Polygons Labelling of Minimum Leader Length

Michael A. Bekos1  Michael Kaufmann2  Katerina Potika1  Antonios Symvonis1

1 National Technical University of Athens, School of Applied Mathematical & Physical Sciences, 15780 Zografou, Athens, Greece
mikebekos@math.ntua.gr, symvonis@math.ntua.gr, epotika@cs.ntua.gr
2University of Tübingen, Institute for Informatics, Sand 13, 72076 Tübingen, Germany
mk@informatik.uni-tuebingen.de

Abstract

We study a variation of the boundary labelling model, with floating sites, labels of uniform size placed in fixed positions on the boundary (that encloses all sites) and special type leaders connecting labels to sites. We seek to obtain a labelling of all sites with leaders that are non-overlapping and have minimum total length. We present an O(n^2 log^2 n) time algorithm for the labelling of polygons.

Keywords: map labelling, boundary labelling, floating sites, polygons.

1 Introduction

Placing extra information, in the form of text labels, next to features of a drawing (map) is an important task in the process of information visualization. Usually, it is desired that the label placement is done so that each label is close to the feature (site) it describes and is not intersecting with any other label. In general it is NP-hard (Formann & Wagner 1991) to obtain optimal label placements. An extensive bibliography about map labelling can be found at (Wolff & Strijk 1996). Besides labelling point feature in a map, some deal with labelling lines such that labels do not intersect (Strijk & van Kreveld 1999).

There are cases, i.e. when the labels are very large or the features are too many, where it is impossible to find a label placement so that the labels are close to the feature they describe. To cope with such cases, one direction is to allow the labels to be placed not close to each feature but in the boundary of the rectangle that encloses all features. Each feature is connected to its label by non-intersecting polygonal lines, called leaders. We call such a label placement legal or crossing free. The boundary labelling was first defined in (Bekos, Kaufmann, Symvonis & Wolff 2005). Bler (Bekos & Symvonis 2005) supports the boundary labelling process.

The sites model features of the drawing. Very often in practice, we want to label an area feature (e.g. a region of a map). Any point inside the area feature can be arbitrarily chosen to represent the area in the input of the boundary labelling problem. However, instead of arbitrarily selecting a point to represent the area feature, we can specify as part of the input a region in which the site is allowed to “float” in any legal solution of the boundary labelling problem. To keep things simple, we specify these regions to be generalized canonical polygons or rectangles or line segments internal to the feature area, and assume that the site “slides” along the boundary of the polygon or on the line segment. 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 1 shows an example where the regions p_i and p_j are represented by GC-POLYGONS.

All labels have the same width and the same height. The labels are placed in distinct places on all four sides of 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 $W = r_R - l_R$ which contains all sites $p_i$ in $P$.

![Figure 1: Leader $c_j$ is oriented towards corner $A$ and leader $c_i$ is oriented away of corner $A$.](image)


The project is co-funded by the European Social Fund (75%) and National Resources (25%) - Operational Program for Educational and Vocational Training II (EPEAEK II) and particularly the Program PYTHAGORAS.
the label on a point on its side that faces the enclosing rectangle. The point where the leader touches the label is called port. We can assume 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) or sliding ports where the leader can touch any point of the label’s side. In the labelling presented in Figure 1, the label port is point q. Let us denote by c_j the leader of site p_j. The labelings in Figures 1-2 use sliding ports.

We want to obtain labelings that optimize some criterion. Keeping in mind that we want to obtain simple and easy to visualize labellings, the following criterion of minimizing the total leader length can be adopted from the areas of VLSI and graph drawing. The length of a leader c_j is the Manhattan (or L_1) distance of site p_j to its label.

A GC-POLYGON p is a set of k corners indexed \{1,\ldots,k\} in clockwise direction that define the boundary of p. (see in Figure 1 GC-POLYGON p_1 has 4 corners while GC-POLYGON p_2 has 5 corners). We extend the notion of general position for points to GC-POLYgons as follows: We say that GC-POLYGONS p_1, p_2,\ldots,p_n are in general position if we can not locate any two corners belonging to GC-POLYGONS p_i and p_j with the same x- or y-coordinate. The point of site p_j that the leader c_j hits, is called port of site p_j (see an example in Figure 1, the port of site p_j is point pp).

Generally, the number of sites is n, the number of labels is m and the maximum number of corners of each site is k. Each label is a rectangle with four corners.

Let’s give some useful definitions for the type-opo labelling. Consider an type-opo leader c which originates from port pp of a GC-POLYGON and is connected with a label on the east side (segment AB) of the rectangle at port q (see Figure 1). The line y_{pp} containing the segment of the leader which is incident to pp and is orthogonal to side AB divides the plane into two half-planes. We say that leader c is oriented towards corner A of the rectangle if port q and corner A are on the same half-plane, otherwise, we say that leader c is oriented away from corner A. In the case of type-o leader, we consider the leader to be oriented towards corner A (and also towards corner B).

This paper is structured as follows: In Section 2, we formally define the boundary labelling problem. Section 3 studies the problem of minimizing the total leader length when type-opo leaders are used and the labels are placed on all four sides of R. In Section 4, we study the problem of minimizing the total leader length when type-po leaders are used and the labels are placed on two opposite sides of R. We conclude in Section 5 with open problems and future work.

2 The Boundary Labelling models

The boundary labelling model is a 7-tuple \( (\text{Side}, \text{LabelSize}, \text{LabelPort}, \text{LabelPos}, \text{Leader}, \text{Site}, \text{Objective}) \), where:

- **Side:** Sides of the enclosing rectangle next to which we place labels. We use any sequence of N, E, W, and S (for North, East, West, and South). In the case of multiple stacks, we use NE, EW, SE, SW when the labels are attached to the North, East, West and South side of R and use \( i,j,k,l \) number of stacks, respectively. If no labels are placed next to a side we omit the letter corresponding to that side, and if only one stack is used we omit the index 1.

- **LabelSize:** UniSize (all labels have the same size), MaxSize (all labels are Uniform of Maximum Size) or NonUniSize (each label \( l_i \) is associated with a height \( h_i \) and a width \( w_i \))

- **LabelPort:** FixedPorts (where a leader can touch a label are predefined) or SlidPorts (points can slide along the label’s edge)

- **LabelPos:** FixedPos (labels have either to be aligned with a predefined fixed set of points on the boundary of the rectangle) or SlidPos (labels can slide along the rectangle’s sides)

- **Leader:** Type of the leader (opo, po or o)

- **Site:** Type of the sites. Each site is a 1-point, line, rectangle, a polygon etc.

- **Objective:** LEGAL (just find a legal label placement), TLLM (find a legal label placement, such that the total leader length is minimum), TBM (find a legal label placement, such that the total number of bends is minimum or, equivalently, the number of type-o leaders is maximum), LSM (find the maximum label size for which a legal label placement is possible), etc.

2.1 Previous Work and Our Results

All the known results on boundary labelling are given in Table 1. Most of the results were presented in (Bekos, Kaufmann, Symvonis & Wolff 2005) where the boundary labelling problem was defined. A variety of models based on the type of leader, the location of the label and the size of the label are studied for legal label placement and leader bend and leader length minimization.

In Table 2 we present the results of this paper.

3 Four-side Labelling of gc-polygons with type-opo Leaders

We consider boundary labelling with “floating” sites, such as GC-POLYGONS, rectangles and lines. According to the notation of Section 2, first we examine the Boundary Labelling (NEWS, UniSize, SlidPort, FixedPos, opo, GC-POLYGON, TLLM) problem. We assume that we have fixed labels of uniform size, placed on all four sides of rectangle R, sliding ports and type-opo leaders. We present a polynomial time algorithm, that returns a legal labelling of minimum total leader length.

Let \( P = \{p_1, p_2, \ldots, p_n\} \) be the set of GC-POLYGONS and \( L = \{l_1, l_2, \ldots, l_m\} \) be the set of labels. Since the labels have uniform size, each site \( p_i \) can be
Algorithm 1: 4SIDE-AREA-OPO

**Input**: A set of \( n \) GC-POLYGONS \( p_i \) in the plane and a set of \( m \) uniform sized labels \( l_j \).

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

**Step A. Shortest Leader Computation**:
Construct a complete weighted bipartite graph \( G = (P \cup L, E, w) \) between all sites \( p \in P \) and all labels \( l \in L \). The weight of an edge \( (p_j, l_j) \in E \) is the Manhattan length of the shortest (under the Manhattan metric) leader, say \( d_{ij} \), which connects \( p_i \) with \( l_j \).

**Step B**.

Proceed by applying to graph \( G \), Vaidya’s algorithm (Vaidya 1989) for minimum-cost bipartite matching under the Manhattan metric. It computes a matching between sites and labels that minimizes the total Manhattan distance of the matched pairs.

**Step C. Obtain a labelling \( M \) as follows**:

**If** an edge \( (p_i, l_j) \in E \) is selected in the matching **then** connect site \( p_i \) to label \( l_j \) with a leader of length \( d_{ij} \).

**Step D. Crossing Free Procedure**:
Eliminate all crossings of leaders and obtain a crossing free labelling \( M' \).

3.1 Shortest Leader Computation

We propose Algorithm 2 for computing the minimum Manhattan distance between every site and every label (Step A of Algorithm 1).

<table>
<thead>
<tr>
<th>Model</th>
<th>Time complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(NESS, UnifSize, SlidPort, FixedPos, ojo, GC-POLYGON TLLM)</td>
<td>( O(n^2 \log n) )</td>
</tr>
<tr>
<td>(EW, UnifSize, SlidPort, FixedPos, po, Rectangle/Line, TLLM)</td>
<td>( O(n^4 \log^3 n) )</td>
</tr>
<tr>
<td>(EW, UnifSize, SlidPort, FixedPos, po, GC-POLYGON TLLM)</td>
<td>( O(n^4 \log^3 n) )</td>
</tr>
</tbody>
</table>

Table 2: The results presented in this paper.

**Theorem 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 \( R \). Moreover this algorithm runs in \( O(n(k' + m) \log k') \) time, where \( m \) is the number of labels, \( n \) is the number of GC-POLYGONS, \( k' = O(k + m) \) and \( k \) is the maximum number of corners that a site of type GC-POLYGON can have.

**Proof**: The number of points in each set \( p_i \) is \( O(m + k) \) and each set \( p_i \) is computed in \( O(n(m + k)) \) time. Note that set \( p_i \) contains candidate site ports. In Step 1 we construct the Voronoi diagram under the Manhattan distance of the set \( p_i \cup p_i' \) (\( k' \) points total), where \( k' = O(k + m) \). The construction of the Voronoi diagram can be done in \( O(k' \log k') \) time (Lee 1980).

<table>
<thead>
<tr>
<th>Model</th>
<th>Time complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(NESS, UnifSize, SlidPort, FixedPos, ojo, GC-POLYGON TLLM)</td>
<td>( O(n^2 \log^2 n) )</td>
</tr>
<tr>
<td>(NESS, UnifSize, SlidPort, FixedPos, po, Rectangle/Line, TLLM)</td>
<td>( O(n^4 \log^3 n) )</td>
</tr>
</tbody>
</table>

Table 1: Known results on boundary labelling. \( H \) is the height of the enclosing rectangle.
Theorem 2 Algorithm 3 computes the minimum distance under the Manhattan distance between any label and any convex polygon when the labels are placed in fixed positions in all four sides of rectangle $R$. Moreover this algorithm runs in $O(n(m+k))$ time, where $m$ is the number of labels, $n$ is the number of sites and $k$ is the maximum number of corners that a site of type GC-POLYGON can have.

Proof: In each side we compute the minimum distance for the first pair of label site in $O(k)$ time (Step (2)). Recall that our sites are convex and therefore we can determine the minimum Manhattan distance without checking all corners and edges of a site to a label, just by finding the first point of the convex site where the distance starts again to increase. In the computation of the minimum distance between site $p_i$ and label $l_j$ (Step 3(ii) of Algorithm 3), we need to examine only the part of the site that lies in between the port of the previous label and the north (east | south | west) most corner, because label $l_j$ lies between the previous label and the north (east | south | west - side of rectangle $R$) in clockwise direction. This step requires $O(m+k)$ time. Totally the required time is $O(n(m+k))$.■

3.2 Crossing Free Procedure

After Step (C) of Algorithm 1 some leaders may cross, thus we have not yet a legal label placement.

Lemma 1 Labelling $M$ of Algorithm 1 is of minimum total leader lengths with some crossings. Let $c_i$ and $c_j$ be two leaders that cross each other. Then the following holds (i) the labels of these leaders are on adjacent sides of the rectangle in the west or east side and (ii) leaders $c_i$ and $c_j$ are oriented towards corner $A$ of the rectangle $R$ and can be rerouted so that they do not cross each other with unchanged total leader length.

Proof: Prove of (i): We show that it is impossible to have the labels placed at the same or opposite sides of $R$. Suppose that the labels lie on the same side, say the east side, and the leaders intersect or overlap.

If leaders $c_i$ and $c_j$ overlap, then we can slide on of the two leaders, by choosing new site port and probably new label port. Then either the total leader length is reduced (see an example in Figure 3), which is a contradiction, or it remains the same. Note that in the latter case, in a GC-POLYGON we can use a new leader with site port a point that is next to the site port of the old leader, so that the new leader length is equal to the old leader length.

If the leaders $c_i$ and $c_j$ intersect, then the intersection takes place outside rectangle $R$ (in the track routing area). This implies that, along the east side, the order of the sites is the reverse of the order of their associated labels. However, by swapping the labels to which each site is connected, we can reduce the total leader length (and also eliminate the crossing), a contradiction since we assumed that the total leader length of the labelling is minimum (see an example in Figure 4).

Consider now the case where the leaders 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 sites) have to overlap. Again, if the leaders overlap then by swapping the labels to which each 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 labelling is minimum (see an example in Figure 5).

We showed that leaders which have their associated labels lying on the same or on opposite sides of rectangle $R$ can not cross and therefore, if we find two crossing leaders, their associated labels must lie on adjacent sides of the rectangle $R$. Prove of (ii):

Let $c_i$ and $c_j$ be oriented towards corner $A$. First we show that in a labelling of minimum total leader length, it is impossible to have one (Case (a)) or both leaders oriented away of corner $A$ (Case (b)). We proceed to consider these two cases.

Case a: Exactly one leader, say $c_j$, is oriented away from corner $A$. This case is described in the left of Figure 6. Rerouting the leaders as described in Figure 6 results in a reduction of the total leader length, a contradiction since we assumed that the total leader length of the labelling is minimum.

Case b: Both leaders $c_i$ and $c_j$ are oriented away of corner $A$ (see left of Figure 7). When both leaders are oriented away of corner $A$, rerouting 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 shown in Figure 7.
Having eliminated the 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 are oriented towards corner A (see an example in Figure 8).

Case c: Leaders \(c_i\) and \(c_j\), that are oriented towards the same corner, say A, can be rerouted (see Figure 8) so that they do not cross each other and the sum of their leader lengths remains unchanged. Partition the first segment of each leader \(c_i\) and \(c_j\) into two sub-segments in their crossing point. 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. To complete the proof of the lemma, we note that whenever we perform a rerouting, we never change the position of a label or site port. So, since the used port would also be available in the case where the fixed-port model is used, the lemma applies to fixed (label) ports, as stated.

If we had a reduction of the total leader length, by taking as site or label ports other points, this would be a contradiction since we assumed that the total leader length of the labelling is minimum.

We proceed by showing that given a labelling of minimum total leader length which may contain crossings, we can efficiently construct a crossing free labelling of identical total leader length, by first identifying such a crossing and then eliminating the crossing (with the help of Lemma 1(ii)).

**Theorem 3** Step D of Algorithm 1 produces a crossing free labelling \(M'\) of minimum total leader length in \(O(n \log n)\) time.

**Proof:** We can resolve all crossings (as in (Bekos, Kaufmann, Potika & Symvonis 2005)) in \(O(n \log n)\) additional time and keep the total leader length unchanged. We show how to eliminate all crossings in labelling \(M\) by rerouting the crossing leaders. Our method performs two passes over the sites, one in the west-to-east and one in the east-to-west direction.

Consider first the west-to-east pass. In the west-to-east pass of labelling \(M\), we consider all sites with labels on the east side of the rectangle. We examine the sites from west-to-east and we are interested only on those who have crossing leaders. Let \(p_i\) be the west-most such site and let \(c_i\) be the leader that connects it with its corresponding label on the east side of the rectangle (see Figure 9). Lemma 1(i) implies that leader \(c_i\) intersects only with leaders that are connected with labels on the north and south sides of rectangle \(R\). Without loss of generality, assume that \(c_i\) is oriented towards the east-south corner of the rectangle, say A. Then all leaders that intersect \(c_i\) have their labels on the south side of \(R\) and are also oriented towards \(A\) (by Lemma 1(ii)). Let \(c_k\) be the west-most leader that intersects \(c_i\), and let \(p_k\) be its incident site. According to Lemma 1(ii), we can reroute leaders \(c_i\) and \(c_k\) so that the total leader length remains unchanged (Figure 10). The total number of crossings is reduced and the next west-most site with intersecting leader and connected to a label on the east side of the rectangle is located to the east of site \(p_k\). Finally, all west-to-east crossings are eliminated. It is also impossible to introduce a east-to-west crossing when the west-to-east pass is executed, as an example see leaders \(c'_i\) and \(c\) in Figure 11. Both leaders \(c'_i\) and \(c\) must be oriented towards corner \(B\) (by Lemma 1(i)), a contradiction since leader \(c'_i\) is oriented away of corner \(A\) (and towards corner \(A\)).

After the west-to-east and the east-to-west pass, we obtain a labelling \(M'\) without any crossings and of total leader length equal to that of \(M\), that is, minimum.

By employing a dynamic priority search tree based on half-balanced trees [7, pp. 209] we can answer question of the form: ‘given a point \(p\) return the segment that intersects line \(y_p\) and is the nearest in the east site of \(p\); insert and delete operations in \(O(\log n)\) time. Thus, identifying the (at most \(n\)) pairs of leaders to be rerouted during the west-to-east pass takes only \(O(n \log n)\) time, resulting to a total time complexity of \(O(n \log n)\) for the production of the crossing free boundary labelling \(M'\).
Theorem 4 Algorithm 1 solves problem Boundary Labelling(NESW, UnifSize, SlidPort, FixedPos, opo, gc-polygon, TLLM) in $O(n^2 \log^3 n)$ time.

Proof: Let $L$ be the set of the $n$ labels (we assume that the number of labels is equal to the number of sites). We assume that all labels are around the boundary of the rectangle. We construct a complete bipartite graph between all sites $p_i \in P$ and all labels $l_j \in L$, with edge weights to be the Manhattan length $d_{ij}$ of the corresponding leaders computed with the help of Algorithm 2 (or Algorithm 3). This step costs $O(n(k + n) \log(k + n))$ time, where $k$ is the maximum number of corners that a site of type gc-polygon can have (Theorem 1). Or $O(n(n + k))$ time (Theorem 2).

In Step B we proceed by applying the Vaidya’s algorithm (Vaidya 1989) for minimum-cost bipartite matching for points in the plane under the Manhattan metric. It runs in $O(n^2 \log^3 n)$ time and finds a matching between sites and labels that minimizes the total Manhattan distance of the matched pairs.

The leaders in the produced solution might cross. However, in Step D based on Theorem 3 we can obtain a crossing free solution in $O(n \log n)$ additional time.

Since a rectangle is a GC-POLYGON with only four corners and a line-segment of two corners, we have:

Corollary 2 Problem Boundary Labelling(NESW, UnifSize, SlidPort, FixedPos, opo, rectangle/line, TLLM) can be solved in $O(n^2 \log^3 n)$ time.

Remark Notice that Algorithm 1 works for any kind of polygon. We choose the special kind of GC-POLYGONS because these polygons offer better visualization and because one can easily find an alternative leader, in the case of overlapping leaders.

3.3 Sample Labelling of type-opo Leaders

Figures 12 and 13 depict the regions of Germany and a boundary labelling on opposite sides (east and west) of $R$ with type-opo leaders. Figure 12 is produced by the algorithm for boundary labelling points and provides an optimal solution. The labelling of Figure 12 is visually improved in Figure 13 by replacing the points with rectangles within each region. The labelling of Figure 13 is optimal, with the use of less total leader length (37% less pixels) than Figure 12. Note that we achieved to reduce the number of leader bends to 5 (in Figure 13) from 8 (in Figure 12), just by the use of rectangle sites instead of points.

Figure 12: A regional map of Germany; a point is the representative of each region.

Figure 13: A visually improved map; a rectangle is the representative of each region.
leaders. Again, our objective is to minimize the total leader length.

To deal with this problem, we use the (matching based) Algorithm 1 for the case of type-opo leaders to get a label placement of minimum total leader length. As already mentioned, this can be done in $O(n^2 \log^3 n)$ time (Theorem 4). Instead of placing type-opo leaders we use type-po leaders. Note that connecting a site to its label with a type-opo or a type-po leader requires the same leader length under the Manhattan metric, assuming that we keep the same ports. Therefore, the returned solution remains optimal, but might contain crossings.

Crossings of leaders oriented towards the same side can occur, however, they can be resolved following a similar strategy as in (Bekos, Kaufmann, Symvonis & Wolff 2005), without changing the total leader length. This can be done in $O(n^2)$ additional time. Crossings of leaders to opposites sides cannot occur, since swapping the labels of the sites that have crossing leaders, would result in a solution with smaller total leader length. This is a contradiction, because we assumed that the original solution minimizes the total leader length.

The same strategy can be applied when labels are placed in only one side of rectangle $R$ or when the sites are line segments. However, this strategy does not result in a crossing free solution, in the case where labels are placed on all four sides of rectangle $R$, since a crossing between two leaders that are oriented towards two adjacent sides of the enclosing rectangle can not always be eliminated. We conclude:

**Corollary 3** Problem Boundary Labelling(EW, UniSize, SlidPort, FixedPos, po, rectangle/line, TLLM) can be solved in $O(n^2 \log^3 n)$ time.

### 4.1 Sample Labelling of type-po Leaders

Figures 14 and 15 depict labelling of the regions of France with type-po leaders. Restricting the sites to by rectangles leads to labelling with reduced total leader length than the use of points (10.73% less pixels). The order of the labels is the same as the order of the correspondent sites in Figure 15, in contrast to Figure 14. Note also that we achieved to reduce the number of leader bends to 1 (in Figure 15) from 14 (in Figure 14).

### 5 Open Problems and Future Work

We have presented results for boundary labelling of gc-polygons, rectangle or lines instead of points with the objective of minimizing the total leader length. It is interesting to see the visual result of the labelling if we choose another objective as the one of minimizing the number of bends of the leaders. Another future research is to find efficient ways of determine for each region of a map a representative gc-polygon site that has less than $k$ corners.

**References**


