Approximation Algorithms for Minimum-Width Annuli and Shells*

Pankaj K. Agarwal† Boris Aronov‡ Sariel Har-Peled§
Micha Sharir¶

Abstract

Let S be a set of n points in $\mathbb{R}^d$. The "roundness" of S can be measured by computing the width $\omega^* = \omega^*(S)$ of the thinnest spherical shell (or annulus in $\mathbb{R}^2$) that contains S. This paper contains two main results related to computing ω^*: (i) For $d = 2$, we can compute in $O(n \log n)$ time an annulus containing S whose width is at most $2\omega^*(S)$. We extend this algorithm, so that, for any given parameter $\varepsilon > 0$, an annulus containing S whose width is at most $(1+\varepsilon)\omega^*$ is computed in time $O(n \log n + n/\varepsilon^2)$. (ii) For $d \geq 3$, given a parameter $\varepsilon > 0$, we can compute a shell containing S of width at most $(1+\varepsilon)\omega^*$ either in time $O\big(\frac{n}{\varepsilon^d} \log(\frac{\Delta}{\omega^* \varepsilon})\big)$ or in time $O\big(\frac{n}{\varepsilon^{d-2}}\big(\log n + \frac{1}{\varepsilon}\big) \log\big(\frac{\Delta}{\omega^* \varepsilon}\big)\big)$, where Δ is the diameter of S.

*Work by P.A. was supported by Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, by NSF grants EIA–9870724 and CCR–9732787, by an NYI award, and by a grant from the U.S.-Israeli Binational Science Foundation. Work by B.A. was supported by a Sloan Research Fellowship and by a grant from the U.S.-Israeli Binational Science Foundation. Work by M.S. was supported by NSF Grants CCR-97-32101, CCR-94-24398, by grants from the U.S.-Israeli Binational Science Foundation, the G.I.F., the German-Israeli Foundation for Scientific Research and Development, and the ESPRIT IV LTR project No. 21957 (CGAL), and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University. Part of the work by P.A. and B.A. on the paper was done when they visited Tel Aviv University in May 1998.

†Center for Geometric Computing, Department of Computer Science, Box 90129, Duke University, Durham, NC 27708-0129, USA. E-mail: pankaj@cs.duke.edu

‡Department of Computer and Information Science, Polytechnic University, Brooklyn, NY 11201-3840, USA. E-mail: aronov@ziggy.poly.edu

§School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: sariel@math.tau.ac.il SHOULD THIS BE UPDATED !?!?!?!?!?!?

¶School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel; and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA. E-mail: sharir@math.tau.ac.il

1 Introduction

Let S be a set of n points in $\mathbb{R}^d$. The *roundness* of S can be measured by approximating S with a sphere Γ so that the maximum distance between a point of S and Γ is minimized, i.e., by computing

$$\min_{c\in\mathbb{R}^d, r\in\mathbb{R}} \max_{p\in S} |d(p,c) - r|.$$

For $c \in \mathbb{R}^d$ and for $r, R \in \mathbb{R}$ with $0 \le r \le R$, we define the *spherical shell* (*shell*, for short, and, in the plane, *annulus*) $\mathcal{A}(c, r, R)$ to be the closed region lying between the two concentric spheres of radii r and R centered at c. The *width* of $\mathcal{A}(c, r, R)$ is $R - r$. The problem of measuring the roundness of S is equivalent to computing a shell, $\mathcal{A}^*(S)$, of the smallest width that contains S. See Figure 1.

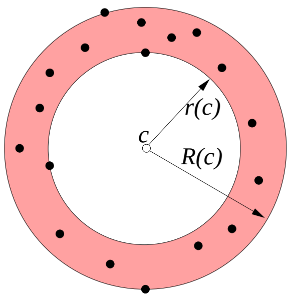

Figure 1: The annulus $\mathcal{A}^*(S)$.

The main motivation for computing a minimum-width shell or annulus comes from metrology. For example, the circularity of a two-dimensional object O in the plane is measured by sampling a set S of points on the surface of O (e.g., using coordinate measurement machines) and computing the width of the thinnest shell containing S [21]. Motivated by this and other applications, the problem of computing $\mathcal{A}^*(S)$ in the plane has been studied extensively [2, 6–8, 18, 19, 22, 26, 28, 30–33, 35, 36, 38]. Ebara *et al.* [18] noticed that in the planar case the center of $\mathcal{A}^*(S)$ is a vertex of the overlay of the nearest- and farthest-neighbor Voronoi diagrams of S. This property was later refined and extended in [32, 36]. These observations immediately lead to an $O(n^2)$-time algorithm for computing $\mathcal{A}^*(S)$ in the plane. Subquadratic algorithms were later developed in [2, 6, 7]. The asymptotically fastest known randomized algorithm, by Agarwal and Sharir [6], computes $\mathcal{A}^*(S)$ in expected time $O(n^{3/2+\varepsilon})$, for any $\varepsilon > 0$. Since the subquadratic algorithms are rather complicated, simpler and faster algorithms have been developed for various special cases [13, 16, 26, 37]. Mehlhorn *et al.* [28] and Kumar and Sivakumar [25] have studied this problem under the *probing* *model* in which the set S of sample points is chosen adaptively; see the original papers for details.

Very little was known about computing $\mathcal{A}^*(S)$ efficiently in higher dimensions. Extending the observation by Ebara *et al.* [18] to $\mathbb{R}^3$, it can be shown that the center of $\mathcal{A}^*(S)$ is the intersection point of an edge of the nearest-neighbor Voronoi diagram of S with a face of the farthest-neighbor Voronoi diagram of S, or vice versa. Using this fact, $\mathcal{A}^*(S)$ can be computed in $O(n^3 \log n)$ time [16]. This idea can also be extended to higher dimensions. Very recently Chan [11] pointed out that the three-dimensional problem can be solved exactly in a very simple manner in time $O(n^2)$; in fact his observation gives a procedure in all dimensions. See the discussion at the end of the paper. $^{\text{bor}}$Any objections?$_{\text{is}}$ ⟵

This paper contains two main results. $^{\text{bor}}$Made an itemize out of this ... If you do not like it, remove itemize and replace (i) and (ii) by words (first, second...)$_{\text{is}}$ ⟵

(i) For $d \geq 2$, given a parameter $\varepsilon > 0$, we present simple algorithms that run either in time $O\big(\frac{n}{\varepsilon^d} \log(\frac{\Delta}{\omega^* \varepsilon})\big)$ or in $O\big(\frac{n}{\varepsilon^{d-2}}\big(\log n + \frac{1}{\varepsilon}\big) \log\big(\frac{\Delta}{\omega^* \varepsilon}\big)\big)$ for computing a shell that contains S and whose width is at most $(1+\varepsilon)\omega^*$, where ω^* is the width of $\mathcal{A}^*(S)$ and $\Delta = \mathrm{diam}(S)$ is the diameter of S (Section 3). If the middle radius (i.e., average of the inner and outer radii) of $\mathcal{A}^*(S)$ is at most $U \cdot \mathrm{diam}(S)$, then the running time of the algorithms are $O((n/\varepsilon^d) \log U)$ and $O\big(\frac{n}{\varepsilon^{d-2}}\big(\log n + \frac{1}{\varepsilon}\big) \log U\big)$, respectively. In most practical situations, U is a constant. For example, if the input points span an angle of at least θ with respect to the center of $\mathcal{A}^*(S)$, $U = O(1/\theta)$. ⟵

$^{\text{bor}}$Technically when we write $\log U$, do we mean $\max\{\log U, 1\}$? Technically, U could be less than 1!$_{\text{is}}$

A main idea used in the algorithms is the observation that, in the plane, the minimum-area annulus containing S can be used to approximate $\mathcal{A}^*(S)$, and while this approximation might not always be good, it can at least be computed in linear time using linear programming. We refine this idea and extend it to higher dimensions to achieve the bounds stated above.

(ii) We describe simpler, faster algorithms for $d = 2$. We first describe in Section 4.1 a very simple $O(n \log n)$-time algorithm for computing an annulus that contains S and whose width is at most twice that of $\mathcal{A}^*(S)$. Duncan *et al.* [16] had described an approximation algorithm under some assumptions on the distribution of input points. No general near-linear time algorithm with constant-factor approximation was previously known.

We then combine this algorithm with the previous one to obtain a $(1+\varepsilon)$-approximation algorithm. Given a parameter $\varepsilon > 0$, we compute in $O(n \log n +$

$n/\varepsilon^2)$ time an annulus that contains S whose width is at most $(1+\varepsilon)\omega^*$, where ω^* is the width of $\mathcal{A}^*(S)$ (Section 4.2).

$^{\text{bor}}$Should we add here: More recently, T.M. Chan obtained a number of new approximation results using fairly simple techniques; refer to [11] and the discussion at the end of this paper. Sufficient? By the way, I am against enumerating the results he obtained in his paper.$_{\text{is}}$ ⟵

2 Geometric Preliminaries

Let S be a set of n points in $\mathbb{R}^d$. For a point $p \in \mathbb{R}^d$, let $\mathrm{r}(p)$ (resp. $\mathrm{R}(p)$) denote the distance between p and its nearest (resp. farthest) neighbor in S. $\mathcal{A}(p, \mathrm{r}(p), \mathrm{R}(p))$ is the shell of smallest width that is centered at p and contains S, which we denote by $\mathcal{A}(p)$. In what follows, unless we consider the problem specifically in the plane, we will use the term "shell" to refer to a spherical shell in dimension higher than two and to an annulus in two dimensions. Set

$$\omega(p) = \mathrm{R}(p) - \mathrm{r}(p) \quad \text{and} \quad r_{\mathrm{mid}}(p) = \frac{\mathrm{R}(p) + \mathrm{r}(p)}{2}.$$

We put $\omega^* = \omega^*(S) = \inf_{p \in \mathbb{R}^d} \omega(p)$ and denote by $\mathcal{A}^* = \mathcal{A}^*(S)$ a shell of width ω^* containing S. Note that the optimum value ω^* may not be attained by any finite point, in which case $\mathcal{A}^*(S)$ is a slab enclosed between two parallel hyperplanes, and $\omega^*(S)$ is then the standard *width* of S. See Figure 2 for an illustration of this case. The following lemma states two simple but useful properties of $r_{\mathrm{mid}}(p)$.

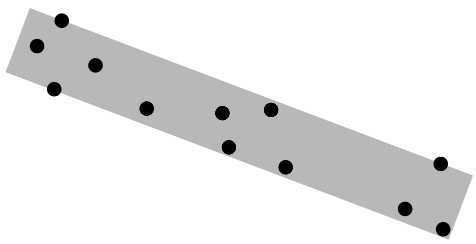

Figure 2: The minimum-width annulus is realized by a center at infinity

Lemma 2.1 *Let S be a finite set of points in $\mathbb{R}^d$. For any $p, q \in \mathbb{R}^d$, we have the following:*

(*i*) $r_{\mathrm{mid}}(p) \geq \mathrm{R}(p)/2 \geq \mathrm{diam}(S)/4$.

(*ii*) $|r_{\mathrm{mid}}(p) - r_{\mathrm{mid}}(q)| \leq d(p,q) \leq r_{\mathrm{mid}}(p) + r_{\mathrm{mid}}(q)$.

Proof: (i) is trivial to prove. To show (ii), use the inequalities

$$\mathrm{r}(p) \le d(p,q) + \mathrm{r}(q),\ \mathrm{R}(p) \le d(p,q) + \mathrm{R}(q),\ d(p,q) \le \mathrm{r}(p) + \mathrm{R}(q),$$

whose proofs are straightforward. □

Let $\mathrm{Vor}_N(S)$ (resp. $\mathrm{Vor}_F(S)$) denote the nearest-neighbor (resp. farthest-neighbor) Voronoi diagram of S. For $d = 2$, let $\mathrm{Vor}_N(S, \ell)$ denote the nearest-neighbor Voronoi diagram of S restricted to a line ℓ. That is, $\mathrm{Vor}_N(S, \ell)$ is the partition of ℓ into maximal intervals so that the same point of S is closest to all points within each interval. The vertices of $\mathrm{Vor}_N(S, \ell)$ are the intersection points of ℓ with the edges of $\mathrm{Vor}_N(S)$. We can obviously compute $\mathrm{Vor}_N(S, \ell)$ in $O(n \log n)$ time by first computing the entire $\mathrm{Vor}_N(S)$ and then intersecting ℓ with it. However, $\mathrm{Vor}_N(S, \ell)$ can be computed directly, in $O(n \log n)$ time, using a considerably simpler algorithm; see e.g. [29]. [bor]Next sentence: Why don't we drop it, if there are no objections?$_{\mathrm{is}}$ As ⟵
an alternative, after having computed $\mathrm{Vor}_N(S)$, we can compute $\mathrm{Vor}_N(S, \ell)$ in $O(n)$ time by tracing ℓ through $\mathrm{Vor}_N(S)$. We define $\mathrm{Vor}_F(S, \ell)$ analogously; it can also be computed either directly in $O(n \log n)$ time or in $O(n)$ time after having computed $\mathrm{Vor}_F(S)$.

3 An Approximation Algorithm in Any Dimension

Let S be a set of n points in $\mathbb{R}^d$; we assume that d is a small constant. Set $\Delta = \mathrm{diam}(S)$. We will first describe an approximation algorithm for computing the thinnest shell $\mathcal{A}(p)$ containing S with the constraint that

$$r_{\mathrm{mid}}(p) = (\mathrm{r}(p) + \mathrm{R}(p))/2 \le U \cdot \Delta$$

for some given parameter $U \in \mathbb{R}$. Let $\mathcal{A}^*(S, U)$ denote this constrained minimum-width shell, and let $\omega^*(S, U)$ denote the width of $\mathcal{A}^*(S, U)$. Computing $\mathcal{A}^*(S, U)$ can be formulated as the following optimization problem in the $d + 2$ variables $x_1, x_2, \dots, x_d, r, R$: [bor]I dropped the parentheses in the previous sentence and refor- ⟵
matted the optimization problems. Feel free to hate me now.$_{\mathrm{is}}$

$$\begin{aligned} &\text{minimize} && R - r \\ &\text{subject to} && r \le \left(\textstyle\sum_{i=1}^{d} (x_i - p_i)^2\right)^{1/2} \le R \qquad \forall p = (p_1, \dots, p_d) \in S \\ & && r + R \le 2U\Delta. \end{aligned}$$

Let C be a d-dimensional hyper-rectangle of the form $\prod_{i=1}^{d} [\alpha_i, \beta_i]$. We define another constrained shell $\mathcal{E}(S, C)$ (which becomes, when $d = 2$, the minimum-area annulus containing S with center constrained to lie in C), in the same variables, as follows:

$$\begin{array}{lll} \text{minimize} & R^2 - r^2 & \\ \text{subject to} & r \le \left(\sum_{i=1}^{d}(x_i - p_i)^2\right)^{1/2} \le R & \forall p = (p_1, \ldots, p_d) \in S \\ & \alpha_i \le x_i \le \beta_i & 1 \le i \le d. \end{array}$$

If we substitute Σ for $R^2 - \sum_{i=1}^{d} x_i^2$ and σ for $r^2 - \sum_{i=1}^{d} x_i^2$, then $\Sigma - \sigma = R^2 - r^2$, and we can restate the optimization problem defining $\mathcal{E}(S, C)$ as:

$$\begin{array}{lll} \text{minimizae} & \Sigma - \sigma & \\ \text{subject to} & \sigma \le -\sum_{i=1}^{d} 2p_i x_i + \sum_{i=1}^{d} p_i^2 \le \Sigma & \forall p = (p_1, \ldots, p_d) \in S \\ & \alpha_i \le x_i \le \beta_i & 1 \le i \le d. \end{array}$$

This is, however, an instance of linear programming with $d+2$ variables, and can be solved in $O(n)$ time [17, 27], provided d is a constant. Let $\hat{\omega}(S, C)$ denote the width of $\mathcal{E}(S, C)$.

We now describe our approximation algorithm. Let $C(p, s)$ be the d-dimensional axis-parallel cube of side length s and centered at p.

Algorithm Approx_Shell (S, U, ε)

1. Compute $\mathcal{E}(S, \mathbb{R}^d)$. If $\hat{\omega}(S, \mathbb{R}^d) = 0$, then return $\mathcal{E}(S, \mathbb{R}^d)$.
2. Pick a point $o \in S$ and set $\mathfrak{C} = C(o, (2U + 2)\Delta)$.
3. Partition $\mathfrak{C}$ into a collection $\mathcal{C} = \{C_1, \ldots, C_k\}$ of axis-parallel cubes so that, for all points p, q inside the same cube C_i, $r_{\text{mid}}(p) \le (1 + \varepsilon) r_{\text{mid}}(q)$.
4. For each $C_i \in \mathcal{C}$, compute $A_i = \mathcal{E}(S, C_i)$. bor Should we use a script A instead of italic here? is ⟵
5. Return the thinnest shell among $A_1, \ldots, A_k$.

Lemma 3.1 Approx_Shell(S, U, ε) *returns a shell whose width is at most* $(1 + \varepsilon)\omega^*(S, U)$.

Proof: If $\hat{\omega}(S, \mathbb{R}^d) = 0$, then the statement is obvious. Otherwise, let p be the center of $\mathcal{A}^*(S, U)$. Since $r_{\text{mid}}(o) \le \mathrm{R}(o) \le \Delta$ and $r_{\text{mid}}(p) \le U\Delta$, we have, by Lemma 2.1(ii), that $p \in \mathfrak{C}$. Let C_i be the cube containing p. Let $q \in C_i$ be the center of $\mathcal{E}(S, C_i)$. Then

$$\mathrm{R}^2(q) - \mathrm{r}^2(q) \le \mathrm{R}^2(p) - \mathrm{r}^2(p), \quad \text{or} \quad r_{\text{mid}}(q)\omega(q) \le r_{\text{mid}}(p)\omega(p).$$

Equivalently,

$$\omega(q) \leq \frac{r_{\mathrm{mid}}(p)}{r_{\mathrm{mid}}(q)}\omega(p) \leq (1+\varepsilon)\omega^*(S,U).$$

□

We now describe how to construct a partition $\mathcal{C}$ of $\mathfrak{C}$. A similar construction is given in [23].

Lemma 3.2 *Let U, ε be two positive numbers. Then $\mathfrak{C} = C(o,(2U+2)\Delta)$ can be partitioned into a set $\mathcal{C}$ of $O((1/\varepsilon)^d \log U)$ cubes so that $r_{\mathrm{mid}}(p) \leq (1+\varepsilon)r_{\mathrm{mid}}(q)$ for all p, q in the same cube of the partition. This tiling can be computed in $O(n + (1/\varepsilon)^d \log U)$ time.*

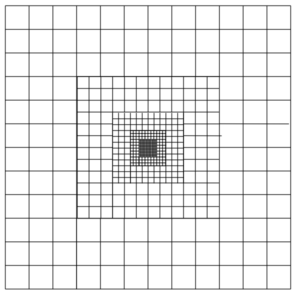

Figure 3: Tiling of $\mathfrak{C}$.

Proof: Compute a real number μ such that $\Delta/2 \leq \mu \leq \Delta$. (See [20] for a simple $O(n)$ algorithm for approximating the diameter to within a factor of $\sqrt{3}$ in any dimension. Alternatively, fix any $p \in S$ and take $\mu = \mathrm{R}(p) \geq \Delta/2$, by Lemma 2.1(i).)

Set $m = \lceil \log_2(U+1) \rceil$. For $i = 1, \ldots, m$, we define

$$B_0 = C(o, 4\mu), \quad B_i = C(o, 2^{i+2}\mu) \setminus C(o, 2^{i+1}\mu).$$

We can tile B_0 by $O(1/\varepsilon^d)$ axis-parallel cubes having side length $r_0 = \mu\varepsilon/(4\sqrt{d})$. Let C be a cube in this tiling. For $p, q \in C$, we have, by Lemma 2.1,

$$\begin{aligned} r_{\mathrm{mid}}(p) &\leq r_{\mathrm{mid}}(q) + d(p,q) \leq r_{\mathrm{mid}}(q) + \mu\varepsilon/4 \\ &\leq (1+\varepsilon)r_{\mathrm{mid}}(q), \end{aligned}$$

since $r_{\mathrm{mid}}(q) \geq \Delta/4 \geq \mu/4$.

Let $r_i = 2^i\mu\varepsilon/\sqrt{d}$, for $i = 1, \ldots, m$. B_i can be tiled by

$$O\left(\left(\frac{2^{i+2}\mu}{r_i}\right)^d\right) = O\left(\left(\frac{2^{i+2}\mu}{2^i\mu\varepsilon/\sqrt{d}}\right)^d\right) = O\left(\frac{1}{\varepsilon^d}\right)$$

axis-parallel cubes with side length r_i, for $i = 1, \ldots, m$.

Let C be a cube in this tiling of B_i, and let p, q be two points in C. Using Lemma 2.1(ii) and the fact that $r_{\text{mid}}(o) \le \Delta \le 2\mu$, we have [bor]should $2^{i+1}\mu$ be $2^{i+1}\mu\sqrt{d}$ below?[is] ⟵

$$r_{\text{mid}}(q) \ge d(q, o) - r_{\text{mid}}(o) \ge 2^{i+1}\mu - 2\mu \ge 2^i\mu.$$

We also have

$$\begin{aligned} r_{\text{mid}}(p) &\le r_{\text{mid}}(q) + d(q, p) \le r_{\text{mid}}(q) + \sqrt{d} r_i \\ &= r_{\text{mid}}(q) + 2^i \mu\varepsilon \le r_{\text{mid}}(q)(1 + \varepsilon). \end{aligned}$$

See Figure 3 for an illustration of the resulting tiling. This completes the proof of the lemma, since B_m [bor]Isn't B_m twice as big as it needs to be?[is] contains $\mathfrak{C}$ and the total number of cubes is $O((1/\varepsilon^d) \log U)$. The bound on the running time of this construction is obvious. ⟵ □

Theorem 3.3 *Let S be a set of n points in $\mathbb{R}^d$, $\varepsilon > 0$, and $U > 0$. One can compute a shell $\mathcal{A} \supset S$ whose width is at most $(1 + \varepsilon)\omega^*(S, U)$ either in time $O((n/\varepsilon^d) \log U)$ or in time*

$$O\left(\frac{n}{\varepsilon^{d-2}}\left(\log n + \frac{1}{\varepsilon}\right) \log U\right).$$

Proof: The first bound on the running time is a consequence of the preceding discussion: We spend $O(n)$ time on each cube of $\mathcal{C}$, and $\mathcal{C}$ has $O((1/\varepsilon^d) \log U)$ cubes. The second bound follows by observing that the execution of the algorithm APPROX_SHELL can be interpreted as follows: We compute a sequence of cubes $\mathcal{C}_1, \ldots, \mathcal{C}_m$, where $m = O(\log U)$. Each such cube is decomposed into $O(1/\varepsilon^d)$ subcubes using an appropriate uniform grid. For each subcube C we obtain $\mathcal{E}(S, C)$ as a solution of an appropriate linear programming problem.

Let $\mathcal{C}_i$ be such a cube, and let $V = \{C_1, \ldots, C_\mu\}$ be the resulting decomposition of $\mathcal{C}_i$ into subcubes. The linear programming instances on each C_j are almost identical except for the $2d$ inequalities restricting the solution to lie inside C_j. This implies that, with the possible exception of one subcube, the solutions to all those linear programming instances must lie on the boundaries of the respective cubes $C_1, \ldots, C_\mu$. Moreover, the solution of the at most one instance of the linear programming that does lie in the interior of its cube, can be computed directly, by solving a single linear-programming instance, without restricting the location of the solution to any subcube (i.e. by dropping the inequalities $\alpha_i \le x_i \le \beta_i$).

In particular, we conclude that we can reduce the d-dimensional problem to a $(d-1)$-dimensional problem, as follows:

- Solve the unrestricted version of the linear programming (i.e., compute the global "minimum area" shell).
- For each axis-parallel $(d-1)$-dimensional hyperplane H of the grid defining the decomposition V, find recursively a $(1+\varepsilon)$-approximate shell containing S whose center is constrained to lie on $H \cap \mathcal{C}_i$. There are $O(d/\varepsilon)$ such hyperplanes.
- Return the shell of minimum width among all those generated by the algorithm.

The recursion bottoms out at $d = 2$, where we proceed as follows. Let H be our two-dimensional plane. We can compute in $O(n \log n)$ time the maps induced on H by the d-dimensional nearest- and furthest-neighbor Voronoi diagrams of S (those maps are called power diagrams [9], they have linear complexity, and they can be computed in $O(n \log n)$ time). Our target is to approximate the minimum difference between the farthest and nearest neighbors of points on H (this is the width of the minimum-width shell whose center is restricted to lie on H). [bor]I am confused. Don't we minimize differences of squares here and not width? Hmmm...[is] We note that we can compute this minimum along a line ℓ in $O(n)$ time, by performing a walk through the overlay of those two diagrams along ℓ. We do this along each line of the grid, and also solve the global linear-programming instance where the center of the shell is restricted to lie on H. Thus, we can solve a two-dimensional instance in $O(n \log n + n/\varepsilon)$ time. ⟵

Overall, the recursive algorithm for the subcubes of $\mathcal{C}_i$ requires $O((n/\varepsilon^{d-2}) \log n + n/\varepsilon^{d-1})$ time. Thus, solving all the linear programming instances for $\mathcal{C}_1, \ldots, \mathcal{C}_m$ requires

$$O\left(\frac{n}{\varepsilon^{d-2}}\left(\log n + \frac{1}{\varepsilon}\right)\log U\right)$$

time. □

Even though Theorem 3.3 is not fully satisfactory, for all practical purposes the assumptions in the theorem are reasonable. For example, in the plane, if the points in S span an angle of at least $\theta \in [0, \pi/2]$ with respect to the center c of $\mathcal{A}^*(S)$, then $r_{\text{mid}}(c) = O(\Delta/\sin\theta) = O(\Delta/\theta)$. In this case we can compute an annulus that contains S and has width at most $(1+\varepsilon)\omega^*(S)$, in time $O(\frac{n}{\varepsilon^2}\log\frac{1}{\theta})$.

For $d = 2$ the algorithm of Theorem 3.3 can be further simplified and improved, by noting that in this case the power diagrams are (regular) nearest- and furthest-neighbor Voronoi diagrams, and that they need to be computed only once. We thus obtain the following.

Theorem 3.4 *Let S be a set of n points in the plane, $\varepsilon > 0$, and $U > 0$. One can compute an annulus $\mathcal{A} \supset S$ of width at most $(1+\varepsilon)\omega^*(S, U)$ in time $O(n \log n + (n/\varepsilon)\log U)$.* [sar]verify new running time![iel] ⟵

We next modify the algorithm Approx_Shell so that it produces in all cases a shell containing S of width at most $(1+\varepsilon)\omega^*(S)$.

Lemma 3.5 *For $U > 6$ we have*

$$\omega^*(S, U) \le \omega^*(S) + \frac{8 \cdot \mathrm{diam}(S)}{U}.$$

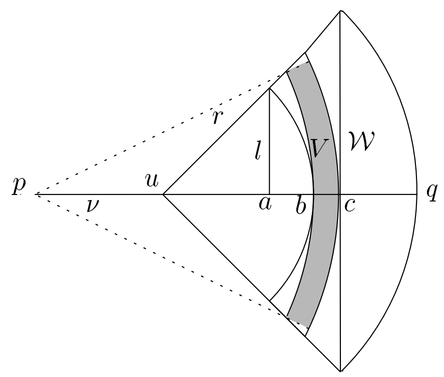

Figure 4: Construction for the proof of Lemma 3.5.

Proof: [bor]Can someone fix the Ipe Figure 4 as follows: move a and b down a bit. Move $\mathcal{W}$ outside of the big circle. In fact, maybe even extend the two rays out of u past the big circle.[is] Let $\mathcal{A}^*$ be a minimum-width shell containing S, with center p and width $\omega^* = \omega^*(S)$. Put $\Delta = \mathrm{diam}(S)$. It suffices to consider the case $\omega^*(S, U) \ne \omega^*(S)$, so we have $r_{\mathrm{mid}}(p) > U\Delta$. ⟵

Let $\mathcal{V}$ be a circular cone centered at p, containing S, and having the smallest opening angle. Let $V = \mathcal{V} \cap \mathcal{A}^*$. Since $r_{\mathrm{mid}}(p) > 6\Delta$, $\mathcal{V}$ spans less than a halfspace. Let ν be the ray emanating from p along the axis of symmetry of $\mathcal{V}$; see Figure 4. Let b and c be the points where ν meets the inner and outer spheres of $\mathcal{A}^*$, respectively. Let u be a point on the segment pb at distance $r = U\Delta/2$ from b. Let $\mathcal{W}$ be the smallest circular cone centered at u, with axis of symmetry along ν and containing V. Let σ be the $(d-2)$-sphere formed by intersecting $\partial\mathcal{W}$ with the sphere of radius r centered at u, and let a and l denote the center and radius of σ, respectively (see Figure 4). Consider the portion of $\mathcal{W}$ lying on the same side as p and u of the hyperplane through c and orthogonal to ν, and let R denote the maximum distance from u to a point in this portion. The shell $\mathcal{A}'$ centered at u with radii r and R, encloses V and thus also covers S. We now estimate $\omega(u)$ by obtaining an upper bound on the width of $\mathcal{A}'$.

Let q be the point on $\mathcal{V}$ at distance R from u, as shown in Figure 4. We have $\omega(u) \leq \omega^* + d(c,q)$. However, $d(u,a) = \sqrt{r^2 - l^2}$ and

$$d(a,b) = r - \sqrt{r^2 - l^2} = \frac{l^2}{r + \sqrt{r^2 - l^2}} \leq \frac{l^2}{r}.$$

By similarity, we have $d(c,q) = d(a,b)\dfrac{r + \omega^*}{d(u,a)}$.

Note that $\omega^* < \Delta < r/3$ and that $l \leq \Delta = 2r/U \leq r/3$. To see the latter inequality, project S centrally, towards u, to the sphere δ of radius r about u. The image $\hat{S}$ of S falls inside the cap $\delta \cap \mathcal{W}$, which, by construction, is a smallest cap on δ enclosing $\hat{S}$ start (indeed, if $\delta \cap \mathcal{W}$ is not minimal, then $\mathcal{V}$ can be also shrunk down, which contradicts its minimality). end ⟵ Chg

bor I do not believe it as written. I do not see a clean way of fixing it. Talk to me if interested to know what I am talking about. The ref is right! is ⟵ Since the projection does not increase the distances between points, the diameter of $\hat{S}$ is at most Δ, which is easily seen to imply that $l \leq \Delta$. This implies that $d(u,a) = \sqrt{r^2 - l^2} \geq r\sqrt{1 - \frac{1}{9}} \geq r/2$. Hence, we have bor Would changing $r/2$ to $2r/3$ and then $2r$ to $4r/3$ get a 4 instead of 8 in the lemma? Or did I make a mistake anywhere? Should we bother? is ⟵

$$d(c,q) \leq \frac{l^2}{r} \cdot \frac{2r}{r/2} = \frac{4l^2}{r}.$$

Putting things together,

$$\begin{aligned} d(b,q) &= \omega^* + d(c,q) \leq \omega^* + \frac{4l^2}{r} \leq \omega^* + \frac{4\Delta^2}{r} \\ &\leq \omega^* + \frac{4\Delta^2}{U\Delta/2} = \omega^* + \frac{8\Delta}{U}. \end{aligned}$$

Note that

$$\begin{aligned} r_{\mathrm{mid}}(u) &\leq r + d(b,q) - \frac{\omega^*}{2} \leq r + \frac{\omega^*}{2} + \frac{8\Delta}{U} < \frac{3r}{2} + \frac{8\Delta}{U} \\ &= \Delta\left(\frac{3U}{4} + \frac{8}{U}\right) < U \cdot \Delta. \end{aligned}$$

Hence $\omega^*(S,U) \leq w(u) \leq \omega^* + \frac{8\Delta}{U}$, as asserted. □

Corollary 3.6 *Let $\varepsilon > 0$, $U > 6$ be two positive constants. One can compute in time $O\big(\big((n/\varepsilon^{d-2})\log n + n/\varepsilon^{d-1}\big)\log U\big)$ or $O(n/\varepsilon^d \log U)$, a shell of width at most*

$$(1+\varepsilon)\left[\omega^*(S) + \frac{8\Delta}{U}\right]$$

that contains S, where $\Delta = \mathrm{diam}(S)$.

Finally, we describe the general approximation algorithm. Let APPROX_DIAM(S) be the procedure that computes in linear time a $\sqrt{3}$-approximation Δ_0 of $\Delta(S) = \mathrm{diam}(S)$ (see [20] or the discussion at the beginning of the proof of Lemma 3.2).

Algorithm APPROX_SHELL_2 (S, ε)

```
ω = Δ_0 = APPROX_DIAM (S);    ω_old = ∞;
while ω < ω_old/2 do
        U = (50√3Δ_0 / ε) · (1/ω);
        A(p) = APPROX_SHELL(S, U, ε/8);
        ω_old = ω;    ω = ω(p);
end while
return A(p);
```

Theorem 3.7 *Given a set S of n points in $\mathbb{R}^d$ and a parameter $0 < \varepsilon < 1$,* APPROX_SHELL_2 *computes a shell of width at most $(1+\varepsilon)\omega^*(S)$. With an appropriate optimization of the calls to* APPROX_SHELL, *the running time is either*

$$O\left(\frac{n}{\varepsilon^d}\log\left(\frac{\Delta}{\omega^*(S)\varepsilon}\right)\right) \quad or \quad O\left(\frac{n}{\varepsilon^{d-2}}\left(\log n + \frac{1}{\varepsilon}\right)\log\left(\frac{\Delta}{\omega^*(S)\varepsilon}\right)\right).$$

Proof: If $\omega^*(S) = 0$, the algorithm terminates after the first iteration. Otherwise, it eventually terminates, as the positive width returned in each call decreases by at least a factor of two, but is no smaller than the optimum width $\omega^*(S)$.

Suppose the while loop is executed m times. Let ω_i, U_i be the values of ω and U computed in the i-th iteration of the loop. Then, putting $\omega^* = \omega^*(S)$,

$$\begin{aligned}
\omega_m &\leq (1+\varepsilon/8)\omega^* + (1+\varepsilon/8)\frac{8\Delta}{U_m} \\
&\leq (1+\varepsilon/8)\omega^* + (1+\varepsilon/8)\frac{8\Delta}{50\sqrt{3}\Delta_0/(\omega_{m-1}\varepsilon)} \\
&\leq (1+\varepsilon/8)\omega^* + (1+\varepsilon/8)\frac{4\varepsilon\omega_{m-1}}{25} \\
&\leq (1+\varepsilon/8)\omega^* + \frac{9\varepsilon\omega_m}{25},
\end{aligned}$$

by Lemma 3.5, and since $w_m \geq w_{m-1}/2$. Thus,

$$\omega_m \leq \frac{1+\varepsilon/8}{1-9\varepsilon/25}\omega^* \leq (1+\varepsilon)\omega^*.$$

Note that for all $i < m$ we have $\omega_i < \frac{\Delta_0\sqrt{3}}{2^i}$. Hence, $\omega^* \leq \omega_{m-1} \leq \frac{\Delta_0\sqrt{3}}{2^{m-1}}$, implying that $m = O(\log\frac{\Delta}{\omega^*})$ and $U_m = O(\Delta/(\omega^*\varepsilon))$.

Note that the i-th call to Approx_Shell (executed, say, by the first algorithm of Theorem 3.3) constructs a tiling of $\mathfrak{C}_i = C(o, (2U_i + 2)\Delta)$, and computes $\mathcal{E}(S, C)$ for each cube C in this tiling. By modifying the algorithm so that it computes $\mathcal{E}(S, C)$ only for the new cubes C in the tiling (that is, ignoring cubes that are covered by cubes produced in earlier iterations), it follows that the running time of the i-th iteration can be improved to $O\left(\frac{n}{\varepsilon^d}\left(1 + \log \frac{U_i}{U_{i-1}}\right)\right)$, for $i = 2, \ldots, m$. Overall, the running time of the algorithm is thus

$$O\left(\frac{n}{\varepsilon^d} \log U_1 + \sum_{i=2}^{m} \frac{n}{\varepsilon^d}\left(1 + \log \frac{U_i}{U_{i-1}}\right)\right)$$
$$= O\left(\frac{n}{\varepsilon^d}(m + \log U_m)\right) = O\left(\frac{n}{\varepsilon^d} \log \frac{\Delta}{\omega^* \varepsilon}\right).$$

The other time bound follows if we execute Approx_Shell using the second algorithm of Theorem 3.3. □

4 Approximation Algorithms in the Plane

Let S be a set of n points in the plane. We first present an $O(n \log n)$-time algorithm that computes an annulus containing S whose width is at most $2\omega^*$. We then describe an algorithm that, given a parameter $\varepsilon > 0$, computes in $O(n \log n + n/\varepsilon^2)$ time an annulus containing S whose width is at most $(1 + \varepsilon)\omega^*$.

4.1 A 2-approximation algorithm

We first compute the width width(S) of S (i.e., the minimum distance between a pair of parallel lines that contain S between them). Next, we compute a diametral pair of S, i.e., a pair $p, q \in S$ such that $d(p,q) = \mathrm{diam}(S) \equiv \max_{p',q' \in S} d(p', q')$. [bor]Is this the only place where we use $\equiv$ to denote definition?[is] Both of these steps take $O(n \log n)$ time. [bor]Should we cite ancient width or diameter algorithms?[is] Let ℓ be the perpendicular bisector of pq. We compute $\mathrm{Vor}_N(S, \ell)$ and $\mathrm{Vor}_F(S, \ell)$, merge the vertices of the two diagrams into a single sorted list V, and compute the point v^* that minimizes $\omega(v)$ over all $v \in \ell$. The latter stages can be done in $O(|V|)$ time because, between any pair of successive points of V, $\omega(v)$ coincides with the difference of distances to two fixed points of S [bor]rephrased[is]. If width$(S) \geq \omega(v^*)$, we return $\mathcal{A}(v^*)$; otherwise, we return a strip of width width(S) that contains S. The algorithm obviously returns an annulus that contains S, and it runs in $O(n \log n)$ time.

←
←
←

Theorem 4.1 *The width of the annulus computed by the above algorithm is at most $2\omega^*$. That is,*

$$\min\{\omega(v^*), \text{width}(S)\} \leq 2\omega^*.$$

Remark 4.2 An easy calculation, which is based on area considerations and uses the fact that pq is a diameter, shows that S can be covered by a strip of width at most $2\,\text{width}(S)$ and bounding lines parallel to pq. Therefore, $\omega(v^*) \leq 2\,\text{width}(S)$, which, in view of Theorem 4.1, implies that $\omega(v^*) \leq 4\omega^*$, so that skipping the width computation in the algorithm gives a 4-approximation of ω^*.

Let $\Delta = \text{diam}(S)$. Let C_O and C_I be the outer and inner circles of an annulus $\mathcal{A}^*$ of width ω^* that contains S, and let c be the center of $\mathcal{A}^*$ (we can clearly assume that c is not at infinity). Let p, q be the diametral pair computed by the algorithm. Without loss of generality, we can assume that c is the origin, $p = (0, 1)$, $1 = d(c, p) \geq d(c, q)$, and $x(q) \geq 0$ (see Figure 5). Let D be the circle of radius $d(p, q) = \Delta$ centered at p.

Lemma 4.3 *If $\Delta \leq 1$, then S in contained in a horizontal strip of width at most $\omega^* + \Delta^2/2$.*

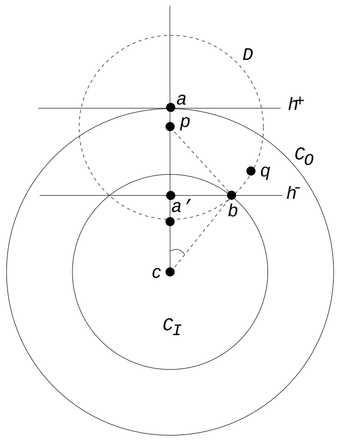

Figure 5: The minimum-width annulus and the strip defined by h^-, h^+.

Proof: Let a be the topmost point of C_O. Since $\Delta \leq 1$, $c \notin \text{int}(D)$, which implies that either D lies fully above C_I (i.e., the horizontal line passing through the topmost point of C_I strictly separates D and C_I) or ∂D and C_I intersect at two points with positive y-coordinates; the case in which ∂D and C_I touch can be handled by essentially the same argument. The first situation is impossible: since $S \subseteq D$, we can grow C_I and still have S lie in the shrunken annulus, contrary to the minimality of $\mathcal{A}^*$. Let b be the intersection point of ∂D and C_I lying to the right of the y-axis. Let h^-, h^+ be the horizontal lines passing through b and a, respectively. Since $S \subseteq \mathcal{A}^* \cap D$, the strip bounded by h^-, h^+ contains S; see Figure 5. Let a' be the intersection point of h^- and the y-axis. Then

$$\begin{aligned} d(a', c) &= d(c, b)\cos(\measuredangle bcp) \\ &= d(c, b)\frac{d(p, c)^2 + d(c, b)^2 - d(p, b)^2}{2d(p, c)d(c, b)} \\ &= \frac{1 + r_I^2 - \Delta^2}{2}, \end{aligned}$$

by the law of cosines, where r_I is the radius of C_I. Therefore the width of the strip is

$$\begin{aligned} d(a, c) - d(a', c) &= r_I + \omega^* - \frac{1 + r_I^2 - \Delta^2}{2} \\ &= \omega^* + \frac{\Delta^2}{2} - \frac{(1 - r_I)^2}{2} \le \omega^* + \frac{\Delta^2}{2}. \end{aligned}$$

□

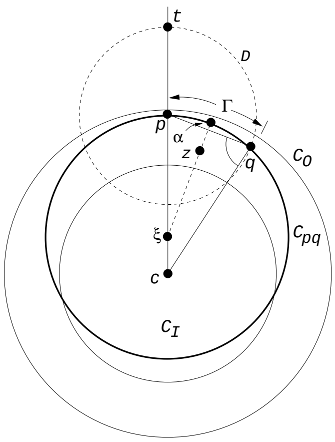

Figure 6: The minimum-width annulus and the circle C_{pq}.

Hence, if $\Delta \le 1$ and $\omega^* \ge \Delta^2/2$, the algorithm computes an annulus (that is, a strip) of width at most $2\omega^*$. We now assume that either $\Delta > 1$ or $\omega^* < \Delta^2/2$.

Let C_{pq} be the circle that passes through p and q and whose center ξ lies on the y-axis; see Figure 6. We will show that all points of S lie within distance ω^* from C_{pq}, which implies that the annulus centered at ξ with the inner radius $d(\xi, p) - \omega^*$ and the outer radius $d(\xi, p) + \omega^*$ contains S. Since ξ lies on the perpendicular bisector of pq, the thinnest annulus that the algorithm computes is certainly no wider than $\mathcal{A}(\xi)$, i.e., its width is at most $2\omega^*$.

Since $d(c,p) \geq d(c,q)$, C_{pq} lies inside the circle passing through p and centered at c, and therefore it also lies inside C_O. But C_{pq} may intersect C_I (as in Figure 6). Let $\Gamma \subseteq C_{pq}$ be the circular arc from p to q in the clockwise direction. A simple calculation shows that the distance from c to the points of Γ decreases monotonically along Γ. Since $p, q \in \mathcal{A}^*$, the entire arc Γ lies inside $\mathcal{A}^*$.

Lemma 4.4 *If* $\Delta > 1$ *or* $\omega^* < \Delta^2/2$, *then* $\measuredangle pqc < \pi/2$.

Proof: If $\Delta > 1$, then $c \in \mathrm{int}(D)$. We then have $\measuredangle pqc < \measuredangle pqm < \measuredangle tqm = \pi/2$, where bor "m" is not on the picture so asking the reader to consult it is kind of odd is m ⟵ is the bottommost point of D; consult Figure 6. Next, assume that $\omega^* < \Delta^2/2$. Since $d(c,p) = 1$, $d(p,q) = \Delta$, and $1 \geq d(c,q) \geq 1 - \omega^*$, we obtain

$$\begin{aligned}
\cos(\measuredangle pqc) &= \frac{d(p,q)^2 + d(c,q)^2 - d(c,p)^2}{2d(p,q)d(c,q)} \\
&= \frac{\Delta^2 + d(c,q)^2 - 1}{2\Delta d(c,q)} \\
&\geq \frac{\Delta^2 + (1-\omega^*)^2 - 1}{2\Delta} \\
&= \frac{\Delta^2 - 2\omega^* + \omega^{*2}}{2\Delta} \\
&> 0.
\end{aligned}$$

The last inequality follows from the assumption that $\omega^* < \Delta^2/2$. This completes the proof of the lemma. □

We now prove that for any point $z \in S$, the distance $d(z, C_{pq})$ between C_{pq} and z is at most ω^*. We will prove the claim for points with positive x-coordinates; the same argument applies to points with negative x-coordinates. Let α be the intersection point of C_{pq} with the ray emanating from ξ in direction $\vec{\xi z}$; see Figure 6. Then $d(z, C_{pq}) = d(z, \alpha)$.

If $z \in \mathrm{int}(C_{pq})$, then let β be the intersection point of C_{pq} with the ray emanating from z in direction $\vec{cz}$ (see Figure 7); otherwise, let β be the intersection point of C_{pq} with the ray emanating from z in direction $\vec{zc}$. The point β exists since c lies inside C_{pq}, as $\measuredangle pqc < \pi/2$. Since α lies on the line passing through z and the center of C_{pq}, i.e., α is the nearest point on C_{pq} from z, $d(z,\alpha) \leq d(z,\beta)$.

Lemma 4.5 $d(z,\beta) < \omega^*$.

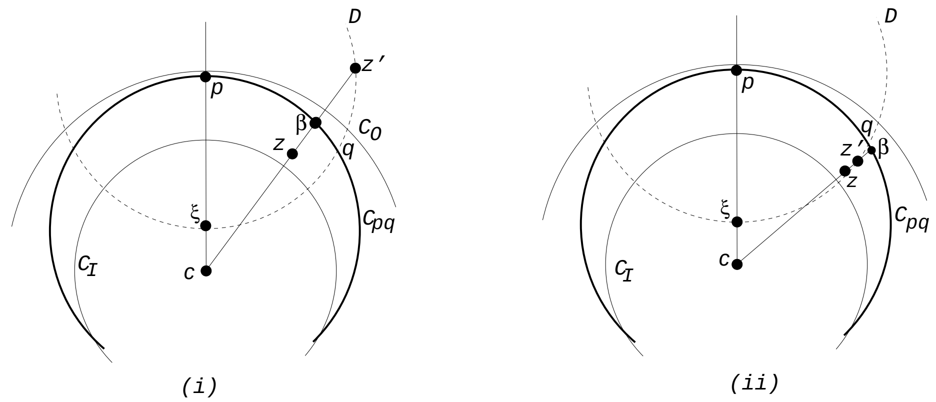

Figure 7: Illustration of the proof of Lemma 4.5. (i) $z' \in D[a,q]$, (ii) $z' \notin D[a,q]$.

Proof: We will prove that β lies in the annulus $\mathcal{A}^*$. Let z' be the intersection point of D with the ray $\vec{cz}$. [bor]I am confused. Why is there only one such intersection? Aren't there always two and you always take the second one? Help![is] ⟵ For two points $x, y \in D$, let $D[x,y] \subseteq D$ denote the circular arc from x to y in the clockwise direction. Let t be the topmost point of D. There are two cases to consider:

Case (i) $z' \in D[t,q]$. By Lemma 4.4, $\measuredangle pqc < \pi/2$, therefore $D[t,q]$ lies in the wedge formed by the positive y-axis and the ray $\vec{cq}$. This in turn implies that $\beta \in \Gamma$ irrespective of whether z lies inside or outside C_{pq}; see Figure 7(i). As noted earlier, $\Gamma \subset \mathcal{A}^*$, so $\beta \in \mathcal{A}^*$, as claimed.

Case (ii) $z' \notin D[t,q]$. Note that q is an intersection point of circles D and C_{pq} and their second point of intersection is the mirror image of q on the other side of y-axis. Therefore the portion of D from q to its bottommost point in the clockwise direction lies inside C_{pq}. Since z' has positive x-coordinate and $z' \notin D[t,q]$, z' lies on the portion [bor]Only if z' is the SECOND intersection point![is] of ∂D inside ⟵ C_{pq}. Therefore β lies after z' on the ray $\vec{cz}$ (see Figure 7(ii)) and

$$r_I \le d(c,z) \le d(c,z') < d(c,\beta) < r_O,$$

where the last inequality follows from the fact that $C_{pq} \subset \text{int}(C_O)$. This implies that $\beta \in \mathcal{A}^*$, as desired.

We thus have $d(z,\beta) < \omega^*$. □

Lemmas 4.3 and 4.5 imply the theorem.

4.2 A $(1+\varepsilon)$-approximation algorithm

In this subsection, we present a $(1+\varepsilon)$-approximation algorithm for the minimum-width annulus. The algorithm is a combination of the approximation techniques developed in the previous subsections.

Algorithm PLANAR_APPROX_SHELL (S, ε)

1. Run the 2-approximation algorithm of Theorem 4.1. Let $\mathcal{A}'$ be the resulting annulus. If the width ω' of $\mathcal{A}'$ is 0 then return $\mathcal{A}'$.

2. Compute the nearest- and farthest-neighbor Voronoi diagrams $\mathrm{Vor}_F(S), \mathrm{Vor}_N(S)$, in $O(n \log n)$ time.

3. Compute, in $O(n \log n + (n/\varepsilon) \log U)$ time, an annulus $\mathcal{A}''$ of width $\le (1 + \varepsilon/2)\omega^*(S, U)$, using the algorithm of Theorem 3.4, with $U = 10000/\varepsilon$. (Either $\mathcal{A}''$ is the required ε-approximation, or $r_{\mathrm{mid}}(\mathcal{A}^*(S)) > U\Delta(S)$.)

4. Compute, in $O(n \log n)$ time, a pair of points $p, q \in S$ that realize the diameter of S. We assume without loss of generality that $p = (-1, 0), q = (1, 0)$. Let $\delta = \varepsilon\omega'/20$, Let $P_p = P(p, \delta, \varepsilon)$, $P_q = P(q, \delta, \varepsilon)$, where

$$P(z, \delta, \varepsilon) = \left\{ z + (0, \delta)i \;\middle|\; i = -\lceil 40/\varepsilon \rceil, \ldots, \lceil 40/\varepsilon \rceil \right\}.$$

   See Figure 8.

5. For each pair $u \in P_p, v \in P_q$ compute the minimum-width annulus whose center lies on the perpendicular bisector of uv. Using the precomputed $\mathrm{Vor}_F(S)$ and $\mathrm{Vor}_N(S)$, this takes $O(n)$ time per pair, as in the algorithm of Theorem 3.3.

6. Output the minimum-width annulus among those computed.

Theorem 4.6 *The width of the annulus output by* PLANAR_APPROX_SHELL (S, ε) *is at most* $(1+\varepsilon)\omega^*(S)$*, and the running time of the algorithm is* $O(n \log n + n/\varepsilon^2)$.

Proof: If $r_{\mathrm{mid}}(\mathcal{A}^*(S)) \le U\Delta(S)$, the correctness and the bound on the running time are consequences of the previous algorithms, so assume that $r_{\mathrm{mid}}(\mathcal{A}^*(S)) > U\Delta(S)$. Let C^* be the middle circle of $\mathcal{A}^*(S)$, and let c^*, r^* denote the center and the radius of C^*, respectively. Without loss of generality, assume that c^* lies (far away) below the x-axis. Let I_p and I_q denote the segments spanned by the points of P_p and of P_q, respectively.

We have that $\omega^*(S) < \Delta(S)/300$ (otherwise, by Lemma 3.5, $\mathcal{A}''$ is the required approximation), which implies that both I_p, and I_q are "short" compared to the

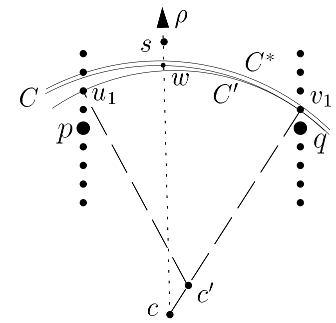

Figure 8: Proof of correctness of PLANAR_APPROX_SHELL

diameter of S. Moreover, the radius of the optimal solution is huge (i.e., at least $(10000/\varepsilon)\Delta(S)$); namely, the sector of the optimal annulus that contains S spans a very small angle. ⟵

[bor]Why exactly can't it miss?[is] It is clear that C^* crosses both I_p and I_q, at two respective points u, v. Let u_1 (resp. v_1) denote the point of P_p (resp. of P_q) that lies immediately below u (resp. v). We first translate C^* downwards, till it first hits either u_1 or v_1. Suppose, without loss of generality, that it first hits v_1. Let C denote the translated circle. Clearly, the center c of C lies vertically below c^* at distance less than δ. In particular, for any $s \in S$ we have $|d(c, s) - d(c^*, s)| \leq d(c, c^*) < \delta$. Put $D(C, S) = \max_{s \in S} d(C, s)$, and $\omega = 2D(C, S)$ and observe that

$$\omega < 2(D(C^*, S) + \delta) = \omega^* + 2\delta \leq (1 + \varepsilon/5)\omega^*.$$

Next, shrink C by moving its center from c towards v_1 while keeping v_1 on the circle, until it also passes through u_1. Let C' denote the new circle and let c' denote its center. See Figure 8.

The distance from c to points on C' decreases monotonically as we traverse C' from v_1 counterclockwise until we reach the point on C' antipodal to v_1. Let s be any point of S. The ray ρ from c towards s crosses C at a point w and C' at a point w'. We have $d(w', s) \leq d(w, s) + d(w, w') \leq \omega/2 + d(w, w')$. It easily follows from the preceding discussion that $d(w, w')$ attains its maximum when w' is near u_1, [bor]Should we add that the logic also works CLOCKWISE of v_1, but we do not have far to go? Literally taken, we have no argument for the other side of v_1 now![is] and this maximum is smaller than 2δ (the later statement is easy to verify, using the fact that the line through w and w' is almost vertical). This implies that ⟵

$$\omega(c') \leq 2D(C', S) \leq \omega + 2\delta \leq (1 + 2\varepsilon/5)\omega^* \leq (1 + \varepsilon)\omega^*.$$

Since c' lies on the perpendicular bisector of $u_1 v_1$, it follows that the width of the annulus output by the algorithm is at most $\omega(c') < (1 + \varepsilon)\omega^*$, as asserted. The bound on the running time is obvious: We have $O(1/\varepsilon^2)$ bisectors to process, and the processing of each of them takes $O(n)$ time, as noted in the algorithm. $\square$

5 Conclusions

We presented simple and efficient approximation algorithms for computing the minimum-width shell containing a set of points in $\mathbb{R}^d$. Although several approximation algorithms were proposed earlier for the planar case, all of them made some assumptions either on the input points or on the minimum-width annulus. In an earlier version of this paper [1], we also presented the first subcubic algorithm for computing a minimum-width shell containing a set of points in $\mathbb{R}^3$. The algorithms was fairly involved and mostly interesting as a confirmation that the problem can be solved in subcubic time. Since then we have learned that a significantly simpler *quadratic* algorithm exists for solving the problem [11]. It was noticed by T. Chan, who also proposes several improvements over the approximation algorithms we described above [11]. $^{\text{bor}}$Is this enough?$_{\text{is}}$ $\longleftarrow$

- Can the running time of our planar approximation algorithm be improved to $O(n \log n + 1/\varepsilon^2)$?
- Can the minimum-width shell containing a set of points in $\mathbb{R}^3$ be computed in near-quadratic time? $^{\text{bor}}$I guess that's settled!$_{\text{is}}$ $\longleftarrow$
- Develop an efficient as algorithm for computing the minimum-width cylindrical shell containing a set of points in $\mathbb{R}^3$. $^{\text{bor}}$Same here? Doesn't a simple exact quadratic algorithm follow from Timothy's stuff?$_{\text{is}}$ $\longleftarrow$

References

[1] P. K. Agarwal, B. Aronov, S. Har-Peled, and M. Sharir, Approximation and exact algorithms for minimum-width annuli and shells, *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, 1999, pp. 380–389.

[2] P. K. Agarwal, B. Aronov, and M. Sharir, Computing envelopes in four dimensions with applications, *SIAM J. Comput.*, 26 (1997), 1714–1732.

[3] P. K. Agarwal, H. Edelsbrunner, O. Schwarzkopf, and E. Welzl, Euclidean minimum spanning trees and bichromatic closest pairs, *Discrete Comput. Geom.*, 6 (1991), 407–422.

[4] P. K. Agarwal and J. Erickson, Geometric range searching and its relatives, in: *Advances in Discrete and Computational Geometry* (B. Chazelle, J. E. Goodman and R. Pollack, eds.), AMS Press, Providence, RI, 1998, pp. 1–56.

[5] P. K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11 (1994), 393–418.

[6] P. K. Agarwal and M. Sharir, Efficient randomized algorithms for some geometric optimization problems, *Discrete Comput. Geom.*, 16 (1996), 317–337.

[7] P. K. Agarwal, M. Sharir, and S. Toledo, Applications of parametric searching in geometric optimization, *J. Algorithms*, 17 (1994), 292–318.

[8] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor, A linear-time algorithm for computing the Voronoi diagram of a convex polygon, *Discrete Comput. Geom.*, 4 (1989), 591–604.

[9] F. Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, 1991.

[10] T. M. Chan, Geometric applications of a randomized optimization technique, *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, 1998, pp. 269–278.

[11] T. M. Chan, Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus, *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, 2000, to appear.

[12] B. Chazelle, H. Edelsbrunner, L. J. Guibas, and M. Sharir, A singly-exponential stratification scheme for real semi-algebraic varieties and its applications, *Theoret. Comput. Sci.*, 84 (1991), 77–105.

[13] M. de Berg, J. Bose, D. Bremner, S. Ramaswami, and G. Wilfong, Computing constrained minimum-width annuli of point sets, *Proc. 5th Workshop Algorithms Data Struct.*, *Lecture Notes Comput. Sci.*, Vol. 1272, Springer-Verlag, 1997, pp. 3–16.

[14] H. Brönnimann, B. Chazelle, and J. Matoušek, Product range spaces, sensitive sampling, and derandomization, *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, 1993, pp. 400–409.

[15] D. P. Dobkin and D. G. Kirkpatrick, A linear algorithm for determining the separation of convex polyhedra, *J. Algorithms*, 6 (1985), 381–392.

[16] C. A. Duncan, M. T. Goodrich, and E. A. Ramos, Efficient approximation and optimization algorithms for computational metrology, *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, 1997, pp. 121–130.

[17] M. Dyer and N. Megiddo, Linear programming in low dimensions, in: *Handbook of Discrete and Computational Geometry* (J. E. Goodman and J. O'Rourke, eds.), CRC Press LLC, Boca Raton, FL, 1997, pp. 699–710.

[18] H. Ebara, N. Fukuyama, H. Nakano, and Y. Nakanishi, Roundness algorithms using the Voronoi diagrams, *Abstracts 1st Canad. Conf. Comput. Geom.*, 1989, p. 41.

[19] H. Edelsbrunner, L. J. Guibas, and J. Stolfi, Optimal point location in a monotone subdivision, *SIAM J. Comput.*, 15 (1986), 317–340.

[20] O. Egecioglu and B. Kalantari, Approximating the diameter of a set of points in the Euclidean space, *Inform. Process. Lett.*, 32 (1989), 205–211.

[21] L. W. Foster, *GEO-METRICS II: The Application of Geometric Tolerancing Techniques*, Addison-Wesley, Reading, MA, 1982.

[22] J. García-Lopez and P. Ramos, Fitting a set of points by a circle, *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, 1997, pp. 139–146.

[23] S. Har-Peled. Constructing approximate shortest path maps in three dimensions. *SIAM J. Comput.*, 28(4):1182–1197, 1999.

[24] D. Haussler and E. Welzl, Epsilon-nets and simplex range queries, *Discrete Comput. Geom.*, 2 (1987), 127–151.

[25] R. Kumar, and D. Sivakumar, Roundness estimation via random sampling, *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, 1999, pp. 603–612.

[26] V. B. Le and D. T. Lee, Out-of-roundness problem revisited, *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-13 (1991), 217–223.

[27] J. Matoušek, M. Sharir, and E. Welzl, A subexponential bound for linear programming, *Algorithmica*, 16 (1996), 498–516.

[28] K. Mehlhorn, T. Shermer, and C. Yap, A complete roundness classification procedure, *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, 1997, pp. 129–138.

[29] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou, The discrete geodesic problem, *SIAM J. Comput.*, 16 (1987), 647–668.

[30] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.

[31] T. J. Rivlin, Approximating by circles, *Computing*, 21 (1979), 93–104.

[32] U. Roy, C. R. Liu, and T. C. Woo, Review of dimensioning and tolerancing: representation and processing, *Comput. Aided Design*, 23 (1991), 466–483.

[33] U. Roy and X. Zhang, Establishment of a pair of concentric circles with the minimum radial separation for assessing roundness error, *Comput. Aided Design*, 24 (1992), 161–168.

[34] M. Sharir and P. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, New York, 1995.

[35] T. C. Shermer and C. K. Yap, Probing for near centers and relative roundness, *Proc. ASME Workshop on Tolerancing and Metrology*, 1995.

[36] M. Smid and R. Janardan. On the width and roundness of a set of points in the plane. *Int. J. Comput. Geom. Appls.*, 9(1):97–108, 1999.

[37] K. Swanson, D. T. Lee, and V. L. Wu, An optimal algorithm for roundness determination on convex polygons, *Comput. Geom. Theory Appl.*, 5 (1995), 225–235.

[38] C. K. Yap and E.-C. Chang, Issues in the metrology of geometric tolerancing, *Algorithms for Robotic Motion and Manipulation* (J.-P. Laumond and M. Overmars, ed.), A.K. Peters, Wellesley, MA, 1997, pp. 393–400.