Computing Instance-Optimal Kernels in Two Dimensions

Pankaj K. Agarwal
Department of Computer Science, Duke University, Durham, NC, USA

Sariel Har-Peled
Department of Computer Science, University of Illinois, Urbana, IL, USA

Abstract

Let P be a set of n points in $\mathbb{R}^2$. For a parameter $\varepsilon \in (0, 1)$, a subset $C \subseteq P$ is an ε-*kernel* of P if the projection of the convex hull of C approximates that of P within $(1-\varepsilon)$-factor in every direction. The set C is a *weak* ε-*kernel* of P if its directional width approximates that of P in every direction. Let $\mathsf{k}_\varepsilon(P)$ (resp. $\mathsf{k}^{\mathsf{w}}_\varepsilon(P)$) denote the minimum-size of an ε-kernel (resp. weak ε-kernel) of P. We present an $O(n\mathsf{k}_\varepsilon(P) \log n)$-time algorithm for computing an ε-kernel of P of size $\mathsf{k}_\varepsilon(P)$, and an $O(n^2 \log n)$-time algorithm for computing a weak ε-kernel of P of size $\mathsf{k}^{\mathsf{w}}_\varepsilon(P)$. We also present a fast algorithm for the Hausdorff variant of this problem.

In addition, we introduce the notion of ε-*core*, a convex polygon lying inside $\mathsf{ch}(P)$, prove that it is a good approximation of the optimal ε-kernel, present an efficient algorithm for computing it, and use it to compute an ε-kernel of small size.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Coreset, approximation, kernel

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.4

Related Version *Full Version*: `https://arxiv.org/abs/2207.07211` [1]

Funding *Pankaj K. Agarwal*: Work on this paper was partially supported by NSF grants IIS-18-14493, CCF-20-07556, and CCF-2223870.
Sariel Har-Peled: Work on this paper was partially supported by a NSF AF award CCF-1907400.

1 Introduction

Coresets have been successfully used as geometric summaries to develop fast approximation algorithms for a wide range of geometric optimization problems. Agarwal et al. [2] introduced the notions of ε-kernels/coresets for approximating the convex hull of a point set P in $\mathbb{R}^d$: For an interval $J = [a, b]$, let $(1-\varepsilon)J = [a+(\varepsilon/2)\,|J|\,, b-(\varepsilon/2)\,|J|]$ be its scaling down by a factor of $1-\varepsilon$ around its center. For a direction $v \in \mathbb{S}$, let $I_v(P)$ denote the projection of $\mathsf{ch}(P)$ in direction v, which is an interval. A subset $C \subseteq P$ is an ε-kernel if $I_v(C) \supseteq (1-\varepsilon)I_v(P)$ for all directions $v \in \mathbb{S}$, see Definition 4. The *weak* ε-*kernels* impose a weaker requirement that $|I_v(C)| \geqslant (1-\varepsilon)\,|I_v(P)|$ for all $v \in \mathbb{S}$, see Definition 6. See Figure 1.

It is known that there exists an ε-kernel (as well as a weak ε-kernel) of P of size $O(\varepsilon^{-(d-1)/2})$ and that it can be computed efficiently [2]. However there may exist an ε-kernel of P of much smaller size, as is often the case in practice, see, e.g. [23]. Let $\mathsf{k}_\varepsilon(P)$ be the minimum size of an ε-kernel of P. An interesting question is whether an ε-kernel of P of size k_ε can be computed efficiently, i.e., computing an *instance-optimal* ε-kernel. A similar question can be asked for weak ε-kernels. These problems are known to be NP-Hard for $d \geqslant 3$. Although it is generally believed that an instance-optimal ε-kernel or weak ε-kernel in the plane can be computed in polynomial time using dynamic programming, we are unaware of any paper that presents such an algorithm. See below for related work on this problem. In this paper, we settle this question by presenting fast algorithms for computing instance-optimal ε-kernels and weak ε-kernels for $d = 2$.

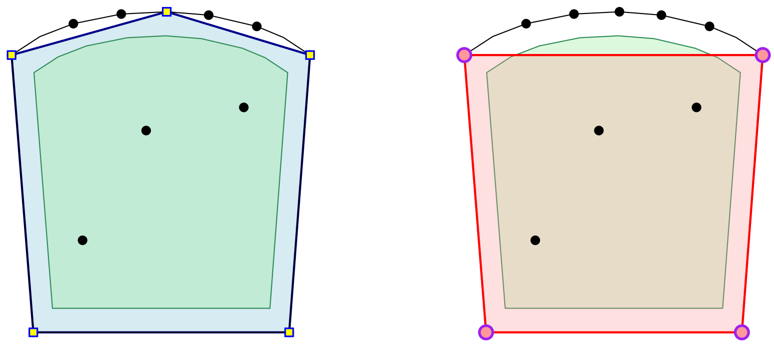


Figure 1 Somewhat oversimplifying the difference, a regular kernel has to conceptually include a "shrunken" middle portion (left), while the weak kernel (right) only has to approximate the projections. Specifically, on the left, the projection interval of the approximation has to include the projection interval of the green region. On the right, the approximation projection interval needs to be sufficiently long but it does not have the inclusion constraint.

Related work. As mentioned above, Agarwal et al. [2] proved the existence of an ε-kernel of size $O(\varepsilon^{-(d-1)/2})$ for any set of points in $\mathbb{R}^d$ and presented fast algorithms for computing such an ε-kernel. These algorithms were subsequently improved and generalized, see [9, 5, 3]. Yu et al. [23] studied practical algorithms for computing coresets/kernels, and suggested an incremental algorithm that seems to provide a good approximation to the optimal kernel.

The NP-Hardness of computing an instance-optimal kernel in $\mathbb{R}^3$ follows from that of polytope approximation [12], see also [4, 8]. Clarkson [11] studied the problem of polytope approximation as a hitting-set problem, providing a logarithmic approximation in the optimal size, that can be used for approximating the optimal kernel. For $d = 3$, the approximation factor can be improved to $O(1)$ [7]. Using a greedy approach, Blum et al. [6] studied the problem of approximating optimal kernels in high dimensions, and presented polynomial-time algorithms for computing an ε-kernel of size $O(d\mathsf{k}_\varepsilon \log \mathsf{k}_\varepsilon)$ or an $(\varepsilon + 8\varepsilon^{1/3})$-kernel of size $O(\mathsf{k}_\varepsilon \varepsilon^{-2/3})$.

More recently, there has been some work on computing variants of ε-kernels of minimum size, though *none* of them compute an instance-optimal ε-kernel. Wang et al. [22] use a different definition of kernel, so comparing the results of this paper to their work is somewhat confusing. Specifically, Wang et al. [22] presented a cubic-time algorithm that computes a minimum-size subset Q of P with the property that $\max_{p\in P}(1-\varepsilon)\langle v, p\rangle \leqslant \max_{q\in Q}\langle v, q\rangle$, assuming that P is α-fat for some constant α; they refer to such a subset as a ε-core-set of P. A shortcoming of this definition is that it is neither translation nor non-uniform-scaling invariant. However, it can be shown that their algorithm computes an ε-kernel of size at most $\mathsf{k}_{\varepsilon/3}$ (observe that $\mathsf{k}_{\varepsilon/3}$ can be much larger than k_ε). Klimenko and Raichel [16] provided an $O(n^{2.53})$ time algorithm for computing a minimum-size subset Q such that $H(\mathsf{ch}(P), \mathsf{ch}(Q))$, the Hausdorff distance between $\mathsf{ch}(P)$ and $\mathsf{ch}(Q)$, is at most ε.[1] They also tackle the case when P is convex, which they solve in $O(n \log^2 n)$ time. The standard approach for computing small kernels, is to apply an affine transformation to the point set to make it "fat", then apply an algorithm for Hausdorff approximation, with parameter ε/c where c depends on the fatness of the mapped point set and its diameter. Using the algorithm

[1] Recall that for two sets $A, B \in \mathbb{R}^2$, $H(A,B) = \max\{h(A,B), h(B,A)\}$, where $h(X,Y) = \max_{x\in X}\min_{y\in Y}\|x-y\|$.

in [16], an ε-kernel of size at most $\mathsf{k}_{\varepsilon/2}$ can be computed in $O(n^{2.53})$ time. We note that since ε is an absolute error, the size of Hausdorff-approximation can be $\Omega(n)$ in the worst case. If we set the error parameter to be $\varepsilon \cdot \mathrm{diam}(P)$, then there exists an ε-Hausdorff approximation Q of size $O(\varepsilon^{-(d-1)/2})$ but Q may not be an ε-kernel since for a direction $v \in \mathbb{S}$, $|I_v(Q)|$ maybe as small as $|I_v(P)| - \varepsilon\mathrm{diam}(P)$, while ε-kernel requires $I_v(Q) \supseteq (1-\varepsilon)I_v(P)$. As such while the width or minimum-enclosing-box of an ε-kernel approximates that of P, a Hausdorff approximation does not offer such a guarantee and thus not always suitable for approximating extent measures of P.

There is also some connection between our problem and minimum-link distance and polygon approximation, see [13, 14, 18, 19, 21, 20] for some relevant results.

Our results. Let P be a set of n points in $\mathbb{R}^2$, and let $\varepsilon > 0$ be a parameter. There are three main results in this paper:

- Optimal kernel. We present (in Section 4) an $O(\mathsf{k}_\varepsilon n \log n)$-time algorithm for computing an ε-kernel of P of size $\mathsf{k}_\varepsilon := \mathsf{k}_\varepsilon(P)$; recall that $\mathsf{k}_\varepsilon = O(\varepsilon^{-1/2})$.
- Optimal weak kernel. We present (in Section 5) an $O(n^2 \log n)$-time algorithm for computing a weak ε-kernel of P of size $\mathsf{k}^{\mathsf{w}}_\varepsilon(P)$, the minimum size of a weak ε-kernel of P.

Our algorithm for computing the optimal kernel can be adapted to computing an optimal Hausdorff approximation of $\mathsf{ch}(P)$:

- Optimal Hausdorff approximation. We present (in [1]) an $O(\mathsf{k}^{\mathsf{h}}_\varepsilon n \log n)$-time algorithm for computing a set $Q \subseteq P$ of size $\mathsf{k}^{\mathsf{h}}_\varepsilon$ such that $H(\mathsf{ch}(P), \mathsf{ch}(Q)) \leqslant \varepsilon$, where $\mathsf{k}^{\mathsf{h}}_\varepsilon$ is the size of the minimum such subset.

We obtain these results by reducing the computation of (weak) optimal kernel to the following two covering problems, which are of independent interest:

- Optimal arc cover. Given a set Ξ of n arcs of the unit circle $\mathbb{S}$, compute its smallest subset that covers $\mathbb{S}$. Lee and Lee [17] had presented an $O(n \log n)$-time algorithm for this problem, which is optimal in the worst case. Here we present a somewhat simpler algorithm with the same running time (see [1]), which is more intuitive and which we adapt to the computation of weak kernels.
- Optimal star cover. Given a polygon $\mathcal{P}$ that is star shaped with respect to the origin o and a set of lines L, compute a smallest subset of lines (i.e., cuts) in L that separate o from $\partial\mathcal{P}$. Alternatively, this can be interpreted as covering $\partial\mathcal{P}$ by the (outer) halfplanes defined by the lines of L. We reduce this problem to the above arc-cover problem, but the number of candidate arcs can be quadratic. We use a greedy algorithm to prune the number of candidate arcs to $O(\mathsf{k}n)$, in $O(\mathsf{k}n \log n)$ time, where k is the size of the optimal solution, and then compute an arc cover in $O(\mathsf{k}n \log n)$ time using the above algorithm. We reduce the computation of ε-kernel to this covering problem by using the polarity transform (see Section 4)

Finally, we introduce (in [1]) the concept of core of a point set, prove its properties, and describe an algorithm for computing it. A convex body C can be represented as the intersection of all the minimal slabs that contains it. The *ε-core* is the result of intersecting all these slabs after shrinking them by a factor of $1-\varepsilon$. It induces an affine-invariant inner approximation of C. For a point set P, its ε-core is a convex polygon lying inside $\mathsf{ch}(P)$. We describe an $O(n \log n)$-time algorithm for computing the ε-core of P.

We show that the convex hull of any ε-kernel of P contains the ε-core of P, and that any subset $C \subseteq P$ whose convex hull contains the ε-core is a 4ε-kernel of P, see [1]. Thus the ε-core is an approximation to the optimal ε-kernel, which has the benefit of being well defined for any bounded convex shape. We believe this notion of ε-core is new, and is of independent interest. We present an $O(n \log n)$-time algorithm for computing the smallest subset of P such that its convex-hull contains the $\varepsilon/4$-core of P, which yields an ε-kernel of P of size at most $\mathsf{k}_{\varepsilon/4}$.

2 Preliminaries

Let P be a set of n points in $\mathbb{R}^2$, and let $\varepsilon \in (0, 1)$ be a parameter. Without loss of generality assume that the origin o lies in the interior of $\mathsf{ch}(P)$, where $\mathsf{ch}(P)$ denotes the convex-hull of P (if $\mathsf{o} \notin \mathsf{ch}(P)$, one can choose three arbitrary points of P and translate P so that their centroid becomes o).

Normal diagram. A direction in $\mathbb{R}^2$ can be represented as a unit vector in $\mathbb{R}^2$. The set of unit vectors (directions) in $\mathbb{R}^2$ is denoted by $\mathbb{S} = \{p \in \mathbb{R}^2 \mid \|p\| = 1\}$.

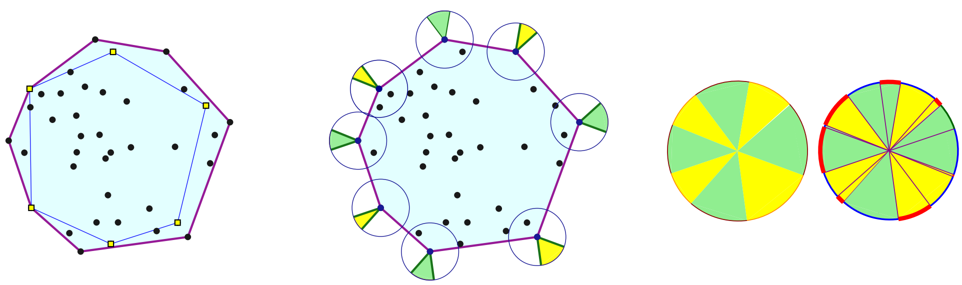
▶ **Definition 1.** *For a line ℓ not passing through the origin, let $\hbar = \hbar(\ell)$ (resp. $\overline{\hbar} = \overline{\hbar}(\ell)$) be the (closed) halfplane bounded by ℓ and containing (resp. not containing) the origin.*

For a direction $v \in \mathbb{S}$ and a point $q \in \mathbb{R}^2$, let $\hbar_v(q)$ be the halfplane that is bounded by the line normal to direction v and passing through q, and that contains o.

▶ **Definition 2** (Extremal point, supporting line)**.** *For a direction $v \in \mathbb{S}$, let p_v be the **extremal point** of P in the direction v. That is $p_v = \arg\max_{p \in P} \langle v, p \rangle$. The point p_v is unique if v is not the outer normal of an edge of $\mathsf{ch}(P)$. Similarly, let ℓ_v be the* supporting line *of $\mathsf{ch}(P)$ normal to v and passing through p_v. Let $\hbar_v = \hbar(\ell_v)$ and $\overline{\hbar}_v = \overline{\hbar}(\ell_v)$. Observe that $\mathsf{ch}(P) \subset \hbar_v$.*

For a real number ψ, let $\hbar_v \ominus \psi$ and $\overline{\hbar}_v \ominus \psi$ be the halfplanes formed by translating $\hbar_v$ and $\overline{\hbar}_v$, respectively, towards the origin by distance ψ.

▶ **Definition 3.** *The **normal diagram** of P is the partition of $\mathbb{S}$ into maximal intervals so that the extremal point p_v remains the same for all directions within an interval. The endpoints of these intervals correspond to the outer normals of the edges of $\mathsf{ch}(P)$. The normal diagram can be further refined so that for all directions v within each interval, both p_v and p_{-v} remain the same. Such a pair of points are* antipodal pairs*. Let $\mathcal{N} = \mathcal{N}(P)$ denote this **refinement** of the normal diagram, and observe that $|\mathcal{N}| \leqslant 2n$. See Figure 2.*



(A) (B) (C)

Figure 2 (A) Point set P, $\mathsf{ch}(P)$, and ε-kernel of P (say, for $\varepsilon = 0.2$). (B) Directions in which a point is extremal. (C) Normal diagram of P and its refinement $\mathcal{N}(P)$, I_v.

Directional width and ε-kernel. For a direction $v \in \mathbb{S}$, let

$$I_v(P) = \left[\min_{p \in P} \langle v, p \rangle, \max_{p \in P} \langle v, p \rangle\right]$$

denote the **projection interval** of P in direction v. Its length $\overline{\omega}(v, P) = \|I_v\|$ is the **directional width** of P in the direction of v. Note that $I_v = -I_{-v}$ and $\overline{\omega}(v, P) = \overline{\omega}(-v, P)$. For an $\varepsilon \in (0, 1)$ and an interval $J = [x, y]$, let $(1 - \varepsilon)J$ be the shrinking of J by a factor of $(1 - \varepsilon)$, i.e., $(1 - \varepsilon)J = [x + (\varepsilon/2)\,|J|\,, y - (\varepsilon/2)\,|J|]$.

▶ **Definition 4.** *A set $X \subseteq \mathsf{ch}(P)$ is an **ε-approximation** of P if $I_u(X) \supseteq (1 - \varepsilon)I_u(P)$ for all directions $u \in \mathbb{S}$. A subset $C \subseteq P$ is a "strong" **ε-kernel** of P if it is an ε-approximation of P. Let $\mathsf{k}_\varepsilon(P)$ denote the minimum size of an ε-kernel of P. See Figure 3 for an example.*

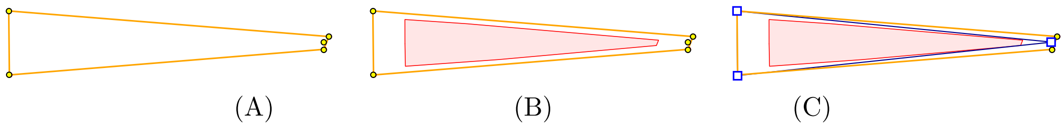


Figure 3 (A) A point set and its convex hull. (B) Its 0.2-core. (C) Its optimal 0.2-kernel – observe that it contains points that are not on the convex-hull.

We emphasize that the shrinking here is done for every direction individually around the center of the projection interval – in particular, there is no center point of the $\mathsf{ch}(P)$ around which we do the scaling – to some extent this gives rise to most of the technical difficulties in constructing and approximating an optimal kernel. The following property of ε-approximation will be useful later on.

▶ **Lemma 5** ([2]). *Let P be a point set in $\mathbb{R}^d$, $X \subseteq \mathsf{ch}(P)$, and T an affine map in $\mathbb{R}^d$. X is an ε-approximation for $P \iff T(X)$ is an ε-approximation of $T(P)$.*

A slightly weaker notion of ε-kernel was used by Agarwal et al. [2], that is potentially (significantly) smaller than their "strong" counterparts but somewhat harder to compute.

▶ **Definition 6.** *A subset $C \subseteq P$ is a **weak ε-kernel** of P if $\overline{\omega}(u, C) \geqslant (1 - \varepsilon)\overline{\omega}(u, P)$ for all $u \in \mathbb{S}$.*

This weaker definition was sufficient for the purposes of Agarwal et al.. However, it is less intuitive than the stronger variant, and it is harder to compute the optimal weak kernel.

Computing optimal circular arc cover. Let Ξ denote a set of n circular arcs on $\mathbb{S}$, each of length less than π, that cover $\mathbb{S}$.[2] As mentioned in the introduction, an $O(n \log n)$-time algorithm for computing the smallest subset of Ξ that cover $\mathbb{S}$ was proposed in [17]. In the full version [1] we present an alternative $O(n \log n)$ time algorithm for computing the smallest-size arc cover from Ξ, which we believe is simpler and more intuitive. The basic idea is to use the greedy algorithm. Picking a start arc, and then going counterclockwise as far one can adding arcs in a greedy fashion results in a cover of size $\mathsf{k} + 1$, where k is the optimal size. After an $O(n \log n)$ preprocessing, the greedy algorithm can be executed in $O(\mathsf{k})$ time. To reduce the size of the solution to k, one has to guess a starting arc that belongs to the optimal solution. We show that the least covered point on the circle is covered

[2] By computing the union of arcs in Ξ, we can decide, in $O(n \log n)$ time, whether Ξ covers $\mathbb{S}$.

by $O(n/\mathsf{k})$ intervals. This implies that one has to try only $O(n/\mathsf{k})$ starting arcs and thus run the greedy algorithm $O(n/\mathsf{k})$ times. The overall running time is thus $O(n \log n)$. See [1] for full details. We will use this algorithm as a subroutine in Section 4 and [1] and a variant of it in Section 5. In particular, we get the following result:

▶ **Theorem 7.** *Let Ξ be a set of n circular arcs on $\mathbb{S}$. The optimal cover of $\mathbb{S}$ by the arcs of Ξ, if there exists one, can be computed in $O(n \log n)$ time.*

3 Covering a Star Polygon by Halfplanes

The input is a set of L of n lines and a polygon $\mathcal{Z}$ with $O(n)$ vertices that is star-shaped with respect to the origin o (i.e., for every point $p \in \mathcal{Z}$, $\mathsf{o}p \subseteq \mathcal{Z}$). Formally, the task at hand is to compute a minimum set of lines $C \subseteq L$, such that for any point $p \in \partial\mathcal{Z}$, $\mathrm{int}(\mathsf{o}p)$ intersects a line of C. Geometrically, $\mathsf{F}_{\mathsf{o}}(C) := \bigcap_{\ell \in C} \hbar(\ell)$, the intersection of inner halfplanes bounded by lines in C, is contained in $\mathcal{Z}$. An alternative interpretation of this problem is that $\partial\mathcal{Z} \subset \bigcup_{\ell \in C} \overline{\hbar}(\ell)$.

3.1 Reduction to arc cover

$\partial\mathcal{Z}$ can be viewed as the image of a function $\mathcal{Z} : \mathbb{S} \to \mathbb{R}^2$. Specifically, for a direction $u \in \mathbb{S}$, $\mathcal{Z}(u)$ is the intersection point of $\partial\mathcal{Z}$ with the ray from the origin in direction u. A line ℓ **blocks** the direction u if ℓ intersects the segment $\mathsf{o}\mathcal{Z}(u)$. A subset $G \subseteq L$ is a **blocking set** of $\mathcal{Z}$ if each direction in $\mathbb{S}$ is blocked by at least one line of G (i.e., $\mathsf{F}_{\mathsf{o}}(G) \subset \mathcal{Z}$).

Fix a line $\ell \in L$. Let $\ell \sqcap \mathcal{Z}$ denote the set of connected components (i.e., segments) of $\ell \cap \mathcal{Z}$. For a segment $s \in \ell \sqcap \mathcal{Z}$, let $\sphericalangle s = \{\mathsf{o}p/\|\mathsf{o}p\| \in \mathbb{S} \mid p \in s\}$ be the circular arc induced by s. All directions in $\sphericalangle s$ are blocked by ℓ. Let $\sphericalangle\ell = \{\sphericalangle s \mid s \in \ell \sqcap \mathcal{Z}\}$ be the set of all circular arcs that are induced by blocking segments of ℓ. Let $\Xi = \bigcup_{\ell \in L} \sphericalangle\ell$ be the set of all circular arcs defined by the lines of L. For a subset $\Gamma \subseteq \Xi$, let $L(\Gamma) = \{\ell \in L \mid \gamma \in \sphericalangle\ell, \gamma \in \Gamma\}$ be the original subset of lines of L supporting the arcs of Γ.

▶ **Lemma 8.** *(i) If $\Gamma \subseteq \Xi$ is an arc cover, i.e., $\bigcup \Gamma = \mathbb{S}$, then $L(\Gamma)$ is a blocking set.*

(ii) There is an arc cover $\Gamma \subseteq \Xi$ of size k if and only if there is a blocking set $G \subseteq L$ of size k.

Proof. (i) If Γ is an arc cover, then for every direction $u \in \mathbb{S}$, there is an arc $\sphericalangle s \in \Gamma$ that blocks the direction u. If $\sphericalangle s \in \sphericalangle\ell$, for a line $\ell \in L(\Gamma)$, then the segment $\mathsf{o}\mathcal{Z}(u)$ intersects ℓ. Since this condition holds for all directions in $\mathbb{S}$, it follows $L(\Gamma) \subseteq L$ is a blocking set.

(ii) If there is an arc cover $\Gamma \subseteq \Xi$ of size k, then by part (i), $L(\Gamma)$ is a blocking set of size at most k. Conversely, let G be a blocking set for $\mathcal{Z}$. Without loss of generality, we can assume that each line of G appears as an edge on the boundary of the face F of $\mathcal{A}(G)$ that contains the origin, because otherwise we can remove the line from G. For each line $\ell \in G$, let $s_\ell \in \ell \sqcap \mathcal{Z}$ be the segment that contains the edge of F lying on ℓ. Since $\mathsf{F} \subseteq \mathcal{Z}$, the segment $\mathsf{o}\mathcal{Z}(u)$ intersects an edge of F for every $u \in \mathbb{S}$. Hence, $\{\sphericalangle s_\ell \mid \ell \in G\}$ is an arc cover of size at most $|G|$. ◀

By Lemma 8, it suffices to compute smallest-size arc cover from Ξ. But $|\Xi| = \Theta(n^2)$ in the worst case. Therefore computing Ξ explicitly and then using Theorem 7 to compute an arc cover take $O(n^2 \log n)$ time. In the following, we show how to improve the running time to $O(n\mathsf{k} \log n)$, where k is the optimal solution size.

3.2 Computing an almost-optimal blocking set

We extend the greedy algorithm used in the circular arc cover (see Section 2 and [1]) to compute an arc cover in Ξ without computing Ξ explicitly. For clarity, we describe the greedy algorithm in terms of computing a blocking set.

For a pair of directions $u, v \in \mathbb{S}$, let $\mathcal{Z}(u, v] \subseteq \mathcal{Z}$ be the semiopen subchain of $\mathcal{Z}$ from $\mathcal{Z}(u)$ to $\mathcal{Z}(v)$ in the counterclockwise direction, which contains the endpoint $\mathcal{Z}(v)$ but not $\mathcal{Z}(u)$. As such, we have $\mathcal{Z}(u, u] = \mathcal{Z}$.

We define a (partial) function $\mathfrak{s} : \mathbb{S} \times L \to \mathbb{R}^4$, as follows. For a pair $u \in \mathbb{S}$ and a line $\ell \in L$, if ℓ does not intersect the segment $\mathsf{o}\mathcal{Z}(u)$, then $\mathfrak{s}(u, \ell)$ is not defined. Otherwise, it is the segment of $\ell \sqcap \mathcal{Z}$ that intersects $\mathsf{o}\mathcal{Z}(u)$. Similarly, we define a (partial) function $\mathfrak{f} : \mathbb{S} \times L \to \mathbb{S}$, that is the first point of $\mathfrak{s}(u, \ell)$ in the counter-clockwise direction after $\mathcal{Z}(u)$ (note, that ℓ might intersect the boundary $\mathcal{Z}$ many times). Set $\lambda(u) = \arg\max_{\ell \in L} \mathfrak{f}(u, \ell)$, i.e., among the feasible segments that intersect $\mathsf{o}\mathcal{Z}(u)$, $\lambda(u)$ is the last one to exit $\mathcal{Z}$ in the counterclockwise direction.

The algorithm consists of the following steps: Set $v_0 := (1, 0)$, $\ell_0 := \lambda(v_0)$, $G := \{\ell_0\}$, and $i := 1$. In the ith iteration, the algorithm does the following: it sets $v_i = \mathfrak{f}(v_{i-1}, \ell_{i-1})$, $\ell_i = \lambda(v_i)$, and $G = G \cup \{\ell_i\}$. The algorithm then continues to the next iteration till $\mathsf{F}_{\mathsf{o}}(G) \subseteq \mathrm{int}(\mathcal{Z})$. Let v'_1 be the first intersection point of ℓ_0 with $\mathcal{Z}$ in the clockwise direction from v_0, i.e., the segment $\mathcal{Z}(v_1)\mathcal{Z}(v'_1)$ lies inside $\mathcal{Z}$. Then the terminating condition is the same as $\mathfrak{f}(v_i, \ell_i)$ lying after v'_1 (from v_i) in the counterclockwise direction. By construction, $\mathsf{F}_{\mathsf{o}}(G) \subset \mathcal{Z}$. Since this is a greedy algorithm for computing an arc cover, $|G| \leqslant \mathsf{k} + 1$. The polygon $\mathcal{Z}$ can be preprocessed, in $O(n \log n)$ time, into a data structure of linear size so that for a pair $u \in \mathbb{S}$ and a line $\ell \in L$, $\mathfrak{f}(u, \ell)$ can be computed in $O(\log n)$ time [10, 15]. The algorithm performs $O(n\mathsf{k})$ such queries, so the total running time is $O(n\mathsf{k} \log n)$.

▶ **Lemma 9.** *Let L be a set of n lines, $\mathcal{Z}$ be a polygon with $O(n)$ vertices that is star shaped with respect to o and that contains $\mathsf{F}_{\mathsf{o}}(L)$, and let k be the size of the smallest blocking set in L for $\mathcal{Z}$. A blocking set $G \subseteq L$ of size at most $\mathsf{k} + 1$ can be computed in $O(\mathsf{k} n \log n)$ time.*

3.3 Computing an optimal solution

Let G be the blocking set computed by the above greedy algorithm. For each line $\ell \in L$ we compute its intersection points with the lines of G. For each such intersection point ξ, if ξ lies inside $\mathcal{Z}$, let $s_\xi \in \mathcal{P} \sqcap \ell$ be the segment that contains ξ. Let S_1 be the set of resulting $O(n\mathsf{k})$ segments. Let

$$S_2 = \bigcup_{\ell \in G} \ell \sqcap \mathcal{P}$$

be the set of all segments induced by the lines of G. Set $S = S_1 \cup S_2$. The computes the set $\Gamma = \{\measuredangle s \mid s \in S\}$, and then computes the minimal size arc cover C of $\mathbb{S}$ by the arcs of Γ. The returned set is $K = \{\ell \in \ell \mid \measuredangle s \in C, s \subset \ell\}$.

▶ **Lemma 10.** *The set Γ contains an arc cover of size k.*

Proof. Suppose for the sake of contradiction that Γ does not contain an arc cover of size k. Let $C \subset \Xi$ be an arc cover of size k. Then C contains an arc $\measuredangle s$ such that s lies on a line of $L \backslash G$ and s does not intersect any line of G, i.e., it lies in the interior of a face of $\mathcal{A}(G)$, the arrangement of G.

If s lies in the face corresponding to $\mathsf{F}_{\mathsf{o}}(G)$, then s must intersect $\partial \mathsf{F}_{\mathsf{o}}(G)$, as the endpoints of s lies on $\partial \mathcal{Z}$ and $\mathcal{I}(G) \subseteq \mathcal{Z}$, contradicting the assumption that s does not intersect any line of G.

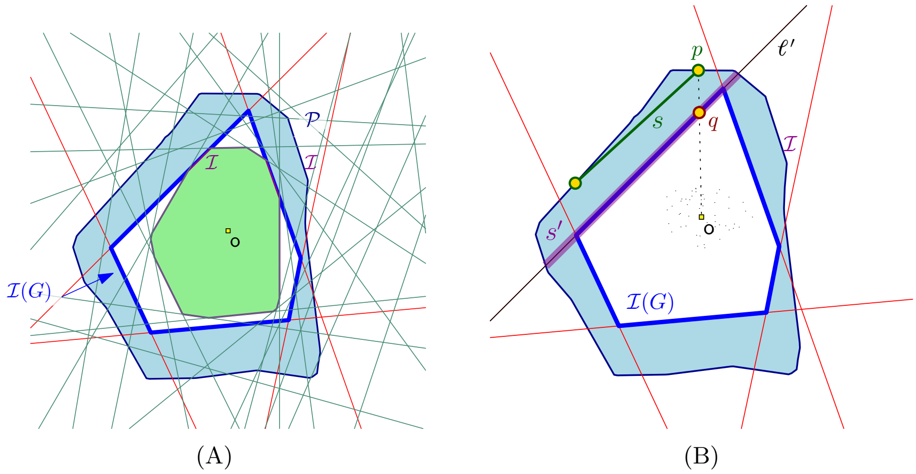


Figure 4 (A) A hitting set G of size at most $\mathsf{k}+1$. (B) Illustration of the proof of Lemma 10.

Next, suppose s lies in some other face of $\mathcal{A}(G)$. Let p be an endpoint of s. The segment $p\mathsf{o}$ must intersect a line $\ell' \in G$ at a point q. In particular, let $s' \in \ell' \cap \mathcal{Z}$ be the segment of ℓ containing q. Clearly, s' is a blocker for all the points on s, so we can obtain another optimal solution by replacing $\measuredangle s$ with $\measuredangle s'$ (see Figure 4), and this solution has one more arc of Γ, a contradiction.

Hence, we can conclude that Γ contains an optimal arc cover. ◀

Computing the set G takes $O(n\mathsf{k}\log n)$ time. Observe that $|S_1| = O(n\mathsf{k})$, as each line of L induces at most $\mathsf{k}+1$ segments in this set. Similarly, as $|G| = \mathsf{k}+1$, we have that $|S_2| = O(n\mathsf{k})$. It follows that computing S_1 and S_2 requires $O(n\mathsf{k})$ ray-shooting queries in $\mathcal{Z}$, and these queries overall take $O(n\mathsf{k}\log n)$ time. Hence, we obtain the following:

▶ **Lemma 11.** *Let L be a set of n lines in the plane, and let $\mathcal{Z}$ be a polygon with $O(n)$ vertices that is star shaped with respect to o and that contains $\mathsf{F}_{\mathsf{o}}(L)$. Then a blocking set from L of $\mathcal{Z}$ of size k can be computed, in $O(\mathsf{k}n\log n)$ time, where k is the size of the optimal solution.*

4 Computing Optimal ε-Kernel

Let P be a set of n points in $\mathbb{R}^2$ and $\varepsilon \in (0,1)$ a parameter. We describe an $O(n\mathsf{k}_\varepsilon \log n)$-time algorithm for computing an ε-kernel of size k_ε. We use polarity to construct a set L of n lines and a star polygon $\mathcal{Z}$ that contains $\mathsf{F}_{\mathsf{o}}(L) = \bigcap_{\ell \in C} \hbar(\ell)$. An ε-kernel of P corresponds to a blocking set in L for $\mathcal{Z}$.

▶ **Definition 12** (ε-shifted supporting line)**.** *For a direction $u \in \mathbb{S}$ and a parameter $\varepsilon > 0$, let $\ell_{u,\varepsilon}$ be the boundary line of $\hbar_{u,\varepsilon} = \hbar_u \ominus (\varepsilon/2)\overline{\omega}(u,P)$, see Definition 2. Let $\overline{\hbar}_{u,\varepsilon}$ be the (closed) complement halfplane to $\hbar_{u,\varepsilon}$.*

Set $\mathcal{H}_\varepsilon = \{\overline{\hbar}_{u,\varepsilon} \mid u \in \mathbb{S}\}$. The following lemma is immediate from the definition of ε-kernel.

▶ **Lemma 13.** *Given a point set P in $\mathbb{R}^2$ and a parameter $\varepsilon \in (0,1)$, a subset $C \subseteq P$ is an ε-kernel of P if and only if $\overline{\hbar}_{v,\varepsilon} \cap C \neq \emptyset$ for all $u \in \mathbb{S}$, i.e., C is a hitting set of $\mathcal{H}_\varepsilon$.*

The problem of computing an ε-kernel thus reduces to computing a minimum-size hitting set of the infinite set $\mathcal{H}_\varepsilon$. It will be convenient to use the polarity transform and work in the mapped plane, so we first describe the polar of ε-kernel and then describe the algorithm.

Polarity. For a point $p \neq \mathsf{o}$, its **inversion**, through the unit circle, is the point $p^{-1} = p/\|p\|^2$. Observe that p, p^{-1}, o are collinear, $\|p\|\,\|p^{-1}\| = 1$, and p and p^{-1} are on the same side of the origin on this line. We use the *polarity transform*, which maps a point $p = (a, b) \neq \mathsf{o}$ to the line

$$p^{\odot} \equiv ax + by - 1 = 0 \equiv \langle p, (x, y) \rangle - 1 = 0 \equiv \left\langle p, (x, y) - \frac{p}{\|p\|^2} \right\rangle = 0.$$

Namely, the line $p^{\odot}$ is orthogonal to the vector $\mathsf{o}p$, and the closest point on $p^{\odot}$ to the origin is p^{-1}. Geometrically, a point p is being mapped to the line passing through the inverted point p^{-1} and orthogonal to the vector $\mathsf{o}p^{-1}$. Similarly, for a line ℓ, its *polar* point $\ell^{\odot}$ is q^{-1}, where q is the closest point to the origin on ℓ. Observe that $(\ell^{\odot})^{\odot} = \ell$ and $(p^{\odot})^{\odot} = p$ for any line ℓ and any point p.

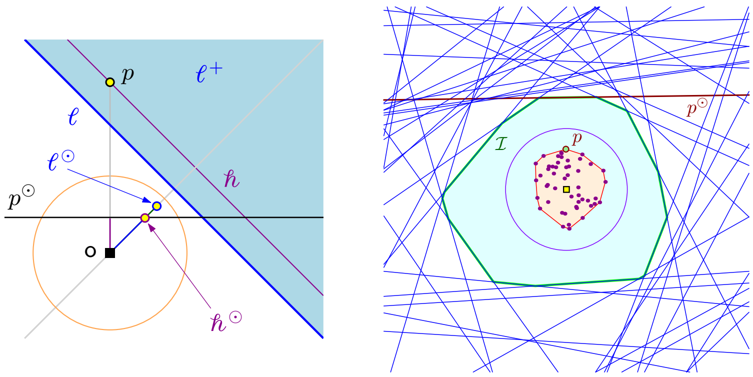


Figure 5 Left: A point p lies in the halfplane $\overline{\hbar}(\ell) \iff p^{\odot}$ intersects the segment $\mathsf{o}\ell^{\odot}$. Right: A convex hull of a point set, and the corresponding "polar" polygon formed by the intersection of halfplanes.

If a point p lies on a line ℓ then $\ell^{\odot} \in p^{\odot}$. If p lies in the halfplane $\overline{\hbar}(\ell)$ (by Definition 1, we have $\mathsf{o} \notin \overline{\hbar}(\ell)$) if and only if $p^{\odot}$ intersects the segment $\mathsf{o}\ell^{\odot}$, see Figure 5 (left). Set $P^{\odot} = \{p^{\odot} \mid p \in P\}$ and $\mathsf{F}_{\mathsf{o}} := \mathsf{F}_{\mathsf{o}}(P^{\odot}) = \bigcap_{p \in P} \hbar(p^{\odot})$. Then the polygon F_{o} is the polar of $\mathsf{ch}(P)$, namely:

I. If $p \in P$ is a vertex of $\mathsf{ch}(P)$ then $p^{\odot}$ contains an edge of F_{o}, see Figure 5 (right).

II. The polar of line ℓ missing (resp. intersecting) $\mathsf{ch}(P)$ is a point lying in (resp. out) F_{o}.

III. For a point $p \in \mathsf{ch}(P)$, $\mathsf{F}_{\mathsf{o}} \subset \hbar(p^{\odot})$.

Consider any direction $u \in \mathbb{S}$. Let p_u be the extremal point of P in direction u, and let ℓ_u be the corresponding supporting line, see Definition 2. The point $\ell_u^{\odot}$ lies on the edge of F_{o} supported by $p_u^{\odot}$, and $\ell_u^{\odot}/\|\ell_u^{\odot}\| = u$. Similarly, the polar of the shifted supporting line $\ell_{u,\varepsilon}$ (see Definition 12), is the point $\ell_{u,\varepsilon}^{\odot}$ which lies outside F_{o} on the ray induced by u (starting at the origin).

Kernel and polarity. Returning to ε-kernels, let $\mathcal{N}$ be the refinement of the normal diagram of $\mathsf{ch}(P)$, see Definition 3. Recall that $\mathcal{N}$ is centrally symmetric. The supporting lines ℓ_u and ℓ_{-u} support the same pair of vertices of $\mathsf{ch}(P)$ for all directions u lying inside an interval of $\mathcal{N}$. For each interval $\gamma \in \mathcal{N}$, let $-\gamma$ denote its antipodal interval. For each interval $\gamma \in \mathcal{N}$, let p_γ be the supporting vertex of $\mathsf{ch}(P)$ for all directions in γ.

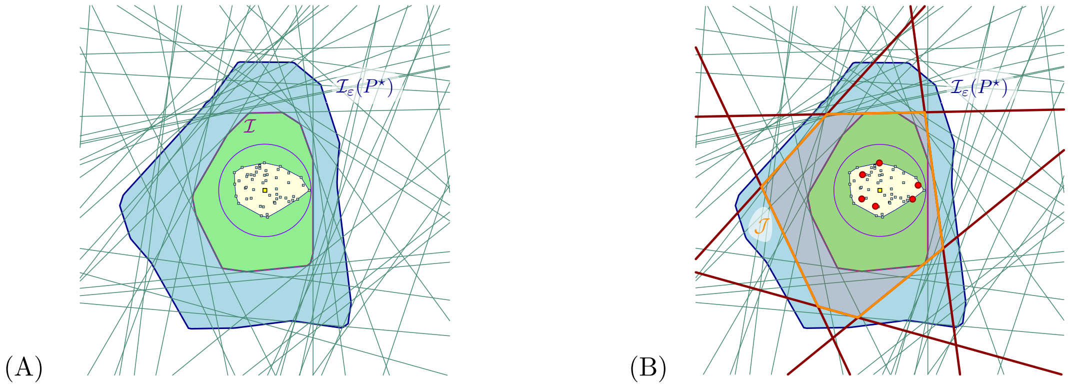


Figure 6 (A) $\mathsf{ch}(P)$, $\mathcal{I}(P^\odot)$, $\mathcal{I}_\varepsilon(P^\odot)$. (B) ε-kernel C and its polar $C^\odot$; $\mathsf{ch}(P) \subseteq \mathcal{I}P^\odot \subseteq \mathcal{I}_\varepsilon(C^\odot)$.

Let $\overline{p}_{\gamma,\varepsilon} = (1-\varepsilon/2)p_\gamma + (\varepsilon/2)p_{-\gamma}$. It can be verified that the line $\ell_{v,\varepsilon}$ for $v \in \gamma$ passes through $\overline{p}_{\gamma,\varepsilon}$. Therefore the polar of the set of lines $\{\ell_{v,\varepsilon} \mid v \in \gamma\}$ is a segment e_γ that lies on the line $(\overline{p}_{\gamma,\varepsilon})^\odot$ and outside $\mathsf{F_o}$. The sequence $\langle e_\gamma \mid \gamma \in \Gamma\rangle$ forms the boundary of a polygon $\mathcal{I}_\varepsilon(P^\odot)$ that is star shaped with respect to o and that contains $\mathsf{F_o}$ in its interior. See Figure 6. Putting everything together, we obtain the following lemma, which characterizes the ε-kernel after polarity.

▶ **Lemma 14.** *Let P be a set of n points in $\mathbb{R}^2$ and $\varepsilon \in (0,1)$ a parameter. The star-shaped polygon $\mathcal{I}_\varepsilon(P^\odot)$ can be computed in $O(n\log n)$ time. Furthermore, a subset $C \subseteq P$ is an ε-kernel of P if and only if $C^\odot$ is a blocking set for $\mathcal{I}_\varepsilon(P^\odot)$ (see Figure 6).*

Computing the smallest set $C \subseteq P$ thus reduces to the star-polygon-cover problem. Using Lemma 11 and that there is an ε-kernel of size $O(\varepsilon^{-1/2})$ [2], we obtain the following:

▶ **Theorem 15.** *Let P be a set of n points in $\mathbb{R}^2$, and let $\varepsilon \in (0,1)$ a parameter. An optimal ε-kernel of P of size k can be computed in $O(\mathsf{k}n\log n)$ time. In the worst case, $\mathsf{k} = O(\varepsilon^{-1/2})$, and the running time is $O(\varepsilon^{-1/2}n\log n)$.*

Below we show that there exists a set P of points such that there are quadratic number of intersections between $P^\odot$ and $\mathcal{I}_\varepsilon(P^\odot)$. This suggest that our somewhat more involved algorithm using greedy algorithm to prune the set of arcs used is necessary even in this case. It will be more convenient to use the duality transform instead of polarity for describing the lower-bound construction.

Duality and ε-kernel. The duality transform provides a similar mapping to polarity. The **dual point** to the line $\ell \equiv y = ax + b$ is the point $\ell^\star = (a, -b)$. Similarly, for a point $p = (c, d)$ its **dual line** is $p^\star \equiv y = cx - d$. Namely, for $p = (a, b)$, the dual line is $p^\star \equiv y = ax - b$, and for a line $\ell \equiv y = c'x + d'$ the dual point is $\ell^\star = (c', -d')$. The following interpretation of kernels in the dual is standard, and goes back to the original work of Agarwal et al. [2]. As such, we state the problem in these settings without proving the equivalence.

For a set of lines $L = P^\star = \{p^\star \mid p \in P\}$ in the plane (i.e., L is a set of affine functions from $\mathbb{R}$ to $\mathbb{R}$), let

$$\Uparrow_L(x) = \max_{f\in L} f(x) \qquad \text{and} \qquad \Downarrow_L(x) = \min_{f\in L} f(x),$$

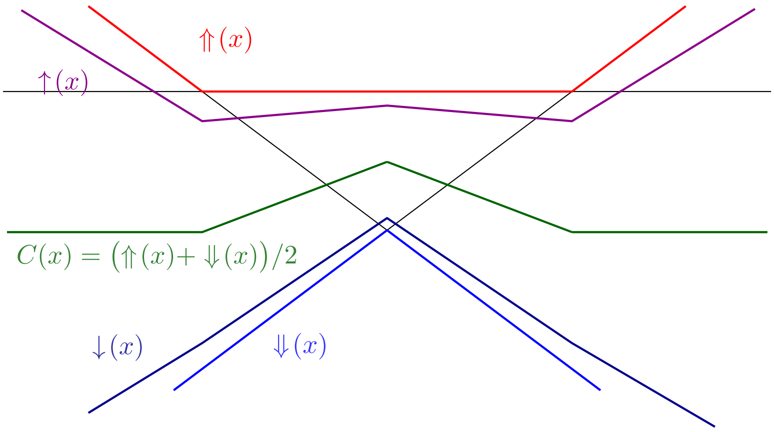


Figure 7 Lower and upper envelopes, and their ε-approximations.

be the upper and lower envelopes of L, respectively. The function $\Uparrow(x)$ is convex, while $\Downarrow(x)$ is concave. The **extent** of L is

$$\Updownarrow_L(x) = \Uparrow_L(x) - \Downarrow_L(x).$$

For a fixed $\varepsilon \in (0,1)$, the *ε-upper envelope* and *ε-lower envelope* are

$$\uparrow_L(x) = \Uparrow_L(x) - \frac{\varepsilon}{2} \Updownarrow_L(x) = \left(1 - \frac{\varepsilon}{2}\right) \Uparrow_L(x) + \frac{\varepsilon}{2} \Downarrow_L(x)$$
$$\downarrow_L(x) = \Downarrow_L(x) + \frac{\varepsilon}{2} \Updownarrow_L(x) = \frac{\varepsilon}{2} \Uparrow_L(x) + \left(1 - \frac{\varepsilon}{2}\right) \Downarrow_L(x),$$

respectively. Unfortunately, these functions are not necessarily convex, as demonstrated in Figure 7.

Computing an optimal ε-kernel for P is equivalent to computing a set of lines $M \subseteq L$, such that $\Uparrow_M(x)$ lies above $\uparrow_L(x)$ (and of course below $\Uparrow_L(x)$), for all x. And similarly, $\Downarrow_M(x)$ lies below $\downarrow_L(x)$, for all x.

Lower-bound construction. Here we show that in the worst case the set $\bigcup_{\ell \in L}(\ell \sqcap \mathcal{P})$ can have quadratic size. In particular, we construct a set of lines L, where the lines of L have quadratic number of intersections with $\uparrow(\cdot)$ and $\downarrow(\cdot)$.

Consider the parabolas $f(x) = \frac{2}{\varepsilon}(x^2+1)$ and $g(x) = -\frac{1}{1-\varepsilon/2}(x^2+1)$. Fix parameters n and ε. Let $p_i = \bigl(i/2n, f(i/2n)\bigr)$ and $q_i = \bigl(i/2n, g(i/2n)\bigr)$, for $i = 0, \ldots, 2n$. For a pair of distinct points $p, q \in \mathbb{R}^2$, let $\ell(p,q)$ denote the line passing through p and q. Let

$$L_f = \{\ell(p_i, p_{i+2}) \mid i = 0, 2, 2n-2\} \qquad \text{and} \qquad L_g = \{\ell(q_i, q_{i+2}) \mid i = 1, 3, 2n-3\}.$$

The upper envelope of L_f in the range $[0,1]$ is above $f(x)$, except for touching it at the points $p_0, p_2, \ldots, p_{2n}$. Similarly, the lower envelope of L_g, in the range $I = [1/2n, 1 - 1/2n]$ lies below g, except for touching it at the points $q_1, q_3, \ldots, q_{2n-1}$.

It is easy to verify that the lines of L_f and L_g do not intersect each other in the range $x \in [0,1]$. As such, the upper envelope (resp. lower envelope) of $L = L_f \cup L_g$ in this range is realized by the upper envelope (resp. lower envelope) of L_f (resp. L_g).

Consider a value $x \in \{1/2n, 3/2n, \ldots, (2n-1)/2n\}$. We have that $\Uparrow_L(x) > f(x)$ and $\Downarrow_L(x) = g(x)$. As such, we have

$$\uparrow_L(x) = \frac{\varepsilon}{2} \Uparrow_L(x) + \left(1 - \frac{\varepsilon}{2}\right) \Downarrow_L(x) > \frac{\varepsilon}{2} f(x) + \left(1 - \frac{\varepsilon}{2}\right) g(x) = x^2 + 1 - (x^2+1) = 0.$$

Similarly, for $x \in \{2/2n, 4/2n, \dots, (2n-2)/2n\}$, we have $\Uparrow_L(x) = f(x)$ and $\Downarrow_L(x) < g(x)$. As such, we have

$$\uparrow_L(x) = \frac{\varepsilon}{2} \Uparrow_L(x) + \left(1 - \frac{\varepsilon}{2}\right) \Downarrow_L(x) < \frac{\varepsilon}{2} f(x) + \left(1 - \frac{\varepsilon}{2}\right) g(x) = 0.$$

We thus obtain the following.

▶ **Lemma 16.** *For any $\varepsilon > 0$ and for any $n \geqslant 1$, there exists a set of $2n$ lines in $\mathbb{R}^2$ whose ε-upper envelope crosses the x-axis at least $2n-2$ times.*

Next, we replicate the x-axis by sufficiently close (almost parallel) n lines that lie between the lower and upper envelopes of L, and we add them to L. Then there are $\Omega(n^2)$ intersection points between $\uparrow_L$ and the lines of L. We thus get the following result.

▶ **Lemma 17.** *There exists a set L of n lines in $\mathbb{R}^2$ such that the number of intersection points between $\mathcal{I}_\varepsilon(L)$ and L is $\Omega(n^2)$.*

5 Optimal Weak Kernel

The above results dealt with the stronger notion of a kernel, but the original work of Agarwal et al. [2] defined a weaker notion of a kernel, see Definition 6. In this section, we present an $O(n^2 \log n)$-time algorithm for computing an optimal weak ε-kernel, by reducing it to computing a smallest arc cover, with some additional properties, in a set of $O(n^2)$ *unit arcs* (i.e., arcs on the unit circle).

Let P be a set of n points in $\mathbb{R}^2$ and $\varepsilon \in (0,1)$ a parameter. We parametrize $\mathbb{S}$ with the orientation in the range $[-\pi, \pi]$ (with the two endpoints of this interval being glued together), and let $u(\theta) = (\cos\theta, \sin\theta)$. Recall that a subset $C \subseteq P$ is an weak ε-kernel of P if

$$\overline{\omega}(u(\theta), C) \geqslant (1-\varepsilon)\overline{\omega}(u(\theta), P) \tag{1}$$

for all $\theta \in [-\pi, \pi]$. Since $\overline{\omega}(u(\theta), P) = \overline{\omega}(u(-\theta), P)$, it suffices to satisfy Eq. (1) for the angular interval $[-\pi/2, \pi/2]$. However, it will be convenient to work with the entire $\mathbb{S}$, so let

$$\Updownarrow_P(\theta) = \overline{\omega}(u(\theta/2), P) \quad \text{for } \theta \in [-\pi, \pi].$$

A subset $C \subseteq P$ is a **weak ε-kernel** if and only if

$$\Updownarrow_C(\theta) \geqslant (1-\varepsilon) \Updownarrow_P(\theta) \quad \forall \theta \in [-\pi, \pi].$$

For a pair $1 \leqslant i < j \leqslant n$ and $\theta \in [-\pi, \pi]$, we define $\gamma_{ij} \in [-\pi, \pi] \to \mathbb{R}_{\geqslant 0}$ as

$$\gamma_{ij}(\theta) := |\langle u(\theta/2), p_i - p_j \rangle| = |(a_i - a_j)\cos(\theta/2) + (b_i - b_j)\sin(\theta/2)|,$$

where $p_i = (a_i, b_i)$. Set $\Gamma = \{\gamma_{ij} \mid 1 \leqslant i < j \leqslant n\}$. It is easily seen that $\Uparrow_\Gamma(\theta) = \Updownarrow_P(\theta)$. For a pair $1 \leqslant i < j \leqslant n$, we define $I_{ij} = \{\theta \in [-\pi, \pi] \mid \gamma_{ij}(\theta) \geqslant (1-\varepsilon) \Uparrow_\Gamma(\theta)\}$.

▶ **Lemma 18.** *The set I_{ij} is a single connected circular arc.*

Proof. It is convenient to reparameterize γ_{ij}. More precisely, we define the function $\xi_{ij} : \mathbb{R} \to \mathbb{R}_{\geqslant 0}$ as

$$\xi_{ij}(x) = |(a_i - a_j) + (b_i - b_j)x| \quad \text{for } x \in \mathbb{R}. \tag{2}$$

Set

$$\Xi = \{\xi_{ij} \mid 1 \leqslant i < j \leqslant n\} \quad \text{and} \quad J_{ij} = \{x \in \mathbb{R} \mid \xi_{ij}(x) \geqslant (1-\varepsilon) \Uparrow_{\Xi} (x)\}.$$

Note that

$$\gamma_{ij}(\theta) = \frac{1}{\sqrt{1+\tan^2(\theta/2)}} \xi_{ij}\big(\tan(\theta/2)\big),$$

therefore $\tan(\theta/2) \in J_{ij}$ if and only if $\theta \in I_{ij}$.

The graph of ξ_{ij} is a cone with axis of symmetry around the y-axis and apex on the x-axis – specifically, there are two numbers α_{ij}, β_{ij} such that $\xi_{ij}(x) = \alpha_{ij}|x - \beta_{ij}|$. The number α_{ij} is the **slope** of ξ_{ij}. The function $(1-\varepsilon) \Uparrow_{\Xi}$ is a convex chain, which is the upper envelope of the functions $(1-\varepsilon)\xi_{ij}$, see Figure 8 (A).

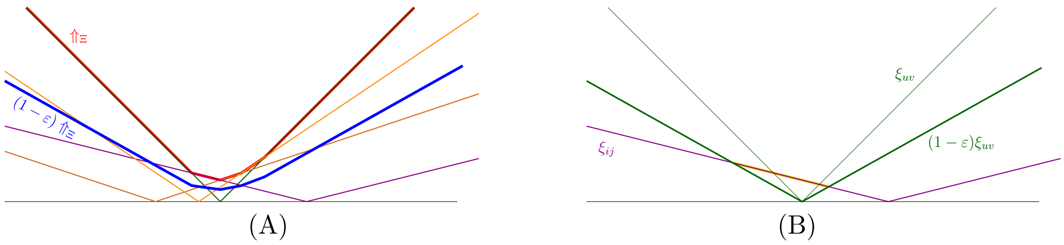


Figure 8 Illustration of the proof of Lemma 18. (A) Upper envelope $\Uparrow_{\Xi}$, and lower-bound curve $(1-\varepsilon) \Uparrow_{\Xi}$. (B) A cone with higher slope "buries" at least one leg of the other cone.

Since the graph of ξ_{ij} is composed of two rays, J_{ij} is potentially the union of two intervals (potentially infinite rays). If J_{ij} does not contain any finite interval, i.e., consists of two rays, then I_{ij} is a single arc containing the orientation π. So assume that J_{ij} contains a finite interval, see Figure 8 (B). This implies that there are indices u, v, such that $(1-\varepsilon)\xi_{uv}$ has higher slope than ξ_{ij}. But then $(1-\varepsilon)\xi_{uv}$ is completely above one of the two rays forming the image of ξ_{ij}, implying that J_{ij} can only be a single interval in this case. This in turn implies that I_{ij} consists of a single arc. This completes the proof of the lemma. ◀

A 2-approximation algorithm. Let $\mathcal{I} = \{I_{ij} \mid 1 \leqslant i < j \leqslant n\}$. Using the algorithm of Theorem 7, we compute, in $O(n^2 \log n)$ time, a minimum arc cover $\mathcal{J} \subseteq \mathcal{I}$. Each interval $I_{ij} \in \mathcal{J}$ corresponds to two points p_i, p_j of P. Set $C := \{p_i, p_j \mid I_{ij} \in \mathcal{J}\}$.

▶ **Lemma 19.** *C is an weak ε-kernel of size at most twice the optimal size.*

Proof. Since $\mathcal{J}$ is an arc cover, for any $\theta \in [-\pi, \pi]$, there is pair $p_i, p_j \in C$ such that $\gamma_{ij}(\theta) \geqslant (1-\varepsilon) \Uparrow_{\Gamma} (\theta)$. Therefore $\Updownarrow_C (\theta) \geqslant (1-\varepsilon) \Uparrow_{\Gamma} (\theta) = (1-\varepsilon) \Updownarrow_P (\theta)$, implying that C is a weak ε-kernel.

Conversely, let C^* be an optimal weak ε-kernel. We construct an arc cover $\mathcal{J}^*$ as follows. The points in C^* are in convex position. Consider $\mathcal{N} = \mathcal{N}(C^*)$ the refined normal diagram of C, which is a centrally symmetric partition of $\mathbb{S}$ into $2|C^*|$ intervals such that each pair of antipodal intervals of is associated with an antipodal pair of points $p_i, p_j \in C^*$. For each such pair p_i, p_j, we add the interval I_{ij} to $\mathcal{J}^*$; $|\mathcal{J}^*| = |C^*|$. For $\theta \in [-\pi, \pi]$, suppose p_i, p_j is the supporting pair in directions $u(\theta/2)$ and $-u(\theta/2)$, respectively. Then $\gamma_{ij}(\theta) = \Updownarrow_{C^*} (\theta)$. Since $\Updownarrow_{C^*} (\theta) \geqslant (1-\varepsilon) \Uparrow_{\Gamma} (\theta)$, $\theta \in I_{ij} \in \mathcal{J}^*$. Hence, $\mathcal{J}^*$ is an arc cover.

We can thus conclude that $|C| \leqslant 2|C^*|$. ◀

An exact algorithm. The above algorithm is a 2-approximation because it uses two potentially new points for each interval. We can change the arc-cover problem to account for this. We label every arc in $\mathcal{I}$ by two indices $i, j \in [\![n]\!]$ – indices of the pair of points in P that define it. An arc cover $\mathcal{J} \subset \mathcal{I}$ of $\mathbb{S}$ is **admissible** if every pair of intersecting arcs in $\mathcal{J}$ share exactly one label. For any admissible arc cover $\mathcal{J}$, the size of the set $\{p_i, p_j \mid I_{ij} \in \mathcal{J}\}$ is at most $|\mathcal{J}|$. Furthermore, the arc cover constructed from a weak kernel in the proof of Lemma 19 is admissible. Therefore it suffices to compute a minimum-size admissible arc cover in $\mathcal{I}$.

To compute the smallest admissible arc cover, we follow the ideas in the algorithm of for the arc-cover [1]. While $|\mathcal{I}| = O(n^2)$, there must be a direction $u \in \mathbb{S}$ that is covered by at most $O(n^2/\mathsf{k})$ intervals of $\mathcal{I}$, where k is the size of the optimal weak ε-kernel. Let $\mathcal{J} \subseteq \mathcal{I}$ be the set of intervals covering u (u and $\mathcal{J}$ can be computed in $O(n^2 \log n)$ time). For each one of these intervals, we now perform the greedy algorithm, as in [1] . The only difference is that instead of having a global data structure for all intervals, we break them into n groups. Specifically, for $i = 1, \ldots, n$, let $\mathcal{I}_i \subset \mathcal{I}$ be the set of all arcs I with i being one of the two indices in its label. Now, we build the necessary data-structure used in [1] for each such group. Now, if the current interval is I_{ij}, the algorithm uses the data-structures for $\mathcal{I}_i$ and $\mathcal{I}_j$ to generate two candidate intervals to be used by the greedy algorithm. The algorithm uses the one that extends further clockwise. The rest of the algorithm is the same as in [1]. This algorithm computes the smallest admissible circular arc cover $\mathcal{J}^*$. We return the set $\{p_i, p_j \mid I_{ij} \in \mathcal{J}^*\}$, which in view of the above discussion is an optimal weak ε-kernel. Putting everything together we obtain the following:

▶ **Theorem 20.** *Given a set P of n points in the plane and a parameter $\varepsilon \in (0, 1)$, an optimal weak ε-kernel of P can be computed in $O(n^2 \log n)$ time.*

6 Conclusions

In this paper, we studied the problem of computing optimal kernels in the plane, both in the strong and weak sense. Surprisingly, this very natural problem had not received much attention when kernels were developed around twenty years ago. The problem has surprisingly non-trivial structure, and getting near linear running time to compute them exactly required non-trivial ideas and care. A natural open question is whether an instance-optimal ε-kernel of n points in $\mathbb{R}^2$ can be computed in $O(n \log n)$ time.

References

1 Pankaj K. Agarwal and Sariel Har-Peled. Computing optimal kernels in two dimensions. *CoRR*, abs/2207.07211, 2022. `doi:10.48550/arXiv.2207.07211`.

2 Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. Assoc. Comput. Mach.*, 51(4):606–635, 2004. `doi:10.1145/1008731.1008736`.

3 Pankaj K. Agarwal, Sariel Har-Peled, and Hai Yu. Robust shape fitting via peeling and grating coresets. *Discret. Comput. Geom.*, 39(1-3):38–58, 2008. `doi:10.1007/s00454-007-9013-2`.

4 Pankaj K. Agarwal, Nirman Kumar, Stavros Sintos, and Subhash Suri. Efficient algorithms for k-regret minimizing sets. In Costas S. Iliopoulos, Solon P. Pissis, Simon J. Puglisi, and Rajeev Raman, editors, *16th Int. Symp. Exper. Alg.,* (SEA), pages 7:1–7:23, 2017. `doi:10.4230/LIPIcs.SEA.2017.7`.

5 Pankaj K. Agarwal and Hai Yu. A space-optimal data-stream algorithm for coresets in the plane. In Jeff Erickson, editor, *Proc. 23rd Annu. Sympos. Comput. Geom.* (SoCG), pages 1–10. ACM, 2007. `doi:10.1145/1247069.1247071`.

6 Avrim Blum, Sariel Har-Peled, and Benjamin Raichel. Sparse approximation via generating point sets. *ACM Trans. Algo.*, 15(3):32:1–32:16, 2019. doi:10.1145/3302249.

7 Hervé Brönnimann and Michael T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete Comput. Geom.*, 14(4):463–479, 1995. doi:10.1007/BF02570718.

8 Wei Cao, Jian Li, Haitao Wang, Kangning Wang, Ruosong Wang, Raymond Chi-Wing Wong, and Wei Zhan. k-regret minimizing set: Efficient algorithms and hardness. In *20th Int. Conf. Data. Theory,* (ICDT), pages 11:1–11:19, 2017. doi:10.4230/LIPIcs.ICDT.2017.11.

9 Timothy M. Chan. Faster core-set constructions and data-stream algorithms in fixed dimensions. *Comput. Geom. Theory Appl.*, 35(1-2):20–35, 2006. doi:10.1016/j.comgeo.2005.10.002.

10 Bernard Chazelle and Leonidas J. Guibas. Visibility and intersection problems in plane geometry. *Discret. Comput. Geom.*, 4:551–581, 1989. doi:10.1007/BF02187747.

11 Kenneth L. Clarkson. Algorithms for polytope covering and approximation. In Frank K. H. A. Dehne, Jörg-Rüdiger Sack, Nicola Santoro, and Sue Whitesides, editors, *Proc. 3th Workshop Algorithms Data Struct.* (WADS), volume 709 of *Lect. Notes in Comp. Sci.*, pages 246–252. Springer, 1993. doi:10.1007/3-540-57155-8_252.

12 Gautam Das and Michael T. Goodrich. On the complexity of optimization problems for 3-dimensional convex polyhedra and decision trees. *Comput. Geom.*, 8:123–137, 1997. doi:10.1016/S0925-7721(97)00006-0.

13 Subir Kumar Ghosh and Anil Maheshwari. An optimal algorithm for computing a minimum nested nonconvex polygon. *Information Processing Letters*, 36(6):277–280, 1990. doi:10.1016/0020-0190(90)90038-Y.

14 Leonidas J. Guibas, John Hershberger, Joseph S. B. Mitchell, and Jack Snoeyink. Approximating polygons and subdivisions with minimum-link paths. *Int. J. Comput. Geom. Appl.*, 3(04):383–415, 1993. doi:10.1142/S0218195993000257.

15 John Hershberger and Subhash Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18(3):403–431, 1995. doi:10.1006/jagm.1995.1017.

16 Georgiy Klimenko and Benjamin Raichel. Fast and exact convex hull simplification. In Mikolaj Bojanczyk and Chandra Chekuri, editors, *Proc. 41th Conf. Found. Soft. Tech. Theoret. Comput. Sci.* (FSTTCS), volume 213 of *LIPIcs*, pages 26:1–26:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.FSTTCS.2021.26.

17 C. C. Lee and D. T. Lee. On a circle-cover minimization problem. *Inf. Process. Lett.*, 18(2):109–115, 1984. doi:10.1016/0020-0190(84)90033-4.

18 Joseph SB Mitchell and Valentin Polishchuk. Minimum-perimeter enclosures. *Information Processing Letters*, 107(3-4):120–124, 2008. doi:10.1016/j.ipl.2008.02.007.

19 Joseph SB Mitchell and Subhash Suri. Separation and approximation of polyhedral objects. *Comput. Geom. Theory Appl.*, 5(2):95–114, 1995. doi:10.1016/0925-7721(95)00006-U.

20 Cao An Wang. Finding minimal nested polygons. *BIT Numerical Mathematics*, 31(2):230–236, 1991. doi:10.1007/bf01931283.

21 Cao An Wang and Edward P. F. Chan. Finding the minimum visible vertex distance between two non-intersecting simple polygons. In Alok Aggarwal, editor, *Proc. 2nd Annu. Sympos. Comput. Geom.* (SoCG), pages 34–42. ACM, 1986. doi:10.1145/10515.10519.

22 Yanhao Wang, Michael Mathioudakis, Yuchen Li, and Kian-Lee Tan. Minimum coresets for maxima representation of multidimensional data. In Leonid Libkin, Reinhard Pichler, and Paolo Guagliardo, editors, *Proc. 40th Symp. Principles Database Sys.* PODS, pages 138–152. ACM, 2021. doi:10.1145/3452021.3458322.

23 Hai Yu, Pankaj K. Agarwal, Raghunath Poreddy, and Kasturi R. Varadarajan. Practical methods for shape fitting and kinetic data structures using coresets. *Algorithmica*, 52(3):378–402, 2008. doi:10.1007/s00453-007-9067-9.