Approximating the Diameter, Width, Smallest Enclosing Cylinder, and Minimum-Width Annulus*

Timothy M. Chan†

October 23, 2000

Abstract

We study $(1+\varepsilon)$-factor approximation algorithms for several well-known optimization problems on a given n-point set: (a) diameter, (b) width, (c) smallest enclosing cylinder, and (d) minimum-width annulus. Among our results are new simple algorithms for (a) and (c) with an improved dependence of the running time on ε, as well as the first linear-time approximation algorithm for (d) in any fixed dimension. All four problems can be solved within a time bound of the form $O(n + \varepsilon^{-c})$ or $O(n \log(1/\varepsilon) + \varepsilon^{-c})$.

Keywords: geometric optimization, approximation algorithms, diameter, width, roundness

1 Introduction

The purpose of this paper is to highlight some useful techniques in the design of efficient approximation algorithms in geometric optimization. For this reason, we have chosen four simple problems, all well-studied from the perspectives of both exact and approximate computation: given a set P of n points in a fixed-dimensional Euclidean space $\mathbb{R}^d$, compute

Diameter: the maximum distance over all pairs of points in P;

Width: the minimum width over all slabs that enclose P, where a *slab* of *width* w refers to a region between two parallel hyperplanes of distance w;

Smallest enclosing cylinder: the minimum radius over all cylinders that enclose P, where a *cylinder* of *radius* z refers to the region of all points of distance z from a line;

Minimum-width annulus: the minimum width over any all annuli that enclose P, where an *annulus* (also called a *spherical shell*) of *width* $|z - y|$ is a region between two concentric spheres of radii y and z.

The last three problems are motivated from statistical analysis and computational metrology, as they respectively ask for the hyperplane, line, and sphere that best fit the data (in the sense of minimizing the maximum distance to the points).

*A preliminary version appeared in *Proc. 16th ACM Sympos. Comput. Geom.*, 2000.

†Dept. of Computer Science, Univ. of Waterloo, Waterloo, Ontario N2L 3G1, Canada, `tmchan@math.uwaterloo.ca`

It is not known how to compute an exact solution to these problems in near-linear time even for low dimensions (specifically, $d \geq 4$ for diameter, $d \geq 3$ for width and smallest enclosing cylinder, and $d \geq 2$ for minimum-width annulus). It is therefore of practical interest to look for faster algorithms that solve the problems approximately by returning a solution that is within a multiplicative factor $1+\varepsilon$ of the optimal value, where $\varepsilon > 0$ is an input parameter. Indeed, for all of the above problems, a $(1+\varepsilon)$-factor approximation can be found in time linear in n (as we will see).

An important consideration in approximation algorithms is the "constant factor" in the running time, which undoubtedly increases as we demand higher accuracy. In our time bounds, we will therefore specify the dependence on both n and ε. Since we are primarily concerned with low dimensions, we will ignore constant factors that depend on d, which usually have an exponential growth. (Efficient high-dimensional algorithms seem to require a different set of techniques; for example, see [11, 22, 24], and for convex bodies, see [25].)

In the sequel, let $E = 1/\varepsilon$, let $\delta > 0$ be an arbitrarily small fixed constant, and let the O^* notation hide $\log^{O(1)} E$ factors. Many algorithms from the literature have time bounds of the form $O^*(E^c n)$ for a small constant c (depending on d). We call such an algorithm a *linear-time approximation scheme (LTAS) of order c*. We are interested in minimizing the order c, because this number dictates how accurate an answer we can get in a reasonable amount of time. (Just imagine an application that only tolerates a relative error of 1%; here, $E = 100$ and a factor like E^2 or E^3 would be substantial.)

Somewhat surprisingly, all of the above problems have algorithms with time bounds of the form $O^*(n + E^c)$ for a constant c. We call such an algorithm a *strong LTAS of order c*. A strong LTAS is interesting, because the running time does not grow asymptotically as long as E is kept below a threshold of $n^{1/c}$. Again, we would like the order c to be as small as possible.

Known Exact Algorithms. The diameter problem has been extensively studied in computational geometry. In the plane, it is quite easy to obtain an optimal $O(n \log n)$ time bound [33]. Much effort was directed to the more difficult $d = 3$ case: Clarkson and Shor [18] were the first to obtain an optimal randomized $O(n \log n)$ algorithm; a deterministic algorithm that matches this performance was announced only recently by Ramos [35], after a long succession of work by various researchers [10, 15, 31, 34]. For $d \geq 4$, we can trivially solve the problem in quadratic time. With known data structures [28, 30] though, a slightly better bound of $O(n^{2-2/(\lceil d/2 \rceil+1)} \log^{O(1)} n)$ can be attained [4].

The width problem has also been extensively studied. Again, in the plane, it is easy to obtain an optimal $O(n \log n)$ time bound [33]. In $d = 3$, Houle and Toussaint [26] were credited as the first to obtain an $O(n^2)$-time algorithm. A series of papers derived improved subquadratic algorithms [3, 5, 6, 15], the best of which required $O(n^{3/2+\delta})$ expected time. We are not aware of any algorithms for $d \geq 4$, although $O(n^{\lceil d/2 \rceil})$ time can be immediately achieved by realizing the solution space as a convex polytope in $d+1$ variables/dimensions (see Section 3) and applying an optimal halfspace intersection algorithm [14, 18].

The smallest enclosing cylinder for $d = 2$ is identical to width, so the first nontrivial case for our third problem is $d = 3$. Two papers studied this case: one by Schömer *et al.* [37], who gave a near-quartic algorithm, and a subsequent one by Agarwal *et al.* [2], who gave a faster $O(n^{3+\delta})$ algorithm. In higher dimensions, a rough bound of $O(n^{2d-1+\delta})$ follows by realizing the solution space as a cell in an arrangement of surfaces in $2d-1$ dimensions (improvements are likely).

The minimum-width annulus problem for $d = 2$ has received much attention because of its relation to testing the roundness of a point set. Quadratic algorithms are easy to obtain by construction of

Voronoi diagrams. A series of papers derived improved subquadratic algorithms [3, 5, 6], culminating in an $O(n^{3/2+\delta})$ randomized algorithm. For $d \geq 3$, it is not difficult to achieve $O(n^{\lfloor d/2 \rfloor+1})$ time, again by realizing the solution space as a convex polytope, this time with $d+2$ variables (see Section 5). Incidently, this renders a recently published (and rather complicated) 3-dimensional $O(n^{3-1/19+\delta})$ algorithm by Agarwal *et al.* [1] unnecessary. Variants and special cases of the problem have also been addressed arising from practical consideration [9, 20, 23, 36].

Known Approximation Algorithms. For the diameter problem, it is quite easy to derive an LTAS that runs in $O(E^{(d-1)/2}n)$ time, as noted by Agarwal *et al.* [4]. Recently (apparently unaware of this), Barequet and Har-Peled [8] described another simple $(1+\varepsilon)$-approximation algorithm; in fact, their algorithm is a *strong* LTAS with a running time of $O(n+E^{2(d-1)})$, which, as they noted, can be improved slightly to $O(n+E^{2(d-1)d/(d+1)})$ if advanced data structures are used.

The approximate width problem was studied by Duncan *et al.* [20], who gave an $O(E^{(d-1)/2}n)$-time LTAS, generalizing an earlier two-dimensional idea they attributed to Janardan [27].

Schömer *et al.*'s and Agarwal *et al.*'s papers on smallest enclosing cylinders [2, 37] also contained approximation algorithms for $d=3$. Schömer *et al.*'s algorithm computes a $(1+\varepsilon)$-factor approximation in $O((EU)^2 n \log(EU))$ time, where U denotes the ratio of the diameter to the minimum cylinder-radius. Note that this ratio can be large when the input points are "almost collinear." In contrast, Agarwal *et al.*'s LTAS is independent of the ratio and takes $O(E^2 n)$ time. Higher dimensions were not discussed.

The approximate minimum-width annulus problem was attacked in the recent paper by Agarwal *et al.* [1]. They described the first near-linear-time algorithm for $d=2$, running in $O(n \log n + E^2 n)$. They also gave algorithms in higher dimensions with running time $O(E^d n \log(EU))$ and $O((E^{d-2} n \log n + E^{d-1} n)\log(EU))$. Here, U may either stand for the ratio of an upper bound on the larger sphere-radius to the diameter, or the ratio of the diameter to the minimum annulus-width. The former ratio can be unbounded if the optimal annulus approaches a slab. On the other hand, the latter ratio can be large when the input point set is "almost round" (an essential case in certain applications). Earlier attempts by Duncan *et al.* [20] again dealt only with special cases or variants of the approximation problem. A general LTAS remained incomplete.

New Approximation Algorithms. We describe new simple algorithms that yield the following for any fixed dimension d:

1. two strong LTASs for the diameter problem with running time $O(n+E^{d-0.5})$, improving the earlier $O(n+E^{2(d-1)})$ result of Barequet and Har-Peled [8];

2. an LTAS for smallest enclosing cylinders with running time $O(E^{(d-1)/2}n)$, improving and generalizing the earlier 3-dimensional $O(E^2 n)$ result by Agarwal *et al.* [2];

3. a strong LTAS for smallest enclosing cylinders, with running time $O(n+E^{3(d-1)/2})$;

4. the first LTAS for minimum-width annuli (the simplest version takes $O(E^{(d-1)/2}n + E^{4d-1})$ time, another version takes $O(E^d n \log E)$ time), improving the 2-dimensional $n \log n$ result by Agarwal *et al.* [2], and eliminating any dependence on distance ratios from their higher-dimensional results.

problem		LTAS		strong LTAS	
		previous	new	previous	new
diameter	$d=4$	1.5	0.5	4.8	2.34
	$d=5$	2	0.8	6.67	3.34
	$d \geq 6$	$\approx d/2$	$\approx d/2$	$\approx 2d$	$\approx d$
width	$d=3$	1	0.34	–	1
	$d=4$	1.5	0.75	–	2.5
	$d \geq 5$	$\approx d/2$	$\approx d/2$	–	$\approx 3d/2$
cylinder	$d=3$	2	< 1	–	< 2.5
	$d=4$	–	< 1.5	–	< 4
	$d \geq 5$	–	$\approx d/2$	–	$\approx 3d/2$
annulus	$d=2$	$1^\dagger$	0.67	–	2
	$d=3$	$2^\dagger$	1.5	–	5
	$d \geq 4$	$\approx d^\dagger$	$\approx d$	–	$\approx d^2/4$

Table 1: The order of LTASs and strong LTASs (the notation $f(d) \approx g(d)$ means $\lim_{d\to\infty} f(d)/g(d) = 1$). The "previous" column describes the best results that were explicitly stated in earlier papers (entries marked † do not technically represent LTASs, as running time has extra $\log n$ and $\log U$ factors for a certain distance ratio U). The "new" column describes the fastest theoretical results mentioned in this paper. Note that not all the new entries are obtained from new ideas (for example, some follow just by using more advanced data structures).

All of these algorithms are not difficult to implement. In Appendix A, we also establish an inequality relating the minimum-width annulus to the "minimum-area" annulus that may be of practical interest.

Since many of the previous papers expressed interest in improving the dependence of the running time on ε, we also point out (in paragraphs marked "Remark") how applying known data structures to the previous or new algorithms lead to the currently best bounds on the order of LTASs and strong LTASs. See Table 1 for a summary; for example, we can obtain a strong LTAS for the width problem in three dimensions with the running time $O^*(n+E)$. It should be emphasized that the data structures used to obtain these results are quite complicated and thus these results represent what is possible in theory only. Nevertheless, the table indicates that determining the optimal order can be nontrivial even for a simple problem like diameter. Lower bounds appear even harder and will be left as open problems.

Techniques. Our algorithms are all obtained by various combinations of known elementary techniques, for instance, of dividing space into *grid* cells, or dividing the space of directions into narrow *cones*. We find another technique to be quite powerful—namely, reducing a geometric approximation problem into a number of instances of fixed-dimensional *linear/convex programming*. These techniques should be applicable to other problems, and we hope that our study here will serve as helpful examples, and at the same time, prompt a closer examination into the dependence on ε.

2 Diameter

We begin with the simplest of the four problems, diameter, mainly to illustrate some of the techniques we are going to use throughout. We first review two almost trivial algorithms: the first based on grids (mentioned by Barequet and Har-Peled [8]), the second based on cones (mentioned by Agarwal *et al.* [4]). Interestingly, neither algorithm gives the best ε-dependence. By combining the two algorithms, we immediately obtain a (new) third algorithm, and by instead "alternating" between the two algorithms, we obtain an even better result.

Let Δ^* denote the actual diameter of our n-point set $P \subset \mathbb{R}^d$. A constant-factor approximation is easy to get in $O(n)$ time. For example, pick any point in P and let Δ_0 be its farthest-point distance. Obviously, $\Delta_0 \leq \Delta^* \leq 2\Delta_0$. We want to compute a $(1+O(\varepsilon))$-factor approximation to Δ^*.

Algorithm 1 (Grid). Here is an algorithm that easily comes to mind [8]: build a uniform grid of side length $\varepsilon\Delta_0$, round each point to the nearest grid point, then compute the diameter of these grid points. Rounding incurs an additive error of $O(\varepsilon\Delta_0) = O(\varepsilon\Delta^*)$ to the diameter, so this algorithm returns a $(1+O(\varepsilon))$-factor approximation of Δ^*.

The analysis is not difficult. Since the points lie within a ball of radius $O(\Delta^*) = O(\Delta_0)$, there are at most $O(E^d)$ grid points. Now, rounding can be done in $O(n)$ time by using the floor function (or in $O(n \log E)$ time without it). Duplicates can be removed by bucketing in $O(n + E^d)$ time. The diameter computation on the $O(E^d)$ grid points can be done by brute force in $O(E^{2d})$ time. We thus have a strong LTAS with a running time of $O(n + E^{2d})$.

A simple modification to the algorithm improves the time bound to $O(n + E^{2(d-1)})$: during rounding, only keep the topmost and bottommost grid points along each vertical line. The reason is that the diameter is unaffected when the other points are pruned. As a result, we are left with only $O(E^{d-1})$ grid points, and the time needed to generate these points is only $O(n+E^{d-1})$ by bucketing.

> **Remark:** Har-Peled (in personal communication) noted that the number of grid points can be further reduced to $O(E^{d-4/3})$ by keeping only the extreme points along each of the $O(E^{d-2})$ parallel grid planes. Since the convex hull of a point set inside an $O(E) \times O(E)$ grid has only $O(E^{2/3})$ vertices [7], each plane has $O(E^{2/3})$ extreme points, and they can be generated in $O(n + E^{d-1})$ total time by Graham scan with pre-sorting [33].
>
> He suggested that the number of grid points can be further reduced to $O(E^{d-3/2})$ by keeping extreme points along each of the $O(E^{d-3})$ parallel grid 3-flats. Each 3-flat now has $O(E^{3/2})$ extreme points [7], and they can be generated in additional $O(E^{d-4/3} \log E)$ total time by an optimal 3-dimensional convex hull algorithm [33]. As a result, the total running time of the grid algorithm is $O(n + E^{2d-3})$.
>
> Of course, one can consider this filtering trick for flats beyond dimension 3 using more complicated higher-dimensional convex hull algorithms (as mentioned in Barequet and Har-Peled's paper [8]), but this theoretical improvement is at best minor and will not be considered here.

Algorithm 2 (Cones + RS). Let $\theta_\varepsilon = \arccos(1/(1+\varepsilon)) = \Theta(\sqrt{\varepsilon})$. The second algorithm [4] is based on the well-known observation [40] that the space of directions can be covered by $O(1/\theta_\varepsilon^{d-1})$ cones of angle θ_ε. In other words, we can form a set V_d of $O(1/\theta_\varepsilon^{d-1})$ unit vectors in $\mathbb{R}^d$ satisfying the following property: for every $x \in \mathbb{R}^d$, there exists $a \in V_d$ such that the angle $\angle(a, x)$ is at most θ_ε.

Note that $\cos \angle(a, x) = a \cdot x \,/\, \|x\|$. The desired property of V_d can therefore be paraphrased as follows: for every $x \in \mathbb{R}^d$,

$$\|x\|/(1+\varepsilon) \;\le\; \max_{a \in V_d} a \cdot x \;\le\; \|x\|. \tag{1}$$

Now, we want a pair of points $p, q \in P$ to maximize $\|p-q\|$. By (1), a $(1+\varepsilon)$-factor approximation can be found by maximizing $a \cdot (p-q)$ over all $p, q \in P$ and $a \in V_d$. This value can be computed by determining for each $a \in V_d$ the point $p \in P$ that maximizes $a \cdot p$ and the point $q \in P$ that minimizes $a \cdot q$. In other words, we have reduced our problem to finding extreme points of P along $O(1/\theta_\varepsilon^{d-1}) = O(E^{(d-1)/2})$ query directions. In the dual, these queries are more commonly known as ray shooting (RS) in a convex polytope.

The trivial method for RS (and the one most suitable for implementation) requires $O(n)$ time per query. So, the running time of this simple algorithm, an LTAS, is $O(E^{(d-1)/2}n)$.

> **Remark:** With advanced data structures [28, 30], we can answer m RS queries on a d-dimensional convex polytope defined by n halfspaces in $O(t_d(n, m))$ time [13], where
>
> $$t_d(n, m) \;:=\; n \log m + (nm)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n + m \log^{O(1)} n. \tag{2}$$
>
> So, the time bound of the algorithm can be improved to $O(t_d(n, E^{(d-1)/2}))$. By straightforward calculations,
>
> $$\min\{n^\alpha, t_d(n, E^\beta)\} \;=\; \begin{cases} O^*(E^{\beta(\alpha-1)\lfloor d/2 \rfloor/(\alpha+(\alpha-1)\lfloor d/2 \rfloor)}n) & \text{if } \alpha \le \lfloor d/2 \rfloor \\ O^*(E^{\beta(\alpha-1)/\alpha}n) & \text{otherwise,} \end{cases} \tag{3}$$
>
> and
>
> $$t_d(n, E^\beta) \;=\; O^*(n + E^{\beta \lfloor d/2 \rfloor}). \tag{4}$$
>
> This yields our fastest LTASs in theory for $d = 4, 5$ as shown in Table 1, since we can bound $\min\{n^{4/3}, t_4(n, E^{3/2})\}$ by $O^*(\sqrt{E}n)$, and we can bound $\min\{n^{3/2}, t_5(n, E^2)\}$ by $O^*(E^{4/5}n)$.

Algorithm 3 (Algorithm 1 + Algorithm 2). Recall that Algorithm 1 reduces the problem to one involving $O(E^{d-1})$ grid points. Instead of applying a brute-force quadratic algorithm to these grid points, we can apply Algorithm 2. (The underlying principle: a $(1+\varepsilon)$-factor approximation of a $(1+\varepsilon)$-factor approximation is a $(1+O(\varepsilon))$-factor approximation.) We thus obtain a more efficient strong LTAS, with a time bound of $O(n + E^{3(d-1)/2})$.

> **Remark:** With advanced data structures, the time bound is $O(n + t_d(E^{d-1}, E^{(d-1)/2}))$ and can be reduced slightly to $O(n + E^{d-1} + t_d(E^{d-3/2}, E^{(d-1)/2}))$ by the first remark. Alternatively, we can consider the following idea (which is better in low dimensions). After rounding, our point set can be decomposed into $O(E^{d-3})$ subsets, each lying in a grid 3-flat. An RS query can be accomplished by querying each of these 3-dimensional subsets separately. The time bound can therefore be rewritten as $O(\sum_{i=1}^{O(E^{d-3})} t_3(n_i, E^{(d-1)/2}))$, where $\sum_i n_i = n$. This is $O^*(n + E^{(3d-7)/2})$.

Algorithm 4 (Grid + Cones + Dimension Reduction). A more efficient strong LTAS is based on the following idea: first apply the grid scheme of Algorithm 1 to reduce the size of the point set to $O(E^{d-1})$; but instead of using d-dimensional cones to solve this reduced problem directly, use 2-dimensional cones to reduce the problem to a number of $(d-1)$-dimensional subproblems and solve these subproblems recursively. The subproblems are formed by projections to $O(E^{1/2})$ hyperplanes corresponding to the 2-dimensional cone directions.

To be precise, given a point x, let x_i denote its i-th coordinate. Now, (1) in $\mathbb{R}^2$ tells us that

$$(x_1^2 + x_2^2)/(1+\varepsilon)^2 \;\le\; \max_{a \in V_2} (a_1 x_1 + a_2 x_2)^2 \;\le\; x_1^2 + x_2^2.$$

Define the projection $\pi_a : \mathbb{R}^d \to \mathbb{R}^{d-1}$: $\pi_a(x) = (a_1 x_1 + a_2 x_2, x_3, \ldots, x_d) \in \mathbb{R}^{d-1}$. The above implies that for every $x \in \mathbb{R}^d$,

$$\|x\|^2/(1+\varepsilon)^2 \;\le\; \max_{a \in V_2} \|\pi_a(x)\|^2 \;\le\; \|x\|^2.$$

Consequently, to find a $(1+O(\varepsilon))$-factor approximation to the diameter of $P \subset \mathbb{R}^d$, it suffices to find a $(1+O(\varepsilon))$-factor approximation to the maximum of $\|\pi_a(p-q)\| = \|\pi_a(p) - \pi_a(q)\|$ over all $p, q \in P$ and $a \in V_2$. In other words, it suffices to approximate recursively the diameter of the projected point set $\pi_a(P) \subset \mathbb{R}^{d-1}$ over each of the vectors $a \in V_2$.

Let $T_d(n)$ be the running time in d dimensions. Since the grid scheme reduces n to $O(E^{d-1})$ and $O(1/\theta_\varepsilon) = O(E^{1/2})$ subproblems in $d-1$ dimensions are generated, we have a recurrence

$$T_d(n) \;=\; O(n + E^{1/2} T_{d-1}(O(E^{d-1}))),$$

which solves to $T_d(n) = O(n + E^{d-1/2})$. We therefore obtain a strong LTAS of order $d - 1/2$.

> **Remark:** As Har-Peled informed the author, by the first remark, the recurrence can be rewritten as $T_d(n) = O(n + E^{d-1} + E^{1/2} T_{d-1}(O(E^{d-3/2})))$, which solves to $T_d(n) = O(n + E^{d-1})$. As we will see, a different approach also enables a similar small improvement.

Algorithm 5 (Grid + Cones + Dimension Reduction). We point out another combination of grid, cones, and induction that lead basically to the same result. The idea is a more refined way to quickly reduce the number of points. We first introduce a definition: given d-dimensional subset $P' \subset P$, we say that P' $(1+\varepsilon)$-*simplifies* P if for any $q \in \mathbb{R}^d$,

$$\max_{p' \in P'} \|p' - q\| \;\ge\; \max_{p \in P} \|p - q\|/(1+\varepsilon).$$

Initially, we form the set P of grid points from Algorithm 1. Next, we find a subset P' of size $O(E^{(d-1)/2})$ that $(1+O(\varepsilon))$-simplifies P, as described by a recursive procedure below. Then by Algorithm 2 in $O(E^{(d-1)/2}|P'|) = O(E^{d-1})$ time, we return a $(1+O(\varepsilon))$-factor approximation to the diameter of P', which is easily seen to be a $(1+O(\varepsilon))$-factor approximation to the diameter of P.

To construct P', we first decompose the grid point set P into $O(E)$ subsets P_i each lying in a grid hyperplane. We then recursively find a subset P_i' of size $O(E^{(d-2)/2})$ that $(1+O(\varepsilon))$-simplifies P_i. A simple argument shows that $\bigcup_i P_i'$ $(1+O(\varepsilon))$-simplifies P: given $q \in \mathbb{R}^d$, say its farthest neighbor p in P lies in the grid hyperplane h containing P_i, and let q_0 be the projection of q to h; then

$$\begin{aligned} \|p - q\|^2 &= \|p - q_0\|^2 + \|q_0 - q\|^2 \;\le\; (1+O(\varepsilon))^2 \max_{p' \in P_i'} \|p' - q_0\|^2 \;+\; \|q_0 - q\|^2 \\ &\le (1+O(\varepsilon))^2 \max_{p' \in P_i'} \|p' - q\|^2. \end{aligned}$$

Now, $\bigcup_i P_i'$ has size $O(E^{d/2})$, which is still too large, so we employ cones to construct a smaller simplifying subset P' from this set: for each $a \in V_d$, find the point $p_a \in \bigcup_i P_i'$ that maximizes $a \cdot p_a$

and form $P' = \{p_a \mid a \in V_d\}$ of size $O(E^{(d-1)/2})$, computable by RS in time $O(E^{(d-1)/2}|\bigcup_i P_i'|) = O(E^{d-1/2})$. It is easy to see from (1) that P' $(1+\varepsilon)$-simplifies $\bigcup_i P_i'$: for any $q \in \mathbb{R}^d$ and $p \in \bigcup_i P_i'$,

$$\begin{aligned} \|p - q\| &\leq (1+\varepsilon) \max_{a \in V_d} a \cdot (p - q) \leq (1+\varepsilon) \max_{a \in V_d} a \cdot (p_a - q) \\ &\leq (1+\varepsilon) \max_{a \in V_d} \|p_a - q\|. \end{aligned}$$

By transitivity, P' $(1+O(\varepsilon))$-simplifies P, as desired.

Let $T_d(n)$ be the running time of the above recursive procedure on an n-point set in a d-dimensional grid. We have the recurrence

$$T_d(n) = \sum_{i=1}^{O(E)} T_{d-1}(n_i) + O(E^{d-1/2}),$$

where $\sum_i n_i = n$. The overall running time is therefore $O(n + E^{d-1/2})$.

Remark: With advanced data structures, the recurrence lowers to $T_d(n) = \sum_{i=1}^{O(E)} T_{d-1}(n_i) + O(t_d(E^{d/2}, E^{(d-1)/2}))$. We can easily check from (2) that $t_d(E^{d/2}, E^{(d-1)/2})$ is at most $O^*(E^{d-5/3})$ for $d \geq 4$, so our best strong LTAS has an $O^*(n + E^{d-5/3})$ time bound, as shown in Table 1.

3 Width

For the width problem, we will not give a new approximation algorithm but rather present a quick reinterpretion of the previous algorithm by Duncan *et al.* [20] so as to set up the basic approach for the subsequent problems.

We first formulate the optimization problem by using $d+2$ variables $x \in \mathbb{R}^d$ and $y, z \in \mathbb{R}$ to parametrize the two parallel hyperplanes that bound the desired slab: $\{\xi \in \mathbb{R}^d \mid x \cdot \xi = y\}$ and $\{\xi \in \mathbb{R}^d \mid x \cdot \xi = z\}$. (Note that one variable can be be eliminated, e.g., by setting $z = y + 1$.) The width of the slab is $|z - y|/\|x\|$. Thus, we want to

$$\begin{array}{ll} \text{minimize} & (z-y)/\|x\| \\ \text{subject to} & y \leq x \cdot p \leq z \quad (\forall p \in P) \\ & x \in \mathbb{R}^d,\ y, z \in \mathbb{R}. \end{array}$$

Although the constraints are all linear, this is not an instance of linear or convex programming, because of the objective function. The obvious way to find the exact optimum is to construct the entire feasible region (a $(d+1)$-dimensional convex polytope, after eliminating a variable), which require $O(n^{\lceil d/2 \rceil})$ time in the worst case (probably less in practice, though). The region can then be triangulated, and the objective function can be optimized within each simplex in constant time.

Algorithm 1 (Cones + LP). If we just want an approximation, a faster algorithm can be obtained by replacing this optimization problem with a number of linear programming problems. The idea is one that we have seen before, namely cones. By (1), the following yields a $(1+\varepsilon)$-factor approximation:

$$\begin{array}{ll} \text{minimize} & (z-y)/|a \cdot x| \\ \text{subject to} & y \leq x \cdot p \leq z \quad (\forall p \in P) \\ & x \in \mathbb{R}^d,\ y, z \in \mathbb{R},\ a \in V_d. \end{array}$$

Changing variables, $X = x/(z-y)$ and $Y = y/(z-y)$, we see that this reduces to a $(d+1)$-dimensional linear program (LP) for each of the $O(E^{(d-1)/2})$ vectors $a \in V_d$:

$$\begin{array}{ll} \text{maximize} & |a \cdot X| \\ \text{subject to} & Y \leq p \cdot X \leq Y + 1 \quad (\forall p \in P) \\ & X \in \mathbb{R}^d,\ Y \in \mathbb{R},\ a \in V_d. \end{array}$$

Several linear-time algorithms [16, 17, 21, 32, 38, 39] are known for fixed-dimensional LP. Consequently, the approximate width problem can be solved by an LTAS in $O(E^{(d-1)/2}n)$ time [20].

> **Remark:** While this result is known, new theoretical results can be obtained by using advanced data structures. We are solving a number of *related* LPs here, all with the same set of constraints in $d+1$ dimensions. Known results on LP queries tell us that a time bound of $O(t_{d+1}(n, E^{(d-1)/2}))$ is achievable for a function t_{d+1} of the same form as (2). The idea is to reduce LP queries to membership queries (special cases of RS) in a convex polytope, via parametric search [29] or Clarkson's randomized LP algorithm [12].
>
> In fact, in our application, the membership queries reduce to membership queries in two separate d-dimensional polytopes $\{\eta \in \mathbb{R}^d \mid p \cdot \eta \geq 1\ (\forall p \in P)\}$ and $\{\eta \in \mathbb{R}^d \mid p \cdot \eta \leq 1\ (\forall p \in P)\}$. This observation allows us to lower the time bound to $O(t_d(n, E^{(d-1)/2}))$, yielding many of the entries in Table 1. For example, according to (3) and (4), for $d = 3$, $\min\{n^{3/2+o(1)}, t_3(n, E)\}$ can be bounded by $O(E^{1/3+o(1)}n)$ or $O^*(n + E)$, and for $d = 4$, $\min\{n^2, t_4(n, E^{3/2})\}$ can be bounded by $O^*(E^{3/4}n)$. By (4), this also proves the existence of a strong LTAS in any dimension (with running time $O^*(n + E^{\lfloor d/2 \rfloor (d-1)/2})$).

Algorithm 2 (Algorithm 1 + Grid). After reading a perliminary draft of this paper, Har-Peled (in personal communication) noted that the grid idea can be adapted for the approximate width problem. Specifically, Barequet and Har-Peled [8] proved the existence of a box B (of arbitrary orientation) containing P such that cB can be translated to fit inside $\mathrm{conv}(P)$ for some constant c depending on d (their paper only stated this lemma in the $d = 3$ case, but according to Har-Peled, it extends to any fixed dimension). Such a box can be computed in linear time.

Now, build a grid where each cell is a translation of $c\varepsilon B$, and replace each point by the vertices of the cell it is in to get a set P' of grid points. If P is contained in a slab S^* of width w^*, then P' is contained in a slab of width $(1 + 2\varepsilon)w^*$ since P' can be translated to fit in $P \oplus c\varepsilon B$, which in turn can be translated to fit inside $P \oplus \varepsilon\mathrm{conv}(P) \subseteq S^* \oplus \varepsilon S^*$.

Therefore, it suffices to approximate the width of P'. The size of P' is clearly $O(E^d)$, which can be further reduced to $O(E^{d-1})$ by keeping only the topmost and bottommost points on each grid line. Applying Algorithm 1 to P' yields a strong LTAS running in time $O(n + E^{3(d-1)/2})$.

> **Remark:** The third remark in Section 2 applies here as well. For example, by decomposing the grid points into $O(E^{d-3})$ 3-dimensional subsets, we can answer the desired LP queries after rounding in total time $O^*(n + E^{(3d-7)/2})$. This yields our best strong LTAS for $d = 4$, as indicated in Table 1.

4 Smallest Enclosing Cylinder

We give two new approximation algorithms for the smallest enclosing cylinder in this section.

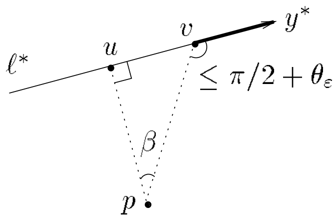

Figure 1: The plane containing the line ℓ^* and the point p.

First we formulate the problem by parametrizing the center line of the cylinder by $2d$ variables $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$ (two of which can be eliminated): $\ell = \{x + ty \mid t \in \mathbb{R}\}$. The radius is represented by an additional variable $z \in \mathbb{R}$. Since the closest point on ℓ to a given point $p \in P$ is given by the expression $x + \left(\frac{y\cdot(p-x)}{y\cdot y}\right) y$, the optimization problem is:

$$\begin{array}{lll} \text{minimize} & z & \\ \text{subject to} & \left\| x + \left(\frac{y\cdot(p-x)}{y\cdot y}\right) y - p \right\| \leq z & (\forall p \in P) \\ & x \in \mathbb{R}^d,\ y \in \mathbb{R}^d,\ z \in \mathbb{R}. & \end{array} \tag{5}$$

Let (x^*, y^*, z^*) be the optimal solution.

Algorithm 1 (Cones + CP). As in Section 3, the idea is to replace this nonconvex optimization problem with an easier problem through cones. We will replace the Euclidean point-line distance function with a more managable distance function based on a "skewed" projection of the point to the line that depends on the cone direction $a \in V_d$. As explained geometrically below, the following turns out to give a $(1+\varepsilon)$-factor approximation:

$$\begin{array}{lll} \text{minimize} & z & \\ \text{subject to} & \left\| x + \left(\frac{a\cdot(p-x)}{a\cdot y}\right) y - p \right\| \leq z & (\forall p \in P) \\ & x \in \mathbb{R}^d,\ y \in \mathbb{R}^d,\ z \in \mathbb{R},\ a \in V_d. & \end{array} \tag{6}$$

Lemma 4.1 *If (x, y, z, a) is the optimal solution for (6), then (x, y, z) is feasible for (5) and $z \leq (1+\varepsilon)z^*$.*

Proof: The first part is easy. For the second part, choose $a^* \in V_d$ such that $\angle(a^*, y^*) \leq \theta_\varepsilon$.

Take any point $p \in P$. Let $u = x^* + \left(\frac{y^*\cdot(p-x^*)}{y^*\cdot y^*}\right) y^*$, and $v = x^* + \left(\frac{a^*\cdot(p-x^*)}{a^*\cdot y^*}\right) y^*$. As noted earlier, u is the closest point on the line $\ell^* = \{x^* + ty^* \mid t \in \mathbb{R}\}$ to p. (Simply check that $u - p$ and y^* have zero dot product and are thus orthogonal.) On the other hand, v is the point on ℓ^* that lies in the hyperplane containing p and perpendicular to the direction a^*. (Simply check that $v - p$ and a^* have zero dot product.) Now, by the triangle inequality for angles,

$$\angle(p - v, y^*) \ \leq\ \angle(p - v, a^*) + \angle(a^*, y^*) \ \leq\ \pi/2 + \theta_\varepsilon.$$

From Figure 1, we see that $\beta := \angle(u-p,\ v-p) \leq \theta_\varepsilon$, and thus $\|v-p\| = \|u-p\|/\cos\beta \leq (1+\varepsilon)\|u-p\|$.

We conclude that $(x^*, y^*, (1+\varepsilon)z^*, a^*)$ is feasible for (6), hence, $z \leq (1+\varepsilon)z^*$. $\square$

Now, a change of variables, $X = x - \left(\frac{a \cdot x}{a \cdot y}\right) y$, $Y = \frac{y}{a \cdot y}$, and $Z = z^2$, reveals that (6) is actually a set of $O(E^{(d-1)/2})$ convex programs (CPs), one for each $a \in V_d$:

$$\begin{array}{ll} \text{minimize} & Z \\ \text{subject to} & \|X + (a \cdot p)Y - p\|^2 \leq Z \quad (\forall p \in P) \\ & a \cdot X = 0, \quad a \cdot Y = 1 \\ & X \in \mathbb{R}^d,\ Y \in \mathbb{R}^d,\ Z \in \mathbb{R},\ a \in V_d. \end{array}$$

(Note: the squared norm of a linear function is convex.) Several linear-time algorithms [16, 17, 39] are known for fixed-dimensional CP. Consequently, we obtain a simple LTAS that runs in $O(E^{(d-1)/2}n)$ time.

> **Remark:** With advanced data structures and the technique of linearization, it is possible to answer m membership queries for the above in a time bound $O(t_c(n, m))$ of the form in (2) for some constant c (depending on d). Known reduction of CP queries to membership queries [12] then yields a running time $O(t_c(n, E^{(d-1)/2}))$. This allows us to improve the order of our LTAS to a number slightly less than $(d-1)/2$, as indicated in Table 1 (although the improvement would probably not be practical).

Algorithm 2 (Algorithm 1 + Grid). We next give a strong LTAS using the simple grid idea. First compute a cylinder enclosing P with radius $z_0 \leq cz^*$ in $O(n)$ time for a constant c by Algorithm 1 (alternatively, as one referee suggested, take the enclosing cylinder with center line through the approximate diametral pair). By rotation, assume that the center line is vertical.

Our algorithm is similar to one in Section 2: build a uniform grid of side length εz_0, round each point to the nearest grid point, keeping only the topmost and bottommost grid point along each vertical line, and finally compute a $(1+\varepsilon)$-approximation to the smallest enclosing cylinder of the reduced set of grid points. Rounding incurs an additive error of $O(\varepsilon z_0) = O(\varepsilon z^*)$, and points between the topmost and bottommost on a line can be pruned by convexity of cylinders. So, the result is indeed a $(1 + O(\varepsilon))$-approximation to z^*.

For the analysis, observe that since the points lie within a distance of z_0 from a vertical line, there are only $O(E^{d-1})$ grid vertical lines (and thus, grid points) to consider. Applying Algorithm 1 to the grid points, we get a total running time of $O(n + E^{3(d-1)/2})$.

Algorithm 3 (Algorithm 1 + Grid). Another grid algorithm with the same time bound can be obtained by following the second width algorithm, using Barequet and Har-Peled's lemma. Our Algorithm 2 is clearly simpler.

> **Remark:** Minor speedups are possible with this approach, however. Har-Peled's comment from the first remark of Section 2 is now applicable to reduce the number of grid points to $O(E^{d-3/2})$. The running time with this enhancement is $O(n + E^{(3d-4)/2})$ (or marginally better by the previous remark), as shown in Table 1.

5 Minimum-Width Annulus

Our last problem, minimum-width annulus, can be formulated as follows, using d variables $x \in \mathbb{R}^d$ to represent the center point, and variables $y \in \mathbb{R}$ and $z \in \mathbb{R}$ to represent the inner and outer radii:

$$\begin{array}{ll} \text{minimize} & z - y \\ \text{subject to} & y \le \|x - p\| \le z \quad (\forall p \in P) \\ & x \in \mathbb{R}^d,\ y \in \mathbb{R},\ z \in \mathbb{R}. \end{array} \tag{7}$$

By translation, assume that one of the points in P is the origin, so that $y \le \|x\| \le z$.

Let (x^*, y^*, z^*) denote the optimal solution and let $w^* = z^* - y^*$. Although we can linearize the constraints by a change of variables (see below) to transform the feasible region into a $(d+2)$-dimensional convex polytope, the resulting objective function is neither linear nor convex. So, the trivial strategy would give a worst-case time bound of $O(n^{\lfloor d/2 \rfloor + 1})$. An alternative objective function $z^2 - y^2$ is known to be linear after the transformation, but unfortunately, does not always approximate $z - y$ well.

Our idea is to divide the problem into two cases. The first case is when the optimal annulus is *narrow*, in the sense that $z^* \le (1+\varepsilon)y^*$. This is the case that is not covered by the previous algorithms (since the annulus may approach a slab or the annulus-width may approach 0). The key is to observe that here, $w^* = (z^{*2} - y^{*2})/(z^* + y^*)$ would be near $(z^{*2} - y^{*2})/(2\|x^*\|)$. We thus consider approximating the objective function $(z^2 - y^2)/\|x\|$, using cones and (1), as in Section 3 for the width problem.

The second case is when the annulus is *wide*, i.e., $z^* > (1+\varepsilon)y^*$. This case turns out to be easy; in fact, we point out at least three approximation algorithms (the first is a simple self-contained grid method, the second is the algorithm by Agarwal *et al.* [1], and the third is a hybrid). Running both a narrow-case and a wide-case algorithm would guarantee that a valid solution is found.

As a warm-up exercise though, we first give a constant-factor approximation algorithm, to illustrate the main elements of this approach. In contrast to the two-dimensional factor-2 algorithm by Agarwal *et al.* [1] (which runs in $O(n \log n)$ time instead of $O(n)$), our correctness proof is quite simple and nongeometric. In the appendix, we also prove that a known heuristic yields another simple linear-time constant-factor algorithm.

Constant-Factor Approximation (Cones + LP). Consider the following optimization problem:

$$\begin{array}{ll} \text{minimize} & (z^2 - y^2)/|a \cdot x| \\ \text{subject to} & y \le \|x - p\| \le z \quad (\forall p \in P) \\ & x \in \mathbb{R}^d,\ y \in \mathbb{R},\ z \in \mathbb{R},\ a \in V_d. \end{array} \tag{8}$$

We prove that this yields a constant-factor approximation.

Lemma 5.1 *If (x, y, z, a) is the optimal solution for (8), then $z - y \le cw^*$ for some constant c.*

Proof: Choose $a^* \in V_d$ so that $a^* \cdot x^* \ge \|x^*\|/(1+\varepsilon)$. Then

$$\frac{z^2 - y^2}{z} \le \frac{z^2 - y^2}{\|x\|} \le \frac{z^2 - y^2}{|a \cdot x|} \le \frac{z^{*2} - y^{*2}}{|a^* \cdot x^*|} \le (1+\varepsilon)\frac{z^{*2} - y^{*2}}{\|x^*\|} \le (1+\varepsilon)\frac{z^{*2} - y^{*2}}{y^*},$$

implying that

$$(1+y/z)(z-y) \;\le\; (1+\varepsilon)(1+z^*/y^*)(z^*-y^*). \tag{9}$$

Case 1: $z^* \le 2y^*$. Then (9) tells us that $z-y \le (3+O(\varepsilon))w^*$.

Case 2: $z^* > 2y^*$. Then $w^* > z^*/2 \ge \Delta^*/4$, where Δ^* denotes the diameter of P. But any annulus with both its inner and outer spheres touching P has width upper-bounded by Δ^*. □

Now, (8) reduces to $O(E^{(d-1)/2})$ number of linear programs through the following change of variables: $X = x/(z^2-y^2)$, $Y = (y^2-\|x\|^2)/(z^2-y^2)$, and $Z = 1/(z^2-y^2)$.

$$\begin{array}{ll} \text{maximize} & |a\cdot X| \\ \text{subject to} & Y \;\le\; -2p\cdot X + (p\cdot p)Z \;\le\; Y+1 \quad (\forall p\in P) \\ & X\in\mathbb{R}^d,\; Y\in\mathbb{R},\; Z\in\mathbb{R},\; a\in V_d. \end{array} \tag{10}$$

We conclude that a constant-factor approximation can be found in $O(n)$ time. We may thus let w_0 be a value satisfying $w_0 \le w^* \le cw_0$ for a constant c.

Narrow Case: Algorithm (Cones + LP). We now refine the preceding algorithm to give a $(1+O(\varepsilon))$-factor approximation assuming that $z^* \le (1+\varepsilon)y^*$.

As hinted earlier, this assumption lets us approximate $2(z^*-y^*)$ by $(z^{*2}-y^{*2})/\|x^*\|$. However, a similar statement cannot be made for $2(z-y)$ in an arbitrary feasible solution unless we impose the condition $z \le (1+\varepsilon)y$ explicitly in (8). Unfortunately, as written, this constraint is nonlinear (and nonconvex). Nevertheless, we overcome the difficulty by considering an alternative constraint, shown in the following, that serves the purpose:

$$\begin{array}{ll} \text{minimize} & (z^2-y^2)/|a\cdot x| \\ \text{subject to} & y \;\le\; \|x-p\| \;\le\; z \quad (\forall p\in P) \\ & w_0 \;\le\; \varepsilon(1+\varepsilon)a\cdot x \\ & x\in\mathbb{R}^d,\; y\in\mathbb{R},\; z\in\mathbb{R},\; a\in V_d. \end{array} \tag{11}$$

The reasoning is given below.

Lemma 5.2 *If (x,y,z,a) is the optimal solution for (11), then $z-y \le (1+O(\varepsilon))w^*$.*

Proof: First observe that (x^*,y^*,z^*,a^*) is feasible for (11), because $w_0 \le w^* \le \varepsilon y^* \le \varepsilon\|x^*\| \le \varepsilon(1+\varepsilon)a^*\cdot x^*$. Therefore, (9) still holds. In particular,

$$z-y \;\le\; (2+O(\varepsilon))w^* \;\le\; O(1)w_0 \;\le\; O(\varepsilon)a\cdot x \;\le\; O(\varepsilon)\|x\| \;\le O(\varepsilon)z.$$

So $z \le (1+O(\varepsilon))y$, and applying (9) a second time yields

$$(2-O(\varepsilon))(z-y) \;\le\; (2+O(\varepsilon))w^*.$$

□

In (10), the additional constraint transforms to a linear one:

$$w_0 Z \;\le\; \varepsilon(1+\varepsilon)a\cdot X.$$

As a result, we obtain a simple $(1+O(\varepsilon))$-factor approximation algorithm for the narrow case that runs in $O(E^{(d-1)/2}n)$ time.

Wide Case: Algorithm 1 (Grid). We now give a simple $(1+O(\varepsilon))$-factor approximation algorithm for the wide case $z^* > (1+\varepsilon)y^*$. Note that $w^* > (1-\frac{1}{1+\varepsilon})z^* = \Omega(\varepsilon z^*)$.

The idea is one we have used several times before—namely, build a uniform grid of side length εw_0, and round each point of P to the nearest grid point. This incurs only an additive error of $O(\varepsilon w_0) = O(\varepsilon w^*)$. For the analysis, observe that since the points lie in an annulus of volume $O(z^{*d} - y^{*d}) = O(w^* z^{*(d-1)}) = O(E^{d-1} w_0^d)$, the number of points reduces to $O(E^{2d-1})$.

We can also restrict the possible center points to grid points. Since the center lies within a radius of $\|x^*\| \leq z^* = O(Ew_0)$ to the origin, there are only $O(E^{2d})$ center points to try. Each requires simply finding the nearest/farthest point to P.

The wide case can thus be solved in $O(n + E^{4d-1})$ time. We now have a complete LTAS for minimum-width annulus, running in $O(E^{(d-1)/2}n + E^{4d-1})$ time.

> **Remark:** With advanced data structures for LP queries, the time bound can be reduced to $O(t_{d+2}(n, E^{(d-1)/2}) + E^{4d-1})$, or with the first remark from Section 3, to $O(t_{d+1}(n, E^{(d-1)/2}) + E^{4d-1})$. While there are several ways that can improve the E^{4d-1} term further, this result is enough to give our best strong LTAS for a sufficiently large d by (4), as shown in Table 1: the running time is $O^*(n + E^{\max\{4d-1, \lceil d/2\rceil(d-1)/2\}})$. We thus focus our attention next on small dimensions like 2 and 3.

Wide Case: Algorithm 2 (Grid + LP). We observe another method for the wide case. Specifically, Agarwal *et al.* [1, Theorem 4.3] gave a simple algorithm that solves the minimum-width annulus problem in $O(E^d n \log U)$ time when $z^* \leq U\Delta^*$. We will not repeat their description here: basically, it involves solving an LP (under the objective function $z^2 - y^2$) for each cell of a certain nonuniform grid.

In the wide case, we know that $z^* = O(Ew^*) = O(E\Delta^*)$, so we can set $U = O(E)$ immediately and get an $O(E^d n \log E)$-time algorithm. Combined with our narrow-case algorithm, the total running time is $O(E^d n \log E)$.

> **Remark:** With advanced data structures, the time bound is $O(t_{d+1}(n, E^d \log E))$. This gives the entries in Table 1 for $d = 2$, as according to (3) and (4), $\min\{n^{3/2+o(1)}, t_3(n, E^2)\}$ can be bounded by $O(E^{2/3+o(1)}n)$ or $O^*(n + E^2)$. For $d = 3$, we also have $\min\{n^2, t_4(n, E^3)\} = O^*(E^{3/2}n)$.

Wide Case: Algorithm 3 (Algorithm 1 + Algorithm 2). For yet another algorithm for the wide case, apply the idea in Algorithm 1 to reduce the problem to one involving $O(E^{2d-1})$ points and then apply Algorithm 2. The overall running time is $O(E^{(d-1)/2}n + E^{3d-1}\log E)$.

> **Remark:** With advanced data structures, the time bound is $O(t_{d+1}(n, E^{(d-1)/2}) + t_{d+1}(E^{2d-1}, E^d \log E))$. Note that the $O(E^{2d-1})$ grid points actually lie in $O(E^{2(d-2)})$ grid planes. So, by decomposing the point set into 2-dimensional subsets in the style of the third remark from Section 2, we can rewrite the time bound as $O^*(t_{d+1}(n, E^{(d-1)/2}) + \sum_{i=1}^{O(E^{2(d-2)})} t_3(n_i, E^d))$, where $\sum_i n_i = n$. This is at most $O^*(n + E^{\max\{3d-4, \lceil d/2\rceil(d-1)/2\}})$, yielding the strong LTAS entry for $d = 3$ in Table 1.

Acknowledgements

I thank Sariel Har-Peled for his valuable comments. Support by NSERC is gratefully acknowledged.

References

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

[2] P. K. Agarwal, B. Aronov, and M. Sharir. Line transversals of balls and smallest enclosing cylinders in three dimensions. *Discrete Comput. Geom.*, 21:373–388, 1999.

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

[4] P. K. Agarwal, J. Matoušek, and S. Suri. Farthest neighbors, maximum spanning trees and related problems in higher dimensions. *Comput. Geom. Theory Appl.*, 1:189–201, 1992.

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

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

[7] G. E. Andrews. A lower bound for the volume of strictly convex bodies with many boundary lattice points. *Trans. Amer. Math. Soc.*, 106:270–279, 1963.

[8] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 82–91, 1999.

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

[10] S. Bespamyatnikh. An efficient algorithm for the three-dimensional diameter problem. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 137–146, 1998.

[11] A. Borodin, R. Ostrovsky, and Y. Rabani. Subquadratic approximation algorithms for clustering problems in high dimensional spaces. In *Proc. 31st ACM Sympos. Theory Comput.*, pages 435–444, 1999.

[12] T. M. Chan. Fixed-dimensional linear programming queries made easy. In *Proc. 12th ACM Sympos. Comput. Geom.*, pages 284–290, 1996.

[13] T. M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. *Discrete Comput. Geom.*, 16:369–387, 1996.

[14] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.

[15] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Diameter, width, closest line pair and parametric searching. *Discrete Comput. Geom.*, 10:183–196, 1993.

[16] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, 21:579–597, 1996.

[17] K. L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*, 42:488–499, 1995.

[18] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[19] O. Devillers and F. P. Preparata. Evaluating the cylindricity of a nominally cylindrical point set. Technical report RR-3747, INRIA, 1999. In *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, 2000.

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

[21] M. E. Dyer. On a multidimensional search technique and its application to the Euclidean one-center problem. *SIAM J. Comput.*, 15:725–738, 1986.

[22] D. V. Finocchiaro and M. Pellegrini. On computing the diameter of a point set in high dimensional Euclidean space. In *Proc. 7th European Sympos. Algorithms*, Lect. Notes in Comput. Sci., vol. 1643, Springer-Verlag, pages 366–377, 1999.

[23] J. Garcia-López, P. A. Ramos, and J. Snoeyink. Fitting a set of points by a circle. *Discrete Comput. Geom.*, 20:389–402, 1998.

[24] A. Goel and P. Indyk. Reductions among high dimensional proximity problems. To appear in *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, 2001.

[25] P. Gritzmann and V. Klee. Computational convexity. In *CRC Handbook on Discrete and Computational Geometry* (ed. J. E. Goodman and J. O'Rourke), CRC Press, Boca Raton, FL, pages 491–515, 1997.

[26] M. E. Houle and G. T. Toussaint. Computing the width of a set. In *Proc. 1st ACM Sympos. Comput. Geom.*, pages 1–7, 1985.

[27] R. Janardan. On maintaining the width and diameter of a planar point-set online. *Int. J. Comput. Geom. Appl.*, 3:331–344, 1993.

[28] J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Appl.*, 2:169–186, 1992.

[29] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993. Also with O. Schwarzkopf in *Proc. 8th ACM Sympos. Comput. Geom.*, pages 16–25, 1992.

[30] J. Matoušek and O. Schwarzkopf. On ray shooting in convex polytopes. *Discrete Comput. Geom.*, 10:215–232, 1993.

[31] J. Matoušek and O. Schwarzkopf. A deterministic algorithm for the three-dimensional diameter problem. *Comput. Geom. Theory Appl.*, 6:253–262, 1996.

[32] N. Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31:114–127, 1984.

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

[34] E. Ramos. Construction of 1-d lower envelopes and applications. In *Proc. 13th ACM Sympos. Comput. Geom.*, pages 57–66, 1997.

[35] E. Ramos. Deterministic algorithms for 3-D diameter and some 2-D lower envelopes. In *Proc. 16th ACM Sympos. Comput. Geom.*, pages 290–299, 2000.

[36] P. A. Ramos. Computing roundness is easy if the set is almost round. In *Proc. 15th ACM Sympos. Comput. Geom.*, pages 307–315, 1999.

[37] E. Schömer, J. Sellen, M. Teichmann, and C.-K. Yap. Smallest enclosing cylinders. *Algorithmica*, 27:170–186, 2000.

[38] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom.*, 6:423–434, 1991.

[39] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Sympos. Theoret. Aspects Comput. Sci.*, Lect. Notes in Comput. Sci., vol. 577, Springer-Verlag, pages 569–579, 1992.

[40] A. C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Comput.*, 11:721–736, 1982.

A An Inequality for the Minimum-Width Annulus

In this appendix, we note another simple constant-factor approximation algorithm for the minimum-width annulus.

As pointed out earlier and is well-known, if the objective is replaced by the function $z^2 - y^2$, then (7) changes to a linear program and thus becomes solvable in $O(n)$ time. (In the two-dimensional case, this optimal annulus is the minimum-*area* annulus.) Let (x_A, y_A, z_A) be the new solution and $w_A = z_A - y_A$. Let x^*, y^*, z^*, w^* be as in Section 5.

The value w_A has been used in practice as an approximation to w^*. For instance, Ramos [36] noted that in many instances arising from metrology applications, the minimum-area annulus is surprisingly the same as the minimum-width annulus. Although in the worst case the approximation can be arbitrarily poor, we nevertheless are able to derive the following inequality relating w^* to w_A, where the third parameter w_S here denotes the minimum width over all enclosing slabs, which we know how to approximate by Section 3:

Lemma A.1 $1/w^* \leq 1/w_A + 2/w_S$.

Proof: We know that $z_A^2 - y_A^2 \leq z^{*2} - y^{*2}$. Let δ be the distance between the two centers x^* and x_A. Note that $y^* - y_A \leq \delta$ and $z^* - z_A \leq \delta$. By a change of coordinate system, we may assume $x^* = (\delta/2, 0, \ldots, 0)$ and $x_A = (-\delta/2, 0, \ldots, 0)$. Then for every point $p = (p_1, \ldots, p_d) \in P$,

$$\begin{array}{lcccl} y^{*2} & \leq & (p_1 - \delta/2)^2 + p_2^2 + \cdots + p_d^2 & \leq & z^{*2} \\ y_A^2 & \leq & (p_1 + \delta/2)^2 + p_2^2 + \cdots + p_d^2 & \leq & z_A^2, \end{array}$$

implying that $y_A^2 - z^{*2} \leq 2\delta p_1 \leq z_A^2 - y^{*2}$. So, we can upper-bound the minimum-slab width by

$$w_S \leq \frac{(z_A^2 - y^{*2}) - (y_A^2 - z^{*2})}{2\delta} \leq \frac{z^{*2} - y^{*2}}{\delta}.$$

On the other hand,

$$\frac{1}{w^*} - \frac{1}{w_A} = \frac{z^* + y^*}{z^{*2} - y^{*2}} - \frac{z_A + y_A}{z_A^2 - y_A^2} \leq \frac{(z^* + y^*) - (z_A + y_A)}{z^{*2} - y^{*2}} \leq \frac{2\delta}{z^{*2} - y^{*2}}.$$

□

Consequently, either w_A or w_S yields a constant-factor approximation to w^*. (Note that a slab may be viewed as an annulus with its center at infinity.)

Corollary A.2 $\min\{w_A, w_S\} \leq 3w^*$. □

The constant 3 above cannot be improved (as can be shown by a simple example), and unlike in the approach in Section 5, we are unable to modify this approach to obtain a $(1+\varepsilon)$-factor method. Still, Lemma A.1 can yield quite an accurate bound on w^* in some practical instances where the given point set is almost round and "well-distributed" near its optimal circle. (For example, imagine when $w_A = 0.01$ and $w_S = 1$, we have $0.0098 \leq w^* \leq 0.01$—an estimate with a 2% relative error at most.)

After discovering our inequality, we learn that Devillers and Preparata [19] earlier have obtained another (similar) inequality relating w^* to w_A. The width w_S is not involved in the statement of their result (but they needed to specify a certain no-empty-sector assumption), and their proof is somewhat lengthier.