

Boundary Labelling of Optimal Total Leader Length*

M.A. Bekos¹, M. Kaufmann², K. Potika¹, and A. Symvonis¹

¹ National Technical University of Athens,
School of Applied Mathematical & Physical Sciences, Athens, Greece
{mikebekos, symvonis}@math.ntua.gr, epotik@cs.ntua.gr
² University of Tübingen, Institute for Informatics, Tübingen, Germany
mk@informatik.uni-tuebingen.de

Abstract. In this paper, we consider the *leader length minimization problem* for *boundary labelling*, i.e. the problem of finding a legal leader-label placement, such that the total leader length is minimized. We present an $O(n^2 \log^3 n)$ algorithm assuming *type-opo leaders* (rectilinear lines with either zero or two bends) and labels of uniform size which can be attached to all four sides of rectangle R . Our algorithm supports *fixed* and *sliding ports*, i.e., the point where each leader is connected to the label (referred to as *port*) may be fixed or may slide along a label edge.

1 Introduction

One of the most challenging tasks in map labelling is the automated visualization of the information on a map, i.e. the association of text labels with graphical features. In order to ensure readability, unambiguity and legibility, cartographers suggest that the labels should be pairwise disjoint and close to the point (also referred to as *site*) to which they belong [5,12]. Unfortunately, the majority of map labelling problems are shown to be *NP*-complete [1,4,6]. Due to this fact, graph drawers and computational geometers have suggested labelling approximations [1,4,10] and heuristics [11], which often try to maximize either the label size or the number of features with labels. A detailed bibliography on map labelling can be found in [9]. It is worth mentioning that the *ACM* Computational Geometry Task Force [3] has identified label placement as an important area of research.

Research on map labelling has been primarily focussed on labelling point-features, where the basic requirement is that the labels should be pairwise disjoint. It is clear that this is not achievable in the case of large labels (or, equivalently, large point sets). Large labels are common in technical drawings or medical atlases where certain site-features are explained with blocks of text.

* This work has partially been supported by the DFG grant Ka 512/8-3, by the German-Greek cooperation program GRC 01/048 and by the Operational Program for Educational and Vocational Training II (EPEAEK II) and particularly the Program PYTHAGORAS (co-funded by the European Social Fund (75%) and National Resources (25%)).

To address this problem, Bekos et. al. defined *boundary labelling*[2]. In boundary labelling, labels are attached on the boundary of a rectangle R which encloses all sites. The main task is to place the labels in distinct positions on the boundary of R so that they do not overlap and, to connect each site with its corresponding label by non-intersecting polygonal lines, so called *leaders*.

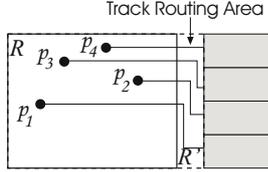
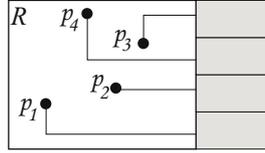
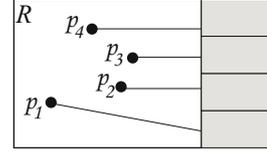
The basic boundary labelling problem can be formally described as follows: We are given an axis-parallel rectangle $R = [l_R, r_R] \times [b_R, t_R]$ and a set P of n sites $p_i = (x_i, y_i)$ in general position, i.e. no three sites lie on a line and no two sites have the same x or y coordinate. We denote with W and H the width and the height of R , respectively. Each site p_i lie in the interior of R (i.e. $l_R < x_i < r_R$ and $b_R < y_i < t_R$) and is associated with an axis-parallel, rectangular label l_i of width w_i and height h_i . Each label should lie outside R but touch the boundary of R . Our task is to place the labels in distinct positions on the boundary of R , so that they do not overlap and, to connect each site with its label, such that no connection (referred to as *leader*) intersect any other connection or site. Such labellings are called *legal leader-label labellings* or *crossing free labellings*.

The point where each leader touches its corresponding label is referred to as *port*. Ports may be fixed (e.g., at the middle of a label edge) or may slide along a label edge. Based on the type of allowed ports (*fixed* or *sliding*) one can define the corresponding variations of the boundary labelling problem. Furthermore, one can define more variations of boundary labelling problems based on the location of the labels. More specifically, labels are usually attached to one, two or all four sides of the enclosing rectangle and are either placed at predefined locations (*fixed labels*) along the sides or can slide (*sliding labels*).

The leaders connecting the sites to their corresponding labels can be of several types, each giving rise to a new boundary labelling model. Of particular interest are two types of leaders: *rectilinear* and *straight* leaders.

- **Rectilinear Leaders:** Each leader consists of a sequence of axis-parallel segments, which are either parallel (p) or orthogonal (o) to the side of R to which the associated label is attached. This suggests that a leader c of type $c_1 c_2 \dots c_k$, where $c_i \in \{o, p\}$ consists of an x - and y -monotone connected sequence (s_1, s_2, \dots, s_k) of segments from the site to the label, where segment s_i is parallel to the side containing the label if $c_i = p$; otherwise it is orthogonal to that side. Our primary focus has been on *opo* and *po* leaders, see Figures 1 and 2, respectively. For each *opo* leader we further insist that the parallel p segment lies immediately outside R , in the so called *track routing area*. Type- o leaders can be considered as either type *opo* or type *po*.
- **Straight Leaders:** Each leader is drawn as a straight line segment (see Figure 3). According to the previous classification scheme, we refer to straight leaders as type s leaders.

Given a boundary labelling problem specified by a set of points inside an enclosing rectangle and a labelling model (as specified by restrictions on the type of ports, the location and type of labels, and the type of leaders), we are interested in finding a solution that is optimal with respect to some objective. We usually aim at:

Fig. 1. Type-*opo* leadersFig. 2. Type-*po* leadersFig. 3. Type-*o* and type-*s* leaders

- **Short Leaders:** Find a legal leader-label placement, such that the total leader length is minimum. Such a labelling minimizes the average leader length.
- **Simple Layout:** Find a legal leader-label placement, such that the total number of bends in minimum. Such a labelling minimizes the average number of bend per leader.

Table 1 summarizes the results of Bekos et. al. [2] on boundary labelling. They examined a variety of models based on the type of leader, the location of the label and the size of the label and presented algorithms for legal leader-label assignments and leader-bend and leader-length minimization. These are the only published results on boundary labelling.

Table 1. Known results on boundary labelling. TLL stands for “Total Leader Length”.

Model	Objective	Time complexity
<i>opo</i> , 1-side, variable size labels	legal #bends	$O(n \log n)$ $O(n^2)$
<i>opo</i> , 4-side, uniform square labels	legal	$O(n \log n)$
<i>po</i> , 1-side, uniform	legal	$O(n^2)$
<i>opo</i> , 2-side (opposite), uniform labels of max-size	TLL	$O(n^2)$
<i>po</i> , 2-side (opposite), uniform labels of max-size	TLL	$O(n^2)$
<i>opo</i> , 4-side, uniform square labels	TLL	$O(n^5)$
<i>opo</i> , 2-side (opposite), variable size labels	TLL	$O(nH^2)$
<i>s</i> , 1-side, uniform labels	legal	$O(n \log n)$
<i>s</i> , 1-side, uniform labels	TLL	$O(n^{2+\delta})$, $\delta > 0$
<i>s</i> , 4-side, uniform square labels	TLL	$O(n^{2+\delta})$, $\delta > 0$

One of the presented algorithms in [2] examines 4-side *opo* labelling with uniform square labels. In $O(n^5)$ time, it computes a boundary labelling of minimum total leader length. The algorithm is based on an $O(n^2 \log^3 n)$ minimum cost bipartite matching algorithm for the production of a minimum total leader length solution which may have crossings, and the subsequent elimination of these crossings in $O(n^5)$ time (based on techniques used in VLSI routing). In

this paper, we present a solution of $O(n^2 \log^3 n)$ time complexity which is also based on minimum cost bipartite matching and an improved $O(n \log n)$ algorithm to eliminate crossings.

2 Four-Sided, Uniform Label, *opo* Boundary Labelling

We show how to compute in $O(n^2 \log^3 n)$ time an *opo* boundary labelling of minimum total leader length where the labels can be placed on all four sides of the enclosing rectangle's boundary. We assume labels of uniform size and sliding ports.

We first make some observations regarding *opo*-labelling (which might contain crossings) of minimum total leader length for the case of four-sided labelling with labels of uniform size and sliding ports. Consider an *opo*-leader c which originates from point p and is connected with a label on side AB of the rectangle at port q (see Figure 4). The line containing the segment of the leader which is incident to site p (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 of corner A . In the case where the *opo*-leader consists of only one segment, i.e., the port lies on the line which defines the two half-planes, we consider the leader to be oriented towards corner A (and also towards corner B).

Lemma 1. *Consider four-sided labelling with labels of uniform size and sliding ports and let L be an *opo*-labelling (which might contain crossings) of minimum total leader length. Let c_i and c_j be two leaders originating from sites p_i and p_j , respectively, which cross each other. Then it holds:*

- (i) *The labels associated with leaders c_i and c_j are located at two adjacent sides of the rectangle incident to, say, corner A .*
- (ii) *Leaders c_i and c_j are oriented towards corner A of the rectangle.*
- (iii) *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.*

Proof. Showing that “the labels associated with leaders c_i and c_j are located at two adjacent sides of the rectangle” is easy. We simply have to show that it is not possible to have the labels located at the same side or opposite sides of the rectangle. For the sake of contradiction, assume first that the labels lie on the same side, say AB , of the rectangle. Then the segments of the leaders which are incident to the sites are parallel to each other. Since the sites have distinct X and Y coordinates, these segments do not overlap each other, and thus, the intersection of the two leaders takes place outside the rectangle (in the track routing area). This implies that, along the direction of side AB , the order of the sites is the reverse of the order of their associated labels. However, by swapping the labels, we can reduce the total leader length (and also eliminate a crossing), a contradiction since we assumed that the total leader length of the labelling is minimum (see Figure 5). Consider now the case where, for the sake of contradiction the labels lie on opposite sides of the rectangle. Then, since

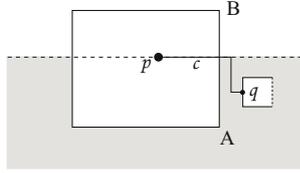


Fig. 4. Orientation of a leader with respect to a corner. Leader c is oriented towards corner A and away of corner B .

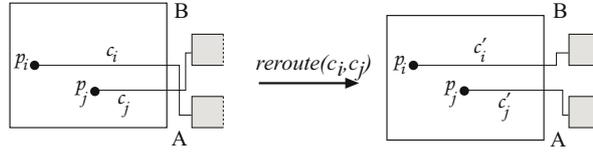


Fig. 5. Rerouting used to prove that in an *opo*-labelling (where crossings are allowed) of minimum total leader length, the labels associated with two crossing leaders do not lie on the same side of the rectangle

the leaders intersect each other, the segments of the leaders which are inside the rectangle (and incident to the sites) have to intersect. However, since these segments are parallel to each other, they have to overlap, and thus have the same X or Y coordinates, a contradiction since we assume that the sites are in general position. Having eliminated the cases that the labels lie on the same or on opposite sides of the rectangle implies that, assuming that we can identify two crossing leaders, their associated labels lie on adjacent sides of the rectangle.

Let A be the corner which is incident to the two sides of the rectangle containing the labels associated with leaders c_i and c_j . In order to show that in a labelling 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 labelling 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 1: Exactly one leaders, say c_i , is oriented away of corner A . This case is described in the left-hand side of Figure 6.a. Rerouting the leaders as described in Figure 6.a results in a reduction of the total leader length, a contradiction since we assumed that the total leader length of the labelling is minimum. Note that, in the figure we only show the sub-case where site p_j is below the horizontal line passing through port q_i . When p_j is on or above the horizontal line passing through port q_i , rerouting again results to a reduction of the total leader length. Thus, a labelling of minimum total leader length does not contain two crossing leaders where one of them is oriented away of the corner A incident to the sides containing their associated labels.

Case 2: Both leaders c_i and c_j are oriented away of corner A . When both leaders are oriented away of corner A , rerouting results in higher reduction of the total leader length, compared to Case 1 where only one leader was oriented away of corner A . The rerouting of the leaders is described in Figure 6.b. Again, only one of the four possible sub-cases based on whether site p_i (p_j) is to the right (below) the vertical (horizontal) line passing through port q_j (q_i) is shown. Given that rerouting results to reduction of the total leader length, we conclude that a labelling of minimum total leader length does not contain two crossing leaders where both of them are oriented away of the corner A incident to the sides containing their associated labels.

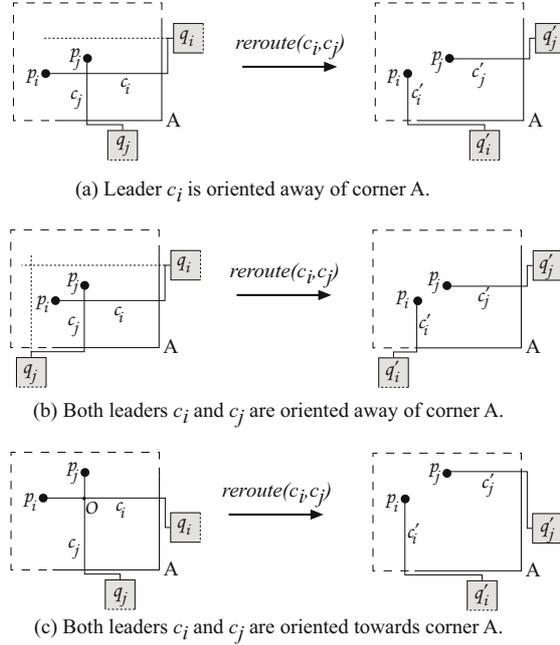


Fig. 6. Rerouting used to prove that in an *opo*-labelling (where crossings are allowed) of minimum total leader length, two crossing leaders are oriented towards the corner incident to the sides of the rectangle containing the associated labels and that their crossing can be eliminated without reducing the sum of their leader length

Having eliminated the cases where one or both crossing leaders are oriented away of corner A , implies that (assuming that we can identify two crossing leaders) they are both oriented towards corner A .

Showing 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” is easy. In the rerouting described in Figure 6.c, use the crossing point O to partition the first segment of each leader c_i and c_j into two sub-segments. Then, leaders c'_i and c'_j can be obtained by a parallel translation of 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 performed a rerouting, we never changed the position of a port. So, since the used port would also be available in the case where the sliding-port model is used, the lemma applies to sliding ports, as stated. \square

Theorem 1. *Consider *opo*-labelling of n sites with uniform labels and sliding ports where crossings are allowed. Then, given a labelling L of minimum total leader length, we can always identify a crossing-free *opo*-labelling L' with total leader length equal to that of L . Moreover, labelling L' can be obtained in $O(n \log n)$ time.*

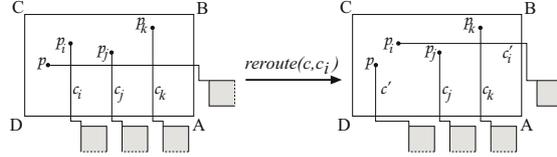


Fig. 7. Rerouting used to eliminate crossings in an *opo*-labelling of minimum total leader length. The crossings to be eliminated are identified in a left-to-right pass of the sites, followed by a right-to-left pass. See proof of Theorem 1.

Proof. We will show how to eliminate all crossings in L by rerouting the intersecting leaders. Our method performs two passes over the 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 labelling L , we consider all sites with labels on the right side of the rectangle. We examine the sites in order from left-to-right and focus only on those which are incident to crossing leaders. Let p be the leftmost such site and let c be the leader that connects it with its corresponding label on the right side of the rectangle (see Figure 7). Given that L is an *opo*-labelling of minimum total leader length, Lemma 1.(i) implies that leader c intersects only with leaders that are connected with labels on the top and bottom sides of the rectangle. Without loss of generality, assume that c is oriented towards the bottom-right corner of the rectangle, say A . Then all leaders that intersect c have their labels on the bottom of the rectangle and are also oriented towards A (Lemma 1.(ii)). Let c_i be the leftmost leader that intersects c , and let p_i be its incident site. According to Lemma 1.(iii), we can reroute leaders c and c_i so that the total leader length remains unchanged (Figure 7). Observe that the rerouting possibly eliminates more than one crossing (e.g., the crossings between leader c and leaders c_j and c_k) but, in general, it might also introduce new crossings (e.g., the crossings between leaders c'_i and c_k). However, the total number of crossings is reduced and, more importantly, the leftmost site incident to an intersecting leader connected to a label on the right side of the rectangle is located to the right of site p . Continuing in the same manner, the leftmost 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.

Another important property is that it is impossible to introduce any “right-to-left” crossing during the left-to-right pass. To see this, assume that such a crossing was introduced and that it involves leader c' and the leader c_l which connects site p_l to a label on the left side of the rectangle (Figure 8). Given that the rerouting does not increase the total leader length, the labelling resulting after all rerouting is still one of minimum total leader length. Then, according to Lemma 1.(i), both leaders c' and c_l must be oriented towards corner D , a contradiction since leader c' is oriented away of corner D (and towards corner A).

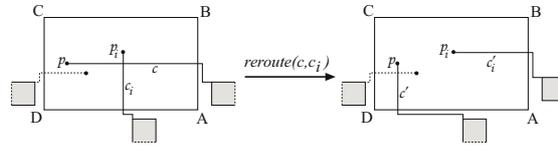


Fig. 8. It is impossible to introduce any right-to-left crossing during the left-to-right pass described in the proof of Theorem 1

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 labelling L' without any crossings and of total leader length equal to that of L , that is, minimum.

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

In order to process site $p = (x_p, y_p)$ and to eliminate the crossings (if any) involving its leader c , we have to identify the leftmost site p_i such that its corresponding leader (say c_i) intersects leader c . Of course, the intersection involves the first segment of leader c_i that is parallel to the Y -axis. The processing of the sites during the left-to-right pass can be accomplished by employing a data structure storing “vertical line segments” and supporting *visibility queries* of the form “given a query point $p_0 = (x_0, y_0)$ return the first line segment to the right of p_0 that is intersected by line $y = y_0$ ”, as well as *insert* (for initialization) and *delete* operations. For the case of vertical line segments of *finite size*, the visibility query can be answered in $O(\log^2 n)$ time by employing a combination of interval trees and priority search trees [7, pp. 211]. This results to a total of $(n \log^2 n)$ time for the left-to-right pass and, consequently, for the elimination of all crossings. However, the time needed to eliminate all crossings can be further reduced to $O(n \log n)$ if we take into account the fact that all vertical segments considered during the left-to-right pass have one of their endpoints on the bottom or the top side of the enclosing rectangle.

Without loss of generality, assume that leader c is oriented towards the bottom-right corner of the enclosing rectangle. (The case where it is oriented towards the top-right corner can be handled in a symmetric manner.) Then, according to Lemma 1.(ii) all leaders intersecting leader c are also oriented towards the bottom-left corner and, thus, their associated labels are placed on the bottom side of the enclosing rectangle. So, leader c can only intersect vertical line segments which have one of their end-points on the bottom side of the enclosing rectangle.

When we have to solve a visibility query on the set of line segments having one of their end-points on the bottom side of the enclosing rectangle, we

can relax the restriction that the segments are of finite size and assume that they are semi-infinite rays having their associated site as their higher endpoint. This is due to the fact that all leader intersections take place inside the enclosing rectangle. Recall that r_R denotes the Y -coordinate of the right side of the enclosing rectangle R . In the case of semi-infinite segments, the visibility query (with $p_0 = (x_0, y_0)$ as the query point) on set of vertical line segments reduces to finding the site of smallest X -coordinate in the semi-infinite vertical strip defined by $x > x_0$, $y \leq y_0$, and $x < r_R$. The *MinXinRectangle* query just described can be answered in time $O(\log n)$ by employing a dynamic priority search tree based on half-balanced trees [7, pp. 209]. Insertions and deletions are also supported in $O(\log n)$ time.

Thus, identifying the (at most n) pairs of leaders to be rerouted during the left-to-right 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 L' . \square

Theorem 2. *Consider four-sided opo-labelling of n sites with uniform labels and sliding ports. A crossing-free solution of minimum total length can be computed in $O(n^2 \log^3 n)$ time.*

Proof. Let M be the set of the n labels around the boundary of the rectangle. We construct a complete bipartite graph $G = (P \cup M, E)$ between all the sites $p \in P$ and all the labels $m \in M$, with edge weights to be the Manhattan length of the corresponding leaders. Note that the length of each leader depends on the type of the port. For the case of sliding ports, the leader typically connects the site to one of the corners of the label. We proceed by applying the Vaidya's algorithm [8] 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 overlap. However, based on Theorem 1 we can eliminate all crossings in $O(n \log n)$ additional time. \square

3 Conclusion

There are several issues that should be considered in future work on boundary labelling. Among them, we distinguish:

- **Labelling “Area Features” of Maps.** Better quality labellings can be produced by allowing a site to slide along a line segment or along the boundary of a polygon. In this case, the solution of the boundary labelling problem has to also specify the final location of each site.
- **Mixed Boundary Labellings.** Examples for type-*opo* and type-*po* leaders show advantages and also some disadvantages of both types. A practical solution might be to mix both types in order to cope with disadvantages while keeping advantages.

References

1. P. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, 11:209–218, 1998.
2. M. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. In Janos Pach, editor, *Proc. 12th Int. Symposium on Graph Drawing (GD'04)*, Lecture Notes in Computer Science, New York, September 2004.
3. B. Chazelle and 36 co-authors. The computational geometry impact task force report. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, vol. 223, pp. 407–463. AMS, 1999.
4. M. Formann and F. Wagner. A packing problem with applications to lettering of maps. In *Proc. 7th ACM Symp. Comp. Geom. (SoCG'91)*, pages 281–288, 1991.
5. E. Imhof. Positioning Names on Maps. *The American Cartographer*, vol.2 (1975), 128-144.
6. C. Iturriaga and A. Lubiw. NP-hardness of some map labeling problems. Technical Report CS-97-18, University of Waterloo, 1997.
7. K. Mehlhorn. *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*, volume 3 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Heidelberg, Germany, 1984.
8. P. M. Vaidya. Geometry helps in matching. *SIAM J. Comput.*, 18:1201–1225, 1989.
9. A. Wolff and T. Strijk. The Map-Labeling Bibliography. <http://i11www.ira.uka.de/map-labeling/bibliography/>, 1996.
10. F. Wagner. Approximate map labeling is in $\Omega(n \log n)$. Technical Report B 93-18, Fachbereich Mathematik und Informatik, Freie Universitat Berlin, 1993.
11. F. Wagner and A. Wolff. Map labeling heuristics: provably good and practically useful Proceedings of the eleventh annual symposium on Computational geometry, p.109-118, 1995
12. P. Yoeli. The Logic of Automated Map Lettering. *The Cartographic Journal* 9, 1972, 99-108