Fast fencing

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

Fast Fencing

Mikkel Abrahamsen[1], Anna Adamaszek[1], Karl Bringmann[2], Vincent Cohen-Addad[3],
Mehran Mehr[4], Eva Rotenberg[5], Alan Roytman[1], and Mikkel Thorup[1]

[1]Department of Computer Science, University of Copenhagen, Denmark
[2]Saarland Informatics Campus, Max Planck Institute for Informatics, Germany
[3]Sorbonne Université, CNRS, Laboratoire d'informatique de Paris 6, LIP6, F-75252 Paris,
France
[4]Department of Computer Science, TU Eindhoven, Netherlands
[5]Department of Applied Mathematics and Computer Science, Technical University of
Denmark, Denmark

Abstract

We consider very natural "fence enclosure" problems studied by Capoyleas, Rote, and Woeg-inger and Arkin, Khuller, and Mitchell in the early 90s. Given a set S of n points in the plane, we aim at finding a set of closed curves such that (1) each point is enclosed by a curve and (2) the total length of the curves is minimized. We consider two main variants. In the first variant, we pay a unit cost per curve in addition to the total length of the curves. An equivalent formulation of this version is that we have to enclose n unit disks, paying only the total length of the enclosing curves. In the other variant, we are allowed to use at most k closed curves and pay no cost per curve.

For the variant with at most k closed curves, we present an algorithm that is polynomial in both n and k. For the variant with unit cost per curve, or unit disks, we present a near-linear time algorithm.

Capoyleas, Rote, and Woeginger solved the problem with at most k curves in $n^{O(k)}$ time. Arkin, Khuller, and Mitchell used this to solve the unit cost per curve version in exponential time. At the time, they conjectured that the problem with k curves is NP-hard for general k. Our polynomial time algorithm refutes this unless P equals NP.

1 Introduction

We consider some very natural "fence enclosure" problems studied by Capoyleas, Rote, and Woeginger [6] and Arkin, Khuller, and Mitchell [3] in the early 90s. Given a set S of n points in the plane, we aim at finding a set of closed curves such that (1) each point is enclosed by a curve and (2) the total length of the curves is minimized. We consider two main variants. In the first variant, we pay an opening cost $\eta > 0$ per curve, which is part of the input. An equivalent formulation is that a circle with radius $\eta/2\pi$ is centered at each point, and we need to enclose these circles with curves of minimal total length, paying no opening cost. The equivalence is illustrated and explained in Figure 1. By a suitable scaling, we may assume that the circles are unit circles. We thus refer to this variant as the *unit disk fencing problem*. In the other variant, we are allowed to use at most k closed curves and pay no cost per curve. We can think of this as dividing the points into k clusters and then viewing the closed curves as perimeters of the convex hulls of the clusters. For this reason, we refer to the variant as the *k-cluster fencing problem* (also referred to as the *minimum perimeter sum problem* in the literature).

Capoyleas, Rote, and Woeginger [6] presented an algorithm for the k-cluster fencing problem that runs in time $\rho(k)n^{O(k)}$, where $\rho(k)$ is the number of nonisomorphic planar graphs on k nodes. This yields a polynomial running time when k is fixed. Arkin et al. [3] conjectured the problem to be NP-hard when k is part of the input and neither an NP-hardness proof nor a polynomial time algorithm has been found so far. To solve the unit disk version, which is equivalent to having a unit cost per cluster, Arkin et al. suggested running their algorithm for the k-cluster fencing problem for all $k \leq n$, adding k to the total perimeter length. These were the best known bounds except in the special case of the k-cluster fencing problem for $k \in \{1, 2\}$ (see the discussion on related work for more details).

1.1 Our Results

We present polynomial time algorithms for both problems. More specifically, for the unit disk fencing problem, we present an efficient near-linear time algorithm (Theorem 1). For the k-cluster fencing problem, we present an algorithm that is polynomial in both n and k (Theorem 3). In particular, this refutes the conjectured hardness unless P = NP.

Our algorithm for the unit disk fencing problem can be generalized to the case where the input consists of objects that are allowed to be disks with different diameters or polygonal objects that have to be fenced. For this variant, our running time increases by a factor that is logarithmic in the ratio between the maximum and the minimum object diameter.

In order to achieve near-linear bounds for the unit disk fencing problem, we introduce new techniques that we believe can have other applications in computational geometry. We give a detailed overview of the techniques below.

Throughout our paper, we assume that it is possible to compare the costs of two different clusterings efficiently. Note that this is a standard assumption in computational geometry.

1.2 Applications

The problem of fencing in disks or objects appears very commonly in the real world. A good example is the protection of trees, either at construction sites to protect the roots, or in the wild to protect rare trees from deer and other animals. When trees are planted by nature, we have no control over their location. In this context, each disk should have a sufficient diameter to protect rare trees from wildlife (see Figure 1).
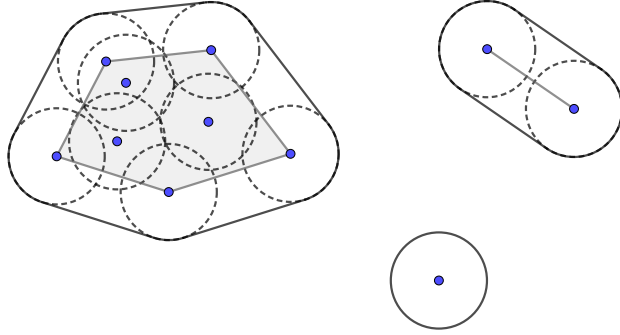
Figure 1: A set of 11 points and the set of enclosing curves minimizing the total length plus the opening cost η per curve. There are three curves, enclosing 1, 2, and 8 points, drawn in gray (one cannot be seen, as it consists of just one input point). A dashed circle centered at each point with radius $\eta/2\pi$ is drawn. For each cluster, the curve enclosing the respective circles is drawn in black. We note that the perimeter of each black curve is exactly η larger than the perimeter of the gray curve, as the linear pieces sum to the perimeter of the gray curve and the circular arcs sum to η. Hence, the problem of enclosing points with curves, minimizing the total length plus an opening cost η per curve, is equivalent to that of enclosing circles with curves of minimal total length.

There are many standards that specify how far fences should be from trees, and even discussions on different advantages of grouping trees beyond the fence cost (e.g., see [8]).

1.3   Our Techniques

Approach for the unit disk fencing problem.   Our approach is as follows. We first partition the plane and recursively subdivide it into four quadrants, using a quadtree dissection-type approach. This divides the plane into *cells* of geometrically decreasing sizes. Each cell of width w consists of four cells of width $w/2$. We find optimal partitions for the smallest cells first (i.e., the lowest level of the quadtree). We then work with increasing cell sizes, solving the problem layer by layer in the quadtree. To obtain a solution at a given cell size, we rely on solutions to smaller cell sizes. For this to work, we need to have precomputed the solutions for various *polyominoes* made of some constant number of cells of smaller sizes.

For a given polyomino, we show how to obtain an optimal clustering of the points within the polyomino by merging the solutions for smaller polyominoes. In some cases, merging is not enough as there can be one large cluster intersecting all the cells of the polyomino, which we do not find by merging solutions of smaller polyominoes.

Hence, we need an efficient algorithm for finding the best cluster C intersecting all the cells of the polyomino. In order to do this, we give a subroutine running in time $O(n \log^3 n)$ that finds the best cluster that intersects all cells. The subroutine works in two steps: it first finds a point x_0 that belongs to the cluster we are looking for together with a point p on the boundary of the cluster. Once we have x_0 and p, the idea is to make an angular sweep of a ray from x_0, and consider the points in the order the ray sweeps over them. For each point v, we calculate the "best" path from p to v, in terms of both its length and how many clusters' opening costs it may save. We prove that the "best" path consists of line segments between points, and save for each v information about the last line segment on the path to v (see the red lines in Figure 2). This information allows us to

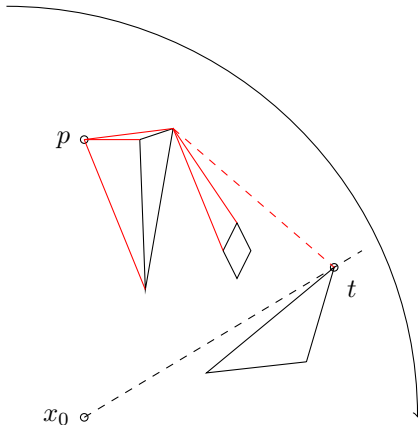finally retrieve the boundary of the convex hull of C recursively.

Figure 2: Given the advice that p lies on the perimeter of the cluster containing x_0 in an optimal clustering, we can compute the cluster containing x_0 with an angular sweep. Here, the angular sweep has reached the point t.

Approach for the k-cluster fencing problem.   Our algorithm for the k-cluster fencing problem shares some similarities with the work of Gibson et al. [10], in which a dynamic programming approach was used to solve the minimum radius sum problem. One main difference is that the running time complexity they obtain is $O(n^{881})$, whereas our approach works in $O(n^{27})$ time. Their technique is to divide a given instance (S, k) of the minimum radius sum problem into subproblems, where S is the set of points to be clustered and k is a constraint on the number of clusters we can use. The problem (S, k) is solvable if we have a solution to all subproblems. However, the number of subproblems is exponential. To get a polynomial time algorithm, they showed that a solution can be found after considering only a polynomial number of subproblems.

We also use a dynamic programming approach for our problem, although we need new techniques since the number of candidate clusters for the minimum radius sum problem is only $O(n^2)$ (as each disk is determined by two points in S, one determining the center and the other determining the radius). In contrast, our problem has an exponential number of candidate clusters (dictated by all subsets of S). We define subproblems based on *boxes*, which are rectangles that cover some portion of the plane and some number of input points from S. Our key observation is that there is some separator of each box (i.e., a vertical line segment or a horizontal line segment) that splits the box into two strictly smaller boxes such that an optimal solution only has a constant number of line segments that intersect this separator (in fact, we give a bound of two on the number of such intersecting line segments).

A dynamic programming approach naturally follows: we simply guess the position of such a separator (for which we argue there are $O(n)$ choices), and then guess which segments crossing this separator belong to an optimal solution. We first obtain solutions for smaller boxes, and then glue together solutions for smaller boxes to obtain solutions for larger boxes.

We note that there is an unpublished solution [12] (i.e., polynomial time algorithm) for the rectilinear version of the problem, where we must enclose n points using k axis-parallel rectangles rather than convex hulls (as in our setting). The solution to the axis-parallel version uses similar

3

ideas. In particular, it is possible to argue the existence of separators that do not cut through any clusters of an optimal solution in this setting. This is not possible for our problem. For the k-cluster fencing problem, it is possible that any such vertical or horizontal separator cuts at least one cluster, and allowing skew separators would result in subproblems of high complexity. In particular, consider an example with n sufficiently large and assume $k < n/3$. We have $n - k + 1$ points spread evenly on a circle as the corners of a regular $(n - k + 1)$-gon, and $k - 1$ points spread evenly on a surrounding circle with the same center. The surrounding points are fairly close yet sufficiently far enough away that the optimal solution is to cluster the inner $n - k + 1$ points together, and open a cluster for each of the $k - 1$ points on the surrounding circle. Cutting away the $k - 1$ points on the outside (with not necessarily axis-aligned separators) creates subproblems defined on polygonal regions with $k - 1$ sides, resulting in high complexity subproblems.

1.4 Related Work

The literature on geometric clustering is vast [2], and thus we focus on the most relevant prior works. Arkin, Khuller, and Mitchell [3] considered many clustering variants related to the problems studied in the present paper. For the variant where points have a value associated with them, they showed that the problem of maximizing profit (i.e., sum of values of points enclosed minus the total perimeter) is NP-hard when values are unrestricted in sign. When values are strictly positive, they gave an $O(n^3)$ time algorithm. For the version in which there is a budget on the total perimeter we can use, the problem of maximizing profit is also NP-hard, even when values are strictly positive (they provided a pseudo-polynomial algorithm when the values are integers).

The k-cluster fencing problem for $k = 1$ is the very well-known problem of computing the convex hull of a set of points in the plane [7]. There has also been some work for the special case of $k = 2$ clusters. The work of Mitchell and Wynters [13] studied four flavors of the problem: minimizing the sum of perimeters, the maximum of the perimeters, the sum of the areas enclosed by the fences, and the maximum of the areas. They gave polynomial time solutions for all four flavors, running in time $O(n^3)$ (for some of them, they gave improved running time bounds of $O(n^2)$). More recently, the work of Abrahamsen et al. [1] gave an algorithm running in time $O(n \log^4(n))$ that solves the case of $k = 2$ clusters, yielding the first subquadratic time algorithm for this setting.

There have been many other papers studying related geometric clustering problems. Capoyleas, Rote, and Woeginger [6] studied a general geometric k-clustering framework in which the cost of a solution is determined by some weight function that assigns real weights to any subset of points in the plane (i.e., each cluster), after which a symmetric k-ary function over k-tuples is applied (e.g., the sum function). For the case when the weight function is the diameter, radius, or perimeter and the symmetric k-ary function is an arbitrary monotone increasing function (such as the sum or the maximum), they gave an algorithm running in time $\rho(k)n^{O(k)}$, where $\rho(k)$ is the number of nonisomorphic planar graphs on k nodes. This is polynomial if k is fixed and not given as input.

In addition, the work of Behsaz and Salavatipour [4] studied objectives such as minimizing the sum of radii and minimizing the sum of diameters subject to the constraint of having at most k clusters. For minimizing the sum of radii, they gave a polynomial time algorithm for clustering points in metric spaces that are induced by unweighted graphs, assuming no singleton clusters are allowed. They also showed that finding the best single cluster for each connected component of the graph yields a $\frac{3}{2}$-approximation algorithm, assuming no singleton clusters are allowed. For the problem of minimizing the sum of diameters, they gave a polynomial time approximation scheme when points lie in the plane with Euclidean distances, along with a polynomial time exact algorithm

when k is constant (for the metric setting).

Many classical clustering problems are NP-hard when k is given as part of the input, though there are some notable exceptions. In 2012, Gibson et al. [10] devised a polynomial time algorithm for finding k disks, each centered at a point in S, such that the sum of the radii of the disks is minimized subject to the constraint that their union must cover S. In their paper, they used a dynamic programming approach to get a running time of $O(n^{881}T(n))$, where $T(n)$ is the time needed to compare two candidate solutions.

2 The Unit Disk Fencing Problem

Given a set of points A in the plane, we denote by $H(A)$ the convex hull of A, and by $h(A)$ the perimeter of $H(A)$.

Let A be a finite set of points in $\mathbb{R}^2$ and let $\eta > 0$ be the *opening cost*. Consider a partition $\mathcal{C} := \{C_1, \ldots, C_\ell\}$ of A. We refer to each set C_i as a *cluster*. The *cost* of $\mathcal{C}$ with respect to η is

$$\eta \cdot \ell + \sum_{i=1}^{\ell} h(C_i).$$

The partition $\mathcal{C}$ is *optimal* for (A, η) if no partition of A has a lower cost. We denote the cost of an optimal partition for (A, η) as $\text{OPT}(A, \eta)$. When the opening cost is clear from the context, we might omit it. In the *unit disk fencing problem*, we are given a set of points A and an opening cost η, and the goal is to find an optimal partition for (A, η).

Observation 1. *In an optimal partition, the clusters have pairwise disjoint convex hulls.*

We say that A is *indivisible* if $\{A\}$ is an optimal partition for (A, η).

Observation 2. *Each cluster of an optimal partition is indivisible.*

We say an optimal partition $\{C_1, \ldots, C_\ell\}$ for (A, η) is *maximal* if there is no optimal partition $\{C'_1, \ldots, C'_{\ell'}\}$ of (A, η), where $\ell' < \ell$, such that for each $i \in \{1, \ldots, \ell\}$, there is some $j \in \{1, \ldots, \ell'\}$ such that $C_i \subseteq C'_j$.

2.1 Structural Results

Lemma 1. *Let A, B be two indivisible sets of points in $\mathbb{R}^2$ under the opening cost η. If $H(A)$ and $H(B)$ intersect, then the set $A \cup B$ is indivisible.*

Proof. Let $\mathcal{C}$ be a maximal optimal partition of $(A \cup B, \eta)$. Let $\{C_1, \ldots, C_m\}$ be the clusters from $\mathcal{C}$ such that $A \cap C_i \neq \emptyset$ and $B \cap C_i \neq \emptyset$ for $1 \leq i \leq m$. Assume for contradiction that $m = 0$. This means that $A \cap B = \emptyset$ and each cluster of $\mathcal{C}$ is a subset of either A or B, and hence that $|\mathcal{C}| \geq 2$. If $|\mathcal{C}| = 2$, then $\mathcal{C} = \{A, B\}$, and $\mathcal{C}$ cannot be optimal as $H(A)$ and $H(B)$ intersect. If $|\mathcal{C}| > 2$, then $\mathcal{C}$ cannot be maximal as A and B are indivisible. Hence $m \geq 1$.

We want to show that the set $A' := A \cup \bigcup_{i=1}^{m} C_i$ is indivisible. Let $\{A_1, \ldots, A_j\}$, where possibly $j = 0$, be the clusters of $\mathcal{C}$ that are contained in A. Then the set $\mathcal{A} := \{A_1, \ldots, A_j, C_1, \ldots, C_m\}$ is a partition of A'. Note that $\mathcal{A}$ must be an optimal partition of A', as otherwise $\mathcal{C}$ would not be optimal for $A \cup B$.

The cost of $\mathcal{A}$ is thus

$$\text{OPT}(A') = (j + m)\eta + \sum_{i=1}^{j} h(A_i) + \sum_{i=1}^{m} h(C_i).$$
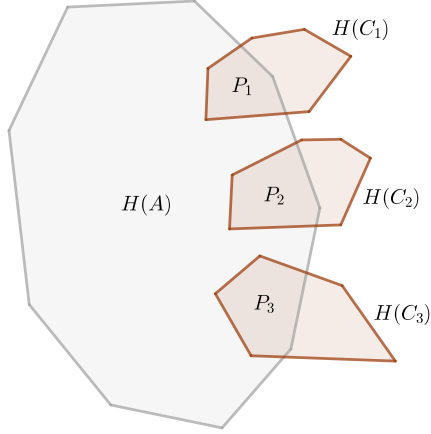
5

Figure 3: The polygon Q from the proof of Lemma 1 is the set $H(A) \cup \bigcup_{i=1}^{3} H(C_i)$. Note that the perimeter is $h(Q) = h(A) + \sum_{i=1}^{3}(h(C_i) - h(P_i))$.

Now, for each cluster C_i, we define a polygon $P_i = H(A) \cap H(C_i)$. As $H(A)$ and $H(C_i)$ are both convex, so is P_i.

Note that all points in A' are contained in the polygon $Q := H(A) \cup \bigcup_{i=1}^{m} H(C_i)$, see Figure 3. It hence follows that

$$h(A') \le h(Q) = h(A) + \sum_{i=1}^{m}(h(C_i) - h(P_i)).$$

Note that $\{A_1, \ldots, A_j, P_1 \cap A, \ldots, P_m \cap A\}$ is a partition of A. Hence, by the indivisibility of A, we have

$$\eta + h(A) \le (j+m)\eta + \sum_{i=1}^{j} h(A_j) + \sum_{i=1}^{m} h(P_i).$$

Combining these two inequalities yields

$$h(A') + \eta \le (j+m)\eta + \sum_{i=1}^{j} h(A_j) + \sum_{i=1}^{m} h(C_i) = \mathrm{OPT}(A'),$$

and so A' is indivisible.

As A' is indivisible, A' is the union of clusters of $\mathcal{C}$, and $\mathcal{C}$ is maximal, it follows that A' is itself a cluster of $\mathcal{C}$, i.e., $m = 1$ and $j = 0$. Recall furthermore that A' contains A and intersects B. In a similar way, it can be shown that $B' := B \cup \bigcup_{i=1}^{m} C_i$ is a cluster of $\mathcal{C}$ that contains B and intersects A. Since $A' \cap B' \ne \emptyset$ and $\mathcal{C}$ is optimal, we must have $A' = B' = A \cup B$, so $\mathcal{C} = \{A \cup B\}$ and $A \cup B$ is indivisible. $\qquad\square$

Lemma 2. *Let $A \subseteq B$ be sets of points in $\mathbb{R}^2$, and let A be indivisible. Let $\mathcal{C} = \{C_1, \ldots, C_\ell\}$ be a maximal optimal partition of B. Then $A \subseteq C_i$ for some $i \in \{1, \ldots, \ell\}$.*

Proof. Let $\mathcal{S} \subseteq \mathcal{C}$ be the set of clusters of $\mathcal{C}$ that intersect A. By Lemma 1, each of the sets $S \cup A$, where $S \in \mathcal{S}$, is indivisible. It thus follows that $A \cup \bigcup_{S \in \mathcal{S}} S = \bigcup_{S \in \mathcal{S}} S$ is also indivisible. Since $\mathcal{C}$ is maximal, it must then be the case that $\mathcal{S}$ consists of a single cluster C_i that contains A. $\qquad\square$

Lemma 3. *Each instance (A, η) of the unit disk fencing problem has a unique maximal optimal partition.*

Proof. Consider two maximal optimal partitions $\mathcal{C} = \{C_1, \ldots, C_\ell\}$ and $\mathcal{C}' = \{C'_1, \ldots, C'_{\ell'}\}$ of A. Lemma 2 gives that each cluster C_i of $\mathcal{C}$ is contained in some cluster C'_j of $\mathcal{C}'$. Likewise, each cluster C'_j of $\mathcal{C}'$ is contained in some cluster C_i of $\mathcal{C}$. Since the clusters of an optimal partition are disjoint, it now follows that there is a one-to-one correspondence between the partitions $\mathcal{C}$ and $\mathcal{C}'$, i.e., they are the same partition of A. □

Lemma 4. *Let $A = \bigcup_{i \in I} A_i$ be a set of points such that $A = \bigcup_{i \in I} A_i^{\complement}$, where $A_i^{\complement} := A \setminus A_i$ for $i \in I$. Let $\mathcal{C}_i$ be the maximal optimal partition of $A_i^{\complement}$, and $\mathcal{C}$ the maximal optimal partition of A. Any cluster of $\mathcal{C}_i$ that intersects a cluster $C \in \mathcal{C}$ is also contained in C, and it follows in particular that each cluster $C \in \mathcal{C}$ is the union of clusters of the partitions $\mathcal{C}_i$. Furthermore, each cluster $C \in \mathcal{C}$ is either a cluster of some partition $\mathcal{C}_i$ or has a non-empty intersection with each set A_i.*

Proof. In order to prove that first part, consider some cluster $C_i \in \mathcal{C}_i$ that intersects $C \in \mathcal{C}$. Since C_i is indivisble, Lemma 2 gives that there is a cluster of $\mathcal{C}$ containing C_i. That cluster must be C, as the clusters of $\mathcal{C}$ are disjoint. It hence also follows that each cluster $C \in \mathcal{C}$ is the union of some clusters of the partitions $\mathcal{C}_i$.

In order to prove the second part, consider a cluster $C \in \mathcal{C}$ that does not intersect some set A_i. Then, $C \subseteq A_i^{\complement}$. Since C is indivisible, Lemma 2 gives that there is a cluster C_i of $\mathcal{C}_i$ such that $C \subseteq C_i$. By the above, we also have that $C_i \subseteq C$. Hence $C = C_i$, i.e., C is a cluster of $\mathcal{C}_i$. □

2.2 Partitioning into Independent Instances

Consider an instance (A, η) of the unit disk fencing problem. Observe that any two points $p_1, p_2 \in A$ such that $\|p_1 p_2\| < \eta/2$ must be in the same cluster in any optimal partition of A. We will prove that we can efficiently decompose the problem instance (A, η) into a collection of independent subinstances (A_i, η), such that for each subinstance $diam(A_i) = \text{poly}(|A_i|) \cdot \eta$.

Lemma 5. *Any n-point instance (A, η) of the unit disk fencing problem can be reduced in time $O(n \log^2 n)$ into a disjoint collection of subinstances (A_i, η), where $A = \bigcup_i A_i$ and each subinstance is bounded by a box of side lengths at most $2|A_i|^2 \cdot \eta$. The subinstances are independent in the sense that an optimal partition for (A, η) is the union of the optimal partitions for (A_i, η).*

Proof. Clearly, the optimal partition for (A, η) has cost of at most $n \cdot \eta$, as a partition of cost $n \cdot \eta$ can be obtained by opening n singleton clusters. Therefore, if any two points are at a distance greater than $n \cdot \eta/2$, they must be in separate clusters of any optimal partition (as the perimeter of any cluster containing such two points would be greater than $\text{OPT}(A)$).

We first sort all points from A with respect to their x-coordinate. Denote the sorted points by $p_1, \ldots, p_n$. Whenever for two consecutive points p_i, p_{i+1} the difference in their x-coordinate is greater than $n \cdot \eta/2$, we know that the sets of points $\{p_1, \ldots, p_i\}$ and $\{p_{i+1}, \ldots, p_n\}$ can be treated separately, i.e., each cluster of any optimal partition will be contained in either $\{p_1, \ldots, p_i\}$ or $\{p_{i+1}, \ldots, p_n\}$. That gives us a partition of $\{p_1, \ldots, p_n\}$ into subinstances, where each subinstance is contained in a vertical slab of width at most $n^2 \cdot \eta/2$. Now, for each subinstance we sort the points according to their y-coordinate, and perform a similar operation. Therefore, in time $O(n \log n)$ we partitioned (A, η) into subinstances (A_i, η), such that $A = \bigcup_i A_i$, each subinstance has a bounding
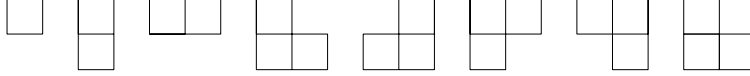
Figure 4: Basic polyominoes.

box of size at most $n^2 \cdot \eta/2 \times n^2 \cdot \eta/2$, and an optimal partition for (A, η) is the union of the optimal partitions for (A_i, η).

Note that if $|A_i| \geq |A|/2$, then $n^2 \cdot \eta/2 = |A|^2 \cdot \eta/2 \leq 2|A_i|^2 \cdot \eta$, and the size of the bounding box is as required. It therefore remains to consider subinstances (A_i, η) for which $|A_i| < |A|/2$ and the smallest bounding box has size at least $2|A_i|^2 \cdot \eta$. In such an instance, there must be two consecutive points in the order of x- or y-coordinates where the respective coordinates differ by at least $\frac{2|A_i|^2 \cdot \eta}{|A_i|-1} > |A_i| \cdot \eta/2$. Thus, the instance (A_i, η) can be recursively partitioned into yet smaller subinstances. Since at each recursive level, at most half of the points from the previous level remain, the depth of the recursive tree is at most $O(\log n)$. The total running time is therefore $O(n \log^2 n)$. □

2.3 Cells and Polyominoes

Consider an instance (A, η) of the unit disk fencing problem with bounding box S (which is an axis-parallel square of side length $\text{poly}(|A|) \cdot \eta$). We will recursively subdivide S into *cells*, starting with a single cell S, and recursively partitioning each cell into four smaller squares, ending when the side length of the cells is at most $\eta/8$. This happens after some $L = O(\log n)$ recursive operations due to the partitioning described in Section 2.2. We call a cell a *level i cell* if it has been obtained after $i - 1$ levels of subdivision. The square S is thus a level 1 cell.

We consider level i cells and define *polyominoes* to be simple polygons consisting of some level i cells. Two cells are *neighbouring* if their boundaries share an edge or a corner. We will be particularly interested in *monominoes*, which consist of a single cell, *dominoes*, which are the union of two cells sharing an edge, *L-trominoes*, which are the union of three pairwise neighbouring cells (i.e., which are in the shape of the letter L), and *square-tetrominoes*, which are a 2×2-square of cells. A *basic polyomino* is a monomino, a domino, an L-tromino, or a square-tetromino. See Figure 4 for all the basic polyominoes. Note that any non-basic polyomino contains two non-neighbouring cells. Polyominoes consisting of level i cells are called level i polyominoes. We say that a polyomino π is *convex* if the intersection of π with any horizontal or vertical line has at most one connected component.

Note that each level i monomino, domino, tromino, or tetromino, for $i < L$, contains $4, 8, 12$, or 16 cells at level $i+1$, respectively. We define a *subpolyomino* at level $i+1$ to be a convex polyomino at level $i + 1$ that is contained in a basic polyomino at level i. Note that each basic polyomino at level $i + 1$ and at level i is also a subpolyomino at level $i + 1$. For all subpolyominoes, see Figure 5.

As we want each input point to belong to exactly one cell of a given level, we define a cell to include its right and bottom edge and the bottom-right corner. The other edges and corners then belong to the neighbouring cells. For any collection of cells π (not necessarily a polyomino), we define $A_\pi := A \cap \pi$ to be the input points in π. We say that a polyomino π is *empty* if $A_\pi = \emptyset$. We will consider subproblems of the original unit disk fencing problem instance (A, η), each subproblem corresponding to the input points A_π lying in a non-empty basic polyomino π of some level. Note that the number of non-empty basic polyominoes of a given level i is $O(n)$, as each point belongs to a constant number of such polyominoes. Therefore, the total number of
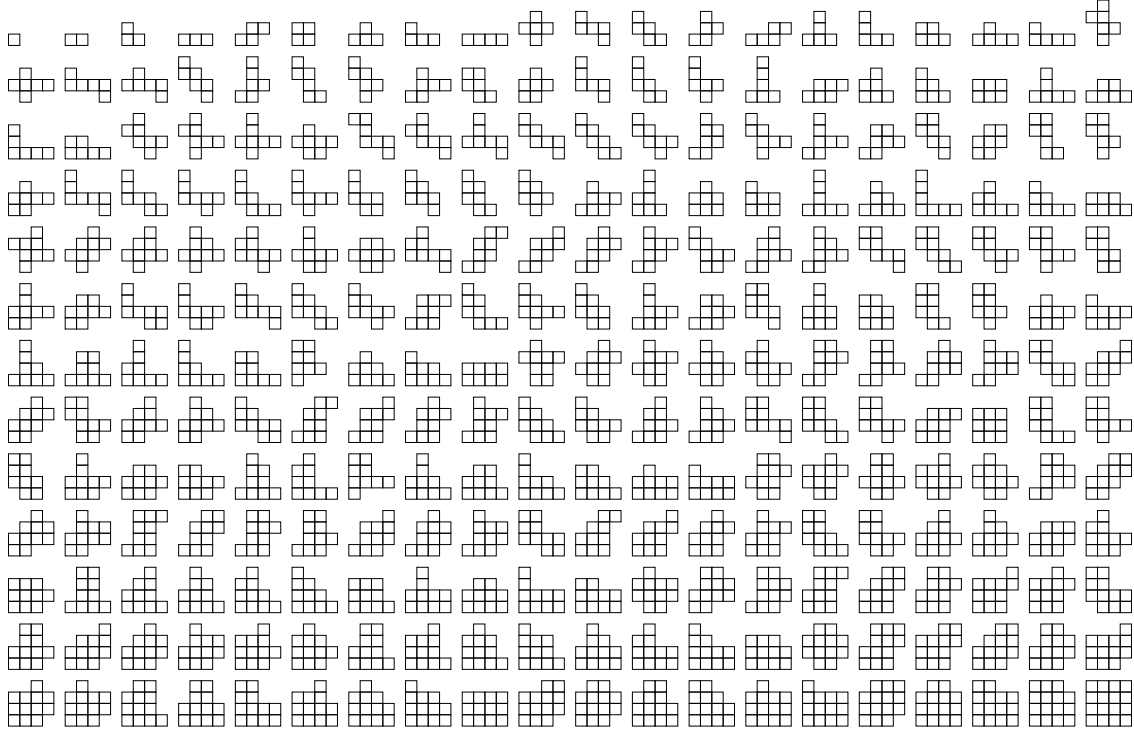
8

Figure 5: The 260 different subpolyominoes (up to rotations and reflections).

non-empty basic polyominoes is at most $O(n \log n)$.

We compute an optimal partition for each subproblem (A_π, η), where π is a non-empty basic polyomino. We start with the polyominoes at level L. At level L, any two points in the same basic polyomino π have distance less than $\eta/2$ and therefore A_π is indivisible. Suppose now that we have already computed the optimal partition for each non-empty basic polyomino at some level $i \le L$. As we will see, this makes it possible to compute the optimal partitions for all non-empty subpolyominoes at level i. Since each basic polyomino at level $i - 1$ is a subpolyomino at level i, we thus also know the optimal partitions of each basic polyomino at level $i - 1$. It follows that the process can be repeated until we reach level 1, i.e., we have computed an optimal partition of A.

To compute and process the polyominoes efficiently, we will use a quadtree construction as described in the next section.

2.4 Quadtree Construction

A *quadtree* is a geometric data structure for objects in the Euclidean plane. The root of the quadtree corresponds to a square containing all the input objects, while the children of each node correspond to the four subsquares of the given square. The leaves correspond to subsquares that have small enough side length, or where the input objects have small complexity, e.g., subsquares containing at most some constant number of input objects. See [11, Chapter 2] for more information on quadtrees, and for their applications.

In our case, the root of the quadtree corresponds to the level 1 cell S. Each node corresponding to a level i cell C has at most 4 children, which correspond to level $i + 1$ cells contained in C. We do not create nodes corresponding to empty cells, i.e., with no points from A. The leaves correspond
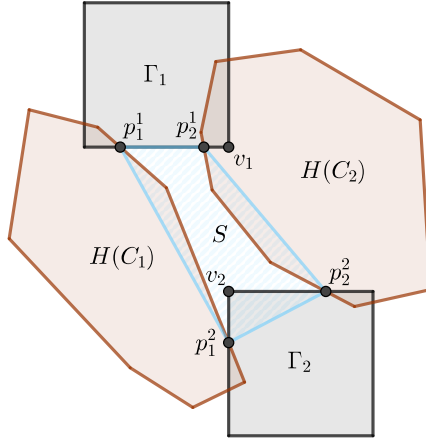
Figure 6: A situation from the proof of Lemma 6.

to the highest level cells, i.e., the side length of the leaf cells is at most $\eta/8$. As there are at most n non-empty cells at each level, the number of nodes of the quadtree is at most $n \cdot L = O(n \log n)$. The quadtree can be constructed in time $O(n \log n)$, as at each of the L levels, we have to compute the subsquares for the n points.

While constructing the quadtree, for each node corresponding to each level i cell, we can compute the nodes corresponding to the eight *neighbouring cells* (if such nodes exist, i.e., the corresponding cells are non-empty). Remembering this information will allow us to construct the polyominoes easier.

From the quadtree, we can construct the set of all non-empty basic polyominoes in time $O(n \log n)$, assigning each basic polyomino π to the nodes of the quadtree corresponding to the cells of π. We do that by considering all nodes of the tree, for each node corresponding to a cell c considering the 21 basic polyominoes containing c, and either constructing a new polyomino, or assigning an existing polyomino to the currently considered node.

2.5   Finding an Optimal Partition for Each Basic Polyomino

We now describe an algorithm for finding an optimal partition for each basic polyomino.

Leaf polyominoes.   Consider a basic polyomino π at level L. As π consists of level L cells, and the side length of such cells is at most $\eta/8$, the distance between any two points in π is smaller than $\eta/2$. Therefore, an optimal partition for (A_π, η) consists of one indivisible set of points A_π. Therefore, optimal partitions for leaf polyominoes can be computed efficiently.

At most one big cluster.   Suppose that we have already computed the optimal partitions for all basic polyominoes at level i. In order to compute the optimal partitions for the basic polyominoes at level $i-1$, we first compute the optimal partitions for all subpolyominoes at level i. This suffices as the basic polyominoes at level $i - 1$ are also subpolyominoes at level i. To find an optimal partition for each subpolyomino π efficiently, we make use the following property.

10

Lemma 6. *Let π be a non-basic polyomino at some level i. Let $\mathcal{C} = \{C_1, \ldots, C_\ell\}$ be a maximal optimal partition of A_π. For any pair Γ_1, Γ_2 of non-neighbouring cells of π, there is at most one cluster $C \in \mathcal{C}$ such that $H(C)$ intersects both Γ_1 and Γ_2. In particular, $\mathcal{C}$ has at most one cluster C such that $H(C)$ intersects all cells of π.*

Proof. Assume that there are two clusters $C_1, C_2 \in \mathcal{C}$ such that $H(C_1)$ and $H(C_2)$ both intersect Γ_1 and Γ_2. The situation is depicted in Figure 6. Since $\mathcal{C}$ is optimal, $H(C_1)$ and $H(C_2)$ are disjoint, and it follows that both boundaries $\partial\Gamma_1$ and $\partial\Gamma_2$ intersect both boundaries $\partial H(C_1)$ and $\partial H(C_2)$. We may then define the point p_j^i, for each choice $i, j \in \{1, 2\}$, to be the intersection point of $\partial\Gamma_i$ with $\partial H(C_j)$ such that (i) $p_1^i p_2^i \cap (H(C_1) \cup H(C_2)) = \{p_1^i, p_2^i\}$, and (ii) $p_j^1 p_j^2 \cap (\Gamma_1 \cup \Gamma_2) \subseteq \{p_j^1, p_j^2\}$. Let S be the quadrilateral with vertices at p_j^i for $i, j \in \{1, 2\}$, and consider the polygon $C_{12} = H(C_1) \cup H(C_2) \cup S$. We will show that $h(C_{12})$ is not larger than $h(C_1) + h(C_2)$. As C_{12} contains all points of $C_1 \cup C_2$, this shows that $\mathcal{C}$ is not optimal or maximal.

In the following, we show the inequality

$$\|p_1^1 p_2^1\| + \|p_1^2 p_2^2\| \leq \|p_1^1 p_1^2\| + \|p_2^1 p_2^2\|. \tag{1}$$

The desired result then follows as the latter sum is at most the length of the perimeter of $H(C_1)$ and $H(C_2)$ contained in S. Let α be the side length of the cells. If Γ_1 and Γ_2 are in the same row or column of cells, then clearly $\|p_1^i p_2^i\| \leq \alpha \leq \|p_j^1 p_j^2\|$, and the inequality 1 holds. Otherwise, let v_1, v_2 be the corners of Γ_1, Γ_2 minimizing the distance between the cells. By considering cases of where the points p_j^i are on $\partial\Gamma_i$, one can observe that it always holds that

$$\|p_1^i p_2^i\| = \sqrt{\|p_1^i v_i\|^2 + \|p_2^i v_i\|^2} \leq \sqrt{\alpha^2 + \|p_j^i v_i\|^2} \leq \sqrt{\alpha^2 + \|p_j^1 v_1\|^2 + \|p_j^2 v_2\|^2} \leq \|p_j^1 p_j^2\|,$$

and inequality (1) follows. $\qquad\square$

Cluster unions. Consider a given subpolyomino π at some level i for which we want to compute the maximal optimal partition of the input points A_π. The overall approach is the following. We use maximal optimal partitions $\mathcal{C}_j$ for A_{π_j} for various smaller collections $\pi_j \subsetneq \pi$ of level i cells. We then construct the *merger* of the partitions $\mathcal{C}_j$, which is the partition of A_π we get when we unite the partitions $\mathcal{C}_j$ and keep merging clusters with overlapping convex hulls. The merger $\mathcal{C}$ thus consists of clusters with pairwise disjoint convex hulls, but is in general not optimal. As we shall see, a maximal optimal partition of A_π can be obtained by uniting one subset of the clusters of $\mathcal{C}$ into one big cluster. This is the motivation for the following definitions.

Let $\mathcal{C} = \{C_1, \ldots, C_\ell\}$ be a partition of a set A of points such that $H(C_i) \cap H(C_j) = \emptyset$ for any $i \neq j$. For any subset $\mathcal{S} \subseteq \mathcal{C}$ consider the set $U_{\mathcal{S}} = \bigcup_{C_i \in \mathcal{S}} C_i$. Let $\mathcal{C}[\mathcal{S}]$ be the partition consisting of the cluster $U_{\mathcal{S}}$ and every $C_i \notin \mathcal{S}$. Consider the set $\mathcal{S}^* \subseteq \mathcal{C}$ minimizing the cost of the partition $\mathcal{C}[\mathcal{S}^*]$. If there is more than one such partition, we are interested in a maximal one. We say that $\mathcal{S}^*$ is an *optimal cluster union* for the clustering $\mathcal{C}$.

Lemma 7. *Let $A = \bigcup_{i \in I} A_i$ be a set of points such that $A = \bigcup_{i \in I} A_i^\complement$, where $A_i^\complement := A \setminus A_i$ for $i \in I$. Let $\mathcal{C}^*$ be the maximal optimal partition of A and suppose that there is at most one cluster in $\mathcal{C}^*$ that intersects each set A_i. Let $\mathcal{C}_i$ be the maximal optimal partition of $A_i^\complement$ and let $\mathcal{C}$ be the merger of the partitions $\mathcal{C}_i$. Let $\mathcal{S}^*$ be an optimal cluster union for $\mathcal{C}$. Then $\mathcal{C}[\mathcal{S}^*]$ and $\mathcal{C}^*$ are the same partition.*

11

Proof. By Lemma 4, each cluster of $\mathcal{C}^*$ is either a cluster of some $\mathcal{C}_i$, or has non-empty intersection with each set A_i. Since there is at most one cluster in $\mathcal{C}^*$ of the latter kind, it follows that $\mathcal{C}^*$ has the form $\{C_1, \ldots, C_k\}$, where each C_i, $i \geq 2$, is contained in a cluster of the partition $\mathcal{C}$. Each cluster C of $\mathcal{C}$ is indivisible by Lemma 1, so it is contained in some cluster $C_i \in \mathcal{C}^*$ by Lemma 2. For each cluster C_i where $i \geq 2$, there must be a cluster $C \in \mathcal{C}$ contained in C_i, and it follows that $C = C_i$. Hence, $\mathcal{C}$ has the form $\{D_1, \ldots, D_\ell, C_2, \ldots, C_k\}$, where $C_1 = \bigcup_{i=1}^{\ell} D_i$. Therefore, the optimal cluster union for $\mathcal{C}$ is $\mathcal{S}^* := \{D_1, \ldots, D_\ell\}$, and $\mathcal{C}[\mathcal{S}^*]$ is the partition $\mathcal{C}^*$. □

Lemma 8. *Let π be a non-basic convex polyomino. There are two cells Γ_1, Γ_2 of π with the following properties:*

(i) Γ_1, Γ_2 are non-neighbouring,

(ii) $\pi \setminus \Gamma_1$ and $\pi \setminus \Gamma_2$ are convex, and either

(iii.a) the horizontal distance between Γ_1 and Γ_2 is at least as large as the vertical distance, Γ_1 is leftmost and Γ_2 is rightmost in π, or

(iii.b) the vertical distance between Γ_1 and Γ_2 is at least as large as the horizontal distance, Γ_1 is topmost and Γ_2 is bottommost in π.

Proof. Since π is non-basic, it has either width of at least 3 cells or height of at least 3 cells. Assume without loss of generality that the width of π is at least as large as the height of π. We will choose Γ_1 to be one of the leftmost cells of π, and Γ_2 to be one of the rightmost cells of π. As we want $\pi \setminus \Gamma_1$ and $\pi \setminus \Gamma_2$ to be convex, Γ_1 and Γ_2 have to be topmost or bottommost in their columns. If there is only one cell in the leftmost (rightmost) column, we take it as Γ_1 (respectively, Γ_2), as then clearly $\pi \setminus \Gamma_1$ (respectively, $\pi \setminus \Gamma_2$) remains a convex polyomino. If there are at least two cells in the column, then, by convexity of π, at least one of them can be removed without disconnecting the polyomino. □

The following lemma states that we can find optimal cluster unions efficiently. The proof is in sections 2.6–2.8.

Lemma 9. *Let π be a collection of cells and Γ_1, Γ_2 two non-neighbouring cells of π such that either*

(i) the horizontal distance between Γ_1 and Γ_2 is at least as large as the vertical distance, Γ_1 is leftmost and Γ_2 is rightmost in π, or

(ii) the vertical distance between Γ_1 and Γ_2 is at least as large as the horizontal distance, Γ_1 is topmost and Γ_2 is bottommost in π.

Let $\mathcal{C}$ be a partition of the points A_π such that for $i \in \{1, 2\}$, $\mathcal{C}$ restricted to the points of $A_{\pi \setminus \Gamma_i}$ is the maximal optimal partition of $A_{\pi \setminus \Gamma_i}$. An optimal cluster union for $\mathcal{C}$ can be found in time $O(n \cdot \operatorname{polylog} n)$, where n is the number of points in A_π.

Solving non-basic subpolyominoes. We can now describe how to find maximal optimal partitions of the points in non-basic subpolyominoes.

Lemma 10. *Let π be a non-basic subpolyomino at level i. Given maximal optimal partitions for basic polyominoes at level i, we can compute an optimal partition of A_π in time $O(n \cdot \operatorname{polylog} n)$, where n is the number of points in A_π.*
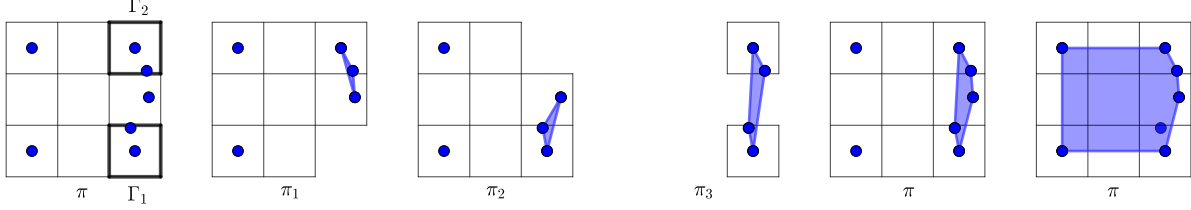
Figure 7: A demonstration of how the optimal partition $\mathcal{C}^*$ of the set A_π of points in the non-basic 3×3-cell polyomino π is obtained from optimal partitions $\mathcal{C}_1^*, \mathcal{C}_2^*, \mathcal{C}_3^*$ of points $A_{\pi_1}, A_{\pi_2}, A_{\pi_3}$ in smaller collections of cells π_1, π_2, π_3, as described the proof of Lemma 10. The cells have width 1 and the opening cost is $\eta := 5/4$. The second, third and fourth figure show the optimal clusterings $\mathcal{C}_1^*, \mathcal{C}_2^*, \mathcal{C}_3^*$. The fifth figure shows the merger $\mathcal{C}$ of these. The final figure shows $\mathcal{C}^*$ (the optimal cluster union consists of all the clusters of $\mathcal{C}$).

Proof. We first find cells Γ_1, Γ_2 of π as in Lemma 8. Let $\pi_1 := \pi \setminus \Gamma_1$, $\pi_2 := \pi \setminus \Gamma_2$, and $\pi_3 := \Gamma_1 \cup \Gamma_2$. As each monomino Γ_i is a basic polyomino at level i, we know the maximal optimal partition of A_{Γ_i} by assumption. Let $\mathcal{C}_3$ be the merger of the maximal optimal partitions of A_{Γ_1} and A_{Γ_2} (which is in fact just the union of the partitions, as the cells are disjoint). Then, $\Gamma_1 \cup \Gamma_2$ together with the partition $\mathcal{C}_3$ satisfy the conditions of Lemma 9, and we can find an optimal cluster union $\mathcal{S}_3$ for $\mathcal{C}_3$ in time $O(n \cdot \text{polylog}\, n)$. Define $\mathcal{C}_3^* := \mathcal{C}_3[\mathcal{S}_3]$. By Lemma 6, the maximal optimal partition of $A_{\Gamma_1 \cup \Gamma_2}$ contains at most one cluster the intersects Γ_1 and Γ_2. Hence, by Lemma 7, $\mathcal{C}_3^*$ is the maximal optimal partition of $A_{\Gamma_1 \cup \Gamma_2}$.

See Figure 7. Denote the maximal optimal partition of A_π as $\mathcal{C}^*$. Consider the sets $A_{\pi_1}, A_{\pi_2}, A_{\pi_3}$, and suppose for now that we know their optimal partitions $\mathcal{C}_1^*, \mathcal{C}_2^*, \mathcal{C}_3^*$. By Lemma 4 (taking $A := A_\pi$, $A_1 := A_{\Gamma_1}$, $A_2 := A_{\Gamma_2}$, and $A_3 := A_{\pi \setminus (\Gamma_1 \cup \Gamma_2)}$), we get that a cluster of $\mathcal{C}^*$ that is not a cluster of any $\mathcal{C}_i^*$ must intersect both Γ_1 and Γ_2. Due to Lemma 6, there is at most one such cluster in $\mathcal{C}^*$. Let $\mathcal{C}$ be the merger of the partitions $\mathcal{C}_1^*, \mathcal{C}_2^*, \mathcal{C}_3^*$. Applying Lemma 9 for π, Γ_1, Γ_2, and $\mathcal{C}$, we obtain an optimal cluster union $\mathcal{S}$ for $\mathcal{C}$. By Lemma 7, we get that $\mathcal{C}[\mathcal{S}] = \mathcal{C}^*$.

As the polyomino π has at most 16 cells, we need to find optimal partitions for a constant number of subpolyominoes π_i before we get down to the basic polyominoes. That gives the total running time of $O(n \cdot \text{polylog}\, n)$. $\square$

Summing up. Consider the subpolyominoes Π_i at some level i. By Lemma 10, the total computation for level i takes time

$$\sum_{\pi \in \Pi_i} O(|A_\pi| \cdot \text{polylog}\, |A_\pi|) \leq \text{polylog}\, n \cdot \sum_{\pi \in \Pi_i} O(|A_\pi|) = O(n \cdot \text{polylog}\, n),$$

where the equality follows since each point belongs to $O(1)$ level i subpolyominoes. Due to the preprocessing described in Lemma 5, the number of levels is $O(\log n)$, so the total running time of the algorithm is $O(n \cdot \text{polylog}\, n)$. We have thus proven the following theorem:

Theorem 1 (The unit disk fencing problem). *There is an algorithm running in $O(n \, \text{polylog}(n))$ time that, given any set A of n points in the plane and an opening cost η, finds a set of ℓ closed curves such that each point in S is enclosed by a curve and the total length of the curves plus $\eta \cdot \ell$ is minimized.*

13

2.6 Finding Optimal Cluster Unions

In order to find optimal cluster unions, we first solve a more specialized problem where we require a special point x_0 to be contained in the cluster union. To be more precise, the optimal cluster union $\mathcal{S}^*$ for the pair $(\mathcal{C}, x_0)$, where $\mathcal{C}$ is a partition, is a maximal subset of the clusters of the partition $\mathcal{C}_{x_0} := \mathcal{C} \cup \{\{x_0\}\}$ such that $\{x_0\} \in \mathcal{S}^*$ and the cost of $\mathcal{C}_{x_0}[\mathcal{S}^*]$ is minimal. Note that we use the terms point and vertex interchangeably in the following.

In this section, we prove the following result.

Lemma 11. *Let* $\mathcal{C} = \{C_1, \ldots, C_\ell\}$ *be a partition of a given set A of points such that $H(C_i) \cap H(C_j) = \emptyset$ for any $i \neq j$, and let x_0 be an arbitrary point. A maximal optimal cluster union for $(\mathcal{C}, x_0)$ can be found in time $O(n \cdot \mathrm{polylog}\, n)$, where n is the number of points in A.*

Our first goal is to solve the following more specialized problem. Given an "internal" point x_0 and a "perimeter" point $p \neq x_0$, and given a set of input points and pre-clustered input points, the goal is to find the optimal cluster containing x_0 with an angle-monotone perimeter seen from x_0 and with p on its perimeter. We present an algorithm to solve this problem. The algorithm can take into account that the cluster must be contained within some delimiting outer boundary.

The idea is to make an angular sweep of a ray from x_0, and consider the points in the order the ray sweeps over them. For each point v, we calculate the *best* path (see Section 2.6.4) from p to v, in terms of both its length and how many clusters' opening costs it may save. In this process, we only store for each vertex its *parent* $\mathrm{par}(v)$, which is an input point with the property that a best path to v ends with the line segment from $\mathrm{par}(v)$ to v. Finally, we calculate the parent of p, and have thus specified the entire cluster and may generate its convex hull by recursively outputting the parent until we end back at p.

2.6.1 Preliminaries

Definition 1. Any closed simple curve σ divides the plane $\mathbb{R}^2$ into two regions: the bounded *interior*, denoted $\mathrm{int}(\sigma)$, and the unbounded *exterior*, denoted $\mathrm{ext}(\sigma)$. We write $\overline{\mathrm{int}(\sigma)} = \sigma \cup \mathrm{int}(\sigma)$ and $\overline{\mathrm{ext}(\sigma)} = \sigma \cup \mathrm{ext}(\sigma)$.

Definition 2. A region R of Euclidean space is *star shaped* if there exists a point $s \in R$ such that for all $r \in R$, the line segment sr is contained in R. We say that R is star shaped *seen from s*.

Definition 3. Given a point s, a curve $\pi \colon [0,1] \longrightarrow \mathbb{R}^2$ is *angle-monotone* with respect to s if $\angle(\pi(0), s, \pi(x)) < \angle(\pi(0), s, \pi(x'))$ for $x < x'$. Here, $\angle(a, b, c)$ is the counterclockwise angle from the unit vector $\frac{a-b}{\|a-b\|}$ to the unit vector $\frac{c-b}{\|c-b\|}$ on the unit circle.

When a cluster only contains one point, we call it *trivial*, otherwise, it is *non-trivial*. If a curve intersects the interior of a cluster, we say that it *dissects* the cluster. Given a partition $\mathcal{C}$ of some points, we denote by $V(\mathcal{C})$ the set of points, $V(\mathcal{C}) := \bigcup_{C \in \mathcal{C}} C$.

We denote by $B_r(x)$ the ball of radius r around the point x.

2.6.2 Problem formalization

We are given:

- special points x_0 and p,
- clusters $\mathcal{C} := \{C_0, \ldots, C_{l-1}\}$ with costs $\mathrm{cost}(C_i)$,

- an *outer limit*, corresponding to the perimeter of an unbounded cluster C_l, which the cluster containing x_0 is not allowed to intersect.

We may assume that no two clusters touch or intersect, and that each cluster is convex. Unclustered points v are treated as trivial clusters $C_i = \{v\}$.

Definition 4. $Q_{\mathcal{C}}(p, x_0)$ is the set of all closed simple curves σ with $p \in \sigma$, not dissecting any cluster in $\mathcal{C}$, and such that σ is angle-monotone from x_0.

The cost of a curve $\sigma \in Q_{\mathcal{C}}(p, x_0)$ is:

$$\text{length}(\sigma) + \sum_{C \in \mathcal{C}, C \subseteq \text{ext}(\sigma)} \text{cost}(C).$$

Note that we sometimes omit the subscript $\mathcal{C}$ when it is clear from context.

Problem. Given $\mathcal{C}$, C_l, p, and x_0 as described above, compute $\inf_{\sigma \in Q_{p,x_0}} \text{cost}(\sigma)$.

Note here that even if no outer limit is given as part of the construction, we may take any bounding box around x_0 containing all of $V(\mathcal{C})$ as an outer limit.

Lemma 12. *We can compute* $\inf_{\sigma \in Q_{p,x_0}} \text{cost}(\sigma)$ *and also output* $\arg\min_{\sigma \in Q_{p,x_0}} \text{cost}(\sigma)$ *in time* $O(n \log n)$, *where n is the total number of vertices in $V(\mathcal{C})$.*

The rest of this section is dedicated to a proof of the lemma above.

2.6.3 Reduction to the case where every cluster is non-trivial.

We reduce the original $\mathcal{C}$ to an instance $\mathcal{C}'$ where every cluster has a non-empty interior.

This reduction follows the framework of symbolic perturbation [9, 15, 14]. We introduce an infinitesimal ε, and replace each vertex $v \in V(\mathcal{C}) \cup \{x_0\}$ by three vertices v', v'', v''' in an equilateral triangle centered at v. Each precluster $C' \in \mathcal{C}'$ will consist of the set $\{v', v'', v''' \mid v \in C\}$ for some $C \in \mathcal{C}$. We thus have that every cluster in $\mathcal{C}'$ has a non-trivial interior.

Finally, we replace every vertex $v \in V(\mathcal{C}')$ by $v + \varepsilon^2 \cdot (\mathcal{N}(0, 1), 0)$, that is, we perturb each point by a very small random number, such that no three points lie on the same line. Therefore, in the following we can assume all the vertices are in general position.

Note that we may disregard any vertex that does not lie on the convex hull of its cluster.

2.6.4 Subproblem structure.

For any cluster C not containing x_0, note that the set of angles $\{\angle(p, x_0, v) \mid v \in H(C)\}$ is either an interval $[a, b]$ with $0 < a < b < 2\pi$, or the union of intervals $[a, 2\pi) \cup [0, b]$, with $0 < b < a < 2\pi$. Because of the symbolic perturbation introduced in Section 2.6.3, these values a and b are realized by unique vertices on the convex hull of C. Denote by $l(C)$ the vertex realizing a, and by $r(C)$ the vertex realizing b.

Definition 5. For $t \notin \{p, x_0\}$, $Q_{\mathcal{C}}(x_0, p, t)$ is the set of all simple angle-monotone curves π from p to t, not dissecting any cluster C.

Denote by $\text{cone}(p, t)$ the cone with apex x_0 through p and t. Denote by $\text{int}(\pi)$ and $\text{ext}(\pi)$ the bounded, and unbounded, region of $\text{cone}(p, t) \setminus \pi$, respectively.

The cost of a curve π is:

$$\text{cost}(\pi) = \text{length}(\pi) + \sum_{C \in \mathcal{C}, l(C) \in \overline{\text{ext}(\pi)}} \text{cost}(C).$$

15

Observation 3. *Let $\mathcal{C}$ and $\mathcal{C}'$ be as in the reduction (Section 2.6.3). Then, since ε was chosen to be infinitesimal, the difference between $\inf_{\pi \in \mathcal{C}} \operatorname{cost}(\pi)$ and $\inf_{\pi' \in \mathcal{C}'} \operatorname{cost}(\pi')$ is also infinitesimal.*

Definition 6. $P_{\mathcal{C}}(x_0, p, t) \subseteq Q_{\mathcal{C}}(x_0, p, t)$ is the subset of *polygonal curves* consisting of line segments between points of $V(\mathcal{C})$.

Lemma 13.
$$\inf_{\pi \in P_{\mathcal{C}}(x_0,p,t)} \operatorname{cost}(\pi) = \inf_{\pi' \in Q_{\mathcal{C}}(x_0,p,t)} \operatorname{cost}(\pi').$$

Proof. For any curve π in $Q_{\mathcal{C}}(x_0, p, t)$, let V^{int} denote the set of internal points, and V^{ext} denote the set of external points. Then, the shortest curve separating V^{int} and V^{ext} will consist of line segments between vertices of $V^{\text{int}} \cup V^{\text{ext}}$. Since $\operatorname{length}(\pi') \leq \operatorname{length}(\pi)$, and since they cover the same clusters, $\operatorname{cost}(\pi') \leq \operatorname{cost}(\pi)$. $\square$

Definition 7. For $p \neq t \neq x_0$, $P_{\mathcal{C}}(x_0, p, t) \neq \emptyset$, let $\pi(t) = \arg\min_{\pi \in P_{\mathcal{C}}(x_0,p,t)} \operatorname{cost}(\pi)$. Since $\pi(t)$ is polygonal, we may write it as the polygonal curve on the vertex set $p = \pi_1, \ldots, \pi_z = t$. We say that π_{z-1} is the *parent* $\operatorname{par}(t)$ of t.

The cost of $\pi(t)$ may be rewritten in the following way:

$$\operatorname{cost}(\pi(t))$$
$$= \operatorname{length}(\pi_1, \ldots, \pi_{z-1}) + \sum_{C \in \mathcal{C}, l(C) \in \overline{\operatorname{ext}(\pi)}} \operatorname{cost}(C) + d(\pi_{z-1}, t).$$

We denote by $\operatorname{fixedcost}(t)$ the expression $\operatorname{length}(\pi_1, \ldots, \operatorname{par}(t)) + \sum_{C \in \mathcal{C}, l(C) \in \overline{\operatorname{ext}(\pi)}} \operatorname{cost}(C)$.

When $P_{\mathcal{C}}(x_0, p, t) = \emptyset$ for some point not contained in any cluster $C \in \mathcal{C}$, we set $\operatorname{par}(t) = \perp$ and $\operatorname{fixedcost}(t) = \infty$. When a point lies in the interior of some cluster, $t \in \operatorname{int}(C)$, or when t lies outside the outer limit, we write $\operatorname{par}(t) = \emptyset$ and $\operatorname{fixedcost}(t) = \infty$.

2.6.5 Calculating parents via a data structure of intervals.

For each point $v \in V(\mathcal{C})$, we may calculate its angle $\angle(p, x_0, v)$. We may sort the vertices of $V(\mathcal{C})$ increasing according to their angle, $p = v_0, v_1, \ldots, v_l = p'$, where p' represents a copy of p with $\angle(p, x_0, p') = 2\pi$, for convenience.

We rotate a sweep-ray $\mathcal{R}$ from x_0 through v_i for $i = 0, 1, \ldots, l$. While doing so, we maintain a data structure that encodes for each $r \in \mathcal{R}$ the values $\operatorname{fixedcost}(r)$ and $\operatorname{par}(r)$. Here, if $\operatorname{par}(v_i) = v_j$, then $j < i$.

The data structure maintains a partitioning of $\mathcal{R}$ into maximal intervals $\{I_0, \ldots, I_l\}$ with the same values of $(\operatorname{fixedcost}(t), \operatorname{par}(t))$ for all $t \in I_i$. The first and last intervals are special in the sense that the first, I_0, is dedicated to the cluster containing x_0, and the last, I_l, is dedicated to ensuring that we do not cross the outer limit specified in the problem instance.

We observe that the boundaries between intervals only change continuously, and we thus only store an implicit representation of them. We only consider the sweep-ray at discrete times when either

- we reach a vertex v_i, corresponding to a vertex insertion in the data structure, or,
- an interval shrinks to $\emptyset$, corresponding to the interval being deleted from the data structure.

16

For any vertex $v \in V(\mathcal{C})$, it dominates a region of $\mathbb{R}^2$ in the sense that every point in that region would have v as its parent. We call these points the *children* of v:

$$\text{children}(v) = \{t \in \mathbb{R}^2 \mid v = \text{par}(t)\}.$$

Lemma 14. *The region* $\text{children}(v)$ *is star shaped seen from* v.

Proof. Let $t \in \text{children}(v)$, denote by π the optimal curve from p to t, and assume for contradiction $t' \notin \text{children}(v)$ for some t' on the line segment between t and v. Then, $\text{par}(t') = v' \neq v$. Let π' be the optimal path from p to t'. Note that π' passes through $\text{par}(t') = v' \neq v$. But then, since both π and π' pass through t', concatenating π' with the line segment $t't$ is better than π. $\qquad\square$

Denote by $P_{\mathcal{R}}$ the partitioning of the ray $\mathcal{R}$ into intervals consisting of points that have the same parent. For an interval $I \in P_{\mathcal{R}}$, denote by $\text{par}(I)$ the shared parent $p(t)$ of all points $t \in I$.

Lemma 15. $|P_{\mathcal{R}}| \leq 4n + 4$ *for all angles.*

Proof. Enumerate the intervals of $\mathcal{R}$ consecutively by $I_0, I_1, \ldots$, such that I_0 has x_0 as an endpoint.
For each cluster in $\mathcal{C}$ with a non-trivial intersection with $\mathcal{R}$, we have an unreachable interval I with $\text{par}(I) = \emptyset$. A cluster intersecting with $\mathcal{R}$ can contribute with at most 1 unreachable interval I with $\text{par}(I) = \perp$, namely, the interval above or below the cluster. For any other interval I we have that $\text{par}(I) \in \{v_0, \ldots, v_l\}$. It follows from Lemma 14 that the intervals form a hierarchy, that is, if $i < j < k$, and $\text{par}(I_i) = \text{par}(I_k) \neq \text{par}(I_j)$, then for any I_l with $\text{par}(I_l) = \text{par}(I_j)$, it must hold that $i < l < k$. It follows that there are at most $2n$ such intervals. $\qquad\square$
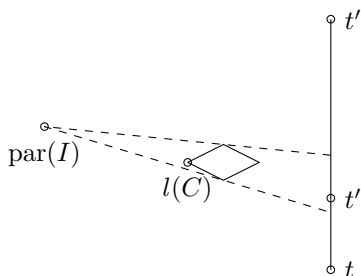

Figure 8: The value of $\text{fixedcost}(t)$ is fixed within an interval.

Lemma 16. *For any* $I \in P_{\mathcal{R}}$, *for* $t, t' \in I$, *we have* $\text{fixedcost}(t) = \text{fixedcost}(t')$.

Proof. Let $t, t' \in I$. Then, $\text{par}(t) = \text{par}(t') = \text{par}(I)$. But then, the only way $\text{fixedcost}(t)$ could be different from $\text{fixedcost}(t')$ would be if the set of clusters $C \in \mathcal{C}, l(C) \in \overline{\text{ext}(\text{par}(t)t)}$ differed from the set of clusters $C \in \mathcal{C}, l(C) \in \overline{\text{ext}(\text{par}(t')t')}$.
Assume for contradiction that there exists a cluster C with $l(C) \in \overline{\text{ext}(\text{par}(I)t')}$ and $l(C) \notin \overline{\text{ext}(\text{par}(I)t)}$ (see Figure 8). Then, there would exist a point $t'' \in \mathcal{R}$ between t and t' with $\text{par}(t'') \neq \text{par}(I)$, since any line from t'' dissects C. But then, $t'' \in I$ with $\text{par}(t'') \neq \text{par}(I)$; contradiction. $\qquad\square$

We denote by $\text{fixedcost}(I)$ the constant value $\text{fixedcost}(t)$ for any $t \in I$.

Determining the boundaries of regions. We may categorize boundaries based on how they arise.

Some are lines due to visibility (see Figure 9, left), while some are found by equating two costs:

$$\text{fixedcost}(I) + d(\text{par}(I), t) = \text{fixedcost}(I') + d(\text{par}(I'), t).$$

One may easily check that this yields a quadratic equation, and thus describes a parabola or a line (see Figure 9, right).
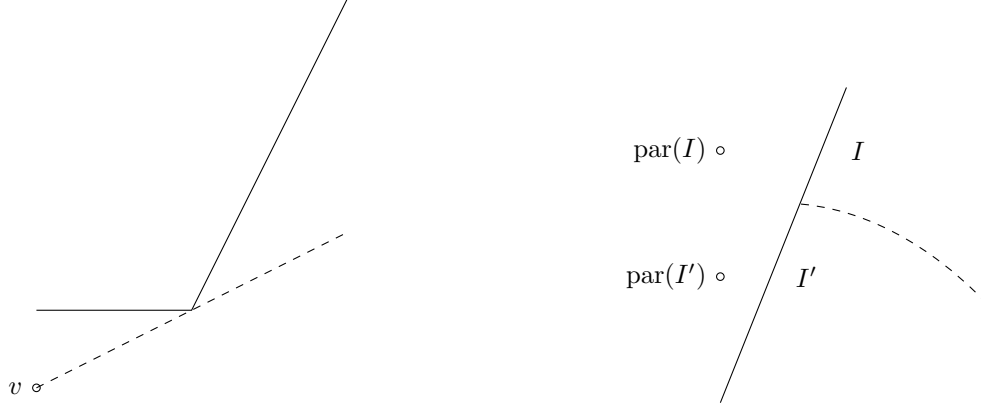

Figure 9: Boundaries between intervals are either a line due to visibility (left) or described by a quadratic equation (right).

Hence, we may store the intervals and the functions describing the boundaries between the intervals on the ray. We may build a balanced binary search tree over these intervals, such that each leaf is an interval, and each non-leaf node stores the function describing the boundary between its children. We call this the *succinct representation* of $P_{\mathcal{R}}$. We note that this succinct representation is enough to determine the parent of a point on the sweep-ray:

Lemma 17. *Given a succinct representation of $P_{\mathcal{R}_\alpha}$ at some angle α, and assuming no boundaries of intervals intersect between angles α and β, then for any $t \in \mathcal{R}_\beta$ we can find $(\text{fixedcost}(t), \text{par}(t))$ in $O(\log n)$ time.*

Proof. This can be done by searching the balanced binary search tree of $\mathcal{R}_\alpha$. At each node, starting with the root, we may evaluate the function describing the boundary between its children at the angle β, and recurse into the child whose range contains t. Since the tree is a balanced binary search tree over the $O(n)$ intervals, it has height $O(\log n)$, and thus the total time is $O(\log n)$. □

Initializing the data structure. The sweep-ray $\mathcal{R}$ is initialized at angle 0 (that is, with $p \in \mathcal{R}$). It is initialized with an interval I_0 containing x_0. For each cluster crossing the line segment between x_0 and p, we have two consecutive intervals: an unreachable interval I_{2m-1} with $\text{par}(I_{2m-1}) = \bot$, and an interval corresponding to the inside of the cluster I_{2m} with $\text{par}(I_{2m}) = \emptyset$. Then, there will be an interval I_p with $\text{par}(I_p) = p$, and then, for each cluster intersecting the ray after p, there will be two consecutive intervals I_{2m} and I_{2m+1} with $\text{par}(I_{2m}) = \emptyset$ and $\text{par}(I_{2m+1}) = \bot$. Finally, the last interval will be delimited by the outer limit given as part of the input.

For any pair of consecutive intervals, exactly one of them corresponds to a cluster, and the boundary between them follows the line segment of the convex hull of the cluster.

18

For all intervals I, fixedcost(I) is initialized to 0.

2.6.6 Updating the data structure.

We now describe how we update the interval structure of the sweep-ray.

There are two types of updates. Vertex insertions, and interval annihilations. Whenever we update the data structure, we can foresee when the next interval annihilation will occur. Thus, when we encounter a point, we know that the topology of $P_{\mathcal{R}}$ has not changed.

Interval annihilations. An interval I_i shrinks to $\emptyset$ if and only if its boundaries to its two neighbors, I_{i-1} and I_{i+1}, intersect (see Figure 10).

For any triplet of consecutive intervals, I_{i-1}, I_i, I_{i+1}, we store the predicted annihilation-angle for I_i, that is, when I_{i-1} and I_{i+1} have dominated it. Whenever an interval disappears, only at most two such triplets are affected, namely, we must make a new predicted annihilation-angle for I_{i-1} and for I_{i+1}, where they take each other into account.

Note that by star shapedness (Lemma 14), par(I_{i-1}) $\neq$ par(I_{i+1}), and thus we never have to merge intervals.
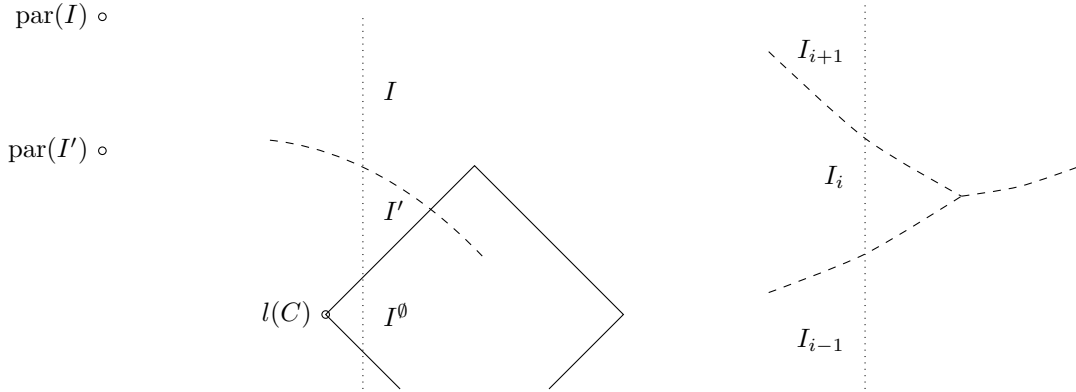

Figure 10: An interval is deleted because it is dominated by its neighbors.

Vertex insertions. Assume we reach a vertex t and update our sweep-ray such that $t \in \mathcal{R}$. Due to Lemma 17, we can determine the interval I containing t, and thus, determine par(t) and fixedcost(t).

Now, t gives rise to at most three intervals in $\mathcal{R}$, which we add to the data structure. Recall that every vertex we consider lies on the convex hull of some non-trivial cluster C. In the following, denote by e^- and e^+ the edges of the perimeter of C incident to t such that e^- comes just before e^+ in a clockwise order. We divide our description into four cases, that depend on how the edges of the convex hull incident to t relate to $\mathcal{R}$ (see Figure 11).

1. $t = l(C)$,
2. t lies on the outer perimeter of C, that is, $\exists t' \in C \cap \mathcal{R}$ with t' closer to x_0 than t,
3. t lies on the inner perimeter of C, that is, $\exists t' \in C \cap \mathcal{R}$ with t' further from x_0 than t,
4. $t = r(C)$.

Figure 11: Cases 1, 2, 3, and 4.

Since C is convex, these are the only cases.

In Case 1, a new interval $I^{t,\emptyset}$ is created, with $\text{par}(I^{t,\emptyset}) = \emptyset$ and $\text{fixedcost}(I^{t,\emptyset}) = \infty$. The succinct representation of the boundaries of $I^{t,\emptyset}$ are just the straight lines prolonging e^- and e^+ (see Figure 12, left and middle).

In Cases 1, 2, and 3, everything that is visible from $\text{par}(t)$ belongs to $\text{children}(\text{par}(t))$, and everything else belongs to $\text{children}(t)$ unless it belongs to $I^{t,\emptyset}$ (see Figure 12).

Lemma 18. *The intervals described above are the correct intervals for Cases 1, 2, and 3.*

Proof. Because of the symbolic perturbation of Section 2.6.3, we may assume that t does not lie on the border between intervals. That is, there exists some $\delta > 0$ such that all of $B_\delta(t) \cap \mathcal{R}$ belongs to the interval I containing t. But then, there exists some δ' with $\delta \geq \delta' > 0$, such that $B_{\delta'}(t) \subseteq \text{children}(\text{par}(t)) \cup \text{children}(t) \cup \text{children}(\emptyset)$.

Then, for any reachable point in $v \in B_{\delta'}(t)$ (i.e., $v \in B_{\delta'}(t)$ such that $P_\mathcal{C}(x_0, p, v) \neq \emptyset$), the optimal path goes through $\text{par}(t)$ either as the second to last, or as the third to last vertex. Thus, if $\text{par}(t)$ is visible from v, then $\text{par}(v) = \text{par}(t)$. Otherwise, necessarily, $\text{par}(v) = t$. $\square$
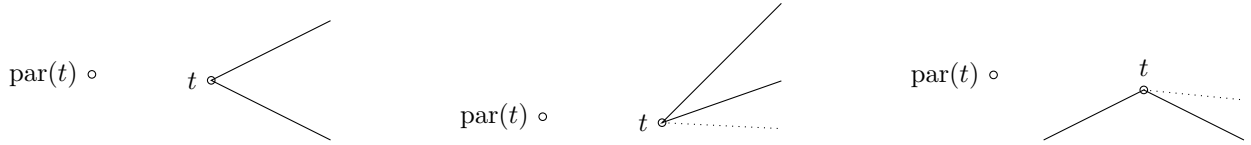

Figure 12: The vertex t gives rise to at most 3 new intervals.

Finally, in Case 4, t lies on the border between the intervals I, I'. We may in constant time calculate and compare the two costs and determine whether $\text{par}(t) = \text{par}(I)$ or $\text{par}(t) = \text{par}(I')$. It follows from the symbolic perturbation in Section 2.6.3 that there are no ties; one is strictly better than the other. Given $\text{par}(t)$, we may calculate $\text{fixedcost}(t)$.

We now introduce a new interval $\widetilde{I}$ that has $\text{par}(\widetilde{I}) = t$ and $\text{fixedcost}(\widetilde{I}) = \text{fixedcost}(t)$. We need to initialize the boundaries of $\widetilde{I}$ correctly: note that an arbitrary number of intervals may be deleted because they are dominated by $\widetilde{I}$, however, they are consecutive intervals, and we may spend constant time on each deletion, since deleted intervals will never resurrect (this follows from the star shapedness property in Lemma 14).

Lemma 19. *The interval described above is the correct interval for Case 4.*

Proof. Assume $\text{par}(t) = \text{par}(I)$, which is strictly better than $\text{par}(I')$. Then, following the same lines as the proof of Lemma 18, there exists a small $\delta' > 0$ such that $B_{\delta'}(t) \subseteq \text{children}(\text{par}(t)) \cup \text{children}(t) \cup \text{children}(\emptyset)$. Thus, it is necessary to introduce the interval $\widetilde{I}$. It follows from star shapedness (Lemma 14) that introducing $\widetilde{I}$ is also sufficient. $\square$
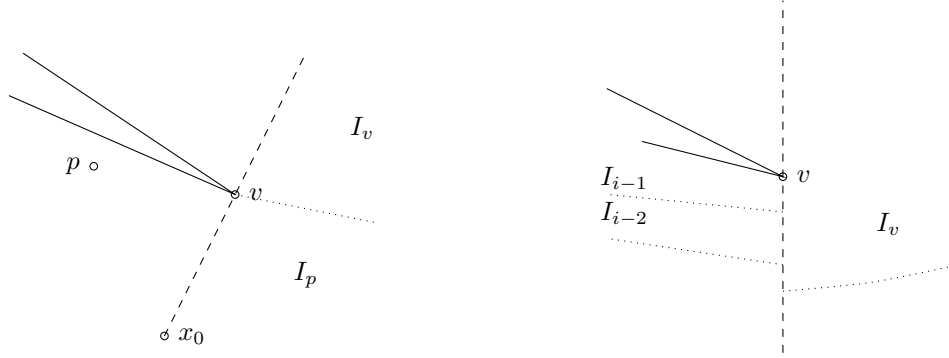
20

Figure 13: The insertion of the vertex $v = r(C)$ may cause the deletion of several intervals.

What remains is to handle the changes to fixedcost efficiently and correctly. The only change to fixedcost of any cluster happens in Case 1: the cost of C needs to be added to fixedcost(I) for every interval I between x_0 and t on $\mathcal{R}$.

Note that no boundaries change because of this: if I' and I'' are intervals that are not separated by t, then their fixedcost either stays the same or is increased by the same amount, that is, cost(C). The only boundary that could be affected is one incident to $I^{t,\emptyset}$. However, the boundaries of $I^{t,\emptyset}$ are lines determined by the convex hull of C.

Finally, we argue that fixedcost can be added to all intervals between x_0 and t in $O(\log n)$ time: since we store $P_\mathcal{R}$ in a balanced binary search tree, we simply update every vertex of the root path of $I^{t,\emptyset}$ with a lazy delta value that they should push down to their children upon inspection.

2.7 Proof of Lemma 11

In this section, we assume that we are given a polyomino, a point x_0 belonging to an optimal cluster union, and a set of clusters of points within the polyomino. We will be concerned with Jordan curves that are not self-intersecting. When we consider a *curve* we only refer to such a curve.

We say that a ray $\mathcal{R}$ is *good* if it satisfies the following conditions:

- it starts at a point x_0 of an optimal cluster union, and
- any input point is at distance at least η/n^2 from the ray.

2.7.1 Finding a good ray.

Lemma 20. *Given a point x_0 that belongs to an optimal cluster union, there exists an $O(n \log n)$ time algorithm that finds a good ray.*

Proof. We know by definition of the point set that x_0 is at distance at least $\eta/2$ from any other point. Thus, we can perform a clockwise scan around x_0 and, by the pigeonhole principle, there must be at least one ray out of x_0 that is at distance at least η/n^2 from all the other points. This ray is a good ray. □

2.7.2 Nested curves.

For any point $p \in \mathbb{R}^2$, denote by $\sigma_{\min}(p)$ the optimal curve found in Lemma 12.

We want to show that we can find a suitable p such that the convex hull $H(C)$ of the cluster C containing x_0 in an optimal clustering is the same curve as $\sigma_{\min}(p)$.

To show this, we need some structural lemmas, stating that the curves for different values $p_1, p_2, \ldots$ along a ray are nested nicely around x_0 without crossings.

Lemma 21. *Consider two points p_1, p_2 on the same ray from x_0, and assume that $d(p_2, x_0) > d(p_1, x_0)$. Then, $\sigma_{\min}(p_1)$ is completely contained in $\overline{\mathrm{int}(\sigma_{\min}(p_2))}$.*
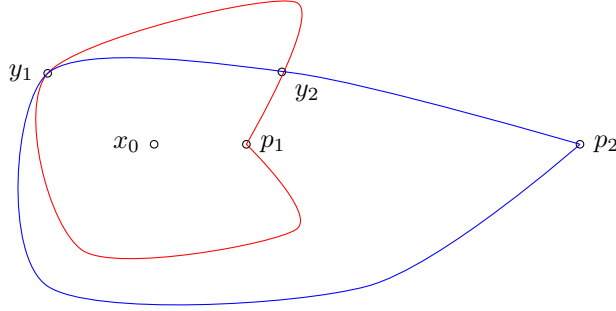

Figure 14: When p_1 and p_2 are on the same ray (here, aligned with the x-axis), their best curves do not cross.

Proof. Assume for contradiction that $C_1 = \sigma_{\min}(p_1)$ intersects the exterior of $C_2 = \sigma_{\min}(p_2)$ (see Figure 14). Since $p_1 \in C_1$ and $p_1 \in \mathrm{int}(C_2)$, we have by Jordan's theorem that the curves cross at least twice. Consider two crossing points y_1, y_2. Denote by $C_i^{|y_2, y_1|}$ the part of $C_i \setminus \{y_1, y_2\}$ containing p_i (for $i = 1, 2$), and let $C_i^{|y_1, y_2|}$ denote the part not containing p_i.

Denote by $\sigma_{\min}(y_1, y_2) = \arg\min_{\pi \in Q_C(x_0, y_1, y_2)}$ the best curve from y_1 to y_2 (see Definition 5). Since there was a crossing, we must have that $\sigma_{\min}(y_1, y_2) \neq C_i^{|y_1, y_2|}$ for some $i \in \{1, 2\}$, and thus $\mathrm{cost}(\sigma_{\min}(y_1, y_2)) < \mathrm{cost}(C_i^{|y_1, y_2|})$.

But then, $\mathrm{cost}(\sigma_{\min}(p_i)) > \mathrm{cost}(C_i^{|y_2, y_1|} \circ \sigma_{\min}(y_1, y_2))$, which is a contradiction since $C_i^{|y_2, y_1|} \circ \sigma_{\min}(y_1, y_2) \in Q_C(p_i, x_0)$. □

Denote by σ^* the best curve containing x_0, that is, the curve describing the convex hull of the cluster C_{x_0} containing x_0 in the optimal clustering.

Lemma 22. *For any point p, $\sigma_{\min}(p)$ and σ^* do not cross.*

Proof. Denote by p' the intersection between σ^* and the ray from x_0 through p. It follows from convexity and boundedness of C_{x_0} that p' is uniquely defined.

Clearly, $\sigma^* = \sigma_{\min}(p')$. But now, p and p' are points on the same ray from x_0, and thus, it follows directly from Lemma 21 that $\sigma_{\min}(p)$ and $\sigma_{\min}(p') = \sigma^*$ do not cross. In fact, we have that σ^* is completely contained in either $\overline{\mathrm{int}(\sigma_{\min}(p))}$ or $\mathrm{ext}(\sigma_{\min}(p))$. □

22

2.7.3 Discretizing the ray.

Consider a good ray $\mathcal{R}$ with origin x_0. For ease of exposition, we rotate the point set and assume that $\mathcal{R}$ is parallel to the x-axis. Define the *landmarks* of $\mathcal{R}$ to be the points of $\mathcal{R}$ at distance $i \cdot \eta/n$ from x_0, for any non-negative integer $0 \leq i \leq n^7$. In particular, we call the point on $\mathcal{R}$ at distance $i \cdot \eta/n$ from x_0 the ith landmark and denote it by x_i.

Denote by H_i the region of the plane containing all the points having x-coordinate greater than the x-coordinate of x_i. Note that $H_1 \supset H_2 \supset H_3 \supset \ldots$. Denote by L (respectively R) the region of the plane that contains all the points with y-coordinate greater (respectively smaller) than the y-coordinate of x_0. Furthermore, we define the ith curve C_i to be the best curve in $\mathbb{R}^2 \setminus H_i$ that intersects $\mathcal{R}$ at the ith landmark.

For any curve C, let $V^{\mathrm{int}}(C)$ and $V^{\mathrm{ext}}(C)$ be the set of points in $\mathrm{int}(C)$ and $\mathrm{ext}(C)$, respectively.

Lemma 23. *There exists an index i^* satisfying the property $n/2 \geq |V^{\mathrm{int}}(C_{i^*-1})|, |V^{\mathrm{ext}}(C_{i^*})|$.*

Proof. The proof follows immediately from Lemmas 21 and 5. Since the curves are nested, it must be that there is such an index i^*. $\qquad\square$

Lemma 24. *Consider the best curve C^*. We have that C^* is contained in*

1. $\overline{\mathrm{int}(C_{i^*-1})}$, *or*
2. $\overline{\mathrm{ext}(C_{i^*})}$, *or*
3. $\overline{\mathrm{int}(C_{i^*})} \cap \overline{\mathrm{ext}(C_{i^*-1})}$.

Define u to be the first input point met by a clockwise walk on C_{i^*} starting at x_{i^*} and similarly define v to be the first input point met by a counterclockwise walk on C_{i^*} starting at x_{i^*}. Consider the angle $\angle(u, x_{i^*}, v)$ and the bisector of this angle. Let s be the line perpendicular to the bisector that goes through x_{i^*-1}. We now define the *tube T* to be the set of points that are in the interior of C_{i^*} and at distance less than $\eta/2$ from s.

We say that a point (w_x, w_y) is *above s* if $w_y \geq g(w_x)$, where g is the equation of line s. Define r to be the point of $C_{i^*} \cap R$ that is above s and the farthest away from s, and ℓ the point of $C_{i^*} \cap L$ that is above s and the farthest away from s.

Lemma 25. *Consider the best curve C^* that crosses $\mathcal{R}$ in the interval $[x_{i^*-1}, x_{i^*})$. We have that either both ℓ and r belong to the tube or there exists a curve with a better cost that lies in $\mathrm{ext}(C_{i^*}) \cup C_{i^*}$.*

Proof. Assume that C^* is the best curve overall. By Lemma 21, C^* lies in $\mathrm{int}(C_{i^*}) \cup C_{i^*}$. Consider C_{i^*} and C^*. Observe that the distance from x_{i^*} to C^* is at most η/n^7. Therefore, the cost of C_{i^*} is at most the cost of C^* plus $2\eta/n^7$.

Now, observe that if the angle at x_{i^*} is at least π, then the lemma holds immediately as the interval $[x_{i^*-1}, x_{i^*})$ has size η/n^7 and the tube has size $\eta/2$ and C^* crosses $[x_{i^*-1}, x_{i^*})$ and remains in $\mathrm{int}(C_{i^*}) \cup C_{i^*}$.

Thus, assume that the angle is at most π, and let Θ denote this angle. Observe that since the maximum length of an edge is bounded by $n^2\eta$, we have that the angle has to be such that $n^2 \cdot \eta \cdot \sin(\Theta) > \eta/2$ in order for r to be outside of the tube. Thus $\sin(\Theta) > 1/(2n^2)$.

We first prove some useful facts regarding $\sin(x)$ and $\cos(x)$. First, we have $\sin(x) \leq x$ for all $x \geq 0$, which holds as the function $f(x) = x - \sin(x)$ satisfies $f(0) = 0$ and $f'(x) = 1 - \cos(x) \geq 0$. Moreover, for all $0 \leq x \leq \pi/2$, we have the property that $\cos(x) \leq 1 - x^2/\pi \leq 1 - x^2/4$. To

see this, consider the function $g(x) = 1 - x^2/\pi - \cos(x)$. This function satisfies $g(0) = 0$ and $g'(x) = -2x/\pi + \sin(x) \geq 0$, where the inequality holds since $\sin(x) \geq 2x/\pi$ for all $0 \leq x \leq \pi/2$ (since $\sin(x)$ is concave on the interval $[0, \pi/2]$, $\sin(0) = 0 = 2\cdot 0/\pi$, and $\sin(\pi/2) = 1 = 2\cdot(\pi/2)/\pi$).

Now, since $\sin(\Theta) > 1/(2n^2)$, we know that $\Theta \leq \pi - 1/(2n^2)$. Otherwise, we would get a contradiction: $1/(2n^2) < \sin(\Theta) = \sin(\pi - \Theta) \leq \pi - \Theta < \pi - (\pi - 1/(2n^2)) = 1/(2n^2)$, where the second to last inequality uses the fact that $\sin(x) \leq x$. We define C' to be the convex hull of C_{i^*}. Since the angle is smaller than π, we have that C' does not intersect x_{i^*}. We now compute the cost difference between C' and C_{i^*}, namely, we define $\delta = C_{i^*} - C' > 0$. We have that the gain δ is at least $\mathrm{dist}(x_{i^*}, r) + \mathrm{dist}(x_{i^*}, \ell) - \mathrm{dist}(r, \ell)$. We claim that if the angle Θ is such that r is outside of the tube, then $\delta > 2\eta/n^7$, and so we have $C' + 2\eta/n^7 < C' + \delta \leq C_{i^*} \leq C^* + 2\eta/n^7$, and therefore $C' < C^*$. This means that C^* is not optimal and so we get a contradiction.

Thus, we aim at showing that $\delta > 2\eta/n^7$. Now, consider the triangle formed by the three points x_{i^*}, ℓ, and r. Let β denote the angle formed by the line segment between the points x_{i^*} and ℓ, along with the line segment between the points ℓ and r. Similarly, let γ denote the angle formed by the line segment between the points x_{i^*} and r, along with the line segment between r and ℓ. Since $\Theta \leq \pi - 1/(2n^2)$, and we have $\Theta + \beta + \gamma = \pi$, we must have $\beta + \gamma = \pi - \Theta \geq \pi - (\pi - 1/(2n^2)) = 1/(2n^2)$. Thus, we know that $\max\{\beta, \gamma\} \geq 1/(4n^2)$.

For ease of notation, let $a = \mathrm{dist}(x_{i^*}, \ell)$, $b = \mathrm{dist}(x_{i^*}, r)$, and $c = \mathrm{dist}(r, \ell)$. Observe that $c = \mathrm{dist}(r, \ell)$ is precisely $\mathrm{dist}(x_{i^*}, \ell)\cos(\beta) + \mathrm{dist}(x_{i^*}, r)\cos(\gamma) = a\cos(\beta) + b\cos(\gamma)$. If $\Theta \leq \pi/2$, then we know $\max\{\beta, \gamma\} \geq \pi/4$, and hence $c = a\cos(\beta) + b\cos(\gamma) \leq \max\{a + b/\sqrt{2}, a/\sqrt{2} + b\}$. This yields $\delta \geq a + b - c \geq a + b - \max\{a + b/\sqrt{2}, a/\sqrt{2} + b\} \geq (1 - 1/\sqrt{2})\min\{a, b\}$. Hence, consider the case when $\Theta > \pi/2$, which implies $0 \leq \beta, \gamma \leq \pi/2$. Now, since we have $\max\{\beta, \gamma\} \geq 1/(4n^2)$, we obtain $c = a\cos(\beta) + b\cos(\gamma) \leq \max\{a + b(1 - 1/(64n^4)), a(1 - 1/(64n^4)) + b\}$ (the inequality follows from $\cos(x) \leq 1 - x^2/4$, which we can apply due to the fact that $\beta, \gamma \leq \pi/2$). This yields $\delta \geq a + b - c \geq \min\{a, b\}/(64n^4)$.

In all, as long as $\min\{\mathrm{dist}(x_{i^*}, r), \mathrm{dist}(x_{i^*}, \ell)\}/(64n^4) > 2\eta/n^7$, we are done. This holds as long as we ensure that the minimum distance of any point to the ray is sufficiently far (by definition of a good ray), as then every input point p has the property that $\mathrm{dist}(x_{i^*}, p)$ is sufficiently large. $\square$

We define the *right tube* as $T^R = T \cap R \cap \mathrm{int}(C_{i^*}) \cap \mathrm{ext}(C_{i^*-1})$. Consider the points of the right tube and order them from their distance to the interval $[x_{i^*-1}, x_{i^*})$. Let $d_1^R, \ldots, d_{|T^R|}^R$ be the points of the right tube in this ordering. Define S_j^R to be the best curve in $\mathrm{int}(C_{i^*}) \cup C_{i^*}$ that has d_j^R on its boundary, contains x_0, and crosses $\mathcal{R}$ in the interval $[x_{i^*-1}, x_{i^*})$.

We have the following lemma.

Lemma 26. *Consider two integers i, j, $i < j$. For any points $d_i^R, d_j^R \in T^R$, we have $S_i^R \subseteq S_j^R \cup \mathrm{int}(S_j^R)$.*

Proof. Assume towards a contradiction that the lemma is not correct. By definition, S_j^R crosses $\mathcal{R}$ in $[x_{i^*-1}, x_{i^*})$; let p^* be the crossing point. By optimality, the segment $\mathcal{L}$ from d_i^R to p^* lies in T^R.

Now, observe that since d_i^R is closer to $[x_{i^*-1}, x_{i^*})$ than d_j^R, the distance from d_i^R to $\mathcal{L}$ is less than $\eta/2$. Thus, including p^* yields a better curve that lies in $C_{i^*} \cup \mathrm{int}(C_{i^*})$, that has d_j^R on its boundary, crosses $\mathcal{R}$ in $[x_{i^*-1}, x_{i^*})$, and contains x_0, a contradiction. $\square$

24

Define S_j^R to be the best curve in $(\text{int}(C_{i^*}) \cap \text{ext}(C_{i^*-1})) \cup C_{i^*-1} \cup C_{i^*}$ starting at d_j^R and crossing $\mathcal{R}$ in $[x_{i-1}, x_i)$.

The lemma below follows immediately by combining Lemma 21 and Lemma 26 with Lemma 23.

Lemma 27. *There exists an index j^R satisfying the property $|V^{\text{int}}(S_{j^R-1})|, |V^{\text{ext}}(S_{j^R})| \leq n/2$.*

We proceed similarly with the left part of the tube. Namely, we define $T^L = T \cap L \cap \text{int}(S_{j^R}) \cap \text{ext}(S_{j^R-1})$. Let $d_1^L, \ldots, d_{|T^L|}^L$ be the points in T^L in order of non-decreasing distances to the interval $[x_{i^*-1}, x_{i^*})$.

We denote by S_j^L the best curve starting at d_j^L and staying in $\text{int}(S_{j^R}) \cap \text{ext}(S_{j^R-1})$. Mimicking the proof of Lemma 26, we have the following.

Lemma 28. *Consider two integers i, j, $i < j$. For any points $d_i^L, d_j^L \in T^L$, we have $S_i^L \subseteq S_j^L \cup \text{int}(S_j^L)$. Moreover, there exists an index j^L such that $|V^{\text{int}}(S_{j^L-1})|, |V^{\text{ext}}(S_{j^L})| \leq n/2$.*

Before concluding the proof, we define $d_{j^L}^L, d_{j^L-1}^L, d_{j^R-1}^R, d_{j^R}^R$ to be the *dominating* points.

Lemma 29. *Consider the best curve C^*. We have that C^* is contained in*

1. $\text{int}(S_{j^L-1}) \cup S_{j^L-1}$, *or*
2. $\text{ext}(S_{j^L}) \cup S_{j^L}$.

Proof. To show the lemma we need to prove that C^* does not intersect $\text{int}(S_{j^L}) \cap \text{ext}(S_{j^L-1})$. Observe that $\text{ext}(C_{i^*}) \cup C_{i^*} \subseteq \text{ext}(S_{j^R}) \cup S_{j^R} \subseteq \text{ext}(S_{j^L}) \cup S_{j^L}$ and $\text{int}(C_{i^*-1}) \cup C_{i^*-1} \subseteq \text{int}(S_{j^R-1}) \cup S_{j^R-1} \subseteq \text{int}(S_{j^L-1}) \cup S_{j^L-1}$. Thus, for C^* to intersect $\text{int}(S_{j^L}) \cap \text{ext}(S_{j^L-1})$ we need that C^* intersects $\text{ext}(C_{i^*-1}) \cap \text{int}(C_{i^*})$ and $\text{ext}(S_{j^R-1}) \cap \text{int}(S_{j^R})$. Hence, by Lemma 22, for C^* to intersect $\text{ext}(C_{i^*-1}) \cap \text{int}(C_{i^*})$, we need that C^* lies in $\text{ext}(C_{i^*-1}) \cap \text{int}(C_{i^*}) \cup C_{i^*} \cup C_{i^*-1}$. Applying Lemma 22 and the definition of S_{j^R-1}, S_{j^R}, for C^* to intersect $\text{int}(S_{j^R}) \cap \text{ext}(S_{j^R-1})$, we have that C^* lies in $\text{int}(S_{j^R}) \cap \text{ext}(S_{j^R-1}) \cup S_{j^R} \cup S_{j^R-1}$.

Finally, if C^* intersects $\text{int}(S_{j^L}) \cap \text{ext}(S_{j^L-1})$, then Lemma 25 applies and we have that C^* has to intersect a point in the tube. Now observe that the only points in the tube in $\text{int}(S_{j^R}) \cup S_{j^R} \cup \text{ext}(S_{j^R-1}) \cup S_{j^R-1}$ are the dominating points. Hence, C^* must be one of $S_{j^R}, S_{j^R-1}, S_{j^L}, S_{j^L-1}$, and the lemma follows. $\square$

Proof of Lemma 11. We can apply the algorithm defined in Lemma 20 to obtain a good ray $\mathcal{R}$ with origin p.

We can also apply a binary search on the landmarks to find i^*. By Lemma 5, the number of landmarks is bounded by $O(n^9)$ – observe that the set of landmarks does not have to be computed explicitly. In particular, we apply a binary search procedure that makes $O(\log n)$ calls to the algorithm of Section 2.6 (Lemma 12) to find i^*.

We compute $d_1^R, \ldots, d_{|T^R|}^R$ and apply a binary search procedure making $O(\log n)$ calls to the algorithm to find j^R. Proceed similarly to find $d_{j^L}^L, d_{j^L-1}^L$, and proceed recursively on $S_{j^L}^L \cup \text{ext}(S_{j^L}^L)$ and $S_{j^L-1}^L \cup \text{int}(S_{j^L-1}^L)$.

By Lemma 27, the depth of the recursion is at most $O(\log n)$ and so, since each vertex appears in at most 2 regions of the plane at each recursive call, each vertex contributes to the running time of the algorithm of Section 2.6 (Lemma 12) at most $O(\log n)$ times. It follows that the overall complexity is at most $O(n \cdot \log^3 n)$. The correctness follows from Lemma 29. $\square$

2.8 Proof of Lemma 9

Given a weighted set of points P, where for each $p \in P$ we have a weight w_p, an *approximate centerpoint* c is a point in the plane such that, for an arbitrary line containing the point, each of the open halfplanes bounded by the line contains a weight of at least $\frac{\sum_{p \in P} w_p}{5}$. Given a tube T (i.e., quadrilateral), let $L_{\max}$ denote its perimeter, and let $L_{\min} \geq \frac{999}{1000} L_{\max}$ denote the assumed minimum perimeter of an optimal cluster union within T. Consider the optimal clusters that are completely contained within the tube, which we denote by $T_1, \ldots, T_m$. For each such cluster T_i, we choose an arbitrary point $p_i \in T_i$, and assign it a weight of $w_{p_i} := h(T_i) + \eta$. Let $W := \sum_{i=1}^m w_{p_i}$. Moreover, for any set of points $S \subseteq P$, we let $w(S) := \sum_{p \in S} w_p$. Since we assume the existence of an optimal cluster union, we assume that $W \geq L_{\min} + \eta$.

In the following, we seek to find an approximate centerpoint efficiently.

Lemma 30. *Let ℓ be a line such that the total weight of points on the line is at least $\frac{L_{\min} + \eta}{100}$. Then there exists a point p^* on ℓ that belongs to an optimal cluster union that we can find in $O(n \log n)$ time.*

Proof. We let W_ℓ denote the total weight of points that lie on the line ℓ, so that $W_\ell \geq \frac{L_{\min} + \eta}{100}$. We first sort the points on the line by ascending x-coordinate. Then, we can find a weighted point p^* on the line in this sorted ordering such that the total weight of points (including the point p^*) that appear earlier in the sorted ordering is at least $\frac{W_\ell}{2}$, and similarly the total weight of points appearing later in the sorted ordering is at least $\frac{W_\ell}{2}$. This can essentially be done by computing the prefix sum of the sorted points, and then performing a binary search. In all, this process takes $O(n \log n)$ time.

We claim that p^* belongs to an optimal cluster union. In particular, suppose towards a contradiction that this is not the case. For ease of notation, let P_1 be the points appearing earlier than (and including) p^* in the ordering, and similarly define P_2 to be those that appear later (including p^*). Then it must be the case that either all of P_1 does not lie in an optimal cluster union, or all of P_2 does not lie in an optimal cluster union. Otherwise, if there is some point $p_1 \in P_1$ in an optimal cluster union and some point $p_2 \in P_2$ in an optimal cluster union, then by convexity the point p^* must also belong to an optimal cluster union. Hence, the total weight of points outside of an optimal cluster union must be at least $\frac{W_\ell}{2} \geq \frac{L_{\min} + \eta}{200}$. Now, an optimal cluster union has weight at least $L_{\min} + \eta$, and hence the total cost of the optimal partition in the tube T is at least $L_{\min} + \eta + \frac{W_\ell}{2} \geq (L_{\min} + \eta)(1 + \frac{1}{200})$. Since we have that $L_{\min} \geq \frac{999}{1000} L_{\max}$, we have that this quantity is at least $(1 + \frac{1}{200})(\frac{999}{1000} L_{\max} + \eta) > L_{\max} + \eta$. This yields a contradiction however, since the supposed optimal partition is not optimal. $\square$

In order to find an approximate centerpoint, we make use of the notion of weighted ham-sandwich cuts as defined in [5]. It is helpful to define the following notation, as developed in [5]. For a line ℓ, we denote by ℓ^+ and ℓ^- the two open halfplanes bounded by ℓ. We say that a line ℓ *bisects* a weighted set of points S if $|w(S \cap \ell^+) - w(S \cap \ell^-)| \leq |w(S \cap \ell)|$ (note that, in our setting, $w(S \cap \ell)$ is always non-negative, and hence $|w(S \cap \ell)| = w(S \cap \ell)$).

Definition 8 (Weighted Ham-Sandwich Cut [5])**.** Let A, B be two sets of weighted points, where for each point $p \in A \cup B$, we are given a weight w_p. A *weighted ham-sandwich cut* for A and B is a line that bisects both weighted sets A and B simultaneously.

We make use of the following result.

Lemma 31 (Theorem 1, [5]). *Given two sets $A, B \subseteq \mathbb{R}^2$ of weighted points, with $|A| + |B| = n$, a weighted ham-sandwich cut can be computed in $O(n \log n)$ time.*

Using Lemma 31, we show how to obtain a point in an optimal cluster union.

Lemma 32. *Consider a tube T. There exists an algorithm running in time $O(n \log n)$ that finds a point belonging to an optimal cluster union contained in T, assuming it exists.*

Proof. Consider a vertical line ℓ_v defined by $x = u$ cutting through the tube T with the property that the total weight of points in T with an x-coordinate of at most u is at least $\frac{W}{2}$, and the total weight of points in T with an x-coordinate of at least u is at least $\frac{W}{2}$. Note that such a line can be found in $O(n \log n)$ time by sorting the weighted points in the tube in ascending order of their x-coordinate, computing the prefix sum of the ordered points, and performing a binary search. By Lemma 30, we can assume that the total weight of points on any such line ℓ_v that belong to the tube is at most $\frac{L_{\min}+\eta}{100} \leq \frac{W}{100}$ (as otherwise, we have already found a point in an optimal cluster union).

For convenience, let P_1 be the set of weighted points with an x-coordinate strictly less than u, and let P_2 be the set of weighted points with an x-coordinate strictly more than u. We can apply Lemma 31 to the two sets P_1, P_2 to get a weighted ham-sandwich cut in time $O(n \log n)$ (note that $|P_1| + |P_2| \leq n$). We denote by ℓ_s the weighted ham-sandwich cut. Observe that if the total weight of points on the line ℓ_s is at least $\frac{L_{\min}+\eta}{100}$, then by Lemma 30, we are done since we can find a point in an optimal cluster union lying on ℓ_s in $O(n \log n)$ time. Note that ℓ_s and ℓ_v intersect, and hence partition the plane into four quadrants. We claim that their point of intersection, denoted by p^*, is an approximate centerpoint.

From now on we assume that the total weight of points lying on ℓ_v is at most $\frac{W}{100}$, and similarly the total weight of points on ℓ_s is at most $\frac{L_{\min}+\eta}{100} \leq \frac{W}{100}$. Consider any line that contains the point p^*, and consider the two open halfplanes induced by such a line. We argue that each such open halfplane has weight at least $\frac{W}{5}$, which gives the lemma.

To this end, observe that $w(P_1), w(P_2) \geq \frac{W}{2} - \frac{W}{100}$. This holds since ℓ_v was chosen to ensure that the total weight lying to the left (and including) u is at least $\frac{W}{2}$. Since the weight on line ℓ_v is at most $\frac{W}{100}$, the total weight of points with an x-coordinate strictly less than u (which is precisely $w(P_1)$) must be at least $\frac{W}{2} - \frac{W}{100}$. A symmetric argument holds for $w(P_2)$.

Since ℓ_s is a weighted ham-sandwich cut, we have $|w(P_1 \cap \ell_s^+) - w(P_1 \cap \ell_s^-)| \leq w(P_1 \cap \ell_s) \leq \frac{W}{100}$, and similarly for the set P_2. Consider the open quadrant to the left of ℓ_v and above ℓ_s given by $\ell_s^+ \cap \ell_v^-$. The total weight of points in this open quadrant is given by $w(P_1 \cap \ell_s^+)$, which we know is at least $w(P_1 \cap \ell_s^-) - \frac{W}{100}$. On the other hand, we also know $w(P_1 \cap \ell_s^-) + w(P_1 \cap \ell_s^+) \geq \frac{W}{2} - \frac{W}{100}$. Hence, we get $w(P_1 \cap \ell_s^+) \geq w(P_1 \cap \ell_s^-) - \frac{W}{100} \geq \frac{W}{2} - \frac{W}{100} - w(P_1 \cap \ell_s^+)$, implying that $w(P_1 \cap \ell_s^+) \geq \frac{W}{4} - \frac{W}{100} \geq \frac{W}{5}$. We can symmetrically show that each of the four open quadrants contains a weight of at least $\frac{W}{5}$.

Finally, the two open halfplanes induced by an arbitrary line containing p^* must fully contain an open quadrant that lies to the left of ℓ_v, and the other open halfplane must fully contain an open quadrant that lies to the right of ℓ_v. Hence, the weight of each such open halfplane is at least $\frac{W}{5}$ and p^* is an approximate centerpoint.

It is now enough to show that the approximate centerpoint p^* lies in an optimal cluster union. As an optimal cluster union is convex, for any point p that is outside of an optimal cluster union, we can draw a line ℓ_p that does not intersect it. As the weight outside of an optimal cluster union is smaller than $\frac{W}{5}$, on one side of ℓ_p we have clusters of weight smaller than $\frac{W}{5}$ and p is not

an approximate centerpoint. Therefore, all approximate centerpoints are contained in an optimal cluster union. □

We are now able to prove Lemma 9.

Proof of Lemma 9. Assume without loss of generality that we are in case (i), and Γ_1 is at least as high as Γ_2. As for $i \in \{1, 2\}$, $\mathcal{C}$ restricted to the points of $A_{\pi \setminus \Gamma_i}$ is a maximal optimal partition of $A_{\pi \setminus \Gamma_i}$, then for an optimal cluster union B for $\mathcal{C}$, $H(B)$ has to intersect both Γ_1 and Γ_2.

Consider a vertical line segment I constructed as follows. The horizontal distance of I to Γ_1 is equal to the horizontal distance of I to Γ_2, the top endpoint of I lies on a line ℓ_t connecting the top-right vertex of Γ_1 with the top-right vertex of Γ_2, and bottom endpoint of I lies on a line ℓ_b connecting the bottom-left vertex of Γ_1 with the bottom-left vertex of Γ_2. Then, as $H(B)$ intersects both Γ_1 and Γ_2, and $H(B)$ is convex, it must intersect I. Let L be the side length of the cells Γ_1 and Γ_2. Then, as the horizontal distance between Γ_1 and Γ_2 is at least as large as the vertical distance, the length of I is at most $2L$. Similarly, we place vertical intervals I_ℓ and I_r as follows. I_ℓ contains the left edge of Γ_1, has its top endpoint at the line connecting the bottom-left corner of Γ_2 and the top-right corner of Γ_1, and its bottom endpoint at the line connecting the bottom-left corner of Γ_2 and the bottom-left corner of Γ_1. I_r contains the right edge of Γ_2, has its top endpoint at the line connecting the top-right corner of Γ_1 and the top-right corner of Γ_2, and its bottom endpoint at the line connecting the top-right corner of Γ_1 and the bottom-left corner of Γ_2. The lengths of I_ℓ and I_r are upper bounded by $4L$.

Let $\alpha = 100$ be a constant. We place 2α equidistant points $p_1, \ldots, p_\alpha$ on I, the first one at the top endpoint and the last one at the bottom endpoint of I. Similarly, we place 4α equidistant points at I_ℓ and I_r, denoted by p_j^ℓ and p_j^r, respectively. Then, the distance between two consecutive points at each of the segments is at most L/α.

We will show that if an optimal cluster union B exists, then either $H(B)$ contains in its interior at least one of the points p_i, or it is contained in a long and thin quadrilateral spanned by some four points $p_j^\ell, p_{j'}^\ell, p_k^r, p_{k'}^r$, where $|p_j^\ell p_{j'}^\ell|, |p_k^r p_{k'}^r| \le 8L/\alpha$.

Assume that an optimal cluster union B exists, and $H(B)$ does not contain any point p_i. Then, $H(B) \cap I$ lies between some two points p_i, p_{i+1}. As $H(B)$ is convex, contains a point of both Γ_1 and Γ_2, and the horizontal distance from Γ_1 and Γ_2 to I is at most $L/2$, the intersection of any vertical line with $H(B)$ is a line segment of width at most $3|p_i p_{i+1}| \le 3L/\alpha$. Consider a line segment J connecting a leftmost point of $H(B)$ with a rightmost point of $H(B)$. Due to convexity of $H(B)$, this line segment is contained in $H(B)$. Therefore, $H(B)$ is contained in a tube around J, of height $6L/\alpha$. We can extend this tube to a tube of height at most $8L/\alpha$, with endpoints at the points of I_ℓ and I_r.

We start by running the algorithm from Lemma 11 for finding an optimal cluster union for $(\mathcal{C}, p_i)$ for $i \in \{1, \ldots, \alpha\}$. If an optimal cluster union exists and contains one of the points p_i, it will be found by this procedure. Then, we consider the set of $O(\alpha^2)$ tubes $\mathcal{T}$, each tube defined by a pair of points p_j^ℓ, p_k^r, and of height at most $8L/\alpha$. For each such tube $T \in \mathcal{T}$, we will find an optimal cluster union contained in T, if such a cluster exists. For this, we construct α equidistant vertical lines within Γ_1, and also α equidistant vertical lines within Γ_2, and for each such pair of lines we construct a sub-tube $T' \subseteq T$, by cutting of the corresponding left and right parts of T. Let $\mathcal{T}'$ be the resulting collection of $O(\alpha^4)$ sub-tubes. For each $T' \in \mathcal{T}'$, we will want to find an optimal cluster union of A_π, assuming that the cluster is contained in T' and not contained in any shorter tube.

Consider a tube $T' \in \mathcal{T}'$, and assume that an optimal cluster union of A_π is contained in T' and not contained in any shorter tube. Let $p(T')$ be the perimeter of T'. Clearly, the perimeter of an optimal cluster union is at most $p(T')$. Also, as an optimal cluster union stretches nearly all the way from the left of the tube to the right of the tube, and the height of T is at most $8L/\alpha$, we have that the perimeter of an optimal cluster union is at least $p(T') - 20L/\alpha$. Additionally, we know that $p(T') \geq 2\alpha$. Therefore, by using the reasoning from Lemma 32, we can find a point in the center of an optimal cluster union. Then, it is enough to run the algorithm from Lemma 11 for this point.

This algorithm uses a constant number of invocations of the algorithm from Lemma 11, therefore its running time is $O(n \cdot \mathrm{polylog}\, n)$. □

3 The k-Cluster Fencing Problem

In the k-cluster fencing problem, we are given a set of n points S in the plane and an integer k as an input instance I, which we denote by the pair $I := (S, k)$. Our main approach is to use dynamic programming to solve this problem. Our subproblems are based on the notion of boxes (i.e., rectangles in the plane), which contain some subset of the input points that we would like to cluster optimally. Great care is needed in order to ensure that there are only polynomially many subproblems we need to consider. To this end, we prove some structural properties regarding an optimal solution that enable us to reduce the complexity of the subproblems that we solve. We first describe some relevant notation and preprocessing that we do to the input.

3.1 Preliminaries

An *edge* e is defined by its distinct head and tail $p, q \in \mathbb{R}^2$ and is denoted by $e := pq$. A loop o is defined by its only point $p \in \mathbb{R}^2$ and is denoted by $o := pp$. An *edge set* is defined to be a set of edges or a set containing a single loop. For a convex polygon P, define ∂P to be the boundary of P and $h(P)$ to be the perimeter of P. We use $\mathcal{E}(P)$ to denote the edge set defining ∂P oriented in counterclockwise order. For the special case where P is a single point $p \in \mathbb{R}^2$, $\mathcal{E}(P) := \{pp\}$.

For a set of geometric objects O, define $H(O)$ to be the convex hull of O. Specifically, for a set S of points in the plane, $H(S)$ is the convex hull of S. With some abuse of notation, we define $\partial S := \partial H(S)$, $h(S) := h(H(S))$, and $\mathcal{E}(S) := \mathcal{E}(H(S))$. For the special case where S is a single point p, define $\mathcal{E}(\{p\}) := \{pp\}$. We denote by $\mathcal{G}(S)$ the set of the $O(n^2)$ oriented segments defined by any ordered pair of points in S.

We define a *k-clustering* of S as a partition $\mathcal{C} := \{S_1, \ldots, S_k\}$ of S into k subsets or *clusters* $S_1, \ldots, S_k \subseteq S$. Let $\mathrm{Part}(S)$ be the set of all possible clusterings of a set S. Let $\Phi(\mathcal{C}) := \sum_{i=1}^{k} h(S_i)$ be the cost of the clustering $\mathcal{C}$. An *optimal k-clustering* is a k-clustering $\mathcal{C}_k^{\mathrm{OPT}}(S)$ such that $\Phi(\mathcal{C}_k^{\mathrm{OPT}}(S)) \leq \Phi(\mathcal{C})$ for any k-clustering $\mathcal{C}$. Here, $\mathcal{C}_k^{\mathrm{OPT}}(S)$ denotes an arbitrary optimal k-clustering of S. We slightly abuse notation and refer to the edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ as the edges induced by the convex hulls of the clusters in $\mathcal{C}_k^{\mathrm{OPT}}(S)$. Let $\mathrm{OPT}_k(S) := \Phi(\mathcal{C}_k^{\mathrm{OPT}}(S))$.

A clustering $\mathcal{C} := \{S_1, \ldots, S_k\}$ is called a *disjoint clustering* if the convex hulls of any two clusters $S_i, S_j \in \mathcal{C}$ are disjoint, i.e., $H(S_i) \cap H(S_j) = \emptyset$.

Observation 4. *Given a set of points S in the plane, $\mathcal{C}_k^{\mathrm{OPT}}(S)$ is a disjoint clustering.*

We first ensure that no two points in S have the same x- or y-coordinate. This is possible to do in $O(n^2)$ time by computing the slopes of segments between all pairs of points in S. If one of the

slopes is vertical or horizontal, we apply a slight rotation to the coordinate system to eliminate all the horizontal and vertical slopes without introducing new ones.

Let $p_1, \ldots, p_n$ be the points in S sorted by their x-coordinates $x_1 < \cdots < x_n$. We construct two vertical lines v_i^- and v_i^+ with each x-coordinate x_i, such that v_i^- formally is to the left of p_i and v_i^+ formally is to the right of p_i. Thus, an edge $p_j p_i$ for $j < i$ intersects v_i^- at p_i but does not intersect v_i^+, whereas $p_i p_j$ for $i < j$ intersects v_i^+ at p_i but does not intersect v_i^-. The set of all lines v_i^-, v_i^+ for all $i \in \{1, \ldots, n\}$ are the *vertical main lines*. Between any two consecutive vertical main lines v_i^+, v_{i+1}^-, we define 19999 *vertical help lines* with x-coordinates $x_i + \frac{x_{i+1} - x_i}{20000} \cdot j$ for $j \in \{1, \ldots, 19999\}$. That is, these 19999 vertical help lines induce 20000 intervals between x_i and x_{i+1}, each of which has the same length given by $\frac{x_{i+1} - x_i}{20000}$. In a similar way, we define two horizontal main lines with the y-coordinate of each input point p, one formally below p and the other formally above p. Let $h_1^-, h_1^+, \ldots, h_n^-, h_n^+$ be the horizontal main lines sorted by ascending y-coordinate. We also define 19999 equidistant horizontal help lines between any two consecutive horizontal main lines.

Boxes. Let $\mathcal{B}(S)$ be the set of all rectangles with edges contained in main or help lines. Note that the size of $\mathcal{B}(S)$ is $O(n^4)$. We use S_B as an abbreviation for $B \cap S$ where $B \in \mathcal{B}(S)$. We denote by $l(B)$ and $r(B)$ the left and right vertical edge of B and by $b(B)$ and $t(B)$ the bottom and top edge of B. We denote by $w(B)$ the *width* of B, i.e., the difference in the x-coordinates of $r(B)$ and $l(B)$. Similarly, we denote by $h(B)$ the *height* of B, i.e., the difference in the y-coordinates of $t(B)$ and $b(B)$. We define the *length* of a box $B \in \mathcal{B}(S)$ as $\max\{w(B), h(B)\}$ and denote it by $length(B)$. Consider an arbitrary k-clustering $\mathcal{C} := \{S_1, \ldots, S_k\}$. For a box $B \in \mathcal{B}(S)$, consider for each $S_i \in \mathcal{C}$ the part of the boundary of $H(S_i)$ that is in B. The cost of $\mathcal{C}$ in B is denoted as $\Phi_B(\mathcal{C})$, and we define it to be the total length of these parts for all the clusters $S_1, \ldots, S_k$.

For a box B, let ℓ be a vertical line defined by an x-coordinate lying strictly between the x-coordinates of $l(B)$ and $r(B)$. We say that the vertical line segment $s := \ell \cap B$ is a *vertical separator* of B. A vertical separator of a box B is *good* if it intersects at most two edges in $\mathcal{C}_k^{\text{OPT}}(S)$. We also analogously define a *horizontal separator*, along with the notion of a *good horizontal separator*. We call a box *elementary* if there are no vertical help or main lines that lie strictly in between $l(B)$ and $r(B)$, and no horizontal help or main lines that lie strictly in between $b(B)$ and $t(B)$.

Finally, we define a *vertical strip* of a box B to be a rectangle T contained in B where the bottom edge of T is contained in $b(B)$ and the top edge of T is contained in $t(B)$ (i.e., the top and bottom edges of T lie on the boundary of B). Similarly, a *horizontal strip* of a box B is a rectangle H contained in B where the left edge of H is contained in $l(B)$ and the right edge of H is contained in $r(B)$. For a vertical strip T of a box B, we denote by $w(T)$ the length of the bottom (or top) edge of T, and for a horizontal strip H of a box B, we denote by $h(H)$ the length of the left (or right) edge of H.

We interpret vertical strips of a box B as the portion of B that lies between two consecutive vertical help lines, or possibly between a vertical help line and a vertical main line. We similarly interpret horizontal strips of a box to be the portion of the box between two consecutive horizontal help lines, or possibly between a horizontal help line and a horizontal main line. Note that, with this interpretation, elementary boxes are precisely those that consist of one vertical strip and one horizontal strip.

3.2 Structural Properties

The main structural property we aim to show is that in a subproblem for a box $B \in \mathcal{B}(S)$, there is a good vertical or horizontal separator contained in a main or help line that divides B into two

smaller boxes. In particular, in order to ensure subproblems of low complexity, we aim to maintain the following invariant.

Definition 9. A box $B \in \mathcal{B}(S)$ satisfies the *box invariant* if each of its edges is intersected by at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$.

Hence, we make some useful observations and prove some lemmas that aid us in this goal. The following observation exploits the general position assumption that no two points in S have the same x- or y-coordinate.

Observation 5. *Let $B, B_1, B_2 \in \mathcal{B}(S)$ be boxes such that $B = B_1 \cup B_2$ and B_1 and B_2 have disjoint interiors. Let $\mathcal{C}$ be any clustering of S. Then $\Phi_B(\mathcal{C}) = \Phi_{B_1}(\mathcal{C}) + \Phi_{B_2}(\mathcal{C})$.*

We also have the property that no point lies on the boundary of a box B.

Observation 6. *Due to the way we have defined the main lines, it follows that no box $B \in \mathcal{B}(S)$ has a boundary containing a point from S. Furthermore, if the convex hull C of a cluster of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ (or an edge e of such a convex hull) intersects the boundary of a box $B \in \mathcal{B}(S)$, then C (or the edge e) is not contained in B.*

The following lemma states that, for an optimal solution, the total cost of the portion of its edges that lie in a box B cannot exceed the perimeter of the box.

Lemma 33. *Let $B \in \mathcal{B}(S)$. Then $\Phi_B(\mathcal{C}_k^{\mathrm{OPT}}(S)) \leq 2w(B) + 2h(B)$.*

Proof. Assume towards a contradiction that this is not the case. Let $S_1, \ldots, S_k$ be the clusters of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ with convex hulls intersecting B. Let $P := B \cup \bigcup_{i=1}^{k} H(S_i)$. Now, P has a perimeter strictly smaller than $\sum_{i=1}^{k} h(S_i)$. However, the merged cluster $\bigcup_{i=1}^{k} S_i$ has a convex hull with a perimeter at most as large as P, which is a contradiction. $\square$

The following lemma is a key ingredient in our algorithm, and it is the main structural property we show in this section.

Lemma 34. *If $B \in \mathcal{B}(S)$ is an elementary box, then there are at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersecting B (in particular, B satisfies the box invariant). Otherwise, let $B \in \mathcal{B}(S)$ be any box satisfying the box invariant. Then B has a good vertical or horizontal separator s contained in a main or help line such that s divides B into two strictly smaller boxes B_l and B_r, both of which satisfy the box invariant.*

We prove the following sequence of lemmas, which we later show how to combine to yield a proof of Lemma 34.

Lemma 35. *Consider any box $B \in \mathcal{B}(S)$. Let X be an arbitrary partition of B into vertical strips, and let Y be an arbitrary partition of B into horizontal strips. Moreover, let $A := \{A_1, \ldots, A_m\}, Z = \{Z_1, \ldots, Z_q\}$ be arbitrary nonempty subsets of X, Y, respectively. Then either there is a vertical separator of B contained in some vertical strip $A_i \in A$ or a horizontal separator of B contained in some horizontal strip $Z_j \in Z$ that intersects at most $2\sqrt{2} \frac{w(B)+h(B)}{\sum_{A_i \in A} w(A_i)+\sum_{Z_j \in Z} h(Z_j)}$ edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$.*

Proof. Define t_1, t_2 to be the values such that the edges $l(B)$ and $r(B)$ are contained in the vertical lines $x = x_1$ and $x = x_2$, respectively. Similarly, define y_1, y_2 to be the values such that the edges $b(B)$ and $t(B)$ are contained in the horizontal lines $y = y_1$ and $y = y_2$, respectively. That is,

x_1, x_2, y_1, y_2 define the coordinates of the bounding box B. In addition, let $x_1 := t_1 < t_2 < \cdots < t_{m+1} := x_2$ be the values such that, for each vertical strip A_i, the left and right edges of A_i are contained in the vertical lines $x = t_i$ and $x = t_{i+1}$, respectively. Similarly, let $y_1 := s_1 < s_2 < \cdots < s_{q+1} := y_2$ be the values such that, for each horizontal strip Z_i, the lower and upper edges of Z_i are contained in the horizontal lines $y = s_i$ and $y = s_{i+1}$, respectively.

Assume towards a contradiction that all vertical separators that are contained in some vertical strip $A_i \in A$ and all horizontal separators that are contained in some horizontal strip $Z_j \in Z$ intersect strictly more than $2\sqrt{2}\frac{w(B)+h(B)}{\sum_{A_i \in A} w(A_i) + \sum_{Z_j \in Z} h(Z_j)}$ edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$. Define the function $f(x)$ to be the number of edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ that have at least some portion in B and intersect the vertical separator at x (for each $x_1 \le x \le x_2$). Similarly, define the function $g(y)$ to be the number of edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ that have at least some portion in B and intersect the horizontal separator at y (for each $y_1 \le y \le y_2$).

Consider an edge e (or portion of an edge) of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ that lies in B, and suppose it has length δ. We want to understand its contribution to the sum of integrals $\int_{x_1}^{x_2} f(x)dx + \int_{y_1}^{y_2} g(y)dy$ (e.g., a vertical segment contributes δ to the sum, along with a horizontal segment). In particular, we want to understand the maximum contribution possible by such a line segment (i.e., edge). To this end, consider the right triangle formed by the two endpoints of the line segment (which is the hypotenuse) along with the point defined by the intersection of the vertical line passing through the upper endpoint and the horizontal line passing through the lower endpoint. Let δ_1 denote the length of one leg and δ_2 denote the length of the other leg, and observe that $\delta_1 + \delta_2$ is precisely the contribution of edge e to the sum of integrals. Hence, we wish to maximize $\delta_1 + \delta_2$ subject to the constraint $\delta^2 = \delta_1^2 + \delta_2^2$, which yields $\delta_1 = \delta_2 = \frac{\delta}{\sqrt{2}}$ (obtained when the triangle is a right isosceles triangle). This implies that an edge of length δ contributes at most $\sqrt{2}\delta$ to the sum of integrals.

By Lemma 33, we know that $\Phi_B(\mathcal{C}_k^{\mathrm{OPT}}(S)) \le 2w(B) + 2h(B)$. Hence, we obtain a contradiction as follows:

$$
\begin{aligned}
2w(B) + 2h(B) \ge \Phi_B(\mathcal{C}_k^{\mathrm{OPT}}(S)) &\ge \frac{\int_{x_1}^{x_2} f(x)dx + \int_{y_1}^{y_2} g(y)dy}{\sqrt{2}} \\
&\ge \frac{\sum_{i=1}^{m} \int_{t_i}^{t_{i+1}} f(x)dx + \sum_{j=1}^{q} \int_{s_j}^{s_{j+1}} g(y)dy}{\sqrt{2}} \\
&> \left(2\frac{w(B)+h(B)}{\sum_{A_i \in A} w(A_i) + \sum_{Z_j \in Z} h(Z_j)}\right) \left(\sum_{A_i \in A} w(A_i) + \sum_{Z_j \in Z} h(Z_j)\right) \\
&= 2w(B) + 2h(B).
\end{aligned}
$$

Here, the second inequality follows from the maximum amount that an edge (or portion of an edge) of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ can contribute to the sum of integrals, the third inequality follows since we are integrating over a smaller domain, and the last inequality follows from our assumption that each vertical separator that belongs to some vertical strip intersects many edges (and similarly for such horizontal separators). $\qquad \square$

We first consider boxes that have many small vertical strips and many small horizontal strips (i.e., many main lines and help lines). We argue that such boxes have either a good vertical separator or a good horizontal separator in the following lemma.

Lemma 36. *Let B be any box and consider the vertical strips induced by the vertical help (and main) lines going through B, along with the horizontal strips induced by the horizontal help (and main lines) going through B. Moreover, suppose that for every such vertical strip T we have $w(T) \leq \frac{1}{200}w(B)$ and for all such horizontal strips H we have $h(H) \leq \frac{1}{200}h(B)$. Then there exists a good vertical separator of B that is contained in a vertical help or main line that splits B into two smaller boxes (i.e., the separator does not lie on $l(B)$ or $r(B)$), or there exists a good horizontal separator of B with similar properties.*

Proof. Consider any such box B (i.e., B has many horizontal and vertical lines that are close). We consider each portion of B that lies between two consecutive vertical help lines (or between a vertical help line and a vertical main line) to be a vertical strip of B. We similarly consider the horizontal strips of B induced by the corresponding horizontal help and main lines. Throughout the proof, we assume inductively that each of the four edges of B is intersected by at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ (and hence, there are at most eight such edges in total intersecting the boundary of B).

One of the vertical strips has a left edge that is precisely $l(B)$, and another one of the vertical strips has a right edge that is precisely $r(B)$. In particular, since we aim to split the box B into two smaller boxes with our separator, we discard these two vertical strips. Moreover, we also discard any vertical strip T that has an edge of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersecting the top edge of T or the bottom edge of T. By our inductive assumption, there are at most four such edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ (since at most two such edges intersect $t(B)$, and at most two such edges intersect $b(B)$). Each such edge of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ can result in discarding at most two vertical strips (in case the edge intersects the corner of a vertical strip, in which case two strips are affected). Hence, in all, we discard at most 10 vertical strips. Similar reasoning applies to horizontal strips, resulting in discarding at most 10 horizontal strips. Note that the total width of vertical strips that are not discarded is at least $(1 - \frac{10}{200})w(B) = \frac{19}{20}w(B)$, and similarly the total height of horizontal strips that are not discarded is at least $\frac{19}{20}h(B)$.

By Lemma 35, either there exists a vertical separator of B contained in some vertical strip that is not discarded or a horizontal separator of B contained in some horizontal strip that is not discarded that intersects at most $2\sqrt{2}\frac{w(B)+h(B)}{\frac{19}{20}(w(B)+h(B))} < 3$ edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$. Suppose there is some such remaining (i.e., not discarded) vertical strip T containing a good vertical separator (as the other case is symmetric). Since T was not discarded, its left edge is not $l(B)$ and its right edge is not $r(B)$. Moreover, since there are no edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersecting the top or bottom edge of T, there can be no edge of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ that intersects the left or right edge of T without also intersecting the good vertical separator contained in T. Hence, either the left or right edge of T serves as a good vertical separator of B. $\qquad\square$

We now argue that boxes that consist of many vertical strips (induced by the corresponding vertical help or main lines) and are much wider than they are tall have a good vertical separator. A similar result holds in the horizontal direction.

Lemma 37. *Let B be any box and consider the vertical strips induced by the vertical help (and main) lines going through B, along with the horizontal strips induced by the horizontal help (and main lines) going through B. Moreover, suppose that $w(B) > 4h(B)$ and all such vertical strips T have $w(T) \leq \frac{1}{200}w(B)$. Then there exists a good vertical separator of B that is contained in a vertical help or main line that splits B into two smaller boxes (i.e., the separator does not lie on*

$l(B)$ or $r(B)$). Likewise, if $h(B) > 4w(B)$ and all such horizontal strips H have $h(H) \leq \frac{1}{200}h(B)$, then there exists a good horizontal separator of B with similar properties.

Proof. Consider any such box B, and suppose $w(B) > 4h(B)$, as the other case is proved similarly. We consider each portion of B that lies between two consecutive vertical help lines (or between a vertical help line and a vertical main line) to be a vertical strip of B. Assume that all such vertical strips T satisfy $w(T) \leq \frac{1}{200}w(B)$. Moreover, throughout the proof, we assume inductively that each of the four edges of B is intersected by at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$. We proceed in a manner similar to the proof of Lemma 36.

One of these vertical strips has a left edge that is precisely $l(B)$, and another one of these vertical strips has a right edge that is precisely $r(B)$. In particular, since we aim to split the box B into two smaller boxes with our separator, we discard these two vertical strips. Moreover, we also discard any vertical strip T that has an edge of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersecting the top edge of T or the bottom edge of T. By our inductive assumption, there are at most four such edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ (since at most two such edges intersect $t(B)$, and at most two such edges intersect $b(B)$). Each such edge of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ can result in discarding at most two vertical strips (in case the edge intersects the corner of a vertical strip, in which case two strips are affected). Hence, in all, we discard at most 10 vertical strips. Note that the total width of vertical strips that are not discarded is at least $\frac{19}{20}w(B)$.

Now, suppose towards a contradiction that all vertical separators in the remaining vertical strips intersect at least three edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$. This implies that the total length of such edges is at least $\frac{57}{20}w(B)$. By Lemma 33, we know that $\Phi_B(\mathcal{C}_k^{\mathrm{OPT}}(S)) \leq 2w(B) + 2h(B)$. Hence, we obtain a contradiction:

$$\frac{57}{20}w(B) \leq \Phi_B(\mathcal{C}_k^{\mathrm{OPT}}(S)) \leq 2w(B) + 2h(B) \leq 2w(B) + \frac{w(B)}{2} = \frac{5}{2}w(B).$$

Hence, there is some remaining vertical strip T that contains a good vertical separator. Since T was not discarded, its left edge is not $l(B)$ and its right edge is not $r(B)$. Moreover, since there are no edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersecting the top or bottom edge of T, there can be no edge of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ that intersects the left or right edge of T without also intersecting the good vertical separator contained in T. Hence, either the left or right edge of T serves as a good vertical separator. $\square$

We now consider the case regarding boxes that either have some wide vertical strip and are not much taller than they are wide, or have some tall horizontal strip and are not much wider than they are tall.

Lemma 38. *Let B be any box and consider the vertical strips induced by the vertical help (and main) lines going through B, along with the horizontal strips induced by the horizontal help (and main lines) going through B. Moreover, suppose that $h(B) \leq 4w(B)$ and there exists some vertical strip T with $w(T) > \frac{1}{200}w(B)$. Then the left edge of T has at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersecting it, and the same holds for the right edge of T. Similarly, if $w(B) \leq 4h(B)$ and there exists some horizontal strip H with $w(H) > \frac{1}{200}h(B)$, then each of the bottom and top edges of H has at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersecting them.*

Proof. Consider any such box B, and suppose $h(B) \leq 4w(B)$, as the other case is proved similarly. We consider each portion of B that lies between two consecutive vertical help lines (or between a vertical help line and a vertical main line) to be a vertical strip of B. Assume that there exists some vertical strip T with $w(T) > \frac{1}{200}w(B)$.

Now, consider the two points p_1, p_2 that give rise to the vertical strip T induced by the corresponding vertical help and main lines, and let x_1, x_2 be the x-coordinates of p_1, p_2 (respectively), with $x_1 < x_2$. Observe that $h(B) \le 4w(B) < 800w(T) = 800\frac{x_2 - x_1}{20000} = \frac{(x_2 - x_1)}{25}$, where the inequalities follow from our assumptions in the lemma and the equality follows from the fact that we divide the interval $[x_1, x_2]$ into 20000 segments of equal length (induced by the vertical help lines).

Suppose towards a contradiction that the left or right edge of T has at least three edges of $\mathcal{C}_k^{\text{OPT}}(S)$ intersecting it. Let s denote this vertical edge, and let u denote the x-coordinate induced by the line s. We must have that at least one of x_1, x_2 must be far away from u (note that $x_1 \le u \le x_2$). In particular, either $x_2 \ge u + \frac{x_2 - x_1}{2}$ or $x_1 \le u - \frac{x_2 - x_1}{2}$. We consider the former case as the latter is symmetric, so suppose $x_2 \ge u + \frac{x_2 - x_1}{2}$.

Let e_1, e_2, e_3 denote three consecutive edges of $\tilde{\mathcal{C}}_k^{\text{OPT}}(S)$ that intersect the line segment s, sorted in decreasing order according to the height of the point of intersection with s (e.g., e_1 is above e_2, which is above e_3 at $x = u$). Note that it cannot be the case that all three edges are contained in the boundary of the convex hull of the same cluster, and hence the three edges lie on the boundaries of at least two convex hulls (induced by at least two clusters of $\mathcal{C}_k^{\text{OPT}}(S)$). Without loss of generality, assume that there exists an optimal cluster C such that $H(C)$ contains both e_1 and e_2 (the case when e_2 and e_3 are on the same boundary is symmetric). Moreover, let C' denote the cluster giving rise to the boundary on which e_3 lies.

We first consider the case when the slope of e_1 is strictly more than that of e_2, and the slope of e_2 is strictly more than that of e_3. We imagine extending the line segments e_1, e_2 to the left until they meet at some point $r_{1,2}$ (they must meet at some point), and denote by $\theta_{1,2}$ the angle in radians formed as a result of extension. We can do a similar process for edges e_2, e_3 to obtain the angle $\theta_{2,3}$ (again, extending line segments e_2 and e_3 to the left must result in an intersection), and also for edges e_1, e_3 to obtain an angle $\theta_{1,3}$. Observe that $\theta_{1,3} \le \pi$, implying that either $\theta_{1,2} \le \frac{\pi}{2}$ or $\theta_{2,3} \le \frac{\pi}{2}$.

Consider the scenario where $\theta_{1,2} \le \frac{\pi}{2}$. We transform the solution as follows. First, we split the cluster C into two clusters, where we take all points in C with an x-coordinate of at most u to one cluster C_1, and all points in C with an x-coordinate of at least x_2 to another cluster C_2 (note that there are no points with an x-coordinate in the open interval (u, x_2)). Lastly, we merge the clusters C_1 and C' by taking the union $C_1 \cup C'$. We note that the total number of clusters, after performing the split and merge, remains unchanged. Hence, we need only argue that the cost after splitting and merging does not result in an increase in cost.

To this end, observe that the points in C_1 all lie inside the polygon P_1 obtained by considering the original cluster C, cutting it with the vertical line $x = u$, and taking the left portion (i.e., all points in $H(C)$ that have an x-coordinate of at most u). By a similar argument, the points in C_2 all lie inside the polygon P_2 obtained by considering the original cluster C, cutting it with the vertical line $x = x_2$, and taking the right portion (i.e., all points in $H(C)$ that have an x-coordinate of at least x_2). Now, consider the polygon Q obtained by merging the polygons P_1 and $H(C')$ as follows: we consider a vertical line segment at $x = u$ going from edge e_2 to edge e_3 (note that both edges intersect line s at $x = u$). Observe that Q contains the points in $C_1 \cup C'$ and P_2 contains the points in C_2, and hence $h(C_1 \cup C')$ is at most the perimeter of Q and $h(C_2)$ is at most the perimeter of P_2. Hence, we need only bound the perimeter of Q and P_2.

Observe that the combined perimeter of Q and P_2 is at most $h(C) + h(C')$, plus the lengths of the vertical line segments at $x = u$ and $x = x_2$ going from edge e_1 to e_2, plus twice the length of the vertical line segment at $x = u$ going from edge e_2 to e_3, minus the lengths of the portion of edges

e_1 and e_2 with x-coordinates in between $x = u$ and $x = x_2$. For ease of notation, we denote by $b_{1,2}$ the length of the vertical line segment at $x = x_2$ (going from e_1 to e_2), by ℓ_1 the length of e_1 in the interval $[u, x_2]$, and by ℓ_2 the length of e_2 in the interval $[u, x_2]$. First, the vertical line segments at $x = u$ (which, joined together, go from e_1 to e_3 at $x = u$) contribute at most $2h(B)$ to the perimeter of Q. Now, we upper bound $b_{1,2}$ by considering the triangle formed by the following three points: $r_{1,2}$, the intersection of e_1 with the vertical line $x = x_2$, and the intersection of e_2 with the vertical line $x = x_2$. Let β denote the angle formed at the intersection of e_1 with $x = x_2$, and γ denote the angle formed at the intersection of e_2 with $x = x_2$. We have that $b_{1,2} = h(B) + \ell_1 \cos(\beta) + \ell_2 \cos(\gamma)$. Since $\theta_{1,2} \leq \frac{\pi}{2}$, we know that $\max\{\beta, \gamma\} \geq \frac{\pi}{4}$. This implies $b_{1,2} = h(B) + \ell_1 \cos(\beta) + \ell_2 \cos(\gamma) \leq h(B) + \max\{\ell_1 + \frac{\ell_2}{\sqrt{2}}, \frac{\ell_1}{\sqrt{2}} + \ell_2\}$. Hence, the new solution's cost, given by the sum of the perimeters of Q and P_2, is at most $h(C) + h(C') + 3h(B) + \max\{\ell_1 + \frac{\ell_2}{\sqrt{2}}, \frac{\ell_1}{\sqrt{2}} + \ell_2\} - \ell_1 - \ell_2$. We bound this expression as follows:

$$3h(B) + \max\left\{\ell_1 + \frac{\ell_2}{\sqrt{2}}, \frac{\ell_1}{\sqrt{2}} + \ell_2\right\} - \ell_1 - \ell_2$$

$$\leq 3h(B) - \left(1 - \frac{1}{\sqrt{2}}\right)\min\{\ell_1, \ell_2\} \leq 3h(B) - \left(1 - \frac{1}{\sqrt{2}}\right)(x_2 - u),$$

where the last inequality follows from the fact that both ℓ_1 and ℓ_2 are at least $x_2 - u$. Hence, as long as $3h(B) - (1 - \frac{1}{\sqrt{2}})(x_2 - u) < 0$, we have a contradiction. This holds as $h(B) < \frac{(x_2 - x_1)}{25}$ and $x_2 - u \geq \frac{x_2 - x_1}{2}$, implying $3h(B) - (1 - \frac{1}{\sqrt{2}})(x_2 - u) < 0$.

In the case that $\theta_{1,2} > \frac{\pi}{2}$, then we know $\theta_{2,3} \leq \frac{\pi}{2}$. In this setting, we obtain a new solution by merging the two clusters C and C' to get $C \cup C'$, and argue that this new solution is cheaper. Consider the polygon P obtained by removing the portion of edges e_2 and e_3 in between $x = u$ and $x = x_2$, and then joining C and C' on the left by a vertical line segment going from the point at which e_2 intersects s to the point at which e_3 intersects s (at $x = u$). Similarly, we join C and C' on the right by a vertical line segment between the two points at which edges e_2 and e_3 intersect at $x = x_2$. Now, since P contains all points in $C \cup C'$, we have $h(C \cup C')$ is at most the perimeter of P. Moreover, we have that the perimeter of P is at most $h(C) + h(C')$, plus the lengths of the left and right vertical segments we added, minus ℓ_2 and ℓ_3. The left vertical segment contributes at most $h(B)$ to the perimeter (since the left vertical line segment is contained in s). By a similar argument as earlier, the right vertical segment is at most $h(B) + \max\{\ell_2 + \frac{\ell_3}{\sqrt{2}}, \frac{\ell_2}{\sqrt{2}} + \ell_3\}$ (using the fact that $\theta_{2,3} \leq \frac{\pi}{2}$ in this case). Hence, using similar reasoning as before, the perimeter of P is at most $h(C) + h(C') + 2h(B) + \max\left\{\ell_2 + \frac{\ell_3}{\sqrt{2}}, \frac{\ell_2}{\sqrt{2}} + \ell_3\right\} - \ell_2 - \ell_3$. We bound this expression as follows:

$$2h(B) + \max\left\{\ell_2 + \frac{\ell_3}{\sqrt{2}}, \frac{\ell_2}{\sqrt{2}} + \ell_3\right\} - \ell_2 - \ell_3 \leq 2h(B) - \left(1 - \frac{1}{\sqrt{2}}\right)\min\{\ell_2, \ell_3\}$$

$$\leq 2h(B) - \left(1 - \frac{1}{\sqrt{2}}\right)(x_2 - u).$$

We again have that this quantity is strictly less than zero, a contradiction.

In the case when either the slope of e_1 is at most the slope of e_2, or the slope of e_2 is at most the slope of e_3, the proof is simpler. In particular, in the former case, we can more easily bound the vertical line segment we add along the line $x = x_2$ by $h(B)$ (instead of some function of ℓ_1, ℓ_2). This holds since the gap between edges e_1 and e_2 shrinks when going from $x = u$ to $x = x_2$. The same holds in the latter case when the slope of e_2 is at most the slope of e_3. $\qquad \square$

Finally, as a type of base case, we argue that boxes that are not much taller than they are wide and consist of one vertical strip have at most two edges of $\mathcal{C}_k^{\text{OPT}}(S)$ intersecting the whole box. Likewise, boxes that are not much wider than they are tall and consist of one horizontal strip satisfy the same property.

Lemma 39. *Let B be any box with $h(B) \leq 4w(B)$ such that no vertical help or main line has an x-coordinate strictly in between the x-coordinates induced by $l(B)$ and $r(B)$. Then there are at most two edges of $\mathcal{C}_k^{\text{OPT}}(S)$ intersecting box B. Similarly, if B is any box with $w(B) \leq 4h(B)$ such that no horizontal help or main line has a y-coordinate strictly in between the y-coordinates induced by $b(B)$ and $t(B)$, then there are at most two edges of $\mathcal{C}_k^{\text{OPT}}(S)$ intersecting box B.*

Proof. Consider any such box B with $h(B) \leq 4w(B)$ (as the proof of the other case is symmetric). We consider each portion of B that lies between two consecutive vertical help lines (or between a vertical help line and a vertical main line) to be a vertical strip of B. In particular, the assumption in the lemma regarding vertical help and main lines implies that B consists of exactly one vertical strip T (induced by $l(B)$ and $r(B)$), and hence we have $w(B) = w(T)$. The following proof uses similar ideas as in the proof of Lemma 38.

Now, consider the two points p_1, p_2 that give rise to the vertical strip T, and let x_1, x_2 be the x-coordinates of p_1, p_2 (respectively), with $x_1 < x_2$. Observe that $h(B) \leq 4w(B) = 4w(T) = 4\frac{x_2-x_1}{20000} < \frac{x_2-x_1}{200}$, where the first inequality and the first equality follow from our assumptions in the lemma, and the second equality follows from the fact that we divide the interval $[x_1, x_2]$ into 20000 segments of equal length (induced by the vertical help lines).

Suppose towards a contradiction that there are (at least) three edges of $\mathcal{C}_k^{\text{OPT}}(S)$ intersecting the interior of B. Let u_1, u_2 denote the x-coordinates induced by the line segments $l(B)$ and $r(B)$, respectively. We must have either $x_2 \geq \frac{u_1+u_2}{2} + \frac{x_2-x_1}{2}$ or $x_1 \leq \frac{u_1+u_2}{2} - \frac{x_2-x_1}{2}$. Otherwise, we get a contradiction: $x_2 - x_1 < \frac{u_1+u_2}{2} + \frac{x_2-x_1}{2} - \frac{u_1+u_2}{2} + \frac{x_2-x_1}{2} = x_2 - x_1$. We consider the former case as the latter is symmetric, so suppose $x_2 \geq \frac{u_1+u_2}{2} + \frac{x_2-x_1}{2}$. In the former case, we focus on the vertical line $x = u_1$, namely the line containing $l(B)$ (in the latter case, we focus on the vertical line $x = u_2$, namely the line containing $r(B)$).

Let e_1, e_2, e_3 denote three edges of $\mathcal{C}_k^{\text{OPT}}(S)$ that intersect the interior of box B. Each such edge must also intersect the vertical lines $x = u_1$ and $x = x_2$. We consider three such edges that are consecutive, sorted in decreasing order according to the height of the point of intersection with the line $x = u_1$ (e.g., e_1 is above e_2, which is above e_3 at $x = u_1$). Note that it cannot be the case that all three edges are contained in the boundary of the convex hull of the same cluster, and hence the three edges lie on the boundaries of at least two convex hulls (induced by at least two clusters of $\mathcal{C}_k^{\text{OPT}}(S)$). Without loss of generality, assume that there exists an optimal cluster C such that $H(C)$ contains both e_1 and e_2 (the case when e_2 and e_3 are on the same boundary is symmetric). We also denote by C' the cluster giving rise to the boundary on which e_3 lies.

For ease of notation, we denote by ℓ_1, ℓ_2, ℓ_3 the lengths of e_1, e_2, e_3 in the interval $[u_1, x_2]$, and by g_1, g_2, g_3 the lengths of e_1, e_2, e_3 in an arbitrary interval of length $\frac{x_2-x_1}{20000}$ in $[u_1, x_2]$. We also denote by $a_{1,2}$ the length of the vertical line segment at $x = u_1$ connecting e_1 and e_2, by $a_{2,3}$ the segment at $x = u_1$ connecting e_2 and e_3, and $a_{1,3} = a_{1,2} + a_{2,3}$ (i.e., the length of the vertical line segment at $x = u_1$ going from e_1 to e_3). We define analogous lengths of vertical line segments $b_{1,2}$ and $b_{2,3}$ at $x = x_2$ between edges e_1, e_2 and edges e_2, e_3, respectively. We note that $\ell_i \geq 1000g_i$ for $1 \leq i \leq 3$, since there are at least 1000 disjoint intervals of length $\frac{x_2-x_1}{20000}$ in the interval $[u_1, x_2]$ (using the fact that $x_2 \geq \frac{u_1+u_2}{2} + \frac{x_2-x_1}{2}$), and in each such interval the length of e_i is precisely g_i.

37

We first argue that $a_{i,j} \leq g_i + g_j + h(B)$ for all $1 \leq i < j \leq 3$. For any two such edges e_i, e_j we argue this by considering the following three line segments. We can go along e_i from the intersection of edge e_i with $x = u_1$ to any intersection point of e_i with box B, and then to any intersection point of e_j with box B, and finally go along e_j back to $x = u_1$. The cost of connecting e_i to e_j along $x = u_1$ is at most the lengths of the projection of these three line segments onto $x = u_1$, which sum to at most $g_i + g_j + h(B)$.

Moreover, we show a tighter bound of $a_{i,j} \leq \min\{g_i, g_j\} + h(B)$ for all $1 \leq i < j \leq 3$ satisfying the property that the slope of e_i is strictly more than that of e_j. If the slopes of e_i, e_j are positive and negative, respectively, then the intersections of e_i, e_j with $x = u_1$ must both lie in B, and hence the cost of connecting them via the vertical line segment at $x = u_1$ is $a_{i,j} \leq h(B)$. If the slope of e_i is negative, in which case the slope of e_j is also negative, then the intersection points of both edges with $x = u_1$ must lie above $b(B)$, and hence we can obtain the bound via the following two line segments. We can go from the intersection of edge e_i with $x = u_1$ to any point at which it intersects box B, and then go to the intersection of $b(B)$ with $x = u_1$. The cost of connecting e_i to e_j along $x = u_1$ is at most the lengths of the projection of these two line segments onto $x = u_1$, which sum to at most $g_i + h(B)$ (note that $g_i = \min\{g_i, g_j\}$, since both slopes are negative and the slope of e_i is strictly more than that of e_j). A symmetric argument shows that, in the case that e_j is positive (in which case the slope of e_i is also positive), we have $a_{i,j} \leq g_j + h(B)$ (we symmetrically have $g_j = \min\{g_i, g_j\}$).

We first consider the case when the slope of e_1 is strictly more than that of e_2, and the slope of e_2 is strictly more than that of e_3. This implies $a_{1,2} \leq \min\{g_1, g_2, g_3\} + h(B)$ and $a_{2,3} \leq \min\{g_1, g_2, g_3\} + h(B)$. In particular, we know $a_{1,2} \leq \min\{g_1, g_2\} + h(B)$, and we also know $a_{1,2} \leq a_{1,3} \leq \min\{g_1, g_3\} + h(B)$, implying $a_{1,2} \leq \min\{g_1, g_2, g_3\} + h(B)$. Symmetrically, we have $a_{2,3} \leq \min\{g_1, g_2, g_3\} + h(B)$.

Now, we imagine extending the line segments e_1, e_2 to the left until they meet at some point $r_{1,2}$ (they must meet at some point), and denote by $\theta_{1,2}$ the angle in radians formed as a result of extension. We can do a similar process for edges e_2, e_3 to obtain the angle $\theta_{2,3}$ (again, extending line segments e_2 and e_3 to the left must result in an intersection), and also for edges e_1, e_3 to obtain an angle $\theta_{1,3}$. Observe that $\theta_{1,3} \leq \pi$, implying that either $\theta_{1,2} \leq \frac{\pi}{2}$ or $\theta_{2,3} \leq \frac{\pi}{2}$.

Consider the scenario where $\theta_{1,2} \leq \frac{\pi}{2}$. We transform the solution as follows. First, we split the cluster C into two clusters, where we take all points in C with an x-coordinate of at most u_1 to one cluster C_1, and all points in C with an x-coordinate of at least x_2 to another cluster C_2 (note that there are no points with an x-coordinate in the open interval (u_1, x_2)). Lastly, we merge the clusters C_1 and C' by taking the union $C_1 \cup C'$. We note that the total number of clusters, after performing the split and merge, remains unchanged. Hence, we need only argue that the cost after splitting and merging does not result in an increase in cost.

To this end, observe that the points in C_1 all lie inside the polygon P_1 obtained by considering the original cluster C, cutting it with the vertical line $x = u_1$, and taking the left portion (i.e., all points in $H(C)$ that have an x-coordinate of at most u_1). By a similar argument, the points in C_2 all lie inside the polygon P_2 obtained by considering the original cluster C, cutting it with the vertical line $x = x_2$, and taking the right portion (i.e., all points in $H(C)$ that have an x-coordinate of at least x_2). Now, consider the polygon Q obtained by merging the polygons P_1 and $H(C')$ via a vertical line segment at $x = u_1$ going from edge e_2 to edge e_3. Observe that Q contains the points in $C_1 \cup C'$ and P_2 contains the points in C_2, and hence $h(C_1 \cup C')$ is at most the perimeter of Q and $h(C_2)$ is at most the perimeter of P_2. Moreover, the combined perimeters of Q and P_2 is at

most $h(C) + h(C') + a_{1,2} + 2a_{2,3} + b_{1,2} - \ell_1 - \ell_2$.

Now, we upper bound $b_{1,2}$ by considering the triangle formed by the following three points: $r_{1,2}$, the intersection of e_1 with the vertical line $x = x_2$, and the intersection of e_2 with the vertical line $x = x_2$. Let β denote the angle formed at the intersection of e_1 with $x = x_2$, and γ denote the angle formed at the intersection of e_2 with $x = x_2$. We have that $b_{1,2} = a_{1,2} + \ell_1 \cos(\beta) + \ell_2 \cos(\gamma)$. Since $\theta_{1,2} \leq \frac{\pi}{2}$, we know that $\max\{\beta, \gamma\} \geq \frac{\pi}{4}$. This implies $b_{1,2} = a_{1,2} + \ell_1 \cos(\beta) + \ell_2 \cos(\gamma) \leq a_{1,2} + \max\{\ell_1 + \frac{\ell_2}{\sqrt{2}}, \frac{\ell_1}{\sqrt{2}} + \ell_2\}$. Hence, we obtain

$$a_{1,2} + 2a_{2,3} + b_{1,2} - \ell_1 - \ell_2 \leq 2(a_{1,2} + a_{2,3}) + \max\left\{\ell_1 + \frac{\ell_2}{\sqrt{2}}, \frac{\ell_1}{\sqrt{2}} + \ell_2\right\} - \ell_1 - \ell_2$$

$$\leq 4(\min\{g_1, g_2\} + h(B)) - \left(1 - \frac{1}{\sqrt{2}}\right)\min\{\ell_1, \ell_2\}$$

$$\leq 4(\min\{g_1, g_2\} + 4w(B)) - 1000\left(1 - \frac{1}{\sqrt{2}}\right)\min\{g_1, g_2\} < 0,$$

where the first inequality follows from substituting our upper bound on $b_{1,2}$, the second inequality follows from substituting for our upper bounds on $a_{1,2}$ and $a_{2,3}$ and simplifying, the third inequality follows by the assumption $h(B) \leq 4w(B)$, along with $\ell_1 \geq 1000g_1$ and $\ell_2 \geq 1000g_2$, and the last inequality follows from the observation that $w(B) \leq \min\{g_1, g_2\}$. Hence, the perimeter sum of Q and P_2 is at most $h(C) + h(C') + a_{1,2} + 2a_{2,3} + b_{1,2} - \ell_1 - \ell_2 < h(C) + h(C')$, a contradiction.

In the case that $\theta_{1,2} > \frac{\pi}{2}$, then we know $\theta_{2,3} \leq \frac{\pi}{2}$. In this setting, we obtain a new solution by merging the two clusters C and C' to get $C \cup C'$, and argue that this new solution is cheaper. Consider the polygon P obtained by removing the portion of edges e_2 and e_3 in between $x = u_1$ and $x = x_2$, and then joining C and C' on the left by a vertical line segment going from the point at which e_2 intersects $x = u_1$ to the point at which e_3 intersects $x = u_1$. Similarly, we join C and C' on the right by a vertical line segment between the two points at which edges e_2 and e_3 intersect at $x = x_2$. Now, since P contains all points in $C \cup C'$, we have $h(C \cup C')$ is at most the perimeter of P. Moreover, we have that the perimeter of P is at most $h(C) + h(C') + a_{2,3} + b_{2,3} - \ell_2 - \ell_3$.

By a similar argument as earlier, we have $b_{2,3} \leq a_{2,3} + \max\{\ell_2 + \frac{\ell_3}{\sqrt{2}}, \frac{\ell_2}{\sqrt{2}} + \ell_3\}$ (using the fact that $\theta_{2,3} \leq \frac{\pi}{2}$ in this case). Hence, using similar reasoning as before, we obtain

$$a_{2,3} + b_{2,3} - \ell_2 - \ell_3 \leq 2a_{2,3} + \max\left\{\ell_2 + \frac{\ell_3}{\sqrt{2}}, \frac{\ell_2}{\sqrt{2}} + \ell_3\right\} - \ell_2 - \ell_3$$

$$\leq 2(\min\{g_2, g_3\} + h(B)) - \left(1 - \frac{1}{\sqrt{2}}\right)\min\{\ell_2, \ell_3\}$$

$$\leq 2(\min\{g_2, g_3\} + 4w(B)) - 1000\left(1 - \frac{1}{\sqrt{2}}\right)\min\{g_2, g_3\} < 0,$$

where the first inequality follows from substituting for $b_{2,3}$, the second inequality follows from substituting for $a_{2,3}$ and simplifying, the third inequality follows from the assumption $h(B) \leq 4w(B)$, along with $\ell_2 \geq 1000g_2$ and $\ell_3 \geq 1000g_3$, and the last inequality follows from $w(B) \leq \min\{g_2, g_3\}$. Thus, the perimeter of P is at most $h(C) + h(C') + a_{2,3} + b_{2,3} - \ell_2 - \ell_3 < h(C) + h(C')$, a contradiction.

In the case that the slope of e_2 is at most that of e_3, then we also argue that merging C and C' to get $C \cup C'$ yields a cheaper solution. In particular, in this case, we know that $b_{2,3} \leq a_{2,3}$ (as we go from $x = u_1$ to $x = x_2$, the vertical gap between the two edges shrinks since the slope of e_2

is smaller than the slope of e_3). Moreover, we have $a_{2,3} \leq g_2 + g_3 + h(B)$. By similar reasoning as before, we obtain

$$a_{2,3} + b_{2,3} - \ell_2 - \ell_3 \leq 2a_{2,3} - \ell_2 - \ell_3 \leq 2(g_2 + g_3 + 4w(B)) - \ell_2 - \ell_3$$
$$\leq 12\max\{g_2, g_3\} - 1000g_2 - 1000g_3 < 0,$$

which again yields a contradiction.

In the case that the slope of e_1 is at most that of e_2, but the slope of e_2 is strictly more than that of e_3, we do a similar transformation as previously seen by splitting C into two clusters C_1 and C_2, and then merging C_1 with C' to obtain the cluster $C_1 \cup C'$. As before, we need to argue $a_{1,2} + 2a_{2,3} + b_{1,2} - \ell_1 - \ell_2 < 0$. Note that $b_{1,2} \leq a_{1,2}$, since the vertical gap between e_1 and e_2 shrinks when going from $x = u_1$ to $x = x_2$. Moreover, we have $a_{2,3} \leq \min\{g_2, g_3\} + h(B) \leq g_2 + h(B)$ and $a_{1,2} \leq \min\{g_1, g_2\} + h(B) \leq g_2 + h(B)$. Hence, we have

$$a_{1,2} + 2a_{2,3} + b_{1,2} - \ell_1 - \ell_2 \leq 2(a_{1,2} + a_{2,3}) - \ell_1 - \ell_2 \leq 4(g_2 + h(B)) - \ell_1 - \ell_2$$
$$\leq 4(g_2 + 4w(B)) - 1000g_1 - 1000g_2 \leq 20g_2 - 1000g_1 - 1000g_2 < 0,$$

yielding a contradiction. $\qquad\square$

We are now ready to prove Lemma 34.

Proof of Lemma 34. Given any box $B \in \mathcal{B}(S)$, we consider each portion of B that lies between two consecutive vertical help lines (or between a vertical help line and a vertical main line) to be a vertical strip of B. Similarly, we consider the horizontal strips induced by the horizontal help and main lines in B.

First, we consider the case when B is an elementary box. In this case, B is composed of precisely one vertical strip and one horizontal strip. In particular, we can apply Lemma 39 to get that at most two edges in $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersect the elementary box B (clearly, we must have either $h(B) \leq w(B) \leq 4w(B)$ or $w(B) \leq h(B) \leq 4h(B)$).

Now, suppose B is a box satisfying the box invariant. If B is not an elementary box, then it has at least two vertical strips or at least two horizontal strips. If all vertical strips T satisfy $w(T) \leq \frac{w(B)}{200}$ and all horizontal strips H satisfy $h(H) \leq \frac{h(B)}{200}$, then we can apply Lemma 36 to get that there exists either a good vertical separator of B that splits it into two smaller boxes or a good horizontal separator of B that splits it into two smaller boxes (and is contained in a help or main line). Since the separator is good, at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersect it, and hence the box invariant continues to be satisfied for each of the two smaller boxes.

Hence, suppose there is some vertical strip T satisfying $w(T) > \frac{w(B)}{200}$ or there is some horizontal strip H satisfying $h(H) > \frac{h(B)}{200}$. If the box B is much wider than it is tall and has many small vertical strips (i.e., $w(B) > 4h(B)$ and $w(T) \leq \frac{w(B)}{200}$ for all vertical strips T), then we can apply Lemma 37. We get that there is a good vertical separator (contained in a vertical help or main line) that splits B into two smaller boxes such that the box invariant continues to hold on each of the smaller boxes (since at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersect the vertical separator). Likewise, if the box is much taller than it is wide and has many small horizontal strips (i.e., $h(B) > 4w(B)$ and $h(H) \leq \frac{h(B)}{200}$ for all horizontal strips H), then we can again apply Lemma 37. We get that there is a good horizontal separator (contained in a horizontal help or main line) that splits B into two smaller boxes, each satisfying the box invariant.

The only other case to consider is when either the box is not much taller than it is wide and there is some wide vertical strip T, or the box is not much wider than it is tall and there is some
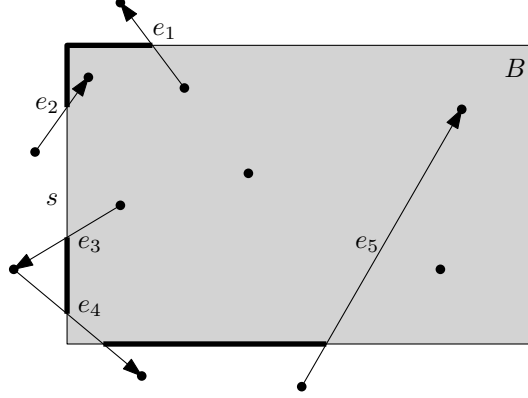
Figure 15: The cut number of the set of edges $E := \{e_1, \ldots, e_5\}$ shown in this figure is 2, i.e. $\Theta_{B,s}(E) = 2$. Note that in this example the border set of E is E, i.e. $\mathcal{M}_B(E) = E$, and it is alternating.

tall horizontal strip H. That is, either $h(B) \leq 4w(B)$ and there is some vertical strip T satisfying $w(T) > \frac{w(B)}{200}$, or $w(B) \leq 4h(B)$ and there is some horizontal strip H satisfying $w(H) > \frac{h(B)}{200}$. Suppose the former is the case, as the proof of the latter is symmetric. Hence, we can apply Lemma 38 to get that either the left edge of T is intersected by at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ and the right edge of T is intersected by at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ (in the latter case, we know the bottom and top edges of H satisfy a similar property).

If either the left edge of T or the right edge of T serves as a good vertical separator of B (i.e., if either edge splits B into two strictly smaller boxes), we are done since we already know such an edge of T is intersected by at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$. Hence, we need only consider the case when the box B consists of one vertical strip (i.e., the left and right edges of T are $l(B)$ and $r(B)$, respectively). Since $h(B) \leq 4w(B)$, we can apply Lemma 39 to get that at most two edges of $\mathcal{C}_k^{\mathrm{OPT}}(S)$ intersect the entire box B. Since box B is not elementary, there is either a vertical help or main line that lies strictly in between $l(B)$ and $r(B)$, or there is a horizontal help or main line that lies strictly in between $b(B)$ and $t(B)$. Regardless, the separators induced by such lines are good.

Tying everything together, for boxes B that are not elementary and satisfy the box invariant, we get that we can always find a good horizontal or vertical separator (contained in a help or main line) that splits B into two strictly smaller boxes, both of which also satisfy the box invariant. $\square$

3.3 Coverings and Signatures

In the rest of this section, we assume we are given a set of points S on the plane, and a number k and we want to solve the k-clustering problem for S. Moreover, we assume the boxes we work with are in $\mathcal{B}(S)$, the clusterings we work with are clusterings in $\mathrm{Part}(S)$, the clusters we work with are subsets of S, and the edges we work with are in $\mathcal{G}(S)$. For brevity, we define $\tilde{\mathcal{C}} := \mathcal{C}_k^{\mathrm{OPT}}(S)$.

Let B be a box and E a set of edges and loops. E is called *interior-disjoint* if the interior of any edge $e \in E$ is not intersected by any other edge in E. Indeed, all sets of edges (and loops) we work with in this section are interior-disjoint. The *border set* of E on B is denoted by $\mathcal{M}_B(E)$ and is defined as the set of all edges in E intersecting both the boundary and the interior of B. An intersection q of an edge $pp' \in \mathcal{M}_B(E)$ and ∂B is called *entering* if pq does not intersect

the interior of B, and is called *exiting* if qp' does not intersect the interior of B. Assuming E is interior-disjoint, all the intersections of the edges in its border set and ∂B are distinct and we can define a cyclic order on these intersections by sorting them in counterclockwise order. A border set is called *alternating* if no two consecutive intersections in this cyclic order are both entering or both exiting. Moreover, every two consecutive exiting and entering intersection define an interval on the boundary of B. For an edge s of ∂B the *cut number* of E on the edge s is denoted by $\Theta_{B,s}(E)$ and is defined as the number of such intervals (if any) intersecting s (see Figure 15).[1] The *border sequence* of E on B is defined as the sequence $e_1, \ldots, e_t$ of all edges in $\mathcal{M}_B(E)$ sorted in counterclockwise order according to their intersection with the boundary of B. Note that some edges might appear twice in this sequence. Moreover, this sequence is not unique as the first edge in this sequence can be chosen arbitrarily.

Let B be a box. A nonempty set of edges E is called a *convex chain* on B of size t, where $t := |E|$, if there exists a sequence of distinct points $p_1, \ldots, p_{t+1}$ such that

- $p_2, \ldots, p_t \in S \cap B$ and $p_1, p_{t+1} \in S \setminus B$,
- $E = \{p_1 p_2, p_2 p_3, \ldots, p_t p_{t+1}\}$,
- $\bigcup_{i=1}^{t} p_i p_{i+1}$ is a simple open curve intersecting the interior of B, and
- for any i, where $1 \leq i < t$, $p_i p_{i+1} p_{i+2}$ is a left turn.

We call this sequence the *vertex sequence* of the convex chain E. The edge $p_1 p_2$ is called the *starting* edge and the edge $p_t p_{t+1}$ is called the *ending* edge of the convex chain. Note that the starting and ending edges of a convex chain are the same when $t = 1$.

Let B be a box. A *cycle* on B is defined to be a nonempty set E with the following properties. We call the set E an *inner* cycle on B of size t if either $t = 1$ and E contains a single loop pp, where $p \in S \cap B$, or if there exists a sequence of distinct points $\sigma := (p_1, \ldots, p_t)$ such that

- $p_1, \ldots, p_t \in S \cap B$,
- $E = \{p_1 p_2, p_2 p_3, \ldots, p_{t-1} p_t, p_t p_1\}$, and therefore $t = |E|$, and
- $p_1, \ldots, p_t$ are the vertices of a convex polygon given in counterclockwise direction.

An inner cycle consisting of a single loop is called a *loop* cycle on B. Moreover, we call the set E a *border* cycle on B if there exists a partition $E = E_1 \cup \cdots \cup E_x$, where $1 \leq x \leq 4$, such that

- each E_i, where $1 \leq i \leq x$, is a convex chain, and
- $e_1, e'_1, \ldots, e_x, e'_x$ is a border sequence of E, where e_i and e'_i are the starting and ending edges of the convex chain E_i.

Note that by the definition of main and helper lines, no point in S can be on the boundary of B. It is not hard to verify that the border set of a border cycle is alternating. The *coverage* of a cycle E is denoted by $\mathcal{R}_B(E)$ and is defined either as the set $\{p\}$ if E is a loop cycle $\{pp\}$, or the intersection of all half planes defined by edges in E and the box B, i.e.,

$$\mathcal{R}_B(E) := B \cap \bigcap_{e \in E} HP(e),$$

where $HP(e)$ is the half-plane defined by the line passing through e located on the left of e. We say a point $b \in B$ is *covered* by E if it is in the coverage of E.

---

[1]To handle the degenerate cases where an edge intersects a corner of a box, we consider s as an open edge by excluding its end points.

For a given box B, an interior-disjoint set E of edges and loops is called a *disjoint covering* or simply a *covering* on B if E satisfies the following properties. The set E is called a *nonempty covering* on B of size t if there exists a partition $E = E_1 \cup \cdots \cup E_t$ such that

- each E_i is a cycle,
- when $1 \le i < j \le t$, the coverages of E_i and E_j are disjoint, i.e., $\mathcal{R}_B(E_i) \cap \mathcal{R}_B(E_j) = \emptyset$,
- $S \cap B \subset \mathcal{R}_B(E_1) \cup \cdots \cup \mathcal{R}_B(E_t)$.

It is not hard to verify that the border set of a covering is alternating. We denote the size of E by $\kappa_B(E)$. By definition, the empty set is a covering on B of size zero if $S \cap B = \emptyset$ and we call it the *empty covering* on B. The following lemma shows that the size of a covering is well defined.

Lemma 40. *For a nonempty covering E defined on an arbitrary box B, the satisfying partition, i.e. the partition satisfying the conditions for a nonempty covering on B, is unique.*

Proof. Note that the coverages of each cycle the satisfying partition is a convex polygon containing all parts of its edges in B. Therefore as all the coverages are disjoint, no two inner cycles or convex chains that belong to different border cycles intersect inside B. Moreover because of the condition on the border sequence of a border cycle, different convex chains of a border cycle cannot intersect in B. Hence, any other satisfying partition (if exists) consists of the same convex chains and inner cycles. Additionally, any two convex chain that are in the same border cycle in the satisfying partition must be in the same border cycle in any other satisfying partition as otherwise the coverages of their border cycles intersect. This shows that any other satisfying partition must have the same inner cycles and border cycles which completes the proof. $\square$

It is not hard to verify that for an edge s of ∂B the *cut number* of E on an edge s of B, i.e. $\Theta_{B,s}(E)$, is equal to the number of the coverages of E intersecting s.

We say a cluster C *intersects* a box B if its convex hull intersects B, and we say it *intersects* the boundary of B or simply intersects ∂B if its convex hull intersects ∂B. The *signature* of C on B or its boundary is denoted by $C \sqcap B$ and $C \sqcap \partial B$ respectively and is defined as the set of all edges of its convex hull intersecting B or the boundary of it, i.e.,

$$C \sqcap B := \{e \in \mathcal{E}(C) \mid e \cap B \ne \emptyset\}, \qquad C \sqcap \partial B := \{e \in \mathcal{E}(C) \mid e \cap \partial B \ne \emptyset\}.$$

With some abuse of notation, the *signature* of a disjoint clustering $\mathcal{C}$ on B or its boundary is denoted by $\mathcal{C} \sqcap B$ and $\mathcal{C} \sqcap \partial B$ respectively and is defined as the set of all edges of convex hulls of its clusters intersecting B or the boundary of it, i.e.,

$$\mathcal{C} \sqcap B := \bigcup_{C \in \mathcal{C}} C \sqcap B, \qquad \mathcal{C} \sqcap \partial B := \bigcup_{C \in \mathcal{C}} C \sqcap \partial B.$$

Note that for any disjoint clustering $\mathcal{C}$, $\mathcal{C} \sqcap B$ is a covering on B (see Figure 16).

3.4 Subproblems

Let B be a box, M be an interior-disjoint set of edges with $|M| \le 8$, $k' \in \{0, \ldots, n\}$, and $\delta \in \{T, F\}$ a boolean value. The tuple $I := \langle B, M, k', \delta \rangle$ is called a *subproblem* on B or simply a subproblem if it satisfies the following properties. The tuple I is called a *special* subproblem if
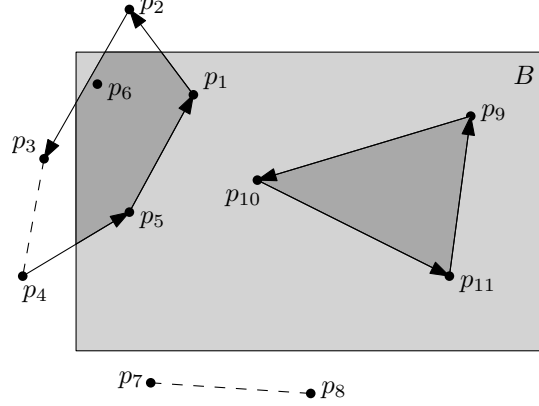
- $\delta = T$,

Figure 16: The signature of the clustering $\mathcal{C} := \{\{p_1, \ldots, p_6\}, \{p_7, p_8\}, \{p_9, p_{10}, p_{11}\}\}$ on B. The edges of the covering $\mathcal{C} \sqcap B$ are solid and the coverage of each cycle is shown in dark grey.

- $M = \emptyset$, and
- $k' = 1$.

The empty covering (on B) is the only *solution* to a special subproblem $\langle B, \emptyset, 1, T \rangle$. The subproblem I is called a *normal* subproblem if

- $\delta = F$,
- M is equal to its border set, i.e. $M = \mathcal{M}_B(M)$, and
- border set of M, i.e. M, is alternating.

A covering E on B is called a *solution* to a normal subproblem $\langle B, M, k', F \rangle$ if $\mathcal{M}_B(E) = M$ and $\kappa_B(E) = k'$. The *cost* of a solution is denoted by $\Phi_B(E)$ and is defined as the sum of the length of $e \cap B$ over all edges $e \in E$. A solution to I is called *optimal* if there is no solution to I with smaller cost. We say a point $b \in B$ is *covered* by a solution E if either b is covered by E or $\delta = T$. Note that the coverage status of any point on the boundary of B is the similar for all solutions.

The *cut number* of a subproblem $I := \langle B, M, k', \delta \rangle$ on an edge s of B is denoted by $\Theta_{B,s}(M, \delta)$ and is defined as

$$\Theta_{B,s}(M, \delta) := \begin{cases} 1, & \delta = T, \\ \Theta_{B,s}(M), & \text{otherwise.} \end{cases}$$

A disjoint clustering $\mathcal{C}$ *respects* a subproblem $I := \langle B, M, k', \delta \rangle$ if either

- $\delta = T$ and B is in the convex hull of a cluster in $\mathcal{C}$, or
- $\delta = F$ and $\mathcal{C} \sqcap B$ is a solution to I.

Observation 7. *Given a box B, the subproblem $I := \langle B, k', \tilde{\mathcal{C}} \sqcap \partial B, \delta \rangle$ is the unique subproblem on B respected by $\tilde{\mathcal{C}}$, where k' is the number of clusters in $\tilde{\mathcal{C}}$ intersecting B and δ is T if B is contained in the convex hull of a cluster in $\tilde{\mathcal{C}}$ and is F otherwise. Moreover, $\tilde{\mathcal{C}} \sqcap B$ is a solution to I.*

Note that the original clustering problem is also a subproblem $I_0 := \langle B_0, \emptyset, k, F \rangle$, where B_0 is defined as the box with edges on $v_1^-, v_n^+, h_1^-, h_n^+$.

Observation 8. *For any solution E to I_0, there is a disjoint clustering $\mathcal{C} \in \text{Part}(S)$ such that $\mathcal{C} \sqcap B_0 = E$. Likewise, for any disjoint clustering $\mathcal{C} \in \text{Part}(S)$, it holds that $\mathcal{C} \sqcap B_0$ is a solution to I_0.*

44

The following property shows that the optimal clustering $\tilde{\mathcal{C}}$ is also optimal for subproblems.

Lemma 41. *Let $I := \langle B, M, k', \delta \rangle$ be a subproblem on a box B respected by $\tilde{\mathcal{C}}$. Then, for any solution E to I, $\Phi_B(\tilde{\mathcal{C}}) \leq \Phi_B(E)$.*

Proof. Assume there exist a solution E to I such that

$$\Phi_B(E) < \Phi_B(\tilde{\mathcal{C}}).$$

It is easy to verify that $E' := ((\tilde{\mathcal{C}} \sqcap B_0) \setminus (\tilde{\mathcal{C}} \sqcap B)) \cup E$ is a solution to I_0 such that $\Phi_{B_0}(E') < \Phi_{B_0}(\tilde{\mathcal{C}})$. As E' consists of k inner cycles on B_0, the set E' is the boundary of a clustering $\mathcal{C}'$ with cost less than $\tilde{\mathcal{C}}$, which is a contradiction. $\square$

3.5 Split and Merge

We say that a box B *splits* into B_l and B_r if there is a separator s that separates B to B_l and B_r. Here, the box B_l is the upper box if s is horizontal and the left box if s is vertical.

Consider two boxes B_1 and B_2 whose boundaries share an edge. We say two subproblems $\langle B_1, M_1, k'_1, \delta_1 \rangle$ and $\langle B_2, M_2, k'_2, \delta_2 \rangle$ are *compatible* if their edges match on their shared boundary, i.e.,

$$\{e \in M_1 | e \cap \partial B_2 \neq \emptyset\} = \{e \in M_2 | e \cap \partial B_1 \neq \emptyset\}.$$

Let B be a box that splits into B_l and B_r by a separator s. We say a subproblem $I := \langle B, M, k', \delta \rangle$ *splits* into $I_l := \langle B_l, M_l, k'_l, \delta_l \rangle$ and $I_r := \langle B_r, M_r, k'_r, \delta_r \rangle$ if I_l and I_r are compatible, $\Theta_{B,s}(M_l, \delta_l) = \Theta_{B,s}(M_r, \delta_r)$, $k' = k'_l + k'_r - \Theta_{B,s}(M_l, \delta_l)$, and $\delta = \delta_l \wedge \delta_r$. Equivalently, we say that I_l and I_r *merge* to I.

Observation 9. *Let B be a box that splits into boxes B_l and B_r by a separator s, and $\mathcal{C}$ be a disjoint clustering. If $\mathcal{C}$ respects all subproblems $I := \langle B, M, k', \delta \rangle$, $I_l := \langle B_l, M_l, k'_l, \delta_l \rangle$, and $I_r := \langle B_r, M_r, k'_r, \delta_r \rangle$, then the subproblem I splits into I_l and I_r.*

The following lemma provides sufficient conditions for a subproblem to have a solution and helps us to find a solution for a subproblem satisfying those conditions.

Lemma 42. *Let $I := \langle B, M, k', \delta \rangle$ be a subproblem that splits into subproblems $I_l := \langle B_l, M_l, k'_l, \delta_l \rangle$ and $I_r := \langle B_r, M_r, k'_r, \delta_r \rangle$. Suppose that E_l and E_r are solutions to I_l and I_r, respectively. Then, $E := E_l \cup E_r$ is a covering on B and a solution to I. Furthermore, $\Phi_B(I) = \Phi_{B_l}(E_l) + \Phi_{B_r}(E_r)$.*

Proof. First assume that none of the subproblems is special. As I_l and I_r are compatible and $\Theta_{B,s}(E_l, \delta_l) = \Theta_{B,s}(E_r, \delta_r)$, each border cycle in E_l intersected by s has a corresponding border cycle in E_r. The set E is a covering on B because each pair of corresponding border cycles intersected by s merge to a cycle on B, and all other cycles from E_l and E_r are also cycles in B. Moreover, as the number of the corresponding pairs of border cycles is $\Theta_{B,s}(E_l, \delta_l)$, the number of cycles in E is $\kappa_B(E) = k'_l + k'_r - \Theta_{B,s}(E_l, \delta_l)$. By the definition of a solution, the covering E is a solution to a subproblem $\langle B, \mathcal{M}_B(E), k'_l + k'_r - \Theta_{B,s}(E_l, \delta_l), F \rangle$. However, as $E = E_l \cup E_r$, the border set $\mathcal{M}_B(E)$ is simply all the edges in M_l and M_r that intersect the boundary of B. As I splits into I_l and I_r, we have $M = \mathcal{M}_B(E)$ and $k' = k'_l + k'_r - \Theta_{B,s}(E_l, \delta_l)$. Moreover, as the edges of E are simply all the edges of E_l and E_r combined, the cost of the covering E satisfies the condition mentioned in the statement of the lemma.

It is easy to verify that in case one or both of the subproblems are special, the claim still holds, which completes the proof of the lemma. $\square$

3.6 Elementary Subproblems

We call a subproblem *elementary* if it is defined on an elementary box. In this section, we show that any elementary subproblem $\langle B, M, k', \delta \rangle$ with at most two edges, i.e., $|M| \leq 2$, has at most one solution. Moreover, if such a subproblem has a solution we can recognize it and find its unique solution in constant time.

An elementary box contains either one point from S or is empty. Therefore, if an elementary box B is empty, all cycles on B are border cycles. If, on the other hand, $B \cap S = \{p\}$, then the only inner cycle on B is the loop cycle $\{pp\}$. Furthermore, as B is defined by two pairs of consecutive main lines, v_i^-, v_i^+ and h_j^-, h_j^+, and its width and height is zero, any edge intersecting B contains p. The only covering on B with an inner cycle is $\{pp\}$, and all other coverings consists of border cycles only. Note that if $\{pp\}$ is a solution to a subproblem, there is no other solution, as in that case $M = \emptyset$ and the loop cycle $\{pp\}$ is the only way p can possibly be covered.

Now, consider a solution without any inner cycles to an elementary subproblem. As an elementary box B contains at most one point, the vertex sequence of a convex chain on B has a size at most three, which means it has at most two edges, and both edges must intersect the boundary of B. Therefore, for a covering E on an elementary box B that does not have any inner cycles, we have $\mathcal{M}_B(E) = E$.

By considering the possible cases, we make the following observation.

Observation 10. *An elementary subproblem $I := \langle B, M, k', \delta \rangle$ has a solution if and only if either*

- $I = \langle B, \emptyset, 1, F \rangle$ *and* $S \cap B = \{p\}$,
- $I = \langle B, \emptyset, 0, F \rangle$ *and* $S \cap B = \emptyset$,
- $I = \langle B, \emptyset, 1, T \rangle$, *or*
- $I = \langle B, M, k', F \rangle$, *where M is a covering on B of size k'.*

Moreover, if a solution exists, it is unique and is equal to $\{pp\}$ in the first case and M otherwise.

Note that when $|M| \leq 2$, we can decide if a set of edges is a solution to an elementary subproblem in constant time.

3.7 Dynamic Programming Algorithm

Definition 10. A subproblem $I := \langle B, M, k', \delta \rangle$ is *compact* if for each edge of B, at most two edges in M intersect it.

Observation 11. *Let B be a box satisfying the box invariant and I be a subproblem on B such that $\tilde{\mathcal{C}}$ respects I. Then I is compact.*

Algorithm 1 solves the minimum perimeter sum problem. For a subproblem $I := \langle B, M, k', \delta \rangle$, the entry Table($I$) either stores a solution to I or the value 'Impossible', which means we have not found any solution to I, though it does not mean a solution to I does not exist. After the execution of the algorithm, Table($\langle B_0, \emptyset, k, F \rangle$) contains the convex hull of an optimal k-clustering. The algorithm iterates over all compact subproblems sorted ordered by the size of the defining box (i.e., the number of elementary boxes contained in the box) and stores a solution for some of them in Table.

- For an elementary subproblem I that has a solution, the algorithm stores the unique solution in Table(I) only if $|M| \leq 2$. The algorithm uses the conditions stated in Observation 10 to

Algorithm 1:

```
 1  for b = 1, ..., (20000n)² do
 2      for all boxes B consisting of b elementary boxes do
 3          for all compact subproblems I := ⟨B, M, k', δ⟩ do
 4              Table(I) := Impossible
 5              if b = 1
 6                  if M = ∅ and k' = 1 and δ = F and B ∩ S = {p}
 7                      Table(I) := {pp}
 8                  else if |M| ≤ 2 and M is a solution to I
 9                      Table(I) := M
10              else
11                  for all compact subproblems I_l and I_r that merge to I do
12                      if Table(I_l) ≠ Impossible and Table(I_r) ≠ Impossible
13                          E := Table(I_l) ∪ Table(I_r)
14                          if Table(I) = Impossible or Φ_B(E) ≤ Φ_B(Table(I))
15                              Table(I) := E
```

find the unique solution to such subproblems. In the proof of Lemma 43, we show that we do not need to find a solution to elementary subproblem with $|M| > 2$.

- For a subproblem I defined on a non-elementary box, the algorithm finds all possible splits of I to two compact subproblems I_l and I_r. For each such split, it combines the coverings (if such exist) in Table(I_l) and Table(I_r) to get a covering E. Then, it stores E in Table(I) only if it is a better solution to I than already stored in Table(I). By Lemma 42, the union of the two solutions to I_l and I_r is a solution to I.

Lemma 43. *Let I be a subproblem on a box B. Consider the value of* Table(I) *as Algorithm 1 terminates. If B satisfies the box invariant and $\tilde{C}$ respects I, the entry* Table(I) *contains a solution to I of cost $\Phi_B(\tilde{C})$. In particular,* Table(I_0) *contains a solution to $I_0 := \langle B_0, \emptyset, k, F \rangle$ of cost $\Phi_{B_0}(\tilde{C})$, and hence, the algorithm solves the k-cluster fencing problem.*

Proof. The proof is by induction on the number b of elementary boxes contained in B. Consider first the case $b = 1$. If I is respected by $\tilde{C}$, Lemma 34 yields that B is intersected by at most two edges of the signature of $\tilde{C}$. By Observation 6, there are no points from S on the boundary of B, which shows $|M| \leq 2$ and therefore I is compact and will be generated. Depending on the subproblem, the algorithm stores either $\{pp\}$ or M as a solution to I in Table(I) only if M is a solution to I. As by Observation 10 the solution to I is unique, it must be $\tilde{C} \sqcap B$. If $|M| > 2$, the algorithm stores 'Impossible' in Table(I) which completes the proof, for the base case.

Suppose the claim holds for boxes consisting of up to $b - 1$ elementary boxes for some $b \geq 2$ and consider a subproblem $I := \langle B, M, k', \delta \rangle$ where B consists of b elementary boxes. Note that if Table(I) is not *Impossible*, Table(I) must contain a value Table(I_l) $\cup$ Table(I_r) for some subproblems I_l and I_r that merge to I and are defined on boxes consisting of less than b elementary boxes. By the induction hypothesis, Table(I_l) and Table(I_r) contain solutions to I_l and I_r and by

47

Lemma 42, $\text{Table}(I) = \text{Table}(I_l) \cup \text{Table}(I_r)$ is a solution to I.

If $\tilde{\mathcal{C}}$ respects I and B satisfies the box invariant, by Observation 11, I is compact. By Lemma 34, B has a separator s contained in a main or helper line and s separates B into two boxes B_l and B_r both satisfying the box invariant. By Observation 7, there are unique subproblems I_l and I_r, on B_l and B_r, such that $\tilde{\mathcal{C}}$ respects them. By Observation 9, subproblem I splits into I_l and I_r. As B_l and B_r consists of less than b elementary boxes, by the induction hypothesis, $\text{Table}(I_l) = \Phi_{B_l}(\tilde{\mathcal{C}})$ and $\text{Table}(I_r) = \Phi_{B_r}(\tilde{\mathcal{C}})$. By Lemma 42,

$$\Phi(\text{Table}(I)) \leq \Phi_{B_l}(\tilde{\mathcal{C}}) + \Phi_{B_r}(\tilde{\mathcal{C}}) = \Phi_B(\tilde{\mathcal{C}}), \tag{2}$$

where the equality comes from Observation 5. Furthermore, by the above mentioned argument, $\text{Table}(I)$ is a solution to I and by Lemma 41,

$$\Phi(\text{Table}(I)) \geq \Phi_B(\tilde{\mathcal{C}}), \tag{3}$$

which completes the proof. $\qquad\square$

3.8   Run-time Analysis

Data structures.   We implement Table as an array, so that we can update and insert values at a given entry $\text{Table}(I)$ in time $O(1)$.

We use a special representation for solutions (set of edges and loops) by making a directed acyclic graph (DAG) to support constant time union operation on solutions (sets). Each solution E to a subproblem $\langle B, M, k', \delta \rangle$ is either the set M, where $|M| \leq 2$, or a set containing a single loop for elementary subproblems, or is the union of two solutions E_l and E_r to two subproblems defined on smaller boxes B_l and B_r for non-elementary subproblems. For elementary subproblems we simply make a node and store if the solution is a loop or not, and if it is a loop we store the loop in the node. For non-elementary subproblems we make a node and we store the (at most two) edges in the solution that intersect the shared edge of B_l and B_r but do not intersect ∂B. We also store two pointers to E_l and E_r in this node; these pointers form a DAG where the number of nodes visible to each node (solution) is linear and therefore it is possible to find the edges and loops of a solution in linear time by simply adding up all the loops and edges inside the visible nodes to M.

For a box B and an edge s of B, we can compute the value of $\Theta_{B,s}(M, \delta)$ for a subproblem $I := \langle B, M, k', \delta \rangle$ in $O(1)$ time as follows. For each subproblem, we store the coverage status of the four corners of the defining box. It is easy to compute those values for elementary subproblems and after merging two solutions we can easily merge the coverage status of their corners as well.[2] Let m_s be the number edges in M intersecting s. As we just work with compact subproblems, $0 \leq m_s \leq 2$. Having the coverage status of the four corners of B, assuming s_1 and s_2 are the endpoints of s, the cut number is given by

$$\Theta_{B,s}(M, \delta) = \begin{cases} 2, & m_s = 2 \text{ and } s_1 \text{ and } s_2 \text{ are covered,} \\ 0, & m_s = 0 \text{ and } s_1 \text{ and } s_2 \text{ are not covered,} \\ 1, & \text{otherwise.} \end{cases}$$

---

[2]To handle degenerate cases where an edge intersects a corner of a box, we can store two coverage values for each corner. One for the coverage of the corner point and a bit more of the boundary in counterclockwise direction and one for the coverage of the corner point and a bit more of the boundary in clockwise direction.

Observation 12. *Given a box B, the number of compact subproblems $I := \langle B, M, k', \delta \rangle$ is at most*

$$(n+1) \cdot \binom{O(n^2)}{8} = O(n^{17}).$$

Here, $n+1$ is a bound on the number k' and the factor $\binom{O(n^2)}{8}$ is a bound on the number of ways the edges of M can be selected for a compact subproblem. There is also one special subproblem on box B and we can consider that case as well.

Observation 13. *Let B be a box that splits into B_l and B_r. The number of ways to split a compact subproblem $I := \langle B, M, k', \delta \rangle$ to two compact subproblems $I_l := \langle B_l, M_l, k'_l, \delta_l \rangle$ and $I_r := \langle B_r, M_r, k'_r, \delta_r \rangle$ is at most*

$$n \cdot \binom{O(n^2)}{2} = O(n^5).$$

Here n is the number of ways we can choose a value for k'_l and $\binom{O(n^2)}{2}$ is the number subsets of at most two edges intersecting $\partial B_l \cap \partial B_r$.

We now describe how Algorithm 1 at line 11 iterates through the $O(n^5)$ pairs of subproblems (I_l, I_r) merging to I. At first, assume δ_l and δ_r have the value F. The edges M specify all edges in M_l and M_r except possibly some edges intersecting the edge $s := \partial M_l \cap \partial M_r$ and not other edges of ∂M_l or ∂M_r. As I_l and I_r are compact and compatible, M_l and M_r can have at most two such edges. Consider now one of these $\binom{O(n^2)}{2}$ choices of edges crossing s. We iterate through each value $k'_l \in \{0, \dots, k'\}$. For a given value of k'_l, the subproblem I_l is completely specified. If I_l does not have any solution in Table(I_l), we can proceed with another choice of I_l. Otherwise, note that I_r is also specified except for the value k'_r. However, we can compute $\Theta_{B_l, s}(E, \delta_l)$, where E is the solution stored as Table(I_l), in constant time. In order for I_l and I_r to merge to I, we now define $k'_r := k' - k'_l + \Theta_{B_l, s}(M_l, \delta_l)$. There are four different choices for the values δ_l and δ_r and it is not hard to consider those cases as well.

Theorem 2. *Algorithm 1 solves the minimum perimeter sum problem in time $O(n^{27})$.*

Proof. Note that B_0, the box with edges on $v_1^-, v_n^+, h_1^-, h_n^+$, satisfies the box invariant. Since $\tilde{\mathcal{C}}$ respects $I_0 := \langle B_0, \emptyset, k, F \rangle$, Lemma 43 gives that as the algorithm terminates, Table(I_0) contains a solution E to I_0 of cost $\Phi_{B_0}(\tilde{\mathcal{C}}) = \Phi(\tilde{\mathcal{C}})$. Moreover, Observation 8 states that there is a clustering $\mathcal{C}$ such that $\mathcal{C} \sqcap B_0 = E$, and hence, $\mathcal{C}$ is an optimal clustering. Furthermore, the clustering $\mathcal{C}$ can easily be computed given E.

The algorithm uses the conditions specified in Observation 10 to verify solutions to elementary subproblems $I := \langle B, M, k', \delta \rangle$ satisfying $|M| \le 2$, which takes constant time and does not affect the asymptotic running time of the algorithm.

There are $O(n^4)$ different boxes in $\mathcal{B}(S)$. By Observation 12, the number of compact subproblems on a B is $O(n^{17})$. For each separator s of B, by Observation 13, the number of ways to split I to I_l and I_l is $O(n^5)$. As we showed in the beginning of this section, merging two solutions can be done in constant time. The cost of each solution can be stored as a single real number and comparing could be assumed to take only $O(1)$ time.[3] The running time of the algorithm thus is

$$O(n^4) \cdot O(n^{17}) \cdot O(n) \cdot O(n^5) \cdot O(1) = O(n^{27}),$$

---

[3] In the RAM model, assuming comparing two solutions can be done in time $T(n)$, the running time is $O(n^{27} \cdot T(n))$.

where $O(n^4)$ is the number of different boxes $B \in \mathcal{B}(S)$, the factor $O(n^{17})$ is the number of subproblems on B, $O(n)$ is the number of separators of B, $O(n^5)$ is the number of ways to split a subproblem into two subproblems for a given separator s, and $O(1)$ is the time needed to compare two solutions. $\qquad\square$

Theorem 3 (The k-cluster fencing problem). *There is a polynomial time algorithm that, given any set S of n points in the plane and an integer k, finds a set of at most k closed curves such that each point in S is enclosed by a curve and the total length of the curves is minimized.*

4 Acknowledgments

References

[1] M. Abrahamsen, M. de Berg, K. Buchin, M. Mehr, and A. D. Mehrabi. Minimum perimeter-sum partitions in the plane. In *Proceedings of the 33rd International Symposium on Computational Geometry*, pages 4:1–4:15, 2017.

[2] P. K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Computing Surveys*, 30(4):412–458, 1998.

[3] E. M. Arkin, S. Khuller, and J. S. B. Mitchell. Geometric knapsack problems. *Algorithmica*, 10(5):399–427, 1993.

[4] B. Behsaz and M. R. Salavatipour. On minimum sum of radii and diameters clustering. *Algorithmica*, 73(1):143–165, 2015.

[5] P. Bose and S. Langerman. Weighted ham-sandwich cuts. In *Japanese Conference on Discrete and Computational Geometry*, pages 48–53, 2004.

[6] V. Capoyleas, G. Rote, and G. Woeginger. Geometric clusterings. *Journal of Algorithms*, 12(2):341–356, 1991.

[7] M. de Berg, M. van Kreveld, M. Overmars, and O. Cheong. *Computational geometry: algorithms and applications (3rd edition)*. Springer-Verlag, 2008.

[8] S. G. Dicke and B. Hubbard. Tree protection standards in construction sites. 2008. The Forest and Wildlife Research Center, Mississippi State University Extension Service, `http://fwrc.msstate.edu/pubs/treeprotection.pdf`.

[9] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics*, 9(1):66–104, 1990.

[10] M. Gibson, G. Kanade, E. Krohn, I. A. Pirwani, and K. Varadarajan. On clustering to minimize the sum of radii. *SIAM Journal on Computing*, 41(1):47–60, 2012.

[11] S. Har-Peled. *Geometric approximation algorithms*, volume 173. American Mathematical Society, 2011.

[12] J. S. B. Mitchell. Private communication, 2017.

[13] J. S. B. Mitchell and E. L. Wynters. Finding optimal bipartitions of points and polygons. In *Proceedings of the 2nd Workshop on Algorithms and Data Structures*, pages 202–213, 1991.

[14] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete & Computational Geometry*, 19(1):1–17, 1998.

[15] C.-K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *Journal of Computer and System Sciences*, 40(1):2–18, 1990.