Many-to-One Boundary Labeling with Backbones

M. A. Bekos[1], S. Cornelsen[2], M. Fink[3], S. Hong[4], M. Kaufmann[1],
M. Nöllenburg[5], I. Rutter[5], and A. Symvonis[6]

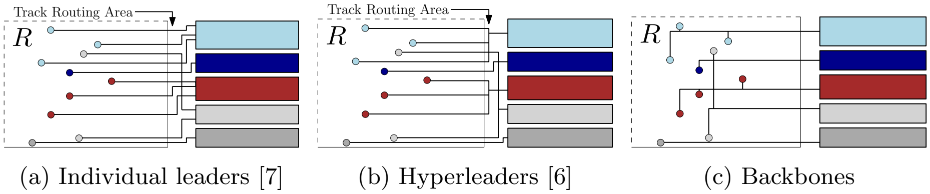
[1] Institute for Informatics, University of Tübingen, Germany
{bekos,mk}@informatik.uni-tuebingen.de
[2] Department of Computer and Information Science, University of Konstanz
sabine.cornelsen@uni-konstanz.de
[3] Lehrstuhl für Informatik I, Universität Würzburg, Germany
martin.a.fink@uni-wuerzburg.de
[4] School of Information Technologies, University of Sydney
shhong@it.usyd.edu.au
[5] Karlsruhe Institute of Technology, Germany
noelle@ira.uka.de, rutter@kit.edu
[6] School of Applied Mathematics and Physical Sciences, NTUA, Greece
symvonis@math.ntua.gr

Abstract. In this paper we study *many-to-one boundary labeling with backbone leaders*. In this model, a horizontal backbone reaches out of each label into the feature-enclosing rectangle. Feature points associated with this label are linked via vertical line segments to the backbone. We present algorithms for label number and leader-length minimization. If crossings are allowed, we aim to minimize their number. This can be achieved efficiently in the case of fixed label order. We show that the corresponding problem in the case of flexible label order is NP-hard.

1 Introduction

Boundary labeling was developed by Bekos et al. [2] as a framework and an algorithmic response to the poor quality (feature occlusion, label overlap) of specific labeling applications. In boundary labeling, labels are placed at the boundary of a rectangle and are connected to their associated features via arcs referred to as *leaders*. Leaders attach to labels at *label ports*. A survey by Kaufmann [4] presents different boundary labeling models that have been studied so far.

In *many-to-one boundary labeling* each label is associated to more than one feature point. This model was formally introduced by Lin et al. [7], who assumed that each label has one port for each connecting feature point (see Fig. 1a) and showed that several crossing minimization problems are NP-complete and, subsequently, developed approximation and heuristic algorithms. In a variant of this model, referred to as *boundary labeling with hyperleaders*, Lin [6] resolved the multiple port issue by joining together all leaders attached to a common label with a vertical line segment in the track-routing area (see Fig. 1b). At the cost of label duplications, leader crossings could be eliminated.



(a) Individual leaders [7] (b) Hyperleaders [6] (c) Backbones

Fig. 1. Different types of many-to-one labelings.

We study *many-to-one boundary labeling with backbone leaders* (for short, *backbone labeling*). In this model, a horizontal backbone reaches out of each label into the feature-enclosing rectangle. Feature points that need to be connected to a label are linked via vertical line segments to the label's backbone (*backbone leaders*; see Fig. 1c). Formally, we are given a set $P = \{p_1, \dots, p_n\}$ of n points in an axis-aligned rectangle R, where each point $p \in P$ is assigned a color $c(p)$ from a color set C. We also assume that the points are in general position and sorted in decreasing order of y-coordinates, with p_1 being the topmost. Our goal is to place colored labels to the left or right side of R and assign each point $p \in P$ to a label $l(p)$ of color $c(p)$. A *backbone labeling* for a set of colored points P is a set $\mathcal{L}$ of colored labels and a mapping of each point $p \in P$ to some $c(p)$-colored label in $\mathcal{L}$, so that (i) each point is connected to a label of the same color, and (ii) there are no backbone leader overlaps. A *crossing-free* backbone labeling is one without leader crossings.

The number of labels of a specific color may be unlimited or bounded by $K \geq |C|$. If $K = |C|$, all points of the same color are associated with a common label. One may restrict the maximum number of allowed labels for each color in C separately by specifying a *color vector* $\boldsymbol{k} = (k_1, \dots, k_{|C|})$. A backbone labeling that satisfies all of the restrictions on the number of labels is called *feasible*.

Our goal is to find feasible backbone labelings that optimize different quality criteria. We study three different quality criteria, *label number minimization* (Section 2), *total leader length minimization* (Section 3), and *crossing minimization* (Section 4). The first two require crossing-free leaders. We consider both *finite backbones* and *infinite backbones*. Finite backbones extend horizontally from the label to the furthest point connected to the backbone, whereas infinite backbones span the whole width of the rectangle (thus one could use duplicate labels on both sides). Our algorithms also vary depending on whether the order of the labels is fixed or flexible and whether more than one label per color class can be used. Note that, due to space constraints some of our proofs are only sketched. Detailed proofs can be found in the technical report [1].

2 Minimizing the Total Number of Labels

In this section we minimize the total number of labels in a crossing-free solution, i.e., we set $K = n$ so that there is effectively no upper bound on the number of labels. We first consider the case of infinite backbones and present an important observation on the structure of crossing-free labelings.

Lemma 1. *Let p_i, p_{i+1} be two vertically consecutive points. Let p_j $(j < i)$ be the first point above p_i with $\mathrm{c}(p_j) \neq \mathrm{c}(p_i)$, and let $p_{j'}$ $(j' > i+1)$ be the first point below p_{i+1} with $\mathrm{c}(p_{j'}) \neq \mathrm{c}(p_{i+1})$, if such points exist. In any crossing-free backbone labeling with infinite backbones, p_i and p_{i+1} are vertically separated by at most 2 backbones and any separating backbone has color $\mathrm{c}(p_i), \mathrm{c}(p_{i+1})$, $\mathrm{c}(p_j)$, or $\mathrm{c}(p_{j'})$.*

Sketch of Proof. In a crossing-free solution any infinite backbone splits the drawing into two independent subinstances above and below the backbone. Clearly, a backbone traversing a point has to be of the same color. On the other hand, we can check that a backbone lying between two points p_i and p_{i+1} can only have color $c(p_i), c(p_{i+1})$, or the color of the next point of distinct color above p_i or of the one below p_{i+1}. □

Clearly, if all points have the same color, one label always suffices. Even in an instance with two colors, one label per color is enough. However, if a third color is involved, then many labels may be required. We sketch how to find an optimum solution in $O(n)$ time. First, we replace any maximal set of identically colored consecutive points by the topmost point in the set. One can show that an optimum solution of the original instance can be easily obtained from an optimum solution of the reduced instance, in which no two consecutive points have the same color. We solve the reduced instance using dynamic programming.

Theorem 1. *Let $P = \{p_1, p_2, \ldots, p_n\}$ be an input point set consisting of n points sorted from top to bottom. Then, a crossing-free labeling of P with the minimum number of infinite backbones can be computed in $O(n)$ time.*

Sketch of Proof. We store a table nl of values $nl(i, \mathrm{cur}, c_{\mathrm{bak}}, c_{\mathrm{free}})$ representing the minimum number of backbones needed above or at point p_i such that the lowest backbone is c_{bak}-colored, the lowest backbone goes through p_i if the flag $\mathrm{cur} = \mathrm{true}$ and lies above p_i otherwise, and the (single) point between p_i and the lowest backbone (if $\mathrm{cur} = \mathrm{false}$) has color c_{free}. By careful case analysis, we can see that any entry of the table can be computed in constant time. □

We now consider finite backbones. First, note that we can slightly shift the backbones in a given solution so that backbones are placed only in gaps between points. We number the gaps from 0 to n where gap 0 is above and gap n is below all points. Suppose a point p_l lies between a backbone of color c in gap g and a backbone of color c' in gap g' with $0 \leq g < l \leq g' \leq n$ such that both backbones horizontally extend to at least the x-coordinate of p_l. Let $R(g, g', l)$ be the rectangle bounded by these two backbones, the vertical line through p_l and the right side of R. Suppose all points except the ones in $R(g, g', l)$ are already labeled. An optimum solution for connecting the points in R cannot reuse any backbone except for the two backbones in gaps g and g'; hence, it is independent of the rest of the solution. We use this observation for solving the problem by dynamic programming.

Theorem 2. *Given a set P of n colored points and a color set C, we can compute a feasible labeling of P with the minimum number of finite backbones in $O(n^4|C|^2)$ time.*

Sketch of Proof. For $0 \leq g \leq g' \leq n$, $l \in \{g, \dots, g'\} \cup \{\emptyset\}$, and two colors c and c' let $T[g, c, g', c', l]$ be the minimum number of additional labels that are needed for labeling all points in the rectangle $R(g, g', l)$ under the assumption that there is a backbone of color c in gap g, a backbone of color c' in gap g', between these two backbones there is no backbone placed yet, and they both extend to the left of p_l. Note that for $l = \emptyset$ the rectangle is empty and $T[g, c, g', c', \emptyset] = 0$. Finally, let $\bar{c} \notin C$ be a dummy color, and let $p_{\bar{l}}$ be the leftmost point. Then, the value $T[0, \bar{c}, n, \bar{c}, \bar{l}]$ is the minimum number of labels needed for labeling all points. By careful case analysis, we can compute each of the $(n+1) \times |C| \times (n+1) \times |C| \times (n+1)$ entries of table T in $O(n)$ time. □

3 Length Minimization

In this section we minimize the total length of all leaders in a crossing-free solution, either including or excluding the horizontal lengths of the backbones. We distinguish between a global bound K on the number of labels or a color vector $\boldsymbol{k}$ of individual bounds per color. We first consider the case of infinite backbones and use a parameter λ to distinguish the two minimization goals, i.e., we set $\lambda = 0$, if we want to minimize only the sum of the length of all vertical segments and we set λ to be the width of the rectangle R if we also take the length of the backbones into account. We further assume that $p_1 > \cdots > p_n$ are the y-coordinates of the input points.

Single Color. If all points have the same color, we seek for a set of at most K y-coordinates where we draw the backbones and connect each point to its nearest one, i.e., we must solve the following problem: Given n points with y-coordinates $p_1 > \ldots > p_n$, find a set S of at most K y-coordinates that minimizes

$$\lambda \cdot |S| + \sum_{i=1}^{n} \min_{y \in S} |y - p_i|. \tag{1}$$

Note that we can optimize the value in Eq. (1) by choosing $S \subseteq \{p_1, \dots, p_n\}$. Hence, the problem can be solved in $O(Kn)$ time if the points are sorted according to their y-coordinates using the algorithm of Hassin and Tamir [3]. Note that the problem corresponds to the K-median problem if $\lambda = 0$.

Multiple Colors. If the input points have different colors, we can no longer assume that all backbones go through one of the given n points. However, by Lemma 1, it suffices to add between any pair of vertically consecutive points two additional candidates for backbone positions, plus one additional candidate above all points and one below all points. Hence, we have a set of $3n$ candidate lines at y-coordinates

$$p_1^- > p_1 > p_1^+ > p_2^- > p_2 > p_2^+ > \cdots > p_n^- > p_n > p_n^+ \tag{2}$$

where for each i the values p_i^- and p_i^+ are as close to p_i as the label heights allow. Clearly, a backbone through p_i can only be connected to points with color $c(p_i)$. If we use a backbone through p_i^- (or p_i^+, respectively), it will have the same color as the first point below p_i (or above p_i, respectively) that has a different color than p_i. Hence, the colors of all candidates are fixed or the candidate will never be used as a backbone. For an easier notation, we denote the ith point in Eq. (2) by y_i and its color by $c(y_i)$. We solve the problem using dynamic programming.

Theorem 3. *A minimum length backbone labeling with infinite backbones for n points with $|C|$ colors can be computed in $O(n^2 \cdot \prod_{i=1}^{|C|} k_i)$ time if at most k_i labels are allowed for color i, $i = 1, \ldots, |C|$ and in $O(n^2 \cdot K)$ time if in total at most K labels are allowed.*

Sketch of Proof. For each $i = 1, \ldots, 3n$, and for each vector $\boldsymbol{k'} = (k'_1, \ldots, k'_{|C|})$ with $k'_1 \leq k_1, \ldots, k'_{|C|} \leq k_{|C|}$, let $L(i, \boldsymbol{k'})$ denote the minimum length of a feasible backbone labeling of $p_1, \ldots, p_{\lfloor \frac{i+1}{3} \rfloor}$ using k'_c infinite backbones of color c for $c = 1, \ldots, |C|$ such that the bottommost backbone is at position y_i, if such a labeling exists. Otherwise $L(i, \boldsymbol{k'}) = \infty$. One can show that the values $L(i, \boldsymbol{k'})$ can be computed recursively in $\mathcal{O}(n^2 \prod_{i=1}^{|C|} k_i)$ time in total. Let S be the set of candidates y_i such that all points below y_i have the same color as y_i. Then, we can compute the minimum total length of a backbone labeling of $p_1, \ldots, p_n$ with at most k_c, $c = 1, \ldots, |C|$ labels per color c by the following formula:

$$\min_{y_i \in S \cup \{p_n^+\}, k'_1 \leq k_1, \ldots, k'_{|C|} \leq k_{|C|}} \left(L(i, k'_1, \ldots, k'_{|C|}) + \sum_{\frac{i+2}{3} \leq x \leq n} (y_i - p_x) \right).$$

If we bound the total number of labels by K, we obtain a similar dynamic program with the corresponding values $L(i, k)$, $i = 1, \ldots, 3n$, $k < K$. □

We now turn our attention to the case of finite backbones and sketch how to modify the dynamic program for minimizing the total number of labels (see Theorem 1) to minimize the total leader length.

Theorem 4. *Given a set P of n colored points, a color set C, and a label bound K (or color vector $\boldsymbol{k}$), we can compute a feasible labeling of P with finite backbones that minimizes the total leader length in time $O(n^7 |C|^2 K^2)$ (or $O(n^7 |C|^2 (\prod_{c \in C} k_c)^2)$).*

Sketch of Proof. We change the meaning of an entry in the table T to denote the additional length of segments and backbones needed for labeling the points of the subinstance. Moreover, the precise positions of backbones matter for length

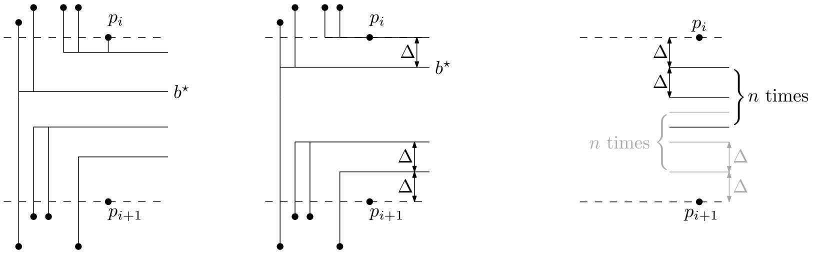


Fig. 2. (a) Longest backbone $b^\star$ splitting the backbones between p_i and p_{i+1}. (b) Backbones placed with the minimum leader length. (c) Candidate positions for backbones inside the gap.

minimization. A clear candidate set is the set of the y-coordinates of input points which may be used by a backbone of the same color. We can also identify candidates for backbones inside a gap between points p_i and p_{i+1}. We observe that the longest backbone $b^\star$ inside the gap splits all other backbones lying between p_i and p_{i+1}; see Fig. 2a. The backbones above $b^\star$ connect only to points above and, hence, must be placed as close to p_i as possible for length minimization. Symmetrically, the backbones below $b^\star$ connect only to the bottom and must be placed as close to p_{i+1} as possible.

For avoiding overlaps and to accommodate labels with fixed heights, we enforce a minimum distance $\Delta > 0$ between pairs of backbones, as well as backbones and differently colored points. Then, for the labels close to p_i, we get a sequence of consecutive candidate positions with distance Δ below p_i; see Fig. 2b and 2c. Symmetrically, there is such a sequence above p_{i+1}. Any such sequence contains up to n points (less if the gap is too small). Note that the two sequences might overlap; we can, however, easily ensure that no two backbones with distance less than Δ are used in the dynamic program. To address entries in T we use the $O(n^2)$ candidate positions (input points and positions in gaps) instead of the gaps; no position can be used twice.

As a final step, we integrate the global value K or the color vector $\boldsymbol{k}$ as a bound on the allowed numbers of labels. To this end, we add additional dimensions for K or for $k_c, c \in C$ to the table that specify the remaining available numbers of labels in the subinstance. □

4 Crossing Minimization

In this section we allow crossings between backbone leaders, which generally allows us to use fewer labels. We concentrate on minimizing the number of crossings for the case $K = |C|$, i.e., one label per color, and distinguish fixed and flexible label orders.

4.1 Fixed y-Order of Labels

In this part, we assume that the color set C is ordered and we require that for each pair of colors $i < j$, the i-colored label is above the j-colored label.

Infinite Backbones. Observe that it is possible to slightly shift the backbones of a solution without increasing the number of crossings so that no backbone contains a point. So, the backbones can be assumed to be in the gaps between vertically consecutive points; we number the gaps from 0 to n, as in Section 2.

Theorem 5. *Given a set P of n colored points and an ordered color set C, a backbone labeling with one label per color, labels in the given color order, infinite backbones, and minimum number of crossings can be computed in $O(n|C|)$ time.*

Proof. Suppose that we fix the position of the i-th backbone to gap g. For $1 \leq i \leq |C|$ and $0 \leq g \leq n$, let $\text{cross}(i, g)$ be the number of crossings of the vertical segments of the non-i-colored points when the color-i backbone is placed at gap g. Note that this number depends only on the y-ordering of the backbones, which is fixed, and not on their actual positions. So, we can precompute the table cross, using dynamic programming, as follows. All table entries of the form $\text{cross}(\cdot, 0)$ can be clearly computed in $O(n)$ time. Then, $\text{cross}(i, g) = \text{cross}(i, g-1)+1$, if the point between gaps $g-1$ and g has color j and $j > i$. In the case where the point between gaps $g-1$ and g has color j and $j < i$, $\text{cross}(i, g) = \text{cross}(i, g-1) - 1$. If it has color i, then $\text{cross}(i, g) = \text{cross}(i, g-1)$. From the above, it follows that the computation of table cross takes $O(n|C|)$ time.

Now, we use another dynamic program to compute the minimum number of crossings. Let $T[i, g]$ denote the minimum number of crossings on the backbones $1, \ldots, i$ in any solution subject to the condition that the backbones are placed in the given ordering and backbone i is positioned in gap g. Clearly $T[0, g] = 0$ for $g = 0, \ldots, n$. Moreover, we have $T[i, g] = \min_{g' \leq g} T[i-1, g'] + \text{cross}(i, g)$. Having pre-computed table cross and assuming that for each entry $T[i, g]$, we also store the smallest entry of row $T[i, \cdot]$ to the left of g, each entry of table T can by computed in constant time. Hence, table T can be filled in time $O(n|C|)$. Then, the minimum crossing number is $\min_g T[|C|, g]$. A corresponding solution can be found by backtracking in the dynamic program. □

Finite Backbones. We can easily modify the approach used for infinite backbones to minimize the number of crossings for finite backbones, if the y-order of labels is fixed, as the following theorem shows.

Theorem 6. *Given a set P of n colored points and an ordered color set C, a backbone labeling with one label per color, labels in the given order, finite backbones, and minimum number of crossings can be computed in $O(n|C|)$ time.*

Proof. We present a dynamic program similar to the one presented in the proof of Theorem 5. Recall that all points of the same color are routed to the same label and the order of the labels is fixed, i.e., the label of the i-colored points is above the label of the j-colored points, when $i < j$. Here, the computation of the number of crossings when fixing a backbone at a certain position should take into consideration that the backbones are not of infinite length. Recall that the dynamic program could precompute these crossings, by maintaining an $n \times |C|$ table cross, in which each entry $\mathrm{cross}(i, g)$ corresponds to the number of crossings of the non-i-colored points when the color-i-backbone is placed at gap g, for $1 \leq i \leq |C|$ and $0 \leq g \leq n$. In our case, $\mathrm{cross}(i, g) = \mathrm{cross}(i, g-1) + 1$, if the point between gaps $g-1$ and g is right of the leftmost of the i-colored points and has color j s.t. $j > i$. In the case, where the point between gaps $g-1$ and g is right of the leftmost of the i-colored points and has color j and $j < i$, $\mathrm{cross}(i, g) = \mathrm{cross}(i, g-1) - 1$. Otherwise, $\mathrm{cross}(i, g) = \mathrm{cross}(i, g-1)$. Again, all table entries of the form $\mathrm{cross}(\cdot, 0)$ can be clearly computed in $O(n)$ time. □

4.2 Flexible y-Order of Labels

In this part the order of labels is no longer given and we need to minimize the number of crossings over all label orders. While there is an efficient algorithm for infinite backbones, the problem is NP-complete for finite backbones.

Infinite Backbones. We give an efficient algorithm for the case that there are $K = |C|$ fixed label positions $y_1, \ldots, y_K$ on the right boundary of R, e.g., uniformly distributed.

Theorem 7. *Given a set P of n colored points, a color set C, and a set of $|C|$ fixed label positions, we can compute in $O(n + |C|^3)$ time a feasible backbone labeling with infinite backbones that minimizes the number of crossings.*

Proof. First observe that if the backbone of color $k, 1 \leq k \leq |C|$ is placed at position $y_i, 1 \leq i \leq |C|$, then the number of crossings created by the vertical segments leading to this backbone is fixed, since all label positions will be occupied by an infinite backbone. This crossing number $\mathrm{cr}(k, i)$ can be determined in $O(n_k + |C|)$ time, where n_k is the number of points of color k. In fact, by a sweep from top to bottom, we can even determine all crossing numbers $\mathrm{cr}(k, \cdot)$ for backbone $k, 1 \leq k \leq |C|$ in time $O(n_k + |C|)$. Now, we construct an instance of a weighted bipartite matching problem, where for each position $y_i, 1 \leq k \leq |C|$ and each backbone $k, 1 \leq k \leq |C|$, we establish an edge (k, i) of weight $cr(k, i)$. In total, this takes $O(n + |C|^2)$ time. The minimum-cost weighted bipartite matching problem can be solved in time $O(|C|^3)$ with the Hungarian method [5] and yields a backbone labeling with the minimal number of crossings. □

Finite Backbones. Next, we consider the variant with finite backbones and prove that it is NP-hard to minimize the number of crossings. For simplicity, we

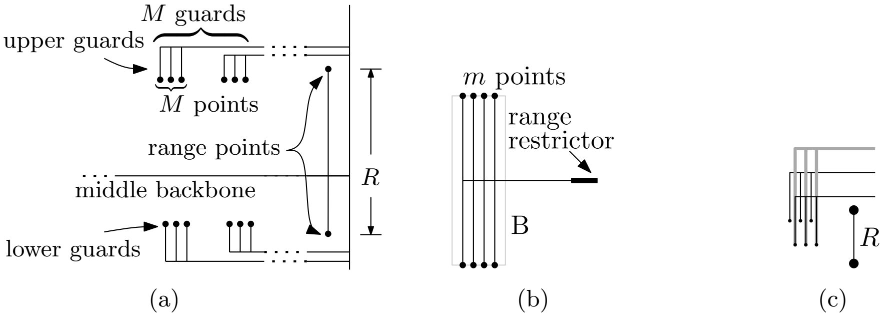


Fig. 3. (a) The range restrictor gadget, (b) a blocker gadget, (c) crossings caused by a pair of an upper and a lower guard that are positioned on the same side outside range R.

allow points that share the same x- or y-coordinates. This can be remedied by a slight perturbation. Our arguments do not make use of this special situation, and hence carry over to the perturbed constructions. We first introduce a number of gadgets that are required for our proof and sketch their properties.

The first one is the *range restrictor gadget*. Its construction consists of the middle backbone, whose position will be restricted to a given range R, and an upper and a lower *guard gadget* that ensure that positioning the middle backbone outside range R creates many crossings. We assume that the middle backbone is connected to at least one point further to the left such that it extends beyond all points of the guard gadgets. The middle backbone is connected to two *range points* whose y-coordinates are the upper and lower boundary of the range R. Their x-coordinates are such that they are on the right of the points of the guard gadgets. A *guard* consists of a backbone that connects to a set of M points, where $M > 1$ is an arbitrary number. The M points of a guard lie left of the range points. The upper guard points are horizontally aligned and lie slightly below the upper bound of range R. The lower guard points are placed such that they are slightly above the lower bound of range R. We place M upper and M lower guards such that the guards form pairs for which the guard points overlap horizontally. The upper (resp. lower) guard gadget is formed by the set of upper (resp. lower) guards. We call M the *size* of the guard gadgets. The next lemma shows properties of the range restrictor.

Lemma 2. *The backbones of the range restrictor can be positioned such that there are no crossings. If the middle backbone is positioned outside the range R, there are at least $M - 1$ crossings.*

Proof. The first statement is illustrated in Fig. 3a. To prove the second statement, assume to the contrary that the middle backbone is positioned outside range R, say w.l.o.g. below range R, and that there are fewer than $M - 1$ crossings. Observe that all guards must be positioned above the middle backbone, as a guard below the middle backbone would create M crossings, namely between the middle backbone and the segments connecting the points of the guard to its backbone. So, the middle backbone is the lowest. Now observe that any guard that is positioned below the upper range point crosses the segment that connects this range point to the middle backbone. To avoid having $M-1$ crossings, at least $M+1$ guards (both upper and lower) must be positioned above range R. Hence, there is at least one pair consisting of an upper and a lower guard that are both positioned above range R. This, independent of their ordering, creates at least $M-1$ crossings, a contradiction; see Fig. 3c, where the two alternatives for the lower guard are drawn in black and bold gray, respectively. □

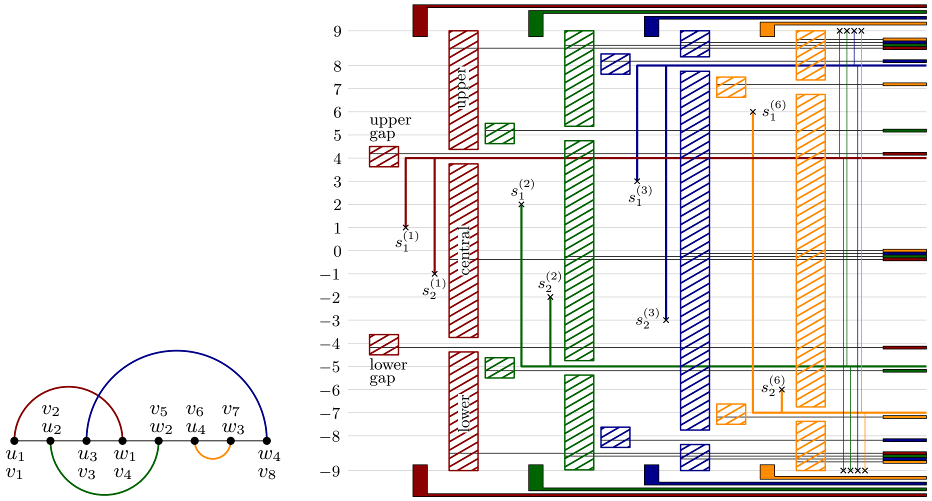
Let B be an axis-aligned rectangular box and R a small interval that is contained in the range of y-coordinates spanned by B. A *blocker gadget* of *width* m consists of a backbone that connects to $2m$ points, half of which are on the top and bottom side of B, respectively. A range restrictor gadget is used to restrict the backbone of the blocker to the range R; see Fig. 3b. Note that, due to the range restrictor, this drawing is essentially fixed. We say that a backbone *crosses* the blocker gadget if its backbone crosses box B. It is easy to see that any backbone that crosses a blocker gadget creates m crossings, where m is the width of the blocker. We are now ready to present the NP-hardness reduction.

Theorem 8. *Given a set of n input points in k different colors and an integer Y it is NP-complete to decide whether a backbone labeling with one label per color and flexible y-order of the labels that has at most Y leader crossings exists.*

Proof. The proof is by reduction from the NP-complete Fixed Linear Crossing Number problem [8]: Given a graph $G=(V,E)$, a bijective function $f\colon V \to \{1,\ldots,|V|\}$, and an integer Z, is there a drawing of G with the vertices placed on a horizontal line (*spine*) in the order specified by f and the edges drawn as semi-circles above or below the spine so that there are at most Z crossings? Masuda et al. [8] showed that the problem is NP-complete, even if G is a matching.

Let G be a matching. Then, the number of vertices is even and we can assume that the vertices $V=\{v_1,\ldots,v_{2n}\}$ are indexed in the order specified by f, i.e., $f(v_i)=i$ for all i. We also direct each edge $\{v_i,v_j\}$ with $i<j$ from v_i to v_j. Let $\{u_1,\ldots,u_n\}$ be the ordered source vertices and let $\{w_1,\ldots,w_n\}$ be the ordered sink vertices; see Fig. 4a. In our reduction we will create an edge gadget for every edge of G. The gadget consists of five blocker gadgets and one *side selector gadget*. Each of the six sub-gadgets uses its own color and thus defines one backbone. The edge gadgets are ordered from left to right according to the sequence of source vertices $(u_1,\ldots,u_n)$; see Fig. 4b.

The edge gadgets are placed symmetrically with respect to the x-axis. We create $2n+1$ special rows above the x-axis and $2n+1$ special rows below, indexed by $-(2n+1),-2n,\ldots,0,\ldots,2n,2n+1$. The gadget for an edge (v_i,v_j) uses five blocker gadgets (denoted as *central, upper, lower, upper gap*, and *lower gap* blockers) in two different columns to create two small gaps in rows j and $-j$, see the hatched blocks in the same color in Fig. 4b. The upper and lower blockers extend vertically to rows $2n+1$ and $-2n-1$. The gaps are intended to create two alternatives for routing the backbone of the side selector. Every



(a) (b)

Fig. 4. (a) An input instance with four edges, (b) Sketch of the reduction for the graph of Fig. 4a. Hatched rectangles correspond to blockers, thick segments to side selectors, and filled shapes to guard gadgets or range restrictor gadgets.

backbone that starts left of the two gap blockers is forced to cross at least one of these five blocker gadgets as long as it is vertically placed between rows $2n + 1$ and $-2n - 1$. The blockers have width $m = 8n^2$. Their backbones are fixed to lie between rows 0 and -1 for the central blocker, between rows $2n$ and $2n + 1$ ($-2n$ and $-2n - 1$) for the upper (resp. lower) blocker, and between rows j and $j + 1$ ($-j$ and $-j - 1$) for the upper (resp. lower) gap blocker.

The *side selector* consists of two horizontally spaced *selector points* $s_1^{(i)}$ and $s_2^{(i)}$ in rows i and $-i$ located between the left and right blocker columns. They have the same color and thus define one joint backbone that is supposed to pass through one of the two gaps in an optimal solution. The n edge gadgets are placed from left to right in the order of their source vertices; see Fig. 4b. The backbone of every selector gadget is vertically restricted to the range between rows $2n + 1$ and $-2n - 1$ in any optimal solution by augmenting each selector gadget with a range restrictor gadget. So, we add two more points for each selector to the right of all edge gadgets, one in row $2n + 1$ and one in row $-2n - 1$. They are connected to the selector backbone. In combination with a corresponding upper and lower guard gadget of size $M = \Omega(n^4)$ between the two selector points $s_1^{(i)}$ and $s_2^{(i)}$ this achieves the range restriction according to Lemma 2.

We now claim that in a crossing-minimal labeling the backbone of the selector gadget for every edge (v_i, v_j) passes through one of its two gaps in rows j or $-j$. The proof of this claim is based on three different options for placing a selector backbone: (a) outside its range restriction, i.e., above row $2n + 1$ or below row

$-2n-1$, (b) between rows $2n+1$ and $-2n-1$, but outside one of the two gaps, and (c) in rows j or $-j$, i.e., inside one of the gaps. By this claim and the fact that violating any range restriction immediately causes M crossings, we can assume that every backbone adheres to the rules, i.e., stays within its range as defined by the range restriction gadgets or passes through one of its two gaps.

One can show that an optimal solution of the backbone labeling instance I_G created for a matching G with n edges has $X + 2Z$ crossings, where X is a constant depending on G, and Z is the minimum number of crossings of G in the Fixed Linear Crossing Number problem. The detailed proof is based on carefully counting crossings in four different cases, depending on which types of backbones and vertical segments intersect. It turns out that almost all crossings are fixed (yielding the number X), except for those of selector backbones with vertical selector segments for which the two underlying edges (v_i, v_j) and (v_k, v_l) with $i < k$ are *interlaced*, i.e., $i < k < j < l$ holds (yielding the number $2Z$). Note that we can guess an order of the backbones and apply Theorem 6 to compute the minimum crossing number, which concludes the NP-completeness proof. □

Acknowledgements. This work was started at the Bertinoro Workshop on Graph Drawing 2013. M. Nöllenburg received financial support by the Concept for the Future of KIT. The work of M. A. Bekos and A. Symvonis is implemented within the framework of the Action "Supporting Postdoctoral Researchers" of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State. We also acknowledge partial support by GRADR – EUROGIGA project no. 10-EuroGIGA-OP-003.

References

1. Bekos, M.A., Cornelsen, S., Fink, M., Hong, S., Kaufmann, M., Nöllenburg, M., Rutter, I., Symvonis, A.: Many-to-one boundary labeling with backbones. CoRR (2013), arXiv:1308.6801
2. Bekos, M.A., Kaufmann, M., Symvonis, A., Wolff, A.: Boundary labeling: Models and efficient algorithms for rectangular maps. Computational Geometry 36(3), 215–236 (2007)
3. Hassin, R., Tamir, A.: Improved complexity bounds for location problems on the real line. Operations Research Letters 10(7), 395–402 (1991)
4. Kaufmann, M.: On map labeling with leaders. In: Albers, S., Alt, H., Näher, S. (eds.) Festschrift Mehlhorn, LNCS, vol. 5760, pp. 290–304. Springer, Heidelberg (2009)
5. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2(1–2), 83–97 (1955)
6. Lin, C.C.: Crossing-free many-to-one boundary labeling with hyperleaders. In: Proc. IEEE Pacific Visualization Symp. (PacificVis'10). pp. 185–192. IEEE (2010)
7. Lin, C.C., Kao, H.J., Yen, H.C.: Many-to-one boundary labeling. Journal of Graph Algorithms and Applications 12(3), 319–356 (2008)
8. Masuda, S., Nakajima, K., Kashiwabara, T., Fujisawa, T.: Crossing minimization in linear embeddings of graphs. IEEE Trans. Computers 39(1), 124–127 (1990)