Minimum perimeter-sum partitions in the plane

Abrahamsen, Mikkel; de Berg, Mark; Buchin, Kevin; Mehr, Mehran; Mehrabi, Ali D.

Published in:
33rd International Symposium on Computational Geometry (SoCG 2017)

DOI:
10.4230/LIPIcs.SoCG.2017.4

Publication date:
2017

Document version
Publisher's PDF, also known as Version of record

Minimum Perimeter-Sum Partitions in the Plane*†

Mikkel Abrahamsen[1], Mark de Berg[2], Kevin Buchin[3], Mehran Mehr[4], and Ali D. Mehrabi[5]

1 **Department of Computer Science, University of Copenhagen, Copenhagen, Denmark**
miab@di.ku.dk

2 **Department of Computer Science, TU Eindhoven, Eindhoven, The Netherlands**
mdberg@win.tue.nl

2 **Department of Computer Science, TU Eindhoven, Eindhoven, The Netherlands**
k.a.buchin@tue.nl

2 **Department of Computer Science, TU Eindhoven, Eindhoven, The Netherlands**
m.mehr@tue.nl

2 **Department of Computer Science, TU Eindhoven, Eindhoven, The Netherlands**
amehrabi@win.tue.nl

Abstract

Let P be a set of n points in the plane. We consider the problem of partitioning P into two subsets P_1 and P_2 such that the sum of the perimeters of $\text{CH}(P_1)$ and $\text{CH}(P_2)$ is minimized, where $\text{CH}(P_i)$ denotes the convex hull of P_i. The problem was first studied by Mitchell and Wynters in 1991 who gave an $O(n^2)$ time algorithm. Despite considerable progress on related problems, no subquadratic time algorithm for this problem was found so far. We present an exact algorithm solving the problem in $O(n \log^4 n)$ time and a $(1+\varepsilon)$-approximation algorithm running in $O(n + 1/\varepsilon^2 \cdot \log^4(1/\varepsilon))$ time.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Computational geometry, clustering, minimum-perimeter partition, convex hull

Digital Object Identifier 10.4230/LIPIcs.SoCG.2017.4

1 Introduction

The clustering problem is to partition a given data set into clusters (that is, subsets) according to some measure of optimality. We are interested in clustering problems where the data set is a set P of points in Euclidean space. Most of these clustering problems fall into one of two categories: problems where the maximum cost of a cluster is given and the goal is to find a clustering consisting of a minimum number of clusters, and problems where the number of clusters is given and the goal is to find a clustering of minimum total cost. In this paper we consider a basic problem of the latter type, where we wish to find a bipartition (P_1, P_2) of a planar point set P. Bipartition problems are not only interesting in their own right, but also because bipartition algorithms can form the basis of hierarchical clustering methods.

* A full version of the paper is available at http://arxiv.org/abs/1703.05549.

† MA is partly supported by Mikkel Thorup's Advanced Grant from the Danish Council for Independent Research under the Sapere Aude research career programme. MdB, KB, MM, and AM are supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 024.002.003, 612.001.207, 022.005025, and 612.001.118 respectively.

33rd International Symposium on Computational Geometry (SoCG 2017).
Editors: Boris Aronov and Matthew J. Katz; Article No. 4; pp. 4:1–4:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

There are many possible variants of the bipartition problem on planar point sets, which differ in how the cost of a clustering is defined. A variant that received a lot of attention is the 2-center problem [8, 11, 12, 15, 20], where the cost of a partition (P_1, P_2) of the given point set P is defined as the maximum of the radii of the smallest enclosing disks of P_1 and P_2. Other cost functions that have been studied include the maximum diameter of the two point sets [3] and the sum of the diameters [14]; see also the survey by Agarwal and Sharir [2] for some more variants.

A natural class of cost function considers the size of the convex hulls $\textsc{ch}(P_1)$ and $\textsc{ch}(P_2)$ of the two subsets, where the size of $\textsc{ch}(P_i)$ can either be defined as the area of $\textsc{ch}(P_i)$ or as the perimeter $\text{per}(P_i)$ of $\textsc{ch}(P_i)$. (The perimeter of $\textsc{ch}(P_i)$ is the length of the boundary $\partial\,\textsc{ch}(P_i)$.) This class of cost functions was already studied in 1991 by Mitchell and Wynters [17]. They studied four problem variants: minimize the sum of the perimeters, the maximum of the perimeters, the sum of the areas, or the maximum of the areas. In three of the four variants the convex hulls $\textsc{ch}(P_1)$ and $\textsc{ch}(P_2)$ in an optimal solution may intersect [17, full version] – only in the *minimum perimeter-sum problem* the optimal bipartition is guaranteed to be a so-called *line partition*, that is, a solution with disjoint convex hulls. For each of the four variants they gave an $O(n^3)$ algorithm that uses $O(n)$ storage and that computes computes an optimal line partition; for all except the minimum area-maximum problem they also gave an $O(n^2)$ algorithm that uses $O(n^2)$ storage. Note that (only) for the minimum perimeter-sum problem the computed solution is an optimal bipartition. Around the same time, the minimum-perimeter sum problem was studied for partitions into k subsets for $k > 2$; for this variant Capoyleas *et al.* [7] presented an algorithm with running time $O(n^{6k})$. Mitchell and Wynters mentioned the improvement of the space requirement of the quadratic-time algorithm as an open problem, and they stated the existence of a subquadratic algorithm for any of the four variants as the most prominent open problem.

Rokne *et al.* [18] made progress on the first question, by presenting an $O(n^2 \log n)$ algorithm that uses only $O(n)$ space for the line-partition version of each of the four problems. Devillers and Katz [10] gave algorithms for the min-max variant of the problem, both for area and perimeter, which run in $O((n+k)\log^2 n)$ time. Here k is a parameter that is only known to be in $O(n^2)$, although Devillers and Katz suspected that k is subquadratic. They also gave linear-time algorithms for these problems when the point set P is in convex position and given in cyclic order. Segal [19] proved an $\Omega(n \log n)$ lower bound for the min-max problems. Very recently, and apparently unaware of some of the earlier work on these problems, Bae *et al.* [4] presented an $O(n^2 \log n)$ time algorithm for the minimum-perimeter-sum problem and an $O(n^4 \log n)$ time algorithm for the minimum-area-sum problem (considering all partitions, not only line partitions). Despite these efforts, the main question is still open: is it possible to obtain a subquadratic algorithm for any of the four bipartition problems based on convex-hull size?

1.1 Our contribution

We answer the question above affirmatively by presenting a subquadratic algorithm for the minimum perimeter-sum bipartition problem in the plane.

As mentioned, an optimal solution (P_1, P_2) to the minimum perimeter-sum bipartition problem must be a line partition. A straightforward algorithm would generate all $\Theta(n^2)$ line partitions and compute the value $\text{per}(P_1)+\text{per}(P_2)$ for each of them. If the latter is done from scratch for each partition, the resulting algorithm runs in $O(n^3 \log n)$ time. The algorithms by Mitchell and Wynters [17] and Rokne *et al.* [18] improve on this by using that the different line bipartitions can be generated in an ordered way, such that subsequent line partitions differ in at most one point. Thus the convex hulls do not have to be recomputed from scratch, but they can be obtained by updating the convex hulls of the previous bipartition. To obtain a subquadratic algorithm a fundamentally new approach is necessary: we need a strategy that generates a subquadratic number of candidate partitions, instead considering all line partitions. We achieve this as follows.

We start by proving that an optimal bipartition (P_1, P_2) has the following property: either there is a set of $O(1)$ canonical orientations such that P_1 can be separated from P_2 by a line with a canonical orientation, or the distance between $\text{CH}(P_1)$ and $\text{CH}(P_2)$ is $\Omega(\min(\text{per}(P_1), \text{per}(P_2))$. There are only $O(1)$ bipartitions of the former type, and finding the best among them is relatively easy. The bipartitions of the second type are much more challenging. We show how to employ a compressed quadtree to generate a collection of $O(n)$ canonical 5-gons – intersections of axis-parallel rectangles and canonical halfplanes – such that the smaller of $\text{CH}(P_1)$ and $\text{CH}(P_2)$ (in a bipartition of the second type) is contained in one of the 5-gons.

It then remains to find the best among the bipartitions of the second type. Even though the number of such bipartitions is linear, we cannot afford to compute their perimeters from scratch. We therefore design a data structure to quickly compute $\text{per}(P \cap Q)$, where Q is a query canonical 5-gon. Brass *et al.* [6] presented such a data structure for the case where Q is an axis-parallel rectangle. Their structure uses $O(n \log^2 n)$ space and has $O(\log^5 n)$ query time; it can be extended to handle canonical 5-gons as queries, at the cost of increasing the space usage to $O(n \log^3 n)$ and the query time to $O(\log^7 n)$. Our data structure improves upon this: it has $O(\log^4 n)$ query time for canonical 5-gons (and $O(\log^3 n)$ for rectangles) while using the same amount of space. Using this data structure to find the best bipartition of the second type we obtain our main result: an exact algorithm for the minimum perimeter-sum bipartition problem that runs in $O(n \log^4 n)$ time. As our model of computation we use the real RAM (with the capability of taking square roots) so that we can compute the exact perimeter of a convex polygon – this is necessary to compare the costs of two competing clusterings. We furthermore make the (standard) assumption that the model of computation allows us to compute a compressed quadtree of n points in $O(n \log n)$ time; see footnote 2 on page 10.

Besides our exact algorithm, we present a linear-time $(1 + \varepsilon)$-approximation algorithm. Its running time is $O(n + T(1/\varepsilon^2)) = O(n + 1/\varepsilon^2 \cdot \log^4(1/\varepsilon))$, where $T(1/\varepsilon^2)$ is the running time of an exact algorithm on an instance of size $1/\varepsilon^2$.

Some arguments are omitted due to limited space. See the full version [1] for the details.

2 The exact algorithm

In this section we present an exact algorithm for the minimum-perimeter-sum partition problem. We first prove a separation property that an optimal solution must satisfy, and then we show how to use this property to develop a fast algorithm.

Let P be the set of n points in the plane for which we want to solve the minimum-perimeter-sum partition problem. An optimal partition (P_1, P_2) of P has the following two basic properties: P_1 and P_2 are non-empty, and the convex hulls $\text{CH}(P_1)$ and $\text{CH}(P_2)$ are disjoint [17, full version]. In the remainder, whenever we talk about a partition of P, we refer to a partition with these two properties.

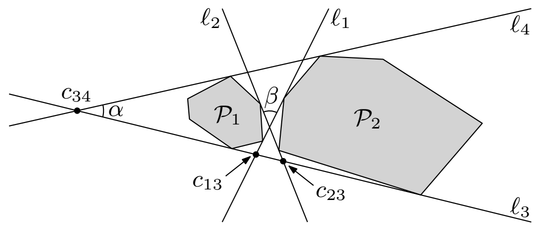

Figure 1 The angles α and β.

2.1 Geometric properties of an optimal partition

Consider a partition (P_1, P_2) of P. Define $\mathcal{P}_1 := \text{CH}(P_1)$ and $\mathcal{P}_2 := \text{CH}(P_2)$ to be the convex hulls of P_1 and P_2, respectively, and let ℓ_1 and ℓ_2 be the two inner common tangents of $\mathcal{P}_1$ and $\mathcal{P}_2$. The lines ℓ_1 and ℓ_2 define four wedges: one containing P_1, one containing P_2, and two empty wedges. We call the opening angle of the empty wedges the *separation angle* of P_1 and P_2. Furthermore, we call the distance between $\mathcal{P}_1$ and $\mathcal{P}_2$ the *separation distance* of P_1 and P_2.

▶ **Theorem 1.** *Let P be a set of n points in the plane, and let (P_1, P_2) be a partition of P that minimizes* $\text{per}(P_1) + \text{per}(P_2)$. *Then the separation angle of P_1 and P_2 is at least $\pi/6$ or the separation distance is at least* $c_{\text{sep}} \cdot \min(\text{per}(P_1), \text{per}(P_2))$, *where* $c_{\text{sep}} := 1/250$.

The remainder of this section is devoted to proving Theorem 1. To this end let (P_1, P_2) be a partition of P that minimizes $\text{per}(P_1) + \text{per}(P_2)$. Let ℓ_3 and ℓ_4 be the outer common tangents of $\mathcal{P}_1$ and $\mathcal{P}_2$. We define α to be the angle between ℓ_3 and ℓ_4. More precisely, if ℓ_3 and ℓ_4 are parallel we define $\alpha := 0$, otherwise we define α as the opening angle of the wedge defined by ℓ_3 and ℓ_4 containing $\mathcal{P}_1$ and $\mathcal{P}_2$. We denote the separation angle of P_1 and P_2 by β; see Fig. 1.

The idea of the proof is as follows. Suppose that the separation distance and the separation angle β are both relatively small. Then the region A in between $\mathcal{P}_1$ and $\mathcal{P}_2$ and bounded from the bottom by ℓ_3 and from the top by ℓ_4 is relatively narrow. But then the left and right parts of ∂A (which are contained in $\partial\mathcal{P}_1$ and $\partial\mathcal{P}_2$) would be longer than the bottom and top parts of ∂A (which are contained in ℓ_3 and ℓ_4), thus contradicting that (P_1, P_2) is an optimal partition. To make this idea precise, we first prove that if the separation angle β is small, then the angle α between ℓ_3 and ℓ_4 must be large. Second, we show that there is a value $f(\alpha)$ such that the distance between $\mathcal{P}_1$ and $\mathcal{P}_2$ is at least $f(\alpha) \cdot \min(\text{per}(P_1), \text{per}(P_2))$. Finally we argue that this implies that if the separation angle is smaller than $\pi/6$, then (to avoid the contradiction mentioned above) the separation distance must be relatively large. Next we present our proof in detail.

Let c_{ij} be the intersection point between ℓ_i and ℓ_j, where $i < j$. If ℓ_3 and ℓ_4 are parallel, we choose c_{34} as a point at infinity on ℓ_3. Assume without loss of generality that neither ℓ_1 nor ℓ_2 separate $\mathcal{P}_1$ from c_{34}, and that ℓ_3 is the outer common tangent such that $\mathcal{P}_1$ and $\mathcal{P}_2$ are to the left of ℓ_3 when traversing ℓ_3 from c_{34} to an intersection point in $\ell_3 \cap \mathcal{P}_1$. Assume furthermore that c_{13} is closer to c_{34} than c_{23}.

For two lines, rays, or segments r_1, r_2, let $\angle(r_1, r_2)$ be the angle we need to rotate r_1 in counterclockwise direction until r_1 and r_2 are parallel. For three points a, b, c, let $\angle(a, b, c) := \angle(ba, bc)$. For $i = 1, 2$ and $j = 1, 2, 3, 4$, let s_{ij} be a point in $P_i \cap \ell_j$. Let $\partial\mathcal{P}_i$ denote the boundary of $\mathcal{P}_i$ and $\text{per}(\mathcal{P}_i)$ the perimeter of $\mathcal{P}_i$. Furthermore, let $\partial\mathcal{P}_i(x, y)$ denote the portion of $\partial\mathcal{P}_i$ from $x \in \partial\mathcal{P}_i$ counterclockwise to $y \in \partial\mathcal{P}_i$, and $\text{length}(\partial\mathcal{P}_i(x, y))$ denote the length of $\partial\mathcal{P}_i(x, y)$.

▶ **Lemma 2.** *We have $\alpha + 3\beta \geqslant \pi$.*

Proof. Since $\mathrm{per}(\mathcal{P}_1) + \mathrm{per}(\mathcal{P}_2)$ is minimum, we know that

$$\mathrm{length}(\partial \mathcal{P}_1(s_{13}, s_{14})) + \mathrm{length}(\partial \mathcal{P}_2(s_{24}, s_{23})) \ \leqslant\ \Psi,$$

where $\Psi := |s_{13}s_{23}| + |s_{14}s_{24}|$. Furthermore, we know that $s_{11}, s_{12} \in \partial \mathcal{P}_1(s_{13}, s_{14})$ and $s_{21}, s_{22} \in \partial \mathcal{P}_1(s_{24}, s_{23})$. We thus have

$$\mathrm{length}(\partial \mathcal{P}_1(s_{13}, s_{14})) + \mathrm{length}(\partial \mathcal{P}_2(s_{24}, s_{23})) \ \geqslant\ \Phi,$$

where $\Phi := |s_{13}s_{11}| + |s_{11}s_{12}| + |s_{12}s_{14}| + |s_{24}s_{21}| + |s_{21}s_{22}| + |s_{22}s_{23}|$. Hence, we must have

$$\Phi \ \leqslant\ \Psi. \tag{1}$$

Now assume that $\alpha+3\beta < \pi$. We will show that this assumption, together with inequality (1), leads to a contradiction, thus proving the lemma. To this end we will argue that if (1) holds, then it must also hold when (i) s_{21} or s_{22} coincides with c_{12}, and (ii) s_{11} or s_{12} coincides with c_{12}. To finish the proof it then suffices to observe that that if (i) and (ii) hold, then $\mathcal{P}_1$ and $\mathcal{P}_2$ touch in c_{12} and so (1) contradicts the triangle inequality.

It remains to argue that if (1) holds, then we can create a situation where (1) holds and (i) and (ii) hold as well. To this end we ignore that the points s_{ij} are specific points in the set P and allow the point s_{ij} to move on the tangent ℓ_j, as long as the movement preserves (1). Moving s_{13} along ℓ_3 away from s_{23} increases Ψ more than it increases Φ, so (1) is preserved. Similarly, we can move s_{14} away from s_{24}, s_{23} away from s_{13}, and s_{24} away from s_{14}.

We first show how to create a situation where (i) holds, and (1) still holds as well. Let $\gamma_{ij} := \angle(\ell_i, \ell_j)$. We consider two cases.

- *Case* (A)*:* $\gamma_{32} < \pi - \beta$.
  Note that $\angle(xs_{23}, \ell_2) \geqslant \gamma_{32}$ for any $x \in s_{22}c_{12}$. However, by moving s_{23} sufficiently far away we can make $\angle(xs_{23}, \ell_2)$ arbitrarily close to γ_{32}, and we can ensure that $\angle(xs_{23}, \ell_2) < \pi - \beta$ for any point $x \in s_{22}c_{12}$. We now let the point x move at unit speed from s_{22} towards c_{12}. To be more precise, let $T := |s_{22}c_{12}|$, let $\mathbf{v}$ be the unit vector with direction from c_{23} to c_{12}, and for any $t \in [0, T]$ define $x(t) := s_{22} + t \cdot \mathbf{v}$. Note that $x(0) = s_{22}$ and $x(T) = c_{12}$.
  Let $a(t) := |x(t)s_{23}|$ and $b(t) := |x(t)s_{21}|$. In the full version [1] we show that
  $$a'(t) = -\cos(\angle(x(t)s_{23}, \ell_2)) \text{ and } b'(t) = \cos(\angle(\ell_2, x(t)s_{21})).$$
  Since $\angle(x(t)s_{23}, \ell_2) < \pi - \beta$ for any value $t \in [0, T]$, we get $a'(t) < -\cos(\pi - \beta)$. Furthermore, we have $\angle(\ell_2, x(t)s_{21}) \geqslant \pi - \beta$ and hence $b'(t) \leqslant \cos(\pi - \beta)$. Therefore, $a'(t) + b'(t) < 0$ for any t and we conclude that $a(T) + b(T) \leqslant a(0) + b(0)$. This is the same as $|s_{21}c_{12}| + |c_{12}s_{23}| \leqslant |s_{21}s_{22}| + |s_{22}s_{23}|$, so (1) still holds when we substitute s_{22} by c_{12}.
- *Case* (B)*:* $\gamma_{32} \geqslant \pi - \beta$.
  Using our assumption $\alpha + 3\beta < \pi$ we get $\gamma_{32} > \alpha + 2\beta$. Note that $\gamma_{14} = \pi - \gamma_{32} + \alpha + \beta$. Hence, $\gamma_{14} < \pi - \beta$. By moving s_{24} and s_{21}, we can in a similar way as in Case (A) argue that (1) still holds when we substitute s_{21} by c_{12}.

We conclude that in both cases we can ensure (i) without violating (1).

Since $\gamma_{42} \leqslant \gamma_{32}$ and $\gamma_{13} \leqslant \gamma_{14}$, we likewise have $\gamma_{42} < \pi - \beta$ or $\gamma_{13} < \pi - \beta$. Hence, we can substitute s_{11} or s_{12} by c_{12} without violating (1), thus ensuring (ii) and finishing the proof. ◀

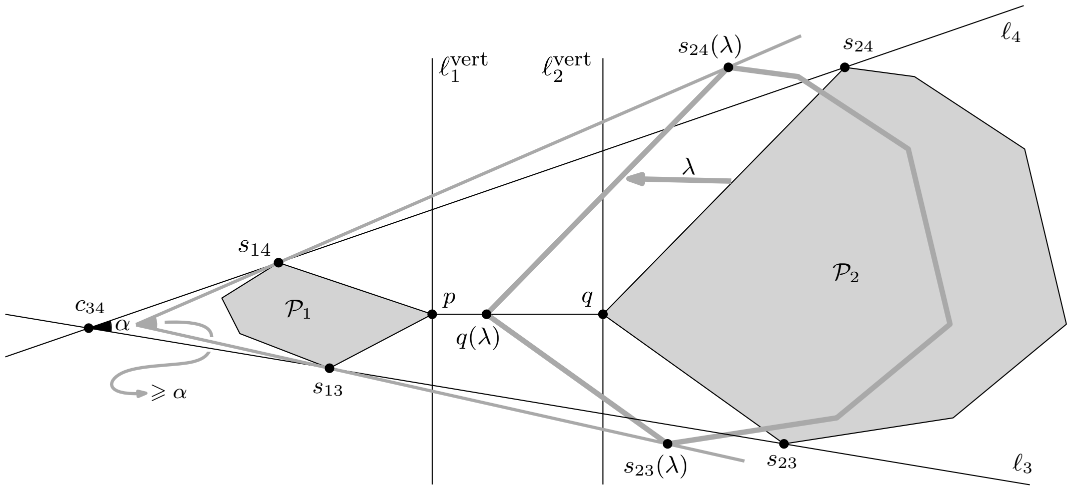

Figure 2 Illustration for the proof of Lemma 3.

Let $\mathrm{dist}(\mathcal{P}_1, \mathcal{P}_2) := \min_{(p,q)\in\mathcal{P}_1\times\mathcal{P}_2}|pq|$ denote the separation distance between $\mathcal{P}_1$ and $\mathcal{P}_2$. Recall that α denotes the angle between the two common outer tangents of $\mathcal{P}_1$ and $\mathcal{P}_2$; see Fig. 1

▶ **Lemma 3.** *We have*

$$\mathit{dist}(\mathcal{P}_1, \mathcal{P}_2) \;\geqslant\; f(\alpha)\cdot \mathrm{per}(\mathcal{P}_1), \tag{2}$$

where $f\colon [0,\pi] \longrightarrow \mathbb{R}$ *is the increasing function*

$$f(\varphi) \;:=\; \frac{\sin(\varphi/4)}{1+\sin(\varphi/4)}\cdot\frac{\sin(\varphi/2)}{1+\sin(\varphi/2)}\cdot\frac{1-\cos(\varphi/4)}{2}.$$

Proof. The statement is trivial if $\alpha = 0$ so assume $\alpha > 0$. Let $p \in \mathcal{P}_1$ and $q \in \mathcal{P}_2$ be points so that $|pq| = \mathrm{dist}(\mathcal{P}_1, \mathcal{P}_2)$ and assume without loss of generality that pq is a horizontal segment with p being its left endpoint. Let ℓ_1^{vert} and ℓ_2^{vert} be vertical lines containing p and q, respectively. Note that $\mathcal{P}_1$ is in the closed half-plane to the left of ℓ_1^{vert} and $\mathcal{P}_2$ is in the closed half-plane to the right of ℓ_2^{vert}. Recall that s_{ij} denotes a point on $\partial\mathcal{P}_i \cap \ell_j$.

▶ **Claim 4.** *There exist two convex polygons $\mathcal{P}'_1$ and $\mathcal{P}'_2$ satisfying the following conditions:*

1. *$\mathcal{P}'_1$ and $\mathcal{P}'_2$ have the same outer common tangents as $\mathcal{P}_1$ and $\mathcal{P}_2$, namely ℓ_3 and ℓ_4.*
2. *$\mathcal{P}'_1$ is to the left of ℓ_1^{vert} and $p \in \partial\mathcal{P}'_1$; and $\mathcal{P}'_2$ is to right of ℓ_2^{vert} and $q \in \partial\mathcal{P}'_2$.*
3. $\mathrm{per}(\mathcal{P}'_1) = \mathrm{per}(\mathcal{P}_1)$.
4. $\mathrm{per}(\mathcal{P}'_1) + \mathrm{per}(\mathcal{P}'_2) \leqslant \mathrm{per}(\textsc{ch}(\mathcal{P}'_1 \cup \mathcal{P}'_2))$.
5. *There are points $s'_{ij} \in \mathcal{P}'_i \cap \ell_j$ for all $i \in \{1,2\}$ and $j \in \{3,4\}$ such that $\partial\mathcal{P}'_1(s'_{13}, p)$, $\partial\mathcal{P}'_1(p, s'_{14})$, $\partial\mathcal{P}'_2(s'_{24}, q)$, and $\partial\mathcal{P}'_2(q, s'_{23})$ each consist of a single line segment.*
6. *Let $s'_{2j}(\lambda) := s'_{2j} - (\lambda, 0)$ and let $\ell'_j(\lambda)$ be the line through s'_{1j} and $s'_{2j}(\lambda)$ for $j \in \{3,4\}$. Then $\angle(\ell'_3(|pq|), \ell'_4(|pq|)) \geqslant \alpha/2$.*

Proof of the Claim. Let $\mathcal{P}'_1 := \mathcal{P}_1$ and $\mathcal{P}'_2 := \mathcal{P}_2$, and let s'_{ij} be a point in $\mathcal{P}'_i \cap \ell_j$ for all $i \in \{1,2\}$ and $j \in \{3,4\}$. We show how to modify $\mathcal{P}'_1$ and $\mathcal{P}'_2$ until they have all the required conditions. Of course, they already satisfy conditions 1–4. We first show how to obtain condition 5, namely that $\partial\mathcal{P}'_1(s'_{13}, p)$ and $\partial\mathcal{P}'_1(p, s'_{14})$ – and similarly $\partial\mathcal{P}'_2(s'_{24}, q)$ and

$\partial\mathcal{P}'_1(q, s'_{23})$ – each consist of a single line segment, as depicted in Fig. 2. To this end, let v_{ij} be the intersection point $\ell_i^{\text{vert}} \cap \ell_j$ for $i \in \{1, 2\}$ and $j \in \{3, 4\}$. Let $s' \in s'_{14}v_{14}$ be the point such that $\text{length}(\partial\mathcal{P}'_1(p, s'_{14})) = |ps'| + |s's'_{14}|$. Such a point exists since

$$|ps'_{14}| \leqslant \text{length}(\partial\mathcal{P}'_1(p, s'_{14})) \leqslant |pv_{14}| + |v_{14}s'_{14}|.$$

We modify $\mathcal{P}'_1$ by substituting $\partial\mathcal{P}'_1(p, s'_{14})$ with the segments ps' and $s's'_{14}$. We can now redefine $s'_{14} := s'$ so that $\partial\mathcal{P}'_1(p, s'_{14}) = ps'_{14}$ is a line segment. We can modify $\mathcal{P}'_1$ in a similar way to ensure that $\partial\mathcal{P}'_1(s'_{13}, p) = s'_{13}p$, and we can modify $\mathcal{P}'_2$ to ensure $\partial\mathcal{P}'_2(s'_{24}, q) = s'_{24}q$ and $\partial\mathcal{P}'_2(q, s'_{23}) = qs'_{23}$. Note that these modifications preserve conditions 1–4 and that condition 5 is now satisfied.

The only condition that $(\mathcal{P}'_1, \mathcal{P}'_2)$ might not satisfy is condition 6. Let $s'_{2j}(\lambda) := s'_{2j} - (\lambda, 0)$ and let $\ell_j(\lambda)$ be the line through $s'_{2j}(\lambda)$ and s'_{1j} for $j \in \{3, 4\}$. Clearly, if the slopes of ℓ_3 and ℓ_4 have different signs (as in Fig. 2), the angle $\angle(\ell_3(\lambda), \ell_4(\lambda))$ is increasing for $\lambda \in [0, |pq|]$, and condition 6 is satisfied. However, if the slopes of ℓ_3 and ℓ_4 have the same sign, the angle might decrease.

Consider the case where both slopes are positive – the other case is analogous. Changing $\mathcal{P}'_2$ by substituting $\partial\mathcal{P}'_2(s'_{23}, s'_{24})$ with the line segment $s'_{23}s'_{24}$ makes $\text{per}(\mathcal{P}'_1) + \text{per}(\mathcal{P}'_2)$ and $\text{per}(\text{CH}(\mathcal{P}'_1 \cup \mathcal{P}'_2))$ decrease equally much and hence condition 4 is preserved. This clearly has no influence on the other conditions. We thus assume that $\mathcal{P}'_2$ is the triangle $qs'_{23}s'_{24}$. Consider what happens if we move s'_{23} along the line ℓ_3 away from c_{34} with unit speed. Then $|s'_{13}s'_{23}|$ grows with speed exactly 1 whereas $|qs'_{23}|$ grows with speed at most 1. We therefore preserve condition 4, and the other conditions are likewise not affected.

We now move s'_{23} sufficiently far away so that $\angle(\ell_3, \ell_3(|pq|)) \leqslant \alpha/4$. Similarly, we move s'_{24} sufficiently far away from c_{34} along ℓ_4 to ensure that $\angle(\ell_4, \ell_4(|pq|)) \leqslant \alpha/4$. It then follows that $\angle(\ell_3(|pq|), \ell_4(|pq|)) \geqslant \angle(\ell_3, \ell_4) - \alpha/2 = \alpha/2$, and condition 6 is satisfied. ◀

Note that condition 2 in the claim implies that $\text{dist}(\mathcal{P}'_1, \mathcal{P}'_2) = \text{dist}(\mathcal{P}_1, \mathcal{P}_2) = |pq|$, and hence inequality (2) follows from condition 3 if we manage to prove $\text{dist}(\mathcal{P}'_1, \mathcal{P}'_2) \geqslant f(\alpha) \cdot \text{per}(\mathcal{P}'_1)$. Therefore, with a slight abuse of notation, we assume from now on that $\mathcal{P}_1$ and $\mathcal{P}_2$ satisfy the conditions in the claim, where the points s_{ij} play the role as s'_{ij} in conditions 5 and 6.

We now consider a copy of $\mathcal{P}_2$ that is translated horizontally to the left over a distance λ; see Fig. 2. Let $s_{24}(\lambda)$, $s_{23}(\lambda)$, and $q(\lambda)$ be the translated copies of s_{24}, s_{23}, and q, respectively, and let $\ell_j(\lambda)$ be the line through s_{1j} and $s_{2j}(\lambda)$ for $j \in \{3, 4\}$. Furthermore, define

$$\Phi(\lambda) := |s_{13}p| + |s_{14}p| + |s_{23}(\lambda)q(\lambda)| + |s_{24}(\lambda)q(\lambda)|$$

and

$$\Psi(\lambda) := |s_{13}s_{23}(\lambda)| + |s_{14}s_{24}(\lambda)|.$$

Note that $\Phi(\lambda) = \Phi$ is constant. By conditions 4 and 5, we know that

$$\Phi \leqslant \Psi(0). \tag{3}$$

Note that $q(|pq|) = p$. In the full version [1] we show that

$$\Phi - \Psi(|pq|) \geqslant \sin(\delta/2) \cdot \frac{1 - \cos(\delta/2)}{1 + \sin(\delta/2)} \cdot (|s_{13}p| + |s_{14}p|), \tag{4}$$

where $\delta := \angle(\ell_3(|pq|), \ell_4(|pq|))$. By condition 6, we know that $\delta \geqslant \alpha/2$. The function $\delta \longmapsto \sin(\delta/2) \cdot \frac{1-\cos(\delta/2)}{1+\sin(\delta/2)}$ is increasing for $\delta \in [0, \pi]$ and hence inequality (4) also holds for $\delta = \alpha/2$.

When λ increases from 0 to $|pq|$ with unit speed, the value $\Psi(\lambda)$ decreases with speed at most 2, i.e., $\Psi(\lambda) \geqslant \Psi(0) - 2\lambda$. Using this and inequalities (3) and (4), we get

$$2|pq| \;\geqslant\; \Psi(0) - \Psi(|pq|) \;\geqslant\; \Phi - \Phi + \sin(\alpha/4) \cdot \frac{1-\cos(\alpha/4)}{1+\sin(\alpha/4)} \cdot (|s_{13}p| + |s_{14}p|),$$

and we conclude that

$$|pq| \;\geqslant\; \frac{1}{2} \cdot \sin(\alpha/4) \cdot \frac{1-\cos(\alpha/4)}{1+\sin(\alpha/4)} \cdot (|s_{13}p| + |s_{14}p|). \tag{5}$$

By the triangle inequality, $|s_{13}p| + |s_{14}p| \geqslant |s_{13}s_{14}|$. Furthermore, for a given length of $s_{13}s_{14}$, the fraction $|s_{13}s_{14}|/(|s_{14}c_{34}| + |c_{34}s_{13}|)$ is minimized when $s_{13}s_{14}$ is perpendicular to the angular bisector of ℓ_3 and ℓ_4. (Recall that c_{34} is the intersection point of the outer common tangents ℓ_3 and ℓ_4; see Fig. 2.) Hence

$$|s_{13}s_{14}| \;\geqslant\; \sin(\alpha/2) \cdot (|s_{14}c_{34}| + |c_{34}s_{13}|)\,. \tag{6}$$

We now conclude

$$\begin{aligned}
|s_{13}p| + |s_{14}p| \;&=\; \tfrac{\sin(\alpha/2)}{1+\sin(\alpha/2)} \cdot \left(\tfrac{|s_{13}p|+|s_{14}p|}{\sin(\alpha/2)} + |s_{13}p| + |s_{14}p| \right) \\
&\geqslant\; \tfrac{\sin(\alpha/2)}{1+\sin(\alpha/2)} \cdot \left(\tfrac{|s_{13}s_{14}|}{\sin(\alpha/2)} + |s_{13}p| + |s_{14}p| \right) && \text{by the triangle inequality} \\
&\geqslant\; \tfrac{\sin(\alpha/2)}{1+\sin(\alpha/2)} \cdot \left(|s_{14}c_{34}| + |c_{34}s_{13}| + |s_{13}p| + |s_{14}p| \right) && \text{by (6)} \\
&\geqslant\; \tfrac{\sin(\alpha/2)}{1+\sin(\alpha/2)} \cdot \mathrm{per}(\mathcal{P}_1),
\end{aligned}$$

where the last inequality follows because $\mathcal{P}_1$ is fully contained in the quadrilateral $s_{14}, c_{34}, x_{13}, p$. The statement (2) in the lemma now follows from (5). ◀

We are now ready to prove Theorem 1.

Proof of Theorem 1. If the separation angle of P_1 and P_2 is at least $\pi/6$, we are done. Otherwise, Lemma 2 gives that $\alpha > \pi/2$, and Lemma 3 gives that $\mathrm{dist}(\mathcal{P}_1, \mathcal{P}_2) \;\geqslant\; f(\pi/2) \cdot \mathrm{per}(\mathcal{P}_1) \geqslant (1/250) \cdot \min(\mathrm{per}(\mathcal{P}_1), \mathrm{per}(\mathcal{P}_2))$. ◀

2.2 The algorithm

Theorem 1 suggests to distinguish two cases when computing an optimal partition: the case where the separation angle is large (namely at least $\pi/6$) and the case where the separation distance is large (namely at least $c_{\mathrm{sep}} \cdot \min(\mathrm{per}(P_1), \mathrm{per}(P_2))$). As we will see, the first case can be handled in $O(n \log n)$ time and the second case in $O(n \log^4 n)$ time, leading to the following theorem.

► **Theorem 5.** *Let P be a set of n points in the plane. Then we can compute a partition (P_1, P_2) of P that minimizes $\mathrm{per}(P_1) + \mathrm{per}(P_2)$ in $O(n \log^4 n)$ time using $O(n \log^3 n)$ space.*

To find the best partition when the separation angle is at least $\pi/6$, we observe that in this case there is a separating line whose orientation is $j \cdot \pi/7$ for some $0 \leqslant j < 7$. For each of these orientations we can scan over the points with a line ℓ of the given orientation, and maintain the perimeters of the convex hulls on both sides. This takes $O(n \log n)$ time in total; see the full version [1].

Next we show how to compute the best partition with large separation distance. We assume without loss of generality that $\mathrm{per}(P_2) \leqslant \mathrm{per}(P_1)$. It will be convenient to treat the case where P_2 is a singleton separately.

▶ **Lemma 6.** *The point $p \in P$ minimizing* $\mathrm{per}(P \setminus \{p\})$ *can be computed in* $O(n \log n)$ *time.*

Proof. The point p we are looking for must be a vertex of $\mathrm{CH}(P)$. First we compute $\mathrm{CH}(P)$ in $O(n \log n)$ time [5]. Let $v_0, v_1, \ldots, v_{m-1}$ denote the vertices of $\mathrm{CH}(P)$ in counterclockwise order. Let Δ_i be the triangle with vertices $v_{i-1}v_iv_{i+1}$ (with indices taken modulo m) and let P_i denote the set of points lying inside Δ_i, excluding v_i but including v_{i-1} and v_{i+1}. Note that any point $p \in P$ is present in at most two sets P_i. Hence, $\sum_{i=0}^{m} |P_i| = O(n)$. It is not hard to compute the sets P_i in $O(n \log n)$ time in total. After doing so, we compute all convex hulls $\mathrm{CH}(P_i)$ in $O(n \log n)$ time in total. Since

$$\mathrm{per}(P \setminus \{v_i\}) = \mathrm{per}(P) - |v_{i-1}v_i| - |v_iv_{i+1}| + \mathrm{per}(P_i) - |v_{i-1}v_{i+1}|,$$

we can now find the point p minimizing $\mathrm{per}(P \setminus \{p\})$ in $O(n)$ time. ◀

It remains to compute the best partition (P_1, P_2) with $\mathrm{per}(P_2) \leqslant \mathrm{per}(P_1)$ whose separation distance is at least $c_{\mathrm{sep}} \cdot \mathrm{per}(P_2)$ and where P_2 is not a singleton. Let (P_1^*, P_2^*) denote this partition. Define the *size* of a square[1] σ to be its edge length. A square σ is a *good square* if (i) $P_2^* \subset \sigma$, and (ii) $\mathrm{size}(\sigma) \leqslant c^* \cdot \mathrm{per}(P_2^*)$, where $c^* := 18$. Our algorithm globally works as follows.

1. Compute a set S of $O(n)$ squares such that S contains a good square.
2. For each square $\sigma \in S$, construct a set H_σ of $O(1)$ halfplanes such that the following holds: if $\sigma \in S$ is a good square then there is a halfplane $h \in H_\sigma$ such that $P_2^* = P(\sigma \cap h)$, where $P(\sigma \cap h) := P \cap (\sigma \cap h)$.
3. For each pair (σ, h) with $\sigma \in S$ and $h \in H_\sigma$, compute $\mathrm{per}(P \setminus P(\sigma \cap h)) + \mathrm{per}(P(\sigma \cap h))$, and report the partition $(P \setminus P(\sigma \cap h), P(\sigma \cap h))$ that gives the smallest sum.

Step 1: Finding a good square. To find a set S that contains a good square, we first construct a set S_{base} of so-called *base squares*. The set S will then be obtained by expanding the base squares appropriately.

We define a base square σ to be *good* if (i) σ contains at least one point from P_2^*, and (ii) $c_1 \cdot \mathrm{diam}(P_2^*) \leqslant \mathrm{size}(\sigma) \leqslant c_2 \cdot \mathrm{diam}(P_2^*)$, where $c_1 := 1/4$ and $c_2 := 4$ and $\mathrm{diam}(P_2^*)$ denotes the diameter of P_2^*. Note that $2 \cdot \mathrm{diam}(P_2^*) \leqslant \mathrm{per}(P_2^*) \leqslant 4 \cdot \mathrm{diam}(P_2^*)$. For a square σ, define $\overline{\sigma}$ to be the square with the same center as σ and whose size is $(1 + 2/c_1) \cdot \mathrm{size}(\sigma)$.

▶ **Lemma 7.** *If σ is a good base square then $\overline{\sigma}$ is a good square.*

Proof. The distance from any point in σ to the boundary of $\overline{\sigma}$ is at least

$$\frac{\mathrm{size}(\overline{\sigma}) - \mathrm{size}(\sigma)}{2} \geqslant \mathrm{diam}(P_2^*).$$

Since σ contains a point from P_2^*, it follows that $P_2^* \subset \overline{\sigma}$. Since $\mathrm{size}(\sigma) \leqslant c_2 \cdot \mathrm{diam}(P_2^*)$, we have

$$\mathrm{size}(\overline{\sigma}) \leqslant (2/c_1 + 1) \cdot c_2 \cdot \mathrm{diam}(P_2^*) = 36 \cdot \mathrm{diam}(P_2^*) \leqslant c^* \cdot \mathrm{per}(P_2^*). \quad ◀$$

To obtain S it thus suffices to construct a set S_{base} that contains a good base square. To this end we first build a compressed quadtree for P. For completeness we briefly review the definition of compressed quadtrees; see also Fig. 3 (left).

[1] Whenever we speak of squares, we always mean axis-parallel squares.

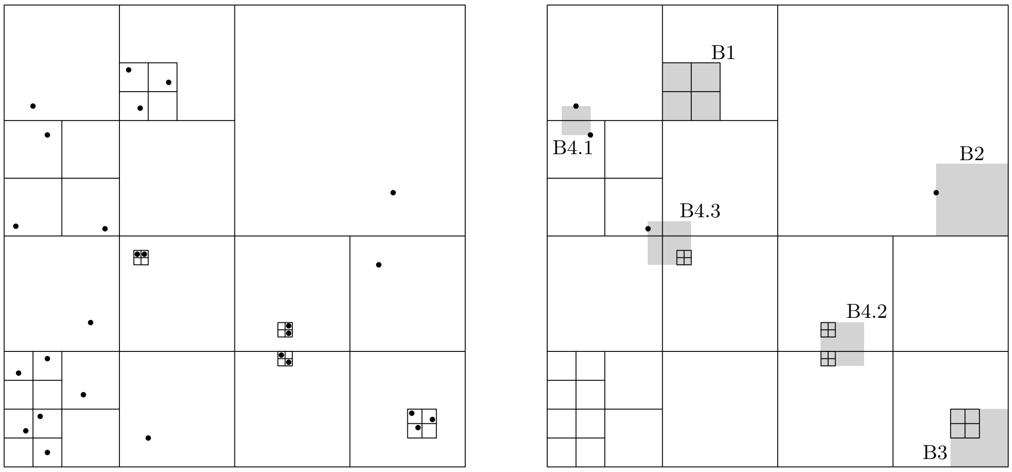

Figure 3 A compressed quadtree and some of the base squares generated from it. In the right figure, only the points are shown that are relevant for the shown base squares.

Assume without loss of generality that P lies in the interior of the unit square $U := [0,1]^2$. Define a *canonical square* to be any square that can be obtained by subdividing U recursively into quadrants. A *compressed quadtree* [13] for P is a hierarchical subdivision of U, defined as follows. In a generic step of the recursive process we are given a canonical square σ and the set $P(\sigma) := P \cap \sigma$ of points inside σ. (Initially $\sigma = U$ and $P(\sigma) = P$.)

- If $|P(\sigma)| \leqslant 1$ then the recursive process stops and σ is a square in the final subdivision.
- Otherwise there are two cases. Consider the four quadrants of σ. The first case is that at least two of these quadrants contain points from $P(\sigma)$. (We consider the quadrants to be closed on the left and bottom side, and open on the right and top side, so a point is contained in a unique quadrant.) In this case we partition σ into its four quadrants – we call this a *quadtree split* – and recurse on each quadrant. The second case is that all points from $P(\sigma)$ lie inside the same quadrant. In this case we compute the smallest canonical square, σ', that contains $P(\sigma)$ and we partition σ into two regions: the square σ' and the so-called *donut region* $\sigma \setminus \sigma'$. We call this a *shrinking step*. After a shrinking step we only recurse on the square σ', not on the donut region.

A compressed quadtree for a set of n points can be computed in $O(n \log n)$ time in the appropriate model of computation[2] [13]. The idea is now as follows. Let $p, p' \in P_2^*$ be a pair of points defining $\mathrm{diam}(P_2^*)$. The compressed quadtree hopefully allows us to zoom in until we have a square in the compressed quadtree that contains p or p' and whose size is roughly equal to $|pp'|$. Such a square will be then a good base square. Unfortunately this does not always work since p and p' can be separated too early. We therefore have to proceed more carefully: we need to add five types of base squares to S_{base}, as explained next and illustrated in Fig. 3 (right).

(B1) Any square σ that is generated during the recursive construction – note that this not only refers to squares in the final subdivision – is put into S_{base}.

(B2) For each point $p \in P$ we add a square σ_p to S_{base}, as follows. Let σ be the square of the final subdivision that contains p. Then σ_p is a smallest square that contains p and that shares a corner with σ.

[2] In particular we need to be able to compute the smallest canonical square containing two given points in $O(1)$ time. See the book by Har-Peled [13] for a discussion.

(B3) For each square σ that results from a shrinking step we add an extra square σ' to S_{base}, where σ' is the smallest square that contains σ and that shares a corner with the parent square of σ.

(B4) For any two regions in the final subdivision that touch each other – we also consider two regions to touch if they only share a vertex – we add at most one square to S_{base}, as follows. If one of the regions is an empty square, we do not add anything for this pair. Otherwise we have three cases.

(B4.1) If both regions are non-empty squares containing points p and p', respectively, then we add a smallest enclosing square for the pair of points p, p' to S_{base}.

(B4.2) If both regions are donut regions, say $\sigma_1 \setminus \sigma'_1$ and $\sigma_2 \setminus \sigma'_2$, then we add a smallest enclosing square for the pair σ'_1, σ'_2 to S_{base}.

(B4.3) If one region is a non-empty square containing a point p and the other is a donut region $\sigma \setminus \sigma'$, then we add a smallest enclosing square for the pair p, σ' to S_{base}.

▶ **Lemma 8.** *The set S_{base} has size $O(n)$ and contains a good base square. Furthermore, S_{base} can be computed in $O(n \log n)$ time.*

Proof. A compressed quadtree has size $O(n)$ so we have $O(n)$ base squares of type (B1) and (B3). Obviously there are $O(n)$ base squares of type (B2). Finally, the number of pairs of final regions that touch is $O(n)$ – this follows because we have a planar rectilinear subdivision of total complexity $O(n)$ – and so the number of base squares of type (B4) is $O(n)$ as well. The fact that we can compute S_{base} in $O(n \log n)$ time follows directly from the fact that we can compute the compressed quadtree in $O(n \log n)$ time [13].

It remains to prove that S_{base} contains a good base square. We call a square σ *too small* when $\text{size}(\sigma) < c_1 \cdot \text{diam}(P_2^*)$ and *too large* when $\text{size}(\sigma) > c_2 \cdot \text{diam}(P_2^*)$; otherwise we say that σ has the *correct size*. Let $p, p' \in P_2^*$ be two points with $|pp'| = \text{diam}(P_2^*)$, and consider a smallest square $\sigma_{p,p'}$, in the compressed quadtree that contains both p and p'. Note that $\sigma_{p,p'}$ cannot be too small, since $c_1 = 1/4 < 1/\sqrt{2}$. If $\sigma_{p,p'}$ has the correct size, then we are done since it is a good base square of type (B1). So now suppose $\sigma_{p,p'}$ is too large.

Let $\sigma_0, \sigma_1, \dots, \sigma_k$ be the sequence of squares in the recursive subdivision of $\sigma_{p,p'}$ that contain p; thus $\sigma_0 = \sigma_{p,p'}$ and σ_k is a square in the final subdivision. Define $\sigma'_0, \sigma'_1, \dots, \sigma'_{k'}$ similarly, but now for p' instead of p. Suppose that none of these squares has the correct size – otherwise we have a good base square of type (B1). There are three cases.

- *Case* (i)*: σ_k and $\sigma'_{k'}$ are too large.*
  We claim that σ_k touches $\sigma'_{k'}$. To see this, assume without loss of generality that $\text{size}(\sigma_k) \leqslant \text{size}(\sigma'_{k'})$. If σ_k does not touch $\sigma'_{k'}$ then $|pp'| \geqslant \text{size}(\sigma_k)$, which contradicts that σ_k is too large. Hence, σ_k indeed touches $\sigma'_{k'}$. But then we have a base square of type (B4.1) for the pair p, p' and since $|pp'| = \text{diam}(P_2^*)$ this is a good base square.
- *Case* (ii)*: σ_k and $\sigma'_{k'}$ are too small.*
  In this case there are indices $0 < j \leqslant k$ and $0 < j' \leqslant k'$ such that σ_{j-1} and $\sigma'_{j'-1}$ are too large and σ_j and $\sigma'_{j'}$ are too small. Note that this implies that both σ_j and $\sigma'_{j'}$ result from a shrinking step, because $c_1 < c_2/2$ and so the quadrants of a too-large square cannot be too small. We claim that σ_{j-1} touches $\sigma'_{j'-1}$. Indeed, similarly to Case (i), if σ_{j-1} and $\sigma'_{j'-1}$ do not touch then $|pp'| > \min(\text{size}(\sigma_{j-1}), \text{size}(\sigma'_{j'-1}))$, contradicting that both σ_{j-1} and $\sigma'_{j'-1}$ are too large. We now have two subcases.
  - The first subcase is that the donut region $\sigma_{j-1} \setminus \sigma_j$ touches the donut region $\sigma'_{j'-1} \setminus \sigma_{j'}$. Thus a smallest enclosing square for σ_j and $\sigma'_{j'}$ has been put into S_{base} as a base square of type (B4.2). Let σ^* denote this square. Since the segment pp' is contained

in σ^* we have

$$c_1 \cdot \mathrm{diam}(P_2^*) \ < \ \mathrm{diam}(P_2^*)/\sqrt{2} \ = \ |pp'|/\sqrt{2} \ \leqslant \ \mathrm{size}(\sigma^*).$$

Furthermore, since σ_j and $\sigma'_{j'}$ are too small we have

$$\mathrm{size}(\sigma^*) \ \leqslant \ \mathrm{size}(\sigma_j) + \mathrm{size}(\sigma'_{j'}) + |pp'| \ \leqslant \ 3 \cdot \mathrm{diam}(P_2^*) \ < \ c_2 \cdot \mathrm{diam}(P_2^*), \tag{7}$$

and so σ^* is a good base square.

- The second subcase is that $\sigma_{j-1} \setminus \sigma_j$ does not touch $\sigma'_{j'-1} \setminus \sigma_{j'}$. This can only happen if σ_{j-1} and $\sigma'_{j'-1}$ just share a single corner, v. Observe that σ_j must lie in the quadrant of σ_{j-1} that has v as a corner, otherwise $|pp'| \geqslant \mathrm{size}(\sigma_{j-1})/2$ and σ_{j-1} would not be too large. Similarly, $\sigma'_{j'}$ must lie in the quadrant of $\sigma'_{j'-1}$ that has v as a corner. Thus the base squares of type (B3) for σ_j and $\sigma'_{j'}$ both have v as a corner. Take the largest of these two base squares, say σ_j. For this square σ^* we have

$$c_1 \cdot \mathrm{diam}(P_2^*) \ < \ \mathrm{diam}(P_2^*)/2\sqrt{2} \ = \ |pp'|/2\sqrt{2} \ \leqslant \ \mathrm{size}(\sigma^*),$$

since $|pp'|$ is contained in a square of twice the size of σ^*. Furthermore, since σ_j is too small and $|pv| < |pp'|$ we have

$$\mathrm{size}(\sigma^*) \ \leqslant \ \mathrm{size}(\sigma_j) + |pv| \ \leqslant \ (c_1 + 1) \cdot \mathrm{diam}(P_2^*) \ < \ c_2 \cdot \mathrm{diam}(P_2^*). \tag{8}$$

Hence, σ^* is a good base square.

- *Case* (iii)*: neither* (i) *nor* (ii) *applies.*
In this case σ_k is too small and $\sigma'_{k'}$ is too large (or vice versa). Thus there must be an index $0 < j \leqslant k$ such that σ_{j-1} is too large and σ_j is too small. We can now follow a similar reasoning as in Case (ii): First we argue that σ_j must have resulted from a shrinking step and that σ_{j-1} touches $\sigma'_{k'}$. Then we distinguish two subcases, namely where the donut region $\sigma_j \setminus \sigma_{j-1}$ touches $\sigma'_{k'}$ and where it does not touch $\sigma'_{k'}$. The arguments for the two subcases are similar to the subcases in Case (ii), with the following modifications. In the first subcase we use base squares of type (B4.3) and in (7) the term $\mathrm{size}(\sigma'_{j'})$ disappears; in the second subcase we use a type (B3) base square for σ_j and a type (B2) base square for p', and when the base square for p' is larger than the base square for σ_j then (8) becomes $\mathrm{size}(\sigma^*) \leqslant 2\,|p'v| < c_2 \cdot \mathrm{diam}(P_2^*)$. ◀

Step 2: Generating halfplanes. Consider a good square $\sigma \in S$. Let Q_σ be a set of $4 \cdot c^*/c_{\mathrm{sep}} + 1 = 18001$ points placed equidistantly around the boundary of σ. Note that the distance between two neighbouring points in Q_σ is less than $c_{\mathrm{sep}}/c^* \cdot \mathrm{size}(\sigma)$. For each pair q_1, q_2 of points in Q_σ, add to H_σ the two halfplanes defined by the line through q_1 and q_2.

▶ **Lemma 9.** *For any good square $\sigma \in S$, there is a halfplane $h \in H_\sigma$ such that $P_2^* = P(\sigma \cap h)$.*

Proof. In the case where $\sigma \cap P_1^* = \emptyset$, two points in Q_σ from the same edge of σ define a half-plane h such that $P_2^* = P(\sigma \cap h)$, so assume that σ contains one or more points from P_1^*.

We know that the separation distance between P_1^* and P_2^* is at least $c_{\mathrm{sep}} \cdot \mathrm{per}(P_2^*)$. Moreover, $\mathrm{size}(\sigma) \leqslant c^* \cdot \mathrm{per}(P_2^*)$. Hence, there is an empty open strip O with a width of at least $c_{\mathrm{sep}}/c^* \cdot \mathrm{size}(\sigma)$ separating P_2^* from P_1^*. Since σ contains a point from P_1^*, we know that $\sigma \setminus O$ consists of two pieces and that the part of the boundary of σ inside O consists of two disjoint portions B_1 and B_2 each of length at least $c_{\mathrm{sep}}/c^* \cdot \mathrm{size}(\sigma)$. Hence the sets $B_1 \cap Q_\sigma$ and $B_2 \cap Q_\sigma$ contain points q_1 and q_2, respectively, that define a half-plane h as desired. ◀

Step 3: Evaluating candidate solutions. In this step we need to compute for each pair (σ, h) with $\sigma \in S$ and $h \in H_\sigma$, the value $\mathrm{per}(P \setminus P(\sigma \cap h)) + \mathrm{per}(P(\sigma \cap h))$. We do this by preprocessing P into a data structure that allows us to quickly compute $\mathrm{per}(P \setminus P(\sigma \cap h))$ and $\mathrm{per}(P(\sigma \cap h))$ for a given pair (σ, h). Recall that the bounding lines of the halfplanes h we must process have $O(1)$ different orientations. We construct a separate data structure for each orientation.

Consider a fixed orientation ϕ. We build a data structure $\mathcal{D}_\phi$ for range searching on P with ranges of the form $\sigma \cap h$, where σ is a square and h is halfplane whose bounding line has orientation ϕ. Since the edges of σ are axis-parallel and the bounding line of the halfplanes h have a fixed orientation, we can use a standard three-level range tree [5] for this. Constructing this tree takes $O(n \log^2 n)$ time and the tree has $O(n \log^2 n)$ nodes.

Each node ν of the third-level trees in $\mathcal{D}_\phi$ is associated with a *canonical subset* $P(\nu)$, which contains the points stored in the subtree rooted at ν. We preprocess each canonical subset $P(\nu)$ as follows. First we compute the convex hull $\mathrm{CH}(P(\nu))$. Let $v_1, \ldots, v_k$ denote the convex-hull vertices in counterclockwise order. We store these vertices in order in an array, and we store for each vertex v_i the value $\mathrm{length}(\partial P(v_1, v_i))$, that is, the length of the part of $\partial\, \mathrm{CH}(P(\nu))$ from v_1 to v_i in counterclockwise order. Note that the convex hull $\mathrm{CH}(P(\nu))$ can be computed in $O(|P(\nu)|)$ from the convex hulls at the two children of ν. Hence, the convex hulls $\mathrm{CH}(P(\nu))$ (and the values $\mathrm{length}(\partial P(v_1, v_i))$) can be computed in $\sum_{\nu \in \mathcal{D}_\phi} O(|P(\nu)|) = O(n \log^3 n)$ time in total, in a bottom-up manner.

Now suppose we want to compute $\mathrm{per}(P(\sigma \cap h))$, where the orientation of the bounding line of h is ϕ. We perform a range query in $\mathcal{D}_\phi$ to find a set $N(\sigma \cap h)$ of $O(\log^3 n)$ nodes such that $P(\sigma \cap h)$ is equal to the union of the canonical subsets of the nodes in $N(\sigma \cap h)$. Standard range-tree properties guarantee that the convex hulls $\mathrm{CH}(P(\nu))$ and $\mathrm{CH}(P(\mu))$ of any two nodes $\nu, \mu \in N(\sigma \cap h)$ are disjoint. Note that $\mathrm{CH}(P(\sigma \cap h))$ is equal to the convex hull of the set of convex hulls $\mathrm{CH}(P(\nu))$ with $\nu \in N(\sigma \cap h)$. In the full version [1] we show that we can compute $\mathrm{per}(P(\sigma \cap h))$ in $O(\log^4 n)$ time.

Observe that $P \setminus P(\sigma \cap h)$ can also be expressed as the union of $O(\log^3 n)$ canonical subsets with disjoint convex hulls, since $\mathbb{R}^2 \setminus (\sigma \cap h)$ is the disjoint union of $O(1)$ ranges of the right type. Hence, we can compute $\mathrm{per}(P \setminus P(\sigma \cap h))$ in $O(\log^4 n)$ time. We thus obtain the following result, which finishes the proof of Theorem 5.

▶ **Lemma 10.** *Step 3 can be performed in $O(n \log^4 n)$ time and using $O(n \log^3 n)$ space.*

3 The approximation algorithm

▶ **Theorem 11.** *Let P be a set of n points in the plane and let (P_1^*, P_2^*) be a partition of P minimizing $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*)$. Suppose we have an exact algorithm for the minimum perimeter-sum problem running in $T(k)$ time for instances with k points. Then for any given $\varepsilon > 0$ we can compute a partition (P_1, P_2) of P such that $\mathrm{per}(P_1) + \mathrm{per}(P_2) \leqslant (1 + \varepsilon) \cdot \big(\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*)\big)$ in $O(n + T(1/\varepsilon^2))$ time.*

Proof. Consider the axis-parallel bounding box B of P. Let w be the width of B and let h be its height. Assume without loss of generality that $w \geqslant h$. Our algorithm works in two steps.

Step 1: Check if $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) \leqslant w/16$. If so, compute the exact solution.
We partition B vertically into four strips with width $w/4$, denoted B_1, B_2, B_3, and B_4 from left to right. If B_2 or B_3 contains a point from P, we have $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) \geqslant w/2 > w/16$ and we go to Step 2. If B_2 and B_3 are both empty, we consider two cases.

Case (i)*: $h \leqslant w/8$.* In this case we simply return the partition $(P \cap B_1, P \cap B_4)$. To see that this is optimal, we first note that any subset $P' \subset P$ that contains a point from B_1 as well as a point from B_4 has $\mathrm{per}(P') \geqslant 2 \cdot (3w/4) = 3w/2$. On the other hand, $\mathrm{per}(P \cap B_1) + \mathrm{per}(P \cap B_4) \leqslant 2 \cdot (w/2 + 2h) \leqslant 3w/2$.

Case (ii)*: $h > w/8$.* We partition B horizontally into four rows with height $h/4$, numbered R_1, R_2, R_3, and R_4 from bottom to top. If R_2 or R_3 contains a point from P, we have $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) \geqslant h/2 > w/16$, and we go the Step 2. If R_2 and R_3 are both empty, we overlay the vertical and the horizontal partitioning of B to get a 4×4 grid of cells $C_{ij} := B_i \cap R_j$ for $i, j \in \{1, \ldots, 4\}$. We know that only the corner cells $C_{11}, C_{14}, C_{41}, C_{44}$ contain points from P. If three or four corner cells are non-empty, $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) \geqslant 6h/4 > w/16$. Hence, we may without loss of generality assume that any point of P is in C_{11} or C_{44}. We now return the partition $(P \cap C_{11}, P \cap C_{44})$, which is easily seen to be optimal.

Step 2: Handle the case where $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) > w/16$.
The idea is to compute a subset $\widehat{P} \subset P$ of size $O(1/\varepsilon^2)$ such that an exact solution to the minimum perimeter-sum problem on $\widehat{P}$ can be used to obtain a $(1+\varepsilon)$-approximation for the problem on P.
We subdivide B into $O(1/\varepsilon^2)$ rectangular cells of width and height at most $c := \varepsilon w/(64\pi\sqrt{2})$. For each cell C where $P \cap C$ is non-empty we pick an arbitrary point in $P \cap C$, and we let $\widehat{P}$ be the set of selected points. For a point $p \in \widehat{P}$, let $C(p)$ be the cell containing p. Intuitively, each point $p \in \widehat{P}$ represents all the points $P \cap C(p)$. Let $(\widehat{P}_1, \widehat{P}_2)$ be a partition of $\widehat{P}$ that minimizes $\mathrm{per}(\widehat{P}_1) + \mathrm{per}(\widehat{P}_2)$. We assume we have an algorithm that can compute such an optimal partition in $T(|\widehat{P}|)$ time. For $i = 1, 2$, define

$$P_i := \bigcup_{p \in \widehat{P}_i} P \cap C(p).$$

Our approximation algorithm returns the partition (P_1, P_2). (Note that the convex hulls of P_1 and P_2 are not necessarily disjoint.) It remains to prove the approximation ratio. First, note that $\mathrm{per}(\widehat{P}_1) + \mathrm{per}(\widehat{P}_2) \leqslant \mathrm{per}(P_1^*) + \mathrm{per}(P_2^*)$ since $\widehat{P} \subseteq P$. For $i = 1, 2$, let $\widetilde{P}_i$ consist of all points in the plane (not only points in P) within a distance of at most $\sqrt{2}c$ from $\textsc{ch}(\widehat{P}_i)$. In other words, $\widetilde{P}_i$ is the Minkowksi sum of $\textsc{ch}(\widehat{P}_i)$ with a disk D of radius $c\sqrt{2}$ centered at the origin. Note that if $p \in \widehat{P}_i$, then $q \in \widetilde{P}_i$ for any $q \in P \cap C(p)$, since any two points in $C(p)$ are at most $\sqrt{2}c$ apart from each other. Therefore $P_i \subset \widetilde{P}_i$ and hence $\mathrm{per}(P_i) \leqslant \mathrm{per}(\widetilde{P}_i)$. Note also that $\mathrm{per}(\widetilde{P}_i) = \mathrm{per}(\widehat{P}_i) + 2c\pi\sqrt{2}$. These observations yield

$$\begin{aligned}
\mathrm{per}(P_1) + \mathrm{per}(P_2) &\leqslant \mathrm{per}(\widetilde{P}_1) + \mathrm{per}(\widetilde{P}_2) \\
&= \mathrm{per}(\widehat{P}_1) + \mathrm{per}(\widehat{P}_2) + 4c\pi\sqrt{2} \leqslant \mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) + 4c\pi\sqrt{2} \\
&= \mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) + 4\pi\sqrt{2} \cdot \left(\varepsilon w/(64\pi\sqrt{2})\right) \\
&\leqslant \mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) + \varepsilon w/16 \leqslant (1+\varepsilon) \cdot (\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*)).
\end{aligned}$$

As all the steps can be done in linear time, the time complexity of the algorithm is $O(n+T(n_\varepsilon))$ for some $n_\varepsilon = O(1/\varepsilon^2)$. ◀

Acknowledgements. This research was initiated when the first author visited the Department of Computer Science at TU Eindhoven during the winter 2015–2016. He wishes to express his gratitude to the other authors and the department for their hospitality.

References

1 M. Abrahamsen, M. de Berg, K. Buchin, M. Mehr, A. D. Mehrabi. Minimum Perimeter-Sum Partitions in the Plane. Preprint, `http://arxiv.org/abs/1703.05549` (2017).

2 P. K. Agarwal, M. Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.* 30(4):412–458 (1998).

3 T. Asano, B. Bhattacharya, M. Keil, and F. Yao. Clustering algorithms based on minimum and maximum spanning trees. In *Proc. 4th ACM Symp. Comput. Geom. (SoCG)*, pages 252–257, 1988.

4 S. W. Bae, H.-G. Cho, W. Evans, N. Saeedi, and C.-S. Shin. Covering points with convex sets of minimum size. *Theor. Comput. Sci.*, in press (2016).

5 M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications (3rd edition)*. Springer-Verlag, 2008.

6 P. Brass, C. Knauer, C.-S. Shin, M. Smid, and I. Vigan. Range-aggregate queries for geometric extent problems. In *Proc. 19th Computing: Australasian Theory Symp. (CATS)*, pages 3–10, 2013.

7 V. Capoyleas, G. Rote, G. Woeginger. Geometric clusterings. *J. Alg.* 12(2):341–356 (1991).

8 T. M. Chan. More planar two-center algorithms. *Comput. Geom. Theory Appl.* 13(2):189–198 (1999).

9 T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (3rd edition)*. MIT Press, 2009.

10 O. Devillers and M. J. Katz. Optimal line bipartitions of point sets. *Int. J. Comput. Geom. Appl.* 9(1):39–51 (1999).

11 Z. Drezner. The planar two-center and two-median problems. *Transportation Science* 18(4):351–361 (1984).

12 D. Eppstein. Faster construction of planar two-centers. In *Proc. 8th ACM-SIAM Symp. Discr. Alg. (SODA)*, pages 131–138 (1997).

13 S. Har-Peled. *Geometric approximation algorithms.* Mathematical surveys and monographs, Vol. 173. American Mathematical Society, 2011.

14 J. Hershberger. Minimizing the sum of diameters efficiently. *Comput. Geom. Theory Appl.* 2(2):111–118 (1992).

15 J. W. Jaromczyk and M. Kowaluk. An efficient algorithm for the Euclidean two-center problem. In *Proc. 10th ACM Symp. Comput. Geom. (SoCG)*, pages 303–311 (1994).

16 D. Kirkpatrick and J. Snoeyink. Computing common tangents without a separating line. In *Proc. 4th Workshop Alg. Data Struct. (WADS)*, LNCS 955, pages 183–193, 1995.

17 J. S. B. Mitchell and E.L. Wynters. Finding optimal bipartitions of points and polygons. In *Proc. 2nd Workshop Alg. Data Struct. (WADS)*, LNCS 519, pages 202–213, 1991. Full version available at `http://www.ams.sunysb.edu/~jsbm/`.

18 J. Rokne, S. Wang, and X. Wu. Optimal bipartitions of point sets. In *Proc. 4th Canad. Conf. Comput. Geom. (CCCG)*, pages 11–16, 1992.

19 M. Segal. Lower bounds for covering problems. *J. Math. Modelling Alg.* 1(1):17–29 (2002).

20 M. Sharir. A near-linear algorithm for the planar 2-center problem. *Discr. Comput. Geom.* 18(2):125–134 (1997).