Multi-Stack Boundary Labeling Problems

Michael A. Bekos[1], Michael Kaufmann[2], Katerina Potika[1] Antonios Symvonis[1]

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

Abstract. The *boundary labeling* problem was recently introduced in [4] as a response to the problem of labeling dense point sets with large labels. In boundary labeling, we are given a rectangle R which encloses a set of n sites. Each site p_i is associated with an axis-parallel rectangular label l_i. 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*. Such a label placement is referred to as *legal leader-label placement*.
In this paper, we study boundary labeling problems along a new line of research. We seek to obtain labelings with labels arranged on more than one stacks placed at the same side of the enclosing rectangle R. We refer to problems of this type as *multi-stack boundary labeling problems*.
We present algorithms for *maximizing the uniform label size* for boundary labeling with two and three stacks of labels. The key component of our algorithms is a technique that combines the merging of lists and the bounding of the search space of the solution. We also present NP-hardness results for multi-stack boundary labeling problems with labels of variable height.

1 Introduction

A common task in the process of information visualization is the placement of extra information, usually in the form of text labels, next to the features of a drawing (diagram, map, technical or graph drawing) they describe. When the labels are small and the features they describe are sparsely distributed in the drawing, it may be feasible to place most labels next to the features they describe so that the labels do not overlap with each other and they do not obscure other drawing features. Obtaining optimal label placements with respect to some optimization criterion is, in general, NP-hard [7]. An extensive bibliography about map labeling can be found at [11].

In the case of very large labels (or, equivalently, dense feature sets), usually it is impossible to find a label placement, i.e. place each label next to the feature it describes. In response to this problem, Bekos, Kaufmann, Symvonis and Wolff [4] (see also [3]) proposed the *boundary labeling model*. In this model the labels are placed on the boundary of a rectangle enclosing all features and each label is connected to its associated feature with polygonal lines, called *leaders*. If the labels are non overlapping and the leaders non intersecting we have a *legal leader-label placement* or *legal labeling*. The boundary labeling model is a realistic model for medical atlases and technical drawings, where certain features of a drawing are explained by blocks of text placed outside the drawing so that no part of the drawing is obscured. SmartDraw [9] provides boundary labelings in a primitive form based on labeling templates. It does not support any form of automated boundary labeling optimization. Bler [5] supports the boundary labeling process and facilitates the annotation of drawings with text labels.

Sites model features of the drawing. If they model a *point-feature* (e.g., a city on a map) they are naturally represented as points (see points in Rectangle R of Figures 1, 2 and 5). So, in its simplest form, a boundary labeling problem specifies as part of its input a set P of n points $p_i = (x_i, y_i)$ on the plane in general position, i.e. no three points lie on a line and no two points have the same x- or y-coordinate. Another interesting variation is the one with two candidate points on the plane for each site (see Figure 3). In practice, several times we want to associate a label with an *area-feature* (e.g., a region on a map, a body part on a medical atlas, a machine part on a technical drawing). To keep things simple, we specify these regions by a closed polygonal line or by a line segment internal to the feature area, and assume that the site "slides" along the boundary of the polygon or on the line segment (see Figure 4).

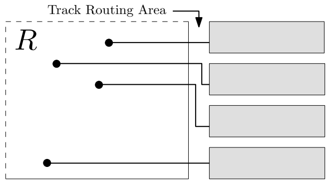


Fig. 1: Type-*opo* leaders.

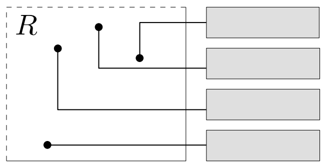


Fig. 2: Type-*po* leaders.

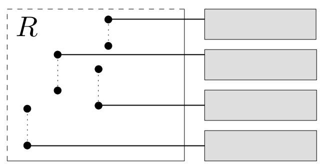


Fig. 3: Sites with 2 candidate positions.

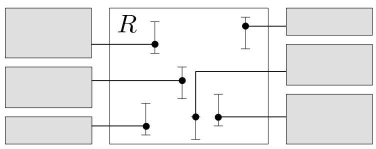


Fig. 4: Sites associated with vertical line segments.

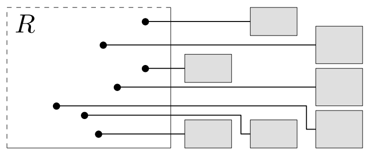


Fig. 5: Three stacks of labels.

In general, each site p_i has a corresponding axis-parallel rectangular, open label l_i of width w_i and height h_i. The labels are to be placed around an axis-parallel rectangle $R = [l_R, r_R]\text{x}[b_R, t_R]$ of height $H = t_R - b_R$ and width $W = r_R - l_R$ which contains all sites p_i in P. While in the general case the labels are of *variable dimensions*, it is natural to consider the restricted cases where the labels are of *uniform size* (height and/or width), or of *maximum uniform size*.

Each site is connected with its corresponding label in a simple and elegant way by using polygonal lines, called *leaders*. In our approach we have leaders that consist of a single straight line segment or a sequence of rectilinear segments. A leader consisting of a straight line segment is referred to as a type-s leader. When a leader is rectilinear, it consists of 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 $\{p, o\}$. We focus on leaders of types-*opo* and *po*, see Figures 1 and 2, respectively. Furthermore, we assume that each type-*opo* leader has the parallel p-segment outside the bounding rectangle R, routed in the so-called *track routing area*. We consider type-o leaders to be of type-*opo* and of type-*po* as well.

A further refinement of the labeling model has to do with the sides of the enclosing rectangle containing the labels. Labels can be placed on one or more sides of the enclosing rectangle (in Figures 1 and 2 all labels are placed on the east side of the enclosing rectangle). Finally, in order to allow for *greater numbers* of *larger labels*, we might have the labels arranged in more than one stack at each side of the enclosing rectangle. This paper is devoted to the case of *multi-stack labelings*. Figure 5 shows a labeling where the labels occupy three stacks to the east side of the enclosing rectangle. Notice that in the case of multiple stacks of labels (say m stacks), a leader of type-*opo* can have its p segment either in between the rectangle R and the first stack (called *first track routing area*) or between the i-th and the $(i+1)$-th stack, where $i < m$ (called $(i+1)$-*th track routing area*).

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 port. We can assume either *fixed ports*, i.e. 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. The labelings in Figures 1, 2, 3 and 4 use sliding ports, while in Figure 5 it uses fixed ports.

Keeping in mind that we want to obtain simple and easy to visualize labelings, the following criteria can be adopted from the areas of VLSI and graph drawing: *minimizing the total number of bends* of the leaders, *minimizing the total leader length*, *minimizing the maximum leader length*. An additional criterion that we consider is the *maximization of the label size* for uniform size labels. This is a quite common optimization criterion in the map labeling literature. In this paper, we seek to obtain labelings with labels of maximum uniform size arranged on more than one stacks of labels at the same side of the enclosing rectangle R

This paper is structured as follows: Section 2, reviews preliminary results required for the development of our algorithms. In Section 3, we present algorithms for obtaining multi-stack labelings of maximum uniform label height for the cases of two and three stacks of labels arranged at the same side of R. In Section 4, we present several NP-hardness results for non-uniform sized labels placed in two stacks. We conclude in Section 5 with open problems and future work.

1.1 Previous Work

Most of the known results on boundary labeling with point sites were presented in [4] (see also [3]). A legal labeling, on one side with type-*opo* (type-*po*) leaders can be achieved in $O(n \log n)$ time (in $O(n^2)$ time, respectively), whereas on all four sides with type-*opo* leaders in $O(n \log n)$ time. The minimization of the total leader length when uniform sized labels can be placed on two opposite sides of R with either type-*opo* and type-*po* leaders needs $O(n^2)$ time. The same problem on two opposite sides of R with type-*opo* leaders and non-uniform label sizes $O(n^2 H)$ is needed. The problem of minimizing the total number of leader bends on one side with type-*opo* leaders can be solved in $O(n^2)$ time. An algorithm for minimizing the total leader length on four sides with type-*opo* leaders in $O(n^2 \log^3 n)$ time is presented for points in [1] and for polygons in [2].

2 Preliminaries

Throughout the paper we use lists that contain pairs of integers. Given a pair (a, b) of integers, a and b are referred to as the *first* and the *second coordinate* of the pair, respectively. Inspired by an idea of Stockmeyer [10] which was subsequently used by Eades et. al. [6], we manage to keep the length of each list bounded by pruning pairs that cannot occur in an optimal solution.

Definition 1. *A list L of pairs of integers is* ***sorted*** *if the pairs it contains are lexicographically sorted in decreasing order with respect to their first coordinate and in increasing order with respect to their second coordinate.*

Definition 2. *Let (a, b) and (c, d) be pairs of integers.*

$$(a, b) \ \textbf{dominates} \ (c, d) \iff a \geq c \text{ and } b \geq d.$$

Suppose we have to solve a problem where the search space of the solution consists of pairs of integers, and let f be a monotone function computing a minimization objective on pairs from the solution search space. Thus, if (a, b) and (c, d) represent possible solutions, then the pair (a, b) can never be involved in an optimal solution and may be safely removed from the solution set. Given a list L of pairs of integers, a pair $(a, b) \in L$ that does not dominate any other pair in L is called an *atom* (with respect to L).

In our algorithms we maintain lists (of pairs) that contain only atoms. A frequently performed operation is the merging of two lists of atoms, resulting in a new list of atoms. The merging algorithm resembles the merging step of merge sort algorithm and can be considered to be folklore. It supports the following lemmas:

Lemma 1. *k sorted lists $L_1, L_2, \ldots, L_k,\ k \geq 2$, of atoms can be merged in $O((k-1)\sum_{i=1}^{k}|L_i|)$ time into a new sorted list L of at most $\sum_{i=1}^{k}|L_i|$ atoms.*

Lemma 2. *Let A and B be two finite set of integers and let $L = \{(a,b)|\ a \in A$ and $b \in B\}$ be a list of atoms. Then, $|L| \leq \min(|A|,|B|)$.*

Finally, we present some notation and terminology that we use in the description of our algorithms. We say that a pair (a,b) obeys the *boundary conditions*, if $a \leq H$ and $b \leq H$, where H is the height of the enclosing rectangle. We also define operator $\oplus_H : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, where:

$$a \oplus_H b = \begin{cases} a+b, \text{ if } a+b \leq H \\ \infty, \quad \text{otherwise} \end{cases}$$

3 Label Size Maximization

3.1 Two Stacks of Labels on the Same Side

We consider boundary labeling with type-*opo* leaders, where the labels are placed at two stacks on the same side (say, the east side) of the enclosing rectangle. We assume that all labels have the same size (width and height) and we seek to maximize the height h of all labels. Our approach is as follows: Given labels of height h, we propose an algorithm, that determines whether there exists a legal labeling. To determine the maximum value of h, we apply a binary search on all possible discrete values for height h. We assume the more general case of sliding labels with sliding ports. The type-*opo* leaders connecting sites to labels that are on the second stack can have their bends (or equivalently their p segments) either in the first or in the second track routing area.

Observe that, in any legal one-side labeling with type-*opo* leaders, the vertical order of the sites is identical to the vertical order of their corresponding labels on both stacks. This, together with the assumption that no two sites share the same y-coordinate, guarantees that leaders do not intersect. So, we assume that the sites are sorted according to increasing y-coordinate.

For a fixed label height h, we propose a dynamic programming algorithm that outputs a boolean value, which indicates whether there exists a legal label placement, when all sites are associated with labels of height h. Our algorithm maintains a table T of size $(n+1)\times(n+1)$. Each entry $T[i,k],\ i \leq k$ of table T describes different possible placements for the subproblem consisting only of the first i sites, such that k out of the i leaders have their bends in the second track routing area. Assuming that we have placed the labels for the first $i-1$ sites, we try to place the i-th label, which corresponds to the i-th site. We distinguish two cases based on whether the label is placed on the first or second stack. Additionally, if l_i is to be placed in the second stack, then we have to check whether this can be done with a leader that has its bends in the first or second track routing area. Obviously, such placements can be obtained from placements of the first $i-1$ sites with either k or $k-1$ leaders, that bend in the second track routing area. For each possible placement, we are interested in the height of two stacks. This information is captured by a pair (a,b), where a (b) is the highest occupied Y-coordinate of the first (respectively second) stack. Thus, entry $T[i,k]$ contains a list of atoms, each corresponding to a different placement. List $T[i,k]$ is empty, when it is impossible to place the first i labels, using k leaders that have their bends in the second track routing area.

Label l_i is placed at the first stack: Let $T_1[i,k]$ be a list of pairs (a,b), where a (b) is the highest occupied Y-coordinate of the first (respectively second) stack, when the labels of the first i sites have been placed, the i-th site has its label at the first stack and k out of the i leaders have their bends in the second track routing area. $T_1[i,k]$ can be computed based on entry $T[i-1,k]$ (see Figure 6a), as follows:

$$T_1[i,k] = \{(a \oplus_H h, b):\ \forall(a,b) \in T[i-1,k]\}$$

(a) Placing l_i in 1st stack; bend in 1st track routing area.

(b) Placing l_i in 2nd stack; bend in 1st track routing area.

(c) Placing l_i in 2nd stack; bend in 1st track routing area.

(d) Placing l_i in 2nd stack; bend in 2nd track routing area.

Fig. 6: Different placements obtained for the label of site i. In Figures 6a, 6b and 6c: $(a, b) \in T[i-1, k]$, whereas in Figure 6d: $(a, b) \in T[i-1, k-1]$.

Label l_i is placed at the second stack - bend at the first track routing area: Let $T_{21}[i, k]$ be a list of pairs (a, b), where a (b) is the highest occupied Y-coordinate of the first (respectively second) stack, when the labels of the first i sites have been placed, the i-th site is connected with a label placed at the second stack using a leader that has its bends at the first track routing area and k out of the i leaders have their bends in the second track routing area. Again, $T_{21}[i, k]$ can be computed based on entry $T[i-1, k]$. This computation is a little bit more complicated. If for some pair $(a, b) \in T[i-1, k]$ it holds that $a \leq b$ (i.e. the occupied area of the first stack is lower or equal than the occupied area of the second stack), then a pair $(b, b \oplus_H h)$ is added in $T_{21}[i, k]$ (see Figure 6b). Otherwise, pair $(a, \max\{b \oplus_H h, a\})$ is added in $T_{21}[i, k]$ (see Figure 6c). So, we conclude that $T_{21}[i, k]$ can be computed by using the following formula:

$$T_{21}[i, k] = A_{21}[i, k] \cup B_{21}[i, k],$$

where:

$$A_{21}[i, k] = \{(b, b \oplus_H h) : \ \forall (a, b) \in T[i-1, k] \text{ s.t. } a \leq b\}$$
$$B_{21}[i, k] = \{(a, \max\{b \oplus_H h, a\}) : \ \forall (a, b) \in T[i-1, k] \text{ s.t. } a > b\}$$

Label l_i is placed at the second stack - bend at the second track routing area: Similarly, let $T_{22}[i, k]$ be a list of pairs (a, b), where a (b) is the highest occupied Y-coordinate of the first (respectively second) stack, when the labels of the first i sites have been placed, the i-th site is connected with a label placed at the second stack using a leader that has its bends at the second track routing area and k out of the i leaders have their bends in the second track routing area. $T_{22}[i, k]$ can be computed based on entry $T[i-1, k-1]$ (see Figure 6d), as follows:

$$T_{22}[i, k] = \{(y_i, b \oplus_H h) : \ \forall (a, b) \in T[i-1, k-1] \text{ s.t. } a < y_i\}$$

All pairs (∞, a), (a, ∞) can be removed from lists $T_1[i, k]$, $T_{21}[i, k]$ and $T_{22}[i, k]$, in linear to their length time, since they do not capture possible placements. The implied lists are merged into list $T[i, k]$ of atoms, based on Lemma 1. We can easily show that $|T[i, k]| \leq 2|T[i-1, k]| + 3$. This implies that $|T[n, k]| = O(2^n)$, $n \geq k$. Also, by Lemma 2, we have that $|T[n, k]| \leq H$. However, by employing the following Lemma 3, we can improve on both of these bounds. Its correctness can easily be shown inductively, by proving that the distinct values that both coordinates of the pairs in $T[i, k]$ can receive are drawn from the sets $\{0, h, 2h, \ldots, ih\}$, $\{y_1, y_2, \ldots, y_i\}$, and $\bigcup_{j=1}^{i}\{y_j + h, y_j + 2h, \ldots, y_j + (i-1)h\}$.

Lemma 3. *List $T[n,k]$, $n \geq k$ contains $O(n^2)$ pairs.*

To prove the correctness of our algorithm, consider a pair $(a,b) \in T[i,k]$ that dominates pair $(c,d) \in T[i,k]$. Assume, for the sake of contradiction, that pair (a,b) yields a solution and pair (c,d) does not. That means that, for at least one pair out of $\{(y_i, b+h), (b, b+h), (a, \max\{b+h, a\}), (a+h, b)\}$ the boundary condition holds while the boundary condition does not hold for any of the pairs $\{(y_i, d+h), (d, d+h), (c, \max\{d+h, c\}), (c+h, d)\}$. This is impossible since $a \geq c$ and $b \geq d$. Therefore (a,b) can never be involved in an optimal solution and can be discarded. This implies that each list $T[i,k]$ should only contain atoms.

Each of the $(n+1) \times (n+1)$ entries of T is computed in $O(n^2)$ time. Thus, our algorithm terminates after $O(n^4)$ time. For a fixed label height h, the algorithm outputs a boolean value, which indicates whether there exists a legal label placement. This is done by identifying whether there exists a non-empty list $T[n,j]$, with $0 \leq j \leq n$. By using an extra table of the same size as T, our algorithm can easily be modified, such that it also computes the label and leader positions.

Theorem 1. *Given a rectangle R of integer height H and a set $P \subset R$ of n points (sites) in general positions, there exists an $O(n^4 \log H)$ time algorithm that produces a legal multi-stack labeling with two stacks of labels and with type-opo leaders such that the uniform integer height of the labels is maximum.*

Proof. In order to solve the *size maximization problem* (i.e. to determine the maximum height of the labels), we can simply apply a binary search on all possible discrete values for height h. To complete the proof, observe that $\frac{H}{n} \leq h \leq \frac{2H}{n}$. □

3.2 Sample Labelings

Figures 7 and 8 are produced from the algorithm of Section 3.1 and depict two regional maps of UK and Italy, respectively. The labels occupy two stacks to the east side of the enclosing rectangle. In both labelings the label size is maximum.

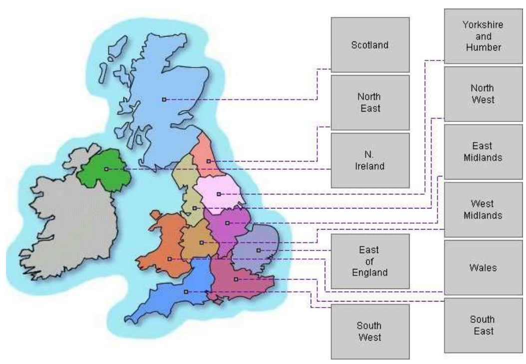


Fig. 7: A regional map of UK.

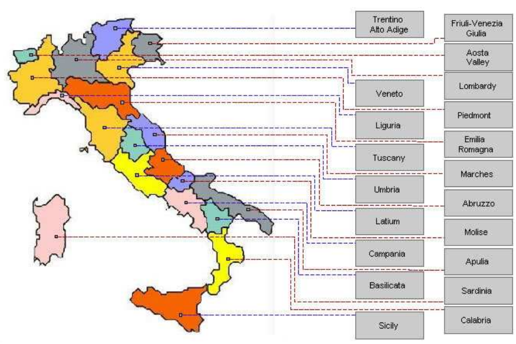
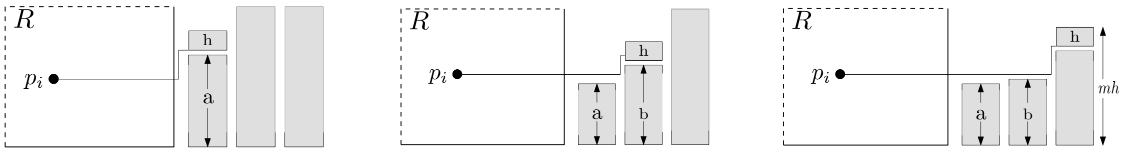


Fig. 8: A regional map of Italy.

3.3 Three Stacks of Labels on the Same Side

In this section, we extend the algorithm of Section 3.1 to support an additional stack of labels. We consider the case, where leaders connected to labels of the i-th stack are restricted to bend in the i-th track routing area. The objective, again, is to maximize the uniform height h of all labels.



(a) Placing l_i in 1st stack. **(b)** Placing l_i in 2nd stack. **(c)** Placing l_i in 3rd stack.

Fig. 9: Different placements obtained for the label of the i-th site. In Figure 9a: $a \in T[i-1,k,m]$, in Figure 9b: $(a,b) \in T[i-1,k-1,m]$, whereas in Figure 9c: $(a,b) \in T[i-1,k,m-1]$

Theorem 2. *Given a rectangle R of integer height H and a set $P \subset R$ of n points (sites) in general positions, there exists an $O(n^4 \log H)$ time algorithm that produces a legal multi-stack labeling with three stacks of labels and with type-opo leaders such that the uniform integer height of the labels is maximum and the leaders connected to labels of the i-th stack are restricted to bend in the i-th track routing area.*

Proof. We use dynamic programming algorithm employing a table T of size $(n+1)\times(n+1)\times(n+1)$. For each $i \geq k+m$, entry $T[i,k,m]$ contains a list of pairs (a,b), where a (b) is the Y-coordinate of the first (second) stack, that is needed to place the first i labels, when m labels are placed in the third stack, k labels are placed in the second stack and $i-k-m$ labels are placed in the first stack. Note that the height of the third stack is mh, since all leaders connected to labels of the third stack are restricted to bend in the third track routing area. List $T[i,k,m]$ is empty, when it is impossible to route the first i labels using k labels in the second stack and m in the third stack. This implies that table entries $T[i,k,m]$, where $i < k+m$, contain empty lists. Following similar arguments as in Section 3.1, entry $T[i,k,m]$ can be computed based on the following recurrence relation:

$$T[i,k,m] = \textsc{Merge}\{T_1[i,k,m], T_2[i,k,m], T_3[i,k,m]\} \tag{1}$$

where:

$$T_1[i,k,m] = \{(a \oplus_H h, b):\ \forall (a,b) \in T[i-1,k,m]\}$$
$$T_2[i,k,m] = \{(y_i, b \oplus_H h):\ \forall (a,b) \in T[i-1,k-1,m] \text{ s.t. } a < y_i\}$$
$$T_3[i,k,m] = \{(y_i, y_i):\ \forall (a,b) \in T[i-1,k,m-1], \text{ s.t. } mh \leq H \text{ and } (a,b) < (y_i,y_i)\}$$

List $T_1[i,k,m]$ of Equation 1 captures placements of the i-th label in the first stack (see Figure 9a). Similarly, list $T_2[i,k,m]$ of Equation 1 captures placements of the i-th label in the second stack. Since we assumed that leaders connected to labels of the second stack are restricted to bend in the second track routing area, this is possible only for pairs $(a,b) \in T[i-1,k-1,m]$ with $a \leq y_i$ (see Figure 9b). Finally, list $T_3[i,k,m]$ of Equation 1 captures placements of the i-th label in the third stack. This is possible only for pairs $(a,b) \in T[i-1,k,m-1]$ with $(a,b) \leq (y_i,y_i)$ (see Figure 9c). To compute entry $T[i,k,m]$, we first remove all pairs (∞,a), (a,∞) from lists $T_1[i,k,m]$, $T_2[i,k,m]$ and $T_3[i,k,m]$ and then we merge the implied lists to $T[i,k,m]$ of atoms, based on Lemma 1.

Lemma 4. *For $n \geq k+m$, $|T[n,k,m]| \leq n+1$.*

Proof. Lists $T_2[i,k,m]$ and $T_3[i,k,m]$ contain pairs of numbers with the same first coordinate. This implies that they contribute at most one non-dominating pair, while list $T_1[i,k,m]$ contains at most i elements, since $|T[i-1,k,m]| \leq i$. Thus, $T[i,k,m] \leq i+1$. □

Each of the $(n+1)\times(n+1)\times(n+1)$ entries of T is computed in $O(n)$ time. Thus, our algorithm terminates after $O(n^4)$ time. For a fixed label height h, the algorithm outputs a boolean value, which indicates whether there exists a legal label placement. This is done by identifying whether there exists a non-empty list $T[n, i, j]$, with $0 \leq i + j \leq n$. By using an extra table of the same size as T, our algorithm can easily be modified, such that it also computes the label and leader positions. In order to solve the *size maximization problem* (i.e. to determine the maximum height of the labels), we can simply apply a binary search on all possible discrete values for height h. To complete the proof, observe that $\frac{H}{n} \leq h \leq \frac{3H}{n}$. □

4 Computational Complexity of Multi-Stack Labeling Problem

In this section, we investigate the computational complexity of several multi-stack boundary labeling problems with either type-*opo* or *po* leaders and labels of arbitrary size, which can be placed at two stacks on the same side of the enclosing rectangle. Without loss of generality, we assume that the labels are located to the east side of the enclosing rectangle. We consider several different type of sites. In the most applicable case, site s_i is associated with a point $p_i = (x_i, y_i)$ on the plane. However, we also consider the cases, where site s_i is associated with either two candidate points $p_i^1 = (x_i^1, y_i^1)$ and $p_i^2 = (x_i^2, y_i^2)$ on the plane (see Figure 3) or with a vertical line segment, so that the site "slides" along the boundary of the proposed line segment (see Figure 4). The assumed models are quite general, since we allow sliding labels with sliding ports.

4.1 Line sites with type-*opo* leaders at two stacks on one side.

We focus on type-*opo* leaders, where each site s_i can slide along a line segment parallel to the y-axis and is associated with a label l_i of height h_i. We seek to find a legal labeling.

Theorem 3. *Given a rectangle R of height H, a set $P \subset R$ of n line segments (sites) that are parallel to the y-axis and a label of height h_i for each site $s_i \in P$, it is NP-hard to place all labels at two stacks on one side of R with non-intersecting type-opo leaders.*

Proof. We reduce the PARTITION problem [8] to our problem. The PARTITION problem is defined as follows: Given positive integers $a_1, a_2, \ldots, a_m$, is there a subset I of $J = \{1, 2, \ldots, m\}$ such that $\sum_{i \in I} a_i = \sum_{i \in J-I} a_i$? We will reduce an instance A of PARTITION to an instance B of our problem, such that instance A can be partitioned if and only if there exists a legal labeling for instance B.

The reduction we propose is somewhat immediate. Our site set $P = \{s_1, s_2, \ldots, s_m\}$ consists of m (parallel to y-axis) line segments of identical length $H = \frac{1}{2} \sum_{i \in J} a_i$ (H is the height of the enclosing rectangle R; see Figure 10). Each site s_i is also associated with a label l_i of height a_i. Both stacks contribute $2H$ height, which is equal to the sum of all label heights.

It is clear that if there exists a labeling for instance B, the labels should be partitioned into two sets such that the sum of the label heights of each set sums up to H. Therefore, the indices of the sites whose labels lie in the first stack imply the desired partition I of J.

Suppose now that there exists a subset I of $J = \{1, 2, \ldots, m\}$ such that $\sum_{i \in I} a_i = \sum_{i \in J-I} a_i$. Without loss of generality, we further suppose that $|I| \geq |J - I|$. For each site s_i with $i \in I$, we choose to place its label to the first stack. The remaining labels are placed to the second stack. To complete the reduction, we have to describe how to connect each site with its corresponding label such that the implied labeling is legal.

The leaders of the sites whose labels lie in the first stack are of type-o, i.e. horizontal line segments with no bends. In this case, the ports of both sites and labels can be chosen arbitrarily. Since the labeling is *tight* (i.e. the sum of the label heights on each stack is equal to H), we use the fact that the labels are open and use the gaps between them as *corridors* to route the leaders, whose labels lie in the second stack. Since we assumed that $|I| \geq |J - I|$, there exist enough corridors to route all leaders, adopting the following scenario: The leader which corresponds to the lowest label that has not been routed yet, can use the lowest available corridor. In this case the site ports are defined based on the corridors, whereas the label ports can be chosen arbitrarily again. □

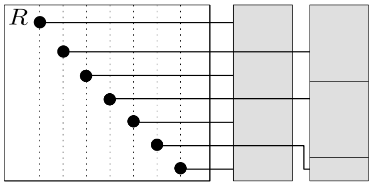


Fig. 10: Instance to which PARTITION is reduced.

4.2 Two candidate points with type-*opo* leaders at two stacks on one side.

We will show that the problem remains NP-hard even if we restrict ourselves in sites, which may have two candidate points, i.e. leader of site s_i connects either point $p_i^1 = (x_i^1, y_i^1)$ or point $p_i^2 = (x_i^2, y_i^2)$ with label l_i. To show NP-hardness, we reduce the following variant of Partition to our problem. The NP-hardness of this problem follows easily from the EVEN ODD PARTITION problem (see [8]).

Lemma 5 (RPartition). *Given $2m$ non-negative integers $a_1, a_2, \ldots a_{2m}$, the problem of finding a subset I of $J = \{1, 2, \ldots 2m\}$ such that the following three conditions are satisfied is NP-hard.*

1. *I contains exactly one of $\{2i-1, 2i\}$ for $i = 1, 2, \ldots m$*
2. $\sum_{i \in I} a_i = \sum_{i \in J-I} a_i$
3. $\sum_{i \in I \& i \leq k} a_i < \sum_{i \in J-I \& i \leq k} a_i$ *for $k = 2, 4, \ldots 2m-2$*

Theorem 4. *Given a rectangle R of height H, a set $P \subset R$ of n sites, each associated with two candidate points, and a label of height h_i for each site $s_i \in P$, it is NP-hard to place all labels at two stacks on one side of R with non-intersecting type-opo leaders.*

Proof. We will reduce an instance A of RPARTITION to an instance B of our problem, such that A can be partitioned if and only if there exists a legal labeling for instance B. Let C be a very large number, e.g. $C = (2m+1)^2 \sum_{i \in J} a_i$. Set $P = \{s_1, s_2, \ldots, s_{2m}\}$ consists of $2m$ sites. Site s_i is associated with $p_i^1 = (x_i, y_i^1)$ and $p_i^2 = (x_i, y_i^2)$. Consecutive sites s_{2i-1} and s_{2i}, $i = 1, 2, \ldots, m$, form m parallelograms r_i, $i = 1, 2, \ldots, m$, such that $y_{2i-1}^1 < y_{2i}^1 < y_{2i-1}^2 < y_{2i}^2$ and $|y_{2i-1}^2 - y_{2i}^1| = \frac{a_{2i-1}+a_{2i}}{2} + 1$ (see Figure 11). We assume that parallelogram r_{i-1} is placed lower than r_i. The vertical distance between two consecutive parallelograms is C, whereas the vertical distance between the bottommost (topmost) parallelogram r_1 (r_m) and the bottommost (topmost respectively) side of the enclosing rectangle R is $C/2$. The height of the enclosing rectangle is $H = m(C+1) + \frac{1}{2}\sum_{i \in J} a_i$. The corresponding label l_i of site s_i, has height $h_i = C + a_i + 1$, thus $\sum_{i \in J} h_i = 2m(C+1) + \sum_{i \in J} a_i$. Observe, that both stacks contribute $2H$ height, which is equal to the sum of all label heights.

The construction ensures that the same number of labels are placed at the two stacks. For the sake of contradiction, suppose that the number of labels of the first stack is $m + \delta$, where $\delta \geq 1$. Then, the sum of the corresponding label heights is at least $(m+\delta)(C+1)$, which is greater than H, since $\delta C > \sum_{i \in J} a_i$, yielding to a contradiction.

If there exists a legal labeling L for instance B, the labels should be partitioned into two sets of equal cardinality, such that the sum of the label heights of each set sums up to H. Since the labeling L is tight, we use the fact that the labels are open and use the gaps between the labels of the first stack as corridors for the leaders of the sites routed at the second stack. Observe, that due to the large label heights and the point locations of the sites, all leaders should bend in the first track routing area. Two consecutive sites s_{2i-1} and s_{2i}, $i = 1, 2, \ldots m$ can not have their labels both at the first stack (consequently at the second stack), because at least one corridor is lost and therefore at least one label at the second stack can not be routed. To avoid leader crossings, the order of indices should be preserved at both stacks, i.e. label l_i will be stacked lower than l_j if

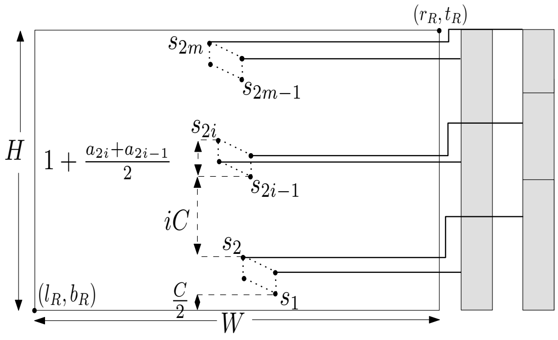


Fig. 11: Instance to which RPartition is reduced.

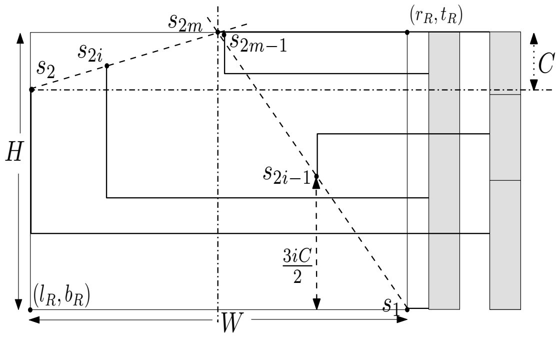


Fig. 12: Instance to which RPartition is reduced.

$i < j$. To connect all sites with their labels, it must either hold $\sum_{i \in I \& i \leq k} h_i < \sum_{i \in J-I \& i \leq k} h_i$ or $\sum_{i \in I \& i \leq k} h_i > \sum_{i \in J-I \& i \leq k} h_i$ for all $k = 2, 4, \ldots 2m-2$, which is equivalent to the third condition of RPartition, since all labels are augmented by $C + 1$. Therefore, the indices of the sites whose labels lie on the first or on the second stack respectively imply the desired partition I of J.

Suppose that there exists a subset I of $J = \{1, 2, \ldots, 2m\}$ of instance A such that all three conditions of RPartition are satisfied. If $i \in I$, then the label of site s_i is placed at the first stack preserving the order of indices, i.e. label l_i will be stacked lower than l_j, if $i < j$. The remaining labels ($i \in J - I$) are placed to the second stack in the same manner. This ensures that all labels of the second stack can be hit by a leader. Suppose for the sake of contradiction that a label of the second stack, say the k-th from the bottom, can not be hit by a leader. This implies that the sum of the label heights of the $k-1$ bottommost labels of the second stack is greater than the corresponding sum of the k bottommost labels of the first stack, which is a contradiction due to the third condition of RPartition, and the facts that all labels are augmented by $C + 1$ and $C > \sum_{i \in J} a_i$. A legal labeling can be obtained as follows: Take the lowest site which has not been routed. If its label is to be placed at the second stack, use the lowest available corridor for its leader, else route it at the first stack with a type-o leader. Since two consecutive sites s_{2i-1} and s_{2i} form a parallelogram r_i, we can determine in constant time which point of both s_{2i-1} and s_{2i} will be used, such that their leaders do not intersect. □

4.3 Type-*po* leaders at two stacks on one side.

We focus on type-*po* leaders, where each site s_i corresponds to a point $p_i = (x_i, y_i)$ of the plane and is associated with a label l_i of height h_i. The labels are located to the east side of the enclosing rectangle at two stacks. We seek to find a legal labeling.

Theorem 5. *Given a rectangle R of height H, a set $P \subset R$ of n and a label of height h_i for each site $s_i \in P$, it is NP-hard to place all labels at two stacks on one side of R with non-intersecting type-po leaders.*

Proof. As in the proof of Theorem 4, we will reduce an instance $A = \{a_1, a_2, \ldots a_m\}$ of RPartition to an instance B of our problem, such that A can be partitioned if and only if there exists a legal labeling for B. Without loss of generality, we further suppose that $a_1 < a_2$.

Let C be a very large number, e.g. $C = (2m+1)^2 \sum_{i \in J} a_i$. Our point set $P = \{s_1, s_2, \ldots, s_{2m}\}$ consists of $2m$ sites. Site s_i is associated with a label l_i of height $h_i = C + a_i$. The height of the enclosing rectangle is $H = mC + \frac{1}{2} \sum_{i \in J} a_i$. Both stacks contribute $2H$ height, which is equal to the sum of all label heights. As shown in proof of Theorem 4, the construction ensures that the same number of labels are placed at the two stacks. In the construction we propose the odd and even indexed sites lie on two separate lines. A detailed description of their exact positions is given below.

- Sites $s_{2i-1},\ i = 1, 2, \ldots m-1$ lie on the line, which is defined from the points (r_R, b_R) and $(l_R + W/2, t_R)$, such that: $y_1 = b_R$ and $y_{2i-1} = b_R + (2i-1)C/2$, $i = 2, 3, \ldots m-1$.

- Sites $s_{2i},\ i = 1, 2, \ldots m$ lie on the line, which is defined from points $(l_R, t_R - C)$ and $(l_R + W/2, t_R)$, such that: $y_2 = t_R - C$ and $y_{2i+2} - y_{2i} = \frac{C}{m}$, $i = 1, 2, \ldots, m-1$.

- Site s_{2m-1} is placed at $x_{2m-1} = \frac{x_{2m} + x_{2m-3}}{2}$ and such that $0 < (y_{2m} - y_{2m-1}) \to 0$

Such a construction is depicted in Figure 12 and can be obtained in linear time. The last condition (i.e. $0 < (y_{2m-1} - y_{2m}) \to 0$) ensures that no horizontal leader can lie between s_{2m-1} and s_{2m}.

Suppose that there exists a subset I of $J = \{1, 2, \ldots, 2m\}$ of instance A, such that all conditions of RPartition are satisfied. If $i \in I$, then the label of site s_i is placed at the first stack preserving the order of indices, i.e. if $i < j$, label l_i will be stacked lower than l_j. The remaining labels are placed at the second stack in the same manner. Following similar arguments as in proof of Theorem 4, we can show that there exists no label which can not be hit by a leader. A legal labeling can be obtained as follows: Take the lowest site which has not been routed. If its label is to be placed at the second stack, use the lowest available corridor for its leader, else route it at the first stack.

In the other direction, suppose that there exists a legal labeling L for instance B. Then, the labels should be partitioned into two sets of equal cardinality, such that the sum of the label heights of each set sums up to H. Since the labeling L is tight, we use the fact that the labels are open and use the gaps between the labels of the first stack as corridors for the leaders of the sites routed at the second stack.

We first focus on the case, where label l_1 is placed at the first stack. Then it holds, for all $k = 1, 2, \ldots, m-1$, that the sum of the label heights of the k bottommost labels of the first stack is less than the sum of the heights of the k bottommost labels of the second stack, otherwise at least one label of the second stack would be unable to be connected to its site. Label l_1 can only be the bottommost label of the first stack, otherwise at least one corridor (the one which will be used for the leader of the bottommost label of the second stack) will be lost.

Similar claim can be proved for label l_2, which corresponds to the leftmost site s_2, i.e. that l_2 is the bottommost label of the second stack. For the sake of contradiction, suppose that l_2 is at the first stack. Obviously it can not be the bottommost label, since this is label l_1. If l_2 is the topmost label and the proposed leader is of

- *type-o*, then several crossings with leaders of even indexed sites or s_{2m-1} exist, a contradiction since L is legal.
- *type-po*, then the o segment of this leader either lies between s_{2m-1} and s_{2m} (a contradiction since $0 < (y_{2m-1} - y_{2m}) \to 0$) or between s_{2i-2} and s_{2i}, and hence several crossing with leaders of even indexed sites exist, a contradiction since L is legal.

Suppose that l_2 is the k-th bottommost label (of the first stack), where $1 < k < m$. In this case labeling L has to contain crossings, because only $k-1$ sites can be connected to the $(k-1)$-th bottommost labels of both stacks (totally $2k-2$ labels) without crossings. Therefore, l_2 should be at the second stack. If l_2 is the topmost label of this stack, then its leader would overlap with site s_{2m}. Using similar arguments as in the case of l_2 being the k-th bottommost label at the first stack, we conclude that label l_2 can only be the bottommost label of the second stack.

Inductively, we can show that two sites s_{2i-1} and s_{2i}, $i = 1, 2, \ldots m$ can not have their labels both at the same stack and that label l_i should be stacked lower than l_j if both l_i and l_j belong to the same stack and $i < j$. So, the indices of the sites whose labels lie at the first stack imply the desired partition I of J, since we have supposed that $a_1 < a_2$ and l_1 lies in the first stack. Similarly, we can prove that if l_1 is placed at the second stack, then the desired partition I of J is formed from the indices of the labels at the second stack. □

Since, each point site can be thought as a line site of zero length, Corollary 1 follows immediately.

Corollary 1. *Given a rectangle R of height H, a set $P \subset R$ of n lines (sites) and a label of height h_i for each site $s_i \in P$, it is NP-hard to place all labels at two stacks on one side of R with non-intersecting type-po leaders.*

Following similar arguments as in proofs of Theorems 4 and 5, one can show that the problem remains NP-hard even if we restrict ourselves in sites, which may have two candidate points, i.e. leader of site s_i connects either point $p_i^1 = (x_i^1, y_i^1)$ or point $p_i^2 = (x_i^2, y_i^2)$ with label l_i.

Theorem 6. *Given a rectangle R of height H, a set $P \subset R$ of n sites, each associated with two candidate points, and a label of height h_i for each site, it is NP-hard to place all labels at two stacks on one side of R with non-intersecting type-po leaders.*

5 Open Problems and Future Work

For multi-stack labeling problems we presented results only for the label size maximization problem and the legal leader-label placement. No results are known regarding the total leader length minimization and the minimization of the total number of bends. Another line of research is to design good approximation algorithms that solve the problems, that are proved to be NP-hard.

References

1. M. A. Bekos, M. Kaufmann, K. Potika, and A. Symvonis. Boundary labelling of optimal total leader length. In *Proc. 10th Panhellenic Conference On Informatics (PCI2005), LNCS 3746*, pages 80–89, 2005.
2. M. A. Bekos, M. Kaufmann, K. Potika, and A. Symvonis. Polygon labelling of minimim leader length. In *Asia Pacific Symposium on Information Visualisation (APVIS2006), CRPIT 60*, pages 15–21, 2006.
3. M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry: Theory and Applications.* To appear.
4. M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. In *Proc. 12th Int. Symposium on Graph Drawing (GD'04), LNCS 3383*, pages 49–59, 2004.
5. M. A. Bekos and A. Symvonis. Bler: A boundary labeller for technical drawings. In *Proc. 13th Int. Symposium on Graph Drawing (GD'05), LNCS 3843*, pages 503–504, 2005.
6. P. Eades, T. Lin, and X. Lin. Minimum size h-v drawings. In *Advanced Visual Interfaces (Proceedings of AVI '92), volume 36 of World Scientific Series in Computer Science*, pages 386–394, 1992.
7. M. Formann and F. Wagner. A packing problem with applications to lettering of maps. In *Proc. 7th Annuual ACM Symposium on Computational Geometry (SoCG'91)*, pages 281–288, 1991.
8. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, NY, 1979, 1979.
9. SmartDraw-7. Product web site: http://www.smartdraw.com.
10. L. Stockmeyer. Optimal orientations of cells in slicing floorplan designs. *Information and Control*, 57(2–3):91–101, 1983.
11. A. Wolff and T. Strijk. The Map-Labeling Bibliography. http://i11www.ira.uka.de/map-labeling/bibliography, 1996.