Boundary Labeling with Octilinear Leaders*,**

M.A. Bekos[1], M. Kaufmann[2], M. Nöllenburg[3], and A. Symvonis[1]

[1] School of Applied Mathematical & Physical Sciences,
National Technical University of Athens, Greece
[2] Institute for Informatics, University of Tübingen, Germany
[3] Faculty of Informatics, Karlsruhe University, Germany

Abstract. A major factor affecting the readability of an illustration that contains textual labels is the degree to which the labels obscure graphical features of the illustration as a result of spatial overlaps. Boundary labeling addresses this problem by attaching the labels to the boundary of a rectangle that contains all features. Then, each feature should be connected to its associated label through a polygonal line, called *leader*, such that no two leaders intersect.

In this paper we study the boundary labeling problem along a new line of research, according to which different pairs of type leaders (i.e. *do* and *pd*, *od* and *pd*) are combined to produce boundary labelings. Thus, we are able to overcome the problem that there might be no feasible solution when labels are placed on different sides and only one type of leaders is allowed. Our main contribution is a new algorithm for solving the total leader length minimization problem (i.e., the problem of finding a crossing free boundary labeling, such that the total leader length is minimized) assuming labels of uniform size. We also present an NP-completeness result for the case where the labels are of arbitrary size.

1 Introduction

Placing extra information—usually in the form of textual labels—next to features of interest within an illustration, constitutes an important task in the process of information visualization. The interest in algorithms that automate this task has increased, due to the large number of applications that stem from diverse areas such as cartography, geographical information systems etc.

Current research on map labeling has been devoted to labeling point-features, so that each label is placed next to the point that it describes (an extensive bibliography about map labeling is maintained by Strijk and Wolff [14]). In this case, the basic requirement is that the labels should be pairwise disjoint. However, this is not always possible, e.g., in the case where the labels are too

---

large or the feature set is too dense. In practice, large labels are quite usual, e.g., in technical drawings, where it is common to explain certain features of the drawing with blocks of text, arranged on its boundary. As a response to this problem, Bekos et al. [4] proposed *boundary labeling*. In boundary labeling, the labels are attached to the boundary of a rectangle R enclosing all features and each feature is connected with its label by using polygonal lines, called *leaders*.

Several authors have proposed algorithms to produce boundary labelings in different settings [2,3,4,5,6,11]. Recently, Benkert et al. [5,6] studied the boundary labeling problem along a new line of research, according to which the leaders are of type *do*, i.e., polygonal lines consisting of two line segments, where the first one is "diagonal" to the side of R containing the label it leads to, whereas the second one is orthogonal to that side (see Figure 1c). Leaders of type *do* maintain a uniform shape and result in simple and easy-to-read labelings. However, in the work reported in [5] and [6], Benkert et al. study the case where the labels can be attached only to one side of R and they state that the production of a boundary labeling with such leaders is not always feasible. Extending their work, we examine the case of four-sided boundary labeling. We also introduce two new types of leaders and we show that by combining them, the boundary labeling problem is always feasible. To the best of our knowledge, this is the first attempt, where different types of leaders are combined to produce boundary labelings.

2   Problem Definition

The *input* of a boundary labeling problem consists of a set P of n points (referred to as *sites*) $s_i = (x_i, y_i)$, $i = 1, 2, \ldots n$. The site set P is enclosed in an axis-parallel rectangle $R = [0, W] \times [0, H]$, which is called *enclosing rectangle*. Each site s_i is associated with an axis-parallel, $w_i \times h_i$ rectangular label l_i.

The *output* of a boundary labeling problem is a placement of the labels at distinct positions on the boundary of R and a set of leaders connecting each site with its associated label, so that i) the labels do not overlap with each other and ii) the leaders do not intersect or overlap with each other. Such labelings are referred to as *legal boundary labelings* (or simply as *legal labelings*).

Following the naming scheme of Bekos et al. [4], we focus on three different types of leaders, each of which consists of two line segments:

Type-*od* leaders: The first line segment of a leader of type *od* is orthogonal (*o*) to the side of R containing the label it leads to. Its second line segment is "diagonal" (*d*) to that side (see Figure 1a).

Type-*pd* leaders: The first line segment of a leader of type *pd* is parallel (*p*) to the side of R containing the label it leads to. Its second line segment is "diagonal" (*d*) to that side (see Figure 1b).

Type-*do* leaders: The first line segment of a leader of type *do* is "diagonal" (*d*) to the side of R containing the label it leads to. Its second line segment is orthogonal (*o*) to that side (see Figure 1c).

(a) *od*-leaders   **(b)** *pd*-leaders   **(c)** *do*-leaders
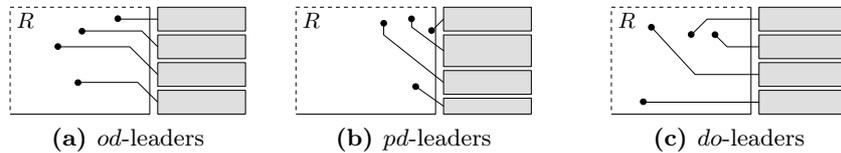
Fig. 1. Different types of leaders

In general, the labels are of arbitrary size (*non-uniform labels*; see Figure 1b). We separately consider the case, where the labels are of the same width and height (*uniform labels*; see Figures 1a and 1c). We further assume that the point where each leader touches its associated label (referred to as *port*) is fixed, e.g., the middle point of the label's side that faces the enclosing rectangle R (see Figures 1a, 1b and 1c). Also, the 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*).

Keeping in mind that we want to obtain simple and easy-to-read labelings, we consider the *leader length minimization problem*, i.e., the problem of determining a legal labeling, such that the total leader length is minimized.

2.1   Preliminaries

We denote the number of sites (and consequently the number of labels) by n. We also denote by c_i the leader of site s_i. A set of sites is considered to be in *general position* if i) no three sites are collinear, ii) no two sites share the same x- or y-coordinate, iii) no two sites lie on the same diagonal line and iv) the horizontal, vertical and diagonal lines that pass through the ports of the labels do not coincide with the sites. In order to avoid leader overlaps, we usually assume that the input site set P is in general position. We also assume that the sites, the leader bends and the label corners have integer coordinates. Consider a leader c_i which originates from site s_i and is connected to a label l_i on the right side AB of R. The horizontal line which coincides with s_i divides the plane into two half-planes (see the dashed line l of Figure 2). We say that leader c_i is *oriented towards* corner A if both A and the port of label l_i are on the same half-plane, otherwise, we say that leader c_i is *oriented away* from corner A.

Consider a site s_i that has to be connected to a label l_i on the right side AB of R. The lines that pass through the port of label l_i and form 45°, 90° and 135° angles with the left side of label l_i, partition R into four regions $R_{i,1}$, $R_{i,2}$, $R_{i,3}$ and $R_{i,4}$, as in Figure 3. If the site s_i lies within a region incident to A or B (i.e., $R_{i,1}$ or $R_{i,4}$; refer to the light-gray colored regions of Figure 3), then it can only be connected to label l_i using a leader of type *pd*. Otherwise (i.e., site s_i lies within $R_{i,2}$ or $R_{i,3}$; refer to the dark-gray colored regions of Figure 3), it can be connected to l_i using either a leader of type *do* or *od*. Also, observe that connecting a site to its label with a leader of type *do*, requires the same leader length as with a leader of type *od*. So, depending on the location of site s_i, one has to use an appropriate leader to connect it to its label l_i.
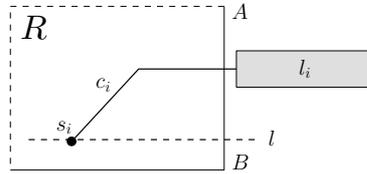
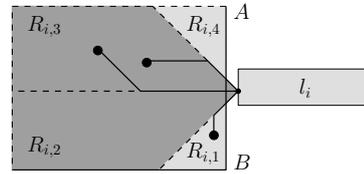**Fig. 2.** c_i is oriented towards corner A

Fig. 3. Connecting site s_i to label l_i

This paper is structured as follows: In Section 3 we prove that the problem of determining a legal boundary labeling of minimum total leader length with leaders of type *do* and *pd* and non-uniform labels is NP-complete. In Sections 4 and 5, we present polynomial time algorithms for obtaining either optimal (in terms of total leader length) or simply legal boundary labelings with labels of uniform size. We conclude in Section 6 with open problems and future work.

3   Boundary Labeling with Non-uniform Labels

In this section, we consider the boundary labeling problem with labels of non-uniform size. We are given a set P of n sites $s_i, i = 1, 2, \ldots n$, each associated with axis-parallel, rectangular label l_i of height h_i. The labels are allowed to be placed on the right side of the enclosing rectangle R. We further assume fixed label ports, i.e., each leader is connected to its corresponding label using the middle point of the label side that faces the enclosing rectangle. For the case where the sites can be placed in arbitrary position, i.e., the general position restriction is relaxed, we can prove:

Theorem 1. *Given a set P of n sites, a label l_i of height h_i for each site s_i and an integer $k \in \mathbb{Z}^+$, it is NP-complete to decide whether there exists a legal boundary labeling of total leader length no more than k assuming type do and pd leaders.*

Proof. Membership in NP follows from the fact that a nondeterministic algorithm needs only guess a positioning of the labels on the boundary of R, a set of leaders connecting each site with its associated label and check in polynomial time that i) the labels do not overlap with each other, ii) the leaders do not intersect with each other and iii) the sum of the lengths of all leaders is no more than k.

We will reduce the following single machine scheduling problem (known as *total discrepancy problem* [8]) to our problem: We are given a set J of $2n + 1$ jobs $J_0, J_1, J_2, \ldots, J_{2n}$, which are to be executed on one machine nonpreemptively and a single *preferred midtime* $M \in \mathbb{Z}^+$, which corresponds to the time at which we would like the first half of each job to be completed. Each job J_i is also associated with a known deterministic processing time p_i. Without loss of generality, we assume that M is large (e.g. $M > \sum_{i=0}^{2n} p_i$) and the jobs are ordered so that $p_i < p_j$, $\forall i < j$. Given a schedule σ, we denote the *starting*

(*completion*) *time* of job J_i in σ by $S_i(\sigma)$ $(C_i(\sigma))$ and we use $M_i(\sigma)$ to denote its *midtime*, i.e., $M_i(\sigma) = S_i(\sigma) + p_i/2$, or equivalently, $M_i(\sigma) = C_i(\sigma) - p_i/2$. Under a schedule σ, a job J_i is considered to be *on-time* if its midtime $M_i(\sigma)$ is equal to the preferred midtime M and in this case, it incurs no penalty. On the other hand, if the midtime $M_i(\sigma)$ of J_i commences prior to M (exceeds M), an earliness (tardiness) penalty $E_i(\sigma) = M - M_i(\sigma)$ $(T_i(\sigma) = M_i(\sigma) - M)$ incurs. The objective is to determine a schedule σ, so that the total earliness-tardiness penalty $\sum_{i=0}^{2n}(E_i(\sigma) + T_i(\sigma)) = \sum_{i=0}^{2n}|M - M_i(\sigma)|$ is minimized[1]. Let σ_{opt} be an optimal schedule of the total discrepancy problem. Then, the following hold [8]:

1) σ_{opt} does not have any gaps between the jobs.
2) $M_0(\sigma_{opt}) = M$.
3) If $A(\sigma_{opt}) = \{J_i : M_i(S) < M\}$ and $B(\sigma_{opt}) = \{J_i : M_i(S) > M\}$, then $|A(\sigma_{opt})| = |B(\sigma_{opt})| = n$.
4) $\sigma_{opt} = [A_n, A_{n-1}, \ldots, A_1, J_0, B_1, B_2, \ldots, B_n]$, where $\{A_i, B_i\} = \{J_{2i}, J_{2i-1}\}$, i.e., if $A_i = J_{2i}$ then $B_i = J_{2i-1}$ otherwise $A_i = J_{2i-1}$ and $B_i = J_{2i}$.
5) The minimum total earliness-tardiness penalty is equal to

$$ETP = \sum_{i=1}^{n}(p_{2i} + p_{2i-1})(n - i + 1/2) + np_0.$$

The reduction we propose, can be achieved in linear time. Let I_S be an instance of the total discrepancy problem mentioned above. We proceed to construct an instance I_L of our problem as follows: For each job J_i, we introduce a site s_i placed at point $(2n + 1 - i, M)$, i.e., the sites are collinear, lie on the horizontal line $y = M$ and the horizontal distance between two consecutive sites is one unit. The label l_i associated with site s_i has height h_i equal to the processing time p_i of job J_i. The bottom left corner of the enclosing rectangle R is $(0, 0)$. The height H of R is equal to $2M$, which ensures that all labels can be placed at the right side of R, since $M > \sum_{i=0}^{2n} p_i$. We seek to exclude the case where a site can be connected to its label through a leader of type pd. So, the enclosing rectangle should be of appropriate width. We set its width W to be equal to $\frac{\sqrt{2}}{2}H + 2n + 1$ (see Figure 4). This ensures that the gray-colored triangular area contains no sites and therefore, all sites can be connected to their associated labels through leaders of type do only.

Then, we can show that we can derive a schedule σ of I_S with total earliness-tardiness penalty ETP if and only if we can determine a legal labeling L of I_L with total leader length at most $(\sqrt{2} - 1)ETP + (2n + 1)(W - n - 1)$.   □

Note. The NP-completeness result of Theorem 1 also holds in the case of boundary labelings with *po* leaders. The proof is almost identical. Instead of measuring the length of each leader using the Euclidean metric, we have to use the Manhattan metric.

---

[1] Surveys on the most important aspects of scheduling research are given at [1,9].
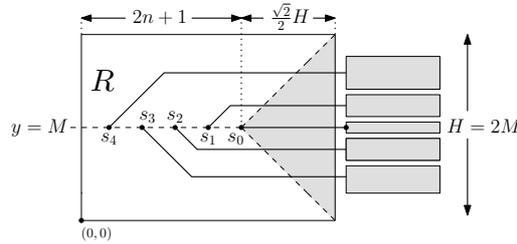
Fig. 4. For each job J_j, we introduce a site s_i placed at $(2n + 1 - i, M)$

4   Boundary Labeling with Uniform Labels

Theorem 1 implies that, unless $P = NP$, we cannot efficiently determine an optimal solution of the boundary labeling problem with non-uniform labels. Therefore, we proceed to consider the case of uniform labels, which is a reasonable assumption, since in real applications the labels usually contain single line texts (for example a place name or an integer used as a legend).

Let $P = \{s_1, s_2, \ldots s_n\}$ and $L = \{l_1, l_2, \ldots l_n\}$ be the sets of sites and labels, respectively. We assume that the sites are in general position and the labels are placed in fixed positions on the boundary of R. Since the labels are of uniform size, each site s_i can be connected to any label l_j. We seek to connect each site s_i to a label l_j, so that the total leader length is minimized.

Initially, we construct a complete weighted bipartite graph $G = (P \cup L, E, w)$ between all sites $s_i \in P$ and all labels $l_j \in L$, where $E = \{(s_i, l_j); s_i \in P, l_j \in L\}$ and $w : E \to \mathbb{R}$ is a cost function (see step A of Algorithm 1). Each edge $e_{ij} = (s_i, l_j) \in E$ of G is assigned a weight $w(e_{ij}) = d_{ij}$, where d_{ij} is equal to the length of the leader which connects site s_i with label l_j. Recall that the type of the leader that will be used to connect site s_i to label l_j depends on their relative positions, as stated in Section 2.1. Also, recall that if a site can be connected to its associated label with a leader of type do, it can also be connected using an od leader. However, in both cases the total length required is the same and, consequently, the edge e_{ij} is assigned the same weight, regardless the type of the leader that will eventually used (i.e., do or od). Observe that G is regular.

We proceed by computing a minimum-cost bipartite matching on G, i.e., a matching between the sites and the labels that minimizes the total weight of the matched pairs (see step B of Algorithm 1). Since G is regular and bipartite, by Hall's theorem a perfect matching exists [10]. Then, we obtain a labeling M of minimum total leader length as follows: If an edge $e_{ij} = (s_i, l_j) \in E$ is selected in the matching, then we connect site s_i with label l_j using a leader of length $w(e_{ij})$ (see step C of Algorithm 1). However, labeling M may contain crossings, which have to be eliminated while keeping the total leader length unchanged, i.e., equal to that of M (see step D of Algorithm 1). The crossing elimination procedure is described in the remainder of this section and depends on i) the location of the labels and ii) the type of the leaders that are used to produce M.

Algorithm 1. Generic Algorithm

---

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

output : A crossing free boundary labeling of minimum total leader length.

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

Step B: *Compute a Minimum Cost Bipartite Matching.*
    Compute a minimum-cost perfect bipartite matching $\mathcal{M}$ of G, i.e., compute a
    matching between sites and labels that minimizes the total distance of the
    matched pairs.

Step C: *Obtain an optimal boundary labeling M.*
    foreach (edge $e_{ij} = (s_i, l_j) \in E$) **do**
      if $e_{ij} = (s_i, l_j) \in \mathcal{M}$ **then**   connect site s_i to label l_j s.t. $length(c_i) = w(e_{ij})$

Step D: *Eliminate crossings.*
    Eliminate all crossings among pairs of leaders and obtain a legal boundary
    labeling M', keeping the total leader length unchanged, i.e., equal to that of M.

4.1   One-Sided Boundary Labeling

We first describe how to eliminate all crossings of labeling M (obtained in Step C of Algorithm 1), assuming that the labels are allowed to be attached to one side of the enclosing rectangle R, say the right side AB. Note that labeling M is of minimum total leader length and the leaders, we have used to produce it in Step C of Algorithm 1, are i) either of type *do* and *pd* or ii) of type *od* and *pd*. Our aim is to eliminate all crossings and obtain a legal labeling M' that keeps the total leader length unchanged.

Lemma 1. *Let M be an optimal one-sided boundary labeling either with type do and pd leaders or with type od and pd leaders (which may contain crossings) obtained in Step C of Algorithm 1. Let c_i and c_j be a pair of intersecting leaders originating from sites s_i and s_j, respectively. Then the following hold:*

  i) Leaders c_i and c_j are of the same type.
  ii) Leaders c_i and c_j are oriented towards the same corner, say A, of the enclosing rectangle R.
 iii) Leaders c_i and c_j can be rerouted so that they do not cross each other, the sum of their leader length remains unchanged, their type remains unchanged and they remain oriented towards corner A of R.

Sketch of proof. Due to space constraints, the detailed proof is omitted. It is based on an exhaustive case analysis on a) the types of the two leaders, b) the orientation of the two leaders (towards the same or different corners) and c) the different regions the two sites may reside. It also makes use of several geometric

properties (e.g. triangle inequality, properties of isosceles triangles, orthogonal triangles etc.). Another important property that is heavily used is the assumption that the sites are in general position. □

Lemma 2. *Let M be an optimal one-sided boundary labeling either with type do and pd leaders or with type od and pd leaders (which may contain crossings) obtained in Step C of Algorithm 1. We can always determine a crossing-free labeling M' with total leader length equal to that of M (step D of Algorithm 1). Moreover, labeling M' can be obtained in $O(n^2)$ time.*

Proof. By Lemma 1, it follows that leaders involved in a crossing are of the same type and oriented towards the same corner of R. We show how to eliminate all crossings of labeling M by rerouting the crossing leaders. Our method performs four passes over the sites. In the first and second pass, we eliminate all crossings among the leaders of type pd, which are oriented towards the top right and bottom right corner of R, respectively. In the third and fourth pass, we eliminate all crossings among the remaining leaders (i.e. either leaders of type do or of type od), which are oriented towards the top right and bottom right corner of R, respectively. Due to space constraints, we describe in detail the first pass only.

We examine these sites from right to left. We are interested only in those sites, that have crossing leaders. Let s_i be the first such site and let c_i be the leader that connects it to its corresponding label on the right side AB of R (see the left part of Figure 5). By Lemma 1.(i) and Lemma 1.(ii), all leaders that intersect c_i are also, of type pd, oriented towards corner A. Let s_k be the site whose leader c_k intersects c_i and its label is placed bottommost. From Lemma 1.(iii), it follows that we can reroute leaders c_i and c_k so that the total leader length remains unchanged (see the right part of Figure 5). Note that the rerouting possibly eliminates more than one crossing but, in general, it may also introduce new crossings with other type pd leaders, oriented towards corner A. However, the crossings are now, located to the left of the vertical line that coincides with s_i (within the gray-colored region of Figure 5). Continuing in the same manner, the line which forms the region containing the crossings in the right-to-left pass is pushed to the left (i.e., the area of this region is reduced at each iteration, in the right-to-left pass), which guarantees that all crossings among leaders of type pd that are oriented towards corner A, are eventually eliminated.

When the four independent passes over the site set are completed, we have eliminated all crossings, resulting in a labeling M' without any crossings and of total leader length equal to that of M, i.e., of minimum total leader length. To complete the proof of the lemma, it remains to explain how to obtain in $O(n^2)$ time the new labeling M'. At each pass, we sort appropriately the site set. This can be done in $O(n \log n)$ time. At each iteration over the sorted sets of sites, we are interested in finding a specific site, which crosses the leader of the site that we currently consider. In a straight-forward manner, this can be computed in $O(n)$ time. This results in a total of $O(n^2)$ time for each pass and, consequently, for the elimination of all crossings. □
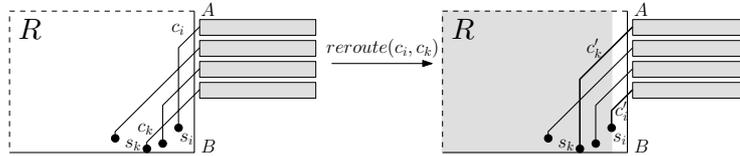
Fig. 5. Rerouting the crossing *pd*-leaders c_i and c_k

Theorem 2. *Given a set P of n sites and a set L of n labels of uniform size placed at fixed positions on one side of the enclosing rectangle R, we can compute in $O(n^3)$ total time a legal boundary labeling of minimum total leader length with type do and pd leaders.*

Proof. In Step A of Algorithm 1, we construct a complete weighted bipartite graph $G = (P \cup L, E, w)$ between all sites $s_i \in P$ and all labels $l_j \in L$, where the weight of an edge $e_{ij} = (s_i, l_j) \in E$ is the length of the leader connecting site s_i to label l_i. The computation of each edge weight requires constant time. Hence, the construction of G can be done in $O(n^2)$ time. In Step B of Algorithm 1, we compute a minimum cost bipartite matching on the graph G, which can be done by means of the Hungarian method in $O(n^3)$ time [12]. Note that we cannot use Vaidya's algorithm [13] to reduce the time complexity of Step B, since the leaders are neither straight lines (Euclidean metric) nor rectilinear (Manhattan metric). The solution obtained in Step C of Algorithm 1 is optimal. However, it may contain crossings. In Step D of Algorithm 1, the crossings are eliminated in $O(n^2)$ time. Thus, the total time complexity of Algorithm 1 for the case of one-sided boundary labeling with type *do* and *pd* leaders is $O(n^3)$.   □

4.2   Two-Sided Boundary Labeling

In this subsection, we consider the case where the labels are allowed to be attached to two opposite sides of R. Again, we use Algorithm 1 to obtain a boundary labeling M (not necessarily crossing-free) of minimum total leader length. This can be done in $O(n^3)$ time. We can observe that a possible crossing, between two leaders that lead to labels located at opposite sides of R, cannot occur, since the rerouting of the leaders c_i and c_j results in a solution with smaller total leader length. This result is summarized in the following lemma.

Lemma 3. *In an optimal two-sided boundary labeling, crossings between leaders that connect labels located at opposite sides of the enclosing rectangle, cannot occur.*

From Lemma 3, it follows that we can independently eliminate the crossings along the two opposite sides of R. The following theorem summarizes our result.

Theorem 3. *Given a set P of n sites and a set L of n labels of uniform size, placed at fixed positions, on two opposite sides of the enclosing rectangle R, we can compute in $O(n^3)$ total time a legal boundary labeling of minimum total leader length with either type do and pd leaders or with od and pd leaders.*

4.3 Four-Sided Boundary Labeling

In this subsection, we consider the general case of determining a legal boundary labeling of minimum total leader length with type *od* and *pd* leaders, where the labels are allowed to be attached to all four sides of R. Again, we use Algorithm 1, to obtain a labeling M of minimum total leader length. By Lemma 1, it follows that crossing leaders that connect labels placed at the same side of R, are of the same type, oriented towards the same corner of R. Crossings between leaders that connect labels placed at opposite sides of R cannot occur, because of Lemma 3. For the case, where the leaders which cross each other connect labels placed on two adjacent sides of R, we can show that the following lemma holds.

Lemma 4. *Let M be an optimal four-sided boundary labeling with type od and pd leaders (which may contain crossings) obtained in Step C of Algorithm 1. Let c_i and c_j be a pair of intersecting leaders originating from sites s_i and s_j, respectively. Let also l_i and l_j be their associated labels, which lie on two adjacent sides of the enclosing rectangle R. Then the following hold:*

- *i) Leaders c_i and c_j are of different type.*
- *ii) Leaders c_i and c_j are oriented towards their incident corner, say corner A.*
- *iii) Leaders c_i and c_j can be rerouted so that they do not cross each other, the sum of their leader length remains unchanged, their type remains unchanged and they remain oriented towards corner A of R.*

Lemma 5. *Let M be an optimal four-sided boundary labeling with type od and pd leaders (which may contain crossings) obtained in Step C of Algorithm 1. We can determine a legal labeling M' with total leader length equal to that of M (step D of Algorithm 1). Moreover, labeling M' can be obtained in $O(n^2)$ time.*

Proof. We partition the site set into four disjoint sets S_{TR}, S_{TL}, S_{BR} and S_{BL}, each of those contains the sites whose leaders are oriented towards the top-right, top-left, bottom-right and bottom-left corner of R, respectively (see Figure 6). From Lemma 1 (one side case), Lemma 3 (two opposite sides case) and Lemma 4 (two adjacent sides case), it follows that possible crossings can only occur between leaders that are oriented towards the same corner of R. Thus, we can independently eliminate the crossings at each of the sets S_{TR}, S_{TL}, S_{BR}, S_{BL}.

We describe in detail how to eliminate the crossings of S_{TR}. The remaining are treated similarly. We further partition S_{TR} into two disjoint subsets $S_{1,TR}$ and $S_{2,TR}$ as follows: $S_{1,TR}$ ($S_{2,TR}$) contains the sites of S_{TR} whose leaders are either i) of type *od* leading to a label placed at the right (top) side of R or ii) of type *pd* leading to a label placed at the top (right) side of R. In Figure 6, the sites that constitute $S_{1,TR}$ are the ones whose leaders are drawn as solid lines.

From Lemma 4.(i), it follows that the leaders, which are involved in a crossing and lead to labels placed at two adjacent sides of R, should be of different type. Furthermore, crossing leaders that connect labels placed at the same side of R, should be of the same type. This directly follows from Lemma 1.(i). Hence, we can independently eliminate the crossings at each of the sets $S_{1,TR}$ and $S_{2,TR}$.
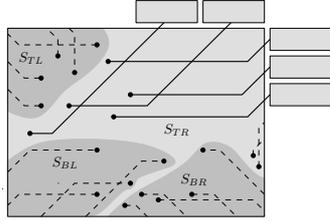
Fig. 6. Sets S_{TR}, S_{TL}, S_{BR} and S_{BL}
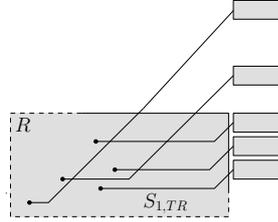
Fig. 7. Extending a *pd* leader

Let $s \in S_{1,TR}$ be a site whose leader c is of type *pd*. Leader c leads to a label at the top side of R. By extending its *d*-segment, leader c can be viewed as an *od* leader leading to a label at the right side of R (see Figure 7). This implies that we can make use of the algorithm described in the proof of Lemma 2 to eliminate all crossings of $S_{1,TR}$. Since all leaders of the sites of $S_{1,TR}$ have the same orientation (this holds because $S_{1,TR} \subseteq S_{TR}$), their *d*-segments are parallel to each other, which guarantees that all crossings will occur within R. This ensures that our approach will find a legal labeling. Similarly, we eliminate the crossings of the set $S_{2,TR}$. The total time needed to eliminate the crossings at each of the sets S_{TR}, S_{TL}, S_{BR} and S_{RL}, is $O(n^2)$. Thus, labeling M' can be obtained in $O(n^2)$. □

Theorem 4. *Given a set P of n sites and a set L of n labels of uniform size placed at fixed positions on all four sides of the enclosing rectangle R, we can compute in $O(n^3)$ total time a legal boundary labeling of minimum total leader length with type od and pd leaders.*

5   An Algorithm for Obtaining Legal Boundary Labelings

In this section, we consider the problem of determining a legal boundary labeling with type *od* and *pd* leaders, i.e., we relax the optimality constraint on the resulting labeling. Our aim is to obtain a more efficient algorithm in terms of time complexity.

Theorem 5. *Given a set P of n sites and a set L of n labels of uniform size placed at fixed positions on all four sides of the enclosing rectangle R, we can compute in $O(n^2)$ total time a legal boundary labeling with type od and pd leaders.*

Sketch of proof. Our basic idea is simple: We first develop an algorithm which determines a legal labeling in the case where the labels are attached to one side of R. Its time complexity is $O(n^2)$. Then, using standard plane sweep algorithms [7], we can in $O(n \log n)$ time partition R into four disjoint regions such that the previous algorithm can be applied to each region separately. To achieve this, we have two requirements for a region A in the partition of R: (a) A must be adjacent to a specific side s_A of R and (b) each site in A can be connected to any label attached to s_A either through a leader of type *do* or of type *pd*. □

6   Conclusions

In this paper, we studied boundary labelings with *do*, *od* and *pd* leaders. The focus of our work was on the leader length minimization problem. The $O(n^3)$ time complexity of the proposed algorithms is dominated by the computation of a minimum-cost bipartite matching. Unfortunately, we cannot use Vaidya's algorithm [13] to reduce it, since the leaders are neither straight lines (Euclidean metric) nor rectilinear (Manhattan metric). It is worth trying to derive a more efficient matching algorithm for this metric. The evaluation of different optimization criteria would also be of particular interest.

References

1. Baker, K.R., Scudder, G.D.: Sequencing with earliness and tardiness penalties: a review. Operations Research 38, 22–36 (1989)
2. Bekos, M.A., Kaufmann, M., Potika, K., Symvonis, A.: Polygons labelling of minimum leader length. In: Kazuo, M., Kozo, S., Jiro, T. (eds.) Proc. Asia Pacific Symposium on Information Visualisation, CRPIT 1960, pp. 15–21 (2006)
3. Bekos, M.A., Kaufmann, M., Symvonis, A., Wolff, A.: Boundary labeling: Models and efficient algorithms for rectangular maps. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 49–59. Springer, New York (2005)
4. Bekos, M.A., Kaufmann, M., Symvonis, A., Wolff, A.: Boundary labeling: Models and efficient algorithms for rectangular maps. Computational Geometry: Theory and Applications 36, 215–236 (2007)
5. Benkert, M., Haverkort, H., Kroll, M., Nöllenburg, M.: Algorithms for multi-criteria one-sided boundary labeling. In: Proc. 15th Int. Symposium on Graph Drawing (GD 2007). LNCS, vol. 4875, pp. 243–254. Springer, Heidelberg (2007)
6. Benkert, M., Nöllenburg, M.: Improved algorithms for length-minimal one-sided boundary labeling. In: 23rd European Workshop on Computational Geometry (EWCG 2007), pp. 190–193 (2007)
7. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer, Heidelberg (2000)
8. Garey, M., Tarjan, R., Wilfong, G.: One-processor scheduling with symmetric earliness and tardiness penalties. Mathematics of Operations Research 13, 330–348 (1988)
9. Gordon, V., Proth, J.-M., Chu, C.: A survey of the state-of-the-art of common due date assignment and scheduling research. European Journal of Operational Research 139(1), 1–25 (2002)
10. Hall, P.: On representation of subsets. Journal of the London Mathematical Society 10, 26–30 (1935)
11. Kao, H.-J., Lin, C.-C., Yen, H.-C.: Many-to-one boundary labeling. In: Proc. Asia Pacific Symposium on Information Visualisation (APVIS 2007), pp. 65–72. IEEE, Los Alamitos (2007)
12. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistic Quarterly 2, 83–97 (1955)
13. Vaidya, P.M.: Geometry helps in matching. SIAM J. Comput. 18, 1201–1225 (1989)
14. Wolff, A., Strijk, T.: The Map-Labeling Bibliography (1996),
    http://i11www.ira.uka.de/map-labeling/bibliography