k elements at each set lⱼ婆 which implies that the number of elements in each set lⱼ婆 is $O(kn)$.

The construction of the Voronoi diagram Hᵢ of Step C.1 of Algorithm 2 can be done in $O(k' \log k')$ time [12], where $k' = O(k + n)$. Finding the nearest neighbor of a point q in the Voronoi diagram Hᵢ costs $O(\log k')$ [12]. Therefore, we compute Step C.2 in a total of $O(kn \log k')$ time. Then, the running time of Algorithm 2 is $O(n(k' + kn) \log k')$ and since we assume that k is constant the running time of Algorithm 2 is $O(n^2 \log n)$.

In Step B of Algorithm 1, we have to compute a minimum-cost perfect bipartite matching on the output graph of Algorithm 2. This problem is also known as assignment problem and can be efficiently solved by means of the Hungarian method in $O(n^3)$ time [10], [11], [14].

3.2. Crossings Elimination

In this section, we describe how to eliminate all crossings from labeling M (obtained from Step C of Algorithm 1). Note that labeling M is of minimum total leader length and we will eliminate all crossings keeping the total leader length unchanged, i.e. optimal.
Algorithm 2: Shortest Leader Computation.

**input**: A set $P = \{s_1, \ldots, s_n\}$ of $n$ GC-polygons on the plane and a set $L = \{l_1, \ldots, l_n\}$ of $n$ uniform labels placed on the boundary of $R$.

**output**: A complete weighted bipartite graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$, where the weight $w(e_{ij})$ of edge $e_{ij} = (s_i, l_j) \in E$ is the length of the shortest (under the Manhattan metric) leader, which connects $s_i$ with $l_j$.

**Step A**: Initiate output graph.
Construct a graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$, where the weight $w(e_{ij})$ of an edge $e_{ij} = (s_i, l_j) \in E$ is initially equal to zero.

**Step B**: Determine all possible site and label-ports.
for each area-site $s_i$ ($1 \leq i \leq n$) do
  Add the corners of $s_i$ to $s^P_i$.
for each label $l_j$ ($1 \leq j \leq n$) do
  if ($l_j$ is located to the right or left side of $R$) then
    Find the intersection points of each edge of $s_i$ with the horizontal lines passing from each corner of label $l_j$ that faces $R$ and in each case select the closest one.
  else
    Find the intersection points of each edge of $s_i$ with the vertical lines passing from each corner of label $l_j$ that faces $R$ and in each case select the closest one.
  end
  Add these points to $s^P_i$.
for each label $l_j$ ($1 \leq j \leq n$) do
  Add the corners of $l_j$ to $l^P_j$.
for each area-site $s_i$ ($1 \leq i \leq n$) do
  Find the intersection points of each edge of $l_j$ that faces $R$ with the perpendicular to it lines passing from each corner of area-site $s_i$. Add these points to $l^P_j$.

**Step C**: Shortest leader computation.
for each area-site $s_i$ ($1 \leq i \leq n$) do
  1. Construct the Voronoi diagram $H_i$ (under the Manhattan distance) for $s^P_i$.
  2. for each element of $l^P_j$ do
     Find the nearest neighbor in $H_i$ and compute their Manhattan distance. Set $w(e_{ij})$ to be the minimum such distance.

**Lemma 2.** Let $\mathcal{M}$ be an opo-labeling (which might contain crossings) obtained from Step C of Algorithm 1. Let $c_i$ and $c_j$ be a pair of intersecting leaders originating from area-sites $s_i$ and $s_j$, respectively. Then, the following hold:
   i) The labels $l_i$ and $l_j$ of these leaders lie on two adjacent sides of the enclosing rectangle $R$. Let $A$ be their incident corner.
   ii) Leader $c_i$ and $c_j$ are oriented towards corner $A$ of the enclosing rectangle $R$.
   iii) Leaders $c_i$ and $c_j$ can be rerouted so that they do not cross each other and the sum of their leader length remains unchanged.

**Proof.**

**Proof of statement (i):** To prove statement (i), we show that labels $l_i$ and $l_j$ can not both lie on the same or opposite sides of $R$. For the sake of contradiction, assume first that the labels $l_i$ and $l_j$ lie on the same side, say the right side, and the leaders $c_i$ and $c_j$ intersect. Then the intersection takes place outside rectangle $R$ (in the track routing area; see Figure 8a). By swapping the labels to which each area-site is connected, we can reduce the total leader length and also eliminate the crossing (see Figure 8b), a contradiction since we assumed that the total leader length of the labeling is minimum.

We now consider the case where the labels $l_i$ and $l_j$ lie on opposite sides of rectangle $R$. Then, since the leaders intersect each other, the segments of the leaders which are inside the rectangle (and incident to the area-sites) have to intersect. However, since these segments are parallel to each other, they have to overlap. Again, by swapping the labels to which each area-site is connected, we can reduce the total leader length (and also eliminate the overlapping), a contradiction since we assumed that the total leader length of the labeling is minimum (see an example in Figure 9).
Proof of statement (ii): Let $A$ be the corner which is incident to the two sides of the rectangle $R$ containing the labels associated with leaders $c_i$ and $c_j$. In order to show that in a labeling of minimum total leader length both leaders $c_i$ and $c_j$ are oriented towards corner $A$, it is enough to show that (in a labeling of minimum total leader length) it is impossible to have one or both leaders oriented away of corner $A$. We proceed to consider these two cases:

Case (a): *Exactly one leader, say $c_j$, is oriented away of corner $A*. This case is described in the left part of Figure 10a. Rerouting both leaders $c_i$ and $c_j$ as in Figure 10a results in a reduction of the total leader length, a contradiction since we assumed that the total leader length of the labeling is minimum.

Case (b): *Both leaders $c_i$ and $c_j$ are oriented away of corner $A*. This case is described in the left part of Figure 10b. When both leaders are oriented away of corner $A$, rerouting both leader $c_i$ and $c_j$ results in higher reduction of the total leader length (compared to Case (a) where only one leader was oriented away of corner $A$). The rerouting of the leaders is depicted in Figure 10b.

Having eliminated cases (a) and (b), where one or both crossing leaders are oriented away of corner $A$, the only case left is the one where both leaders $c_i$ and $c_j$ are oriented towards corner $A$. Such a case is depicted in Figure 10c.

Proof of statement (iii): In order to show that leaders $c_i$ and $c_j$ can be rerouted so that they do not cross each other and the sum of their leader lengths remains unchanged, we partition the first segment of each leader $c_i$ and $c_j$ into two sub-segments from their crossing point to the sides of the enclosing rectangle (see Figure 10c). Then, obtain the new leaders $c_i'$ and $c_j'$ by a sliding the (sub)segments of leaders $c_i$ and $c_j$, leaving their sum unchanged.

In the following, we show that given a labeling of minimum total leader length which may contain crossings of those described in Lemma 2, we can efficiently resolve all crossings yielding to a new crossing free labeling so that the total leader length is unchanged.

**Theorem 3.2.** Let $M$ be an o-po-labelling (which might contain crossings) obtained from Step C of Algorithm 1. We can always identify a crossing-free o-po-labelling $M'$ with total leader length equal to that of $M$ (Step D of Algorithm 1). Moreover, labeling $M'$ can be obtained in $O(n \log n)$ time.

**Proof.** We show how to eliminate all crossings of labeling $M$ by rerouting the crossing leaders. Our method performs two passes over the area-sites, one in the left-to-right and one in the right-to-left direction.

Consider first the left-to-right pass. In the left-to-right pass of labeling $M$, we consider all area-sites with labels on the right side of the rectangle. We examine the area-sites from left-to-right and we are interested only on those who have crossing leaders. Let $s_i$ be the leftmost such area-site and let $c_i$ be the leader that connects it with its corresponding label on the right side of the rectangle (see the left drawing of Figure 11). Lemma 2.i implies that leader $c_i$ intersects only with leaders that are connected with labels either on the top or bottom side of rectangle $R$. Without loss of generality, assume that $c_i$ is oriented towards the bottom-right corner of the rectangle, say $A$. Then all leaders that intersect $c_i$ have their labels on the bottom side of $R$ and are also oriented towards $A$ (by Lemma 2.ii). Let $c_k$ be the leftmost leader that intersects $c_i$, and let $s_k$ be its incident area-site. According to Lemma 2.ii, we can reroute leaders $c_i$ and $c_k$ so that the total leader length remains unchanged (see the right drawing of Figure 11). Note that the rerouting possibly eliminates more than one crossing but, in general, it might also introduce new crossings. However, the total number of crossings is reduced and, more importantly, the leftmost area-site incident to an intersecting leader connected to a label on the right side of the rectangle is located to the right of area-site $s_i$. Continuing in the same manner, the leftmost area-site which participates in a crossing (in the left-to-right pass) is pushed to the right, which guarantees that all “left-to-right” crossings are eventually eliminated.

---

Note that the sorting of the area-sites is based on the coordinates of their ports.

This is because the new area that contains the area-sites with leaders that cross $c_i'$ (refer to the gray colored rectangle of the left part of Figure 11) is fully contained in the area containing the area-sites with leaders intersecting $c_i$ (refer to the gray colored rectangle of the left part of Figure 11).
Also note that it is impossible to introduce new “right-to-left” crossings, when the left-to-right pass is executed. To see this, assume that such a crossing was introduced and that it involves leader $c_i$ and the leader $c$ which connects a area-site $s$ to a label on the left side of the rectangle (see Figure 12). By Lemma 2.i both leaders $c'_i$ and $c'$ must be oriented towards corner $B$, a contradiction since leader $c'_i$ is oriented away of corner $B$ (and towards corner $A$).

From the above discussion, it follows that a left-to-right pass eliminating crossings involving leaders with their associated labels on the right side of the rectangle, followed by a similar right-to-left pass, results to a labeling $M'$ without any crossings and of total leader length equal to that of $M$, that is, minimum. Additionally, since we always keep the position of both site and label-ports unchanged the implied labeling will not contain overlapping leaders. This follows from
Lemma 1 and the assumption that the area-sites are in general position.

To complete the proof of the theorem, it remains to explain how to obtain in $O(n \log n)$ time the new labeling $M'$, given labeling $M$ of minimum total leader length. Consider the left-to-right pass. The analysis for the right-to-left pass is similar. During the pass, we process the area-sites with labels on the right side of the enclosing rectangle in order of increasing $x$-coordinate. Sorting the area-sites in increasing order with respect to the $x$-coordinate of their corresponding site-ports can be done in $O(n \log n)$ time.

In order to process a new area-site $s_i$ and to eliminate the crossings involving its leader $c_i$, we have to identify the leftmost area-site $s_k$ such that its corresponding leader $c_k$ intersects leader $c_i$. Of course, the intersection involves the first $\sigma$-segment of leader $c_i$. Without loss of generality, assume that leader $c_i$ is oriented towards the bottom-right corner of the enclosing rectangle. The case where it is oriented towards the top-right corner can be treated similarly. Then, according to Lemma 2(ii) all leaders intersecting leader $c_i$ are also oriented towards the bottom-right corner and, thus, their associated labels are placed on the bottom side of the enclosing rectangle. So, leader $c_i$ can only intersect vertical line segments which have one of their end-points on the bottom side of the enclosing rectangle. This implies that we can relax the restriction that these segments are of finite size and assume that they are semi-infinite rays having their associated site-ports as their highest endpoints. This is due to the fact that all leader intersections take place inside the enclosing rectangle.

Under this assumption, the processing of the area-sites during the left-to-right pass can be accomplished by employing a data structure supporting queries of the form “given a set of points $Q$ that change under insertions and deletions, a threshold value $y_0$ and a query range $(l, r)$, return the point of $Q$ with the smallest $x$-coordinate that is located within the rectangle $(l, r) \times (y_0, t_R)^5$.” The MinXInRectangle query just described can be answered in time $O(\log n)$ time by employing a dynamic priority search tree based on half-balanced trees [13, pp 209]. The insert (for initialization) and delete operations cost $O(\log n)$ time. This results to a total of $O(n \log n)$ time for the left-to-right pass and, consequently, for the elimination of all crossings.

**Theorem 3.3.** Given a site set $P$ of $n$ GC-Polygons and a label set $L$ of $n$ uniform size labels placed at fixed positions on the boundary of $R$, we can compute in $O(n^3)$ time a legal opo-labeling of minimum total leader length.

**Proof.** In Step A of Algorithm 1, we construct a complete weighted bipartite graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$. Each edge $(s_i, l_j) \in E$ is assigned a weight $d_{ij}$ equal to the Manhattan length of the shortest (under the Manhattan metric) leader, which connects area-site $s_i$ with label $l_j$. According to Theorem 3.1, this step costs $O(n^2 \log n)$ time, assuming that the maximum number of corners of a type GC-POLYGON site is fixed.

In Step B of Algorithm 1, we have to compute a minimum cost bipartite matching on the output graph of Algorithm 1. As already mentioned this will cost $O(n^3)$ time. The solution obtained from Step C of Algorithm 1 might contain crossing leaders. In Step D of Algorithm 1, the crossings are eliminated in $O(n \log n)$ time. Thus, the total time complexity of Algorithm 1 is $O(n^3)$.

### 3.3. Sample Labeling of type-opo Leaders

Figures 13 and 14 depict the regions of Germany. In both Figures, the labels are placed on two opposite sides of $R$ and the leaders are of type opo. In Figure 13 a point is arbitrarily selected as the representative of each region. The labeling of Figure 13 is visually improved in Figure 14 by replacing these points with rectangles within each region. Both labelings are optimal in terms of total leader length. However, the total leader length of Figure 14 is reduced by 37% compare to that of Figure 13. Note that we also achieved to reduce the number of leader bends to 5 (in Figure 14) from 8 (in

---

5Recall that $t_R$ denotes the $y$-coordinate of the top-right corner of the enclosing rectangle $R$

6In Figure 12, the MinXInRectangle query corresponds to determining the leftmost leader which is contained within the gray colored rectangle.
Area-Feature Boundary Labeling

Figure 13: A regional map of Germany; a point is the representative of each region.

Figure 14: A visually improved map; a rectangle is the representative of each region.

(a) Before rerouting.
(b) After rerouting.

Figure 15: The rerouting of two crossing po-leaders does not always eliminate their crossing.

the solution obtained in this manner remains optimal in terms of total leader length, but it might contain crossings.

Possible crossings between leaders to the same side of the enclosing rectangle are resolved following a similar strategy as in [4], without changing the total leader length. This can be done in $O(n^2)$ additional time [4, pp.229]. Moreover, we can easily observe that crossings between leaders that go to opposite sides of the enclosing rectangle cannot occur. This is due to the fact that swapping these crossing leaders would result in a solution with smaller total leader length, a contradiction since we assume that the original solution minimizes the total leader length. Note that the same strategy can be applied in the case where the labels can be placed on only one side of the enclosing rectangle $R$.

The following theorem summarizes our result.

**Theorem 4.1.** Given a site set $P$ of $n$ gc-polygons and a label set $L$ of $n$ uniform size labels placed at fixed positions along two opposite sides of $R$, we can compute in $O(n^3)$ time a legal po-labeling of minimum total leader length.

For the case where the labels can occupy two adjacent sides of the enclosing rectangle $R$, there exists instances of the problem where it is not feasible to determine a crossing-free type-po labeling, irrespective of the labeling optimization criteria. This follows from the observation that the rerouting of two crossing po-leaders does not always eliminate their crossing (see Figure 15). Figure 16 illustrates an instance of the problem where the labels occupy all four sides of $R$. All sites are
Figure 16: An instance which admits no crossing-free labeling.

 contained within the gray colored rectangular area that is formed from i) the top left corner of the enclosing rectangle $R$ and ii) the intersection of the lines that coincide with the bottom side of the topmost label on the left side of $R$ and the right side of the leftmost label on the top side of $R$. Since two sites have to be connected to the dark-gray colored labels (incident to the bottom-right corner of $R$), a crossing as the one of Figure 15 will inevitably arise, which implies that this instance admits no-crossing free solution. The following theorem summarizes this result.

**Theorem 4.2.** Consider the area-feature boundary labeling problem with type-$\text{po}$ leaders, uniform labels placed at fixed positions. If labels occupy two adjacent sides of the enclosing rectangle there exists instances that admit no legal (i.e. crossing-free) labeling irrespectable of the labeling optimization criteria.

### 4.1. Sample Labeling of type-$\text{po}$ Leaders

Figures 17 and 18 depict the regions of France. In both Figures, the labels are restricted on one side of $R$ and the leaders are of type $\text{po}$. In Figure 17 a point is arbitrarily selected as the representative of each region. Both labelings are optimal in terms of total leader length. However, the total leader length of Figure 18 is reduced by $10.73\%$ compared to that of Figure 17. Note that the order of the labels is nearly the same as the order of their corresponding sites in Figure 18, in contrast to Figure 17. We also achieved to reduce the number of leader bends to 1 (in Figure 18) from 14 (in Figure 17), just by the use of rectangular area-sites instead of points.

Figure 17: A regional map of France; a point is the representative of each region.

Figure 18: A visually improved map; a rectangle is the representative of each region.

### 5. CONCLUSIONS

In this paper, we considered the area-feature boundary labeling problem where area-sites are presented as GC-POLYGONS instead of points. We presented an efficient algorithm to determine $\text{opo}$-labelings of minimum total leader length, where the labels are of uniform size, with sliding label-ports, placed on all four sides of the enclosing rectangle $R$. For the special case, where the labels can placed on two opposite sides of the enclosing rectangle $R$, we extended this algorithm to support po-leaders.

It is intuitive that the quality of the labelings can be improved by allowing combinations of different types of leaders. To the best of our knowledge no algorithms exist for this model in the map labeling literature.

All known type-$\text{opo}$ boundary labelings use a track routing area outside the enclosing rectangle to route the $p$-segment of the leader. Routing the $p$-segment inside the enclosing rectangle might lead to visually improved labelings.

The evaluation of different optimization criteria (e.g. the one of minimizing the total number of leader bends)
would also be of particular interest. Furthermore, no results were presented for the cases of sliding labels or non-uniform labels. For the latter problem very few results exists in boundary labeling, in general.

Another line of research would be to try to find efficient ways of determining for each region of a map a representative \(gc\)-polygon that has less than \(k\) corners, where \(k\) is a constant.

REFERENCES


