Smallest Enclosing Cylinders

Elmar Schömer* Jürgen Sellen† Marek Teichmann‡ Chee Yap§

Abstract

This paper addresses the complexity of computing the smallest-radius infinite cylinder that encloses an input set of n points in 3-space. We show that the problem can be solved in time $O(n^4 \log^{O(1)} n)$ in an algebraic complexity model. We also achieve a time of $O(n^4 L \cdot \mu(L))$ in a bit complexity model where L is the maximum bit size of input numbers and $\mu(L)$ is the complexity of multiplying two L bit integers.

These and several other results highlight a general *linearization technique* which transforms non-linear problems into some higher dimensional but linear problems. The technique is reminiscent of the use of Plücker coordinates, and is used here in conjunction with Megiddo's parametric searching.

We further report on experimental work comparing the practicality of an exact with that of a numerical strategy.

1 Introduction

1.1 Motivation

A major topic of geometric optimization is to approximate point sets by simple geometric figures. This includes extensively studied planar problems such as smallest enclosing circles, the minimum width annulus, and the minimum width slab. In higher dimensions, there are few non-trivial complexity results for geometric figures beyond hyperplanes or spheres. In this paper, we consider the following:

> **Smallest Cylinder Problem (P1):** Let I be a given set of n points in 3-space. Find a line ℓ which minimizes $\max\{\ d(\ell, c)\ :\ c \in I\ \}$.

Here, $d(\ell, c)$ denotes the minimum Euclidean distance between c and a point of ℓ. Since cylinders constitute an important primitive shape in computer-aided design and manufacturing, this problem has many applications. We merely give two examples:

The first example is from *assembly planning*. Assume we want to "fit" a given polyhedral object into a cylindrical hole. Obviously, this problem can be solved by computing the smallest enclosing cylinder of the polyhedron. In practical situations, the number n of points defining the polyhedron may be large, but the computation of a suboptimal solution can usually be tolerated.

The second example is from an area of importance to modern high precision engineering, *dimensional tolerancing and metrology* (see [SV, Ya]). Here the task is, given a physical object, to verify

*`schoemer@cs.uni-sb.de`, Universität des Saarlandes.

†`sellen@cs.uni-sb.de`, Universität des Saarlandes. Currently at ESG GmbH, Munich.

‡`marekt@lcs.mit.edu`, Massachusetts Institute of Technology. This work was done while at NYU.

§`yap@cs.nyu.edu`, Courant Institute, NYU.

its conformance to tolerance specifications by taking probes of its surface. The cylinder is one of the basic objects addressed by the ASME tolerancing standards [Asme]. In industry, highly specialized, expensive equipment (called Coordinate Measurement Machine or CMM) is used to perform these probes automatically. In contrast to the previous application, n is small but high numerical accuracy is important [Ya].

1.2 Contributions

We summarize four areas of contribution.

(I) We design efficient algorithms for the smallest cylinder problem:

Theorem 1
The problem (P1) can be solved in time:
(i) $O(n^4 \log^{O(1)} n)$ *in an algebraic model; and*
(ii) $O(L\mu(L)n^4)$ *in a bit model.*

Here, $\mu(L) = O(L \log L \log\log L)$ denotes the complexity of multiplying two L-bit integers. The algebraic and bit complexity models are described below.

(II) We design approximation algorithms for the smallest cylinder problem. Such results may be more useful for certain applications. A cylinder whose radius is within ε of the minimum radius is called an *ε-approximate smallest cylinder*. We obtain complexity trade-offs between n and ε:

Theorem 2
In an algebraic model of computing, an ε-approximate solution of (P1) can be found in times (respectively):

$$O(n\varepsilon^{-2}\log\varepsilon^{-1}),\ O(n^3\varepsilon^{-1}\log\varepsilon^{-1}),\ O(n^4\log\varepsilon^{-1}).$$

(III) We highlight a *linearization technique* for geometric optimization problems. The above result uses Megiddo's parametric search and a new parallel convex hull algorithm in [AGR]. But it also requires an application of the linearization technique, which we believe has wider applicability.

The heart of both approximation and parametric search algorithms is a *decision scheme* for a *fixed optimization parameter*. To obtain efficient decision algorithms, it is often possible to exploit geometric duality transformations, including inversion (as in [FSS]) and Plücker coordinates (as in [ST]). In this paper, we extend these principles to a more general framework, here called *linearization*. We give this an abstract formulation. Let $P(\mathbf{x},\mathbf{y})$ be a polynomial in the real variables $\mathbf{x} = (x_1, \ldots, x_\ell)$ and $\mathbf{y} = (y_1, \ldots, y_m)$.

> **Abstract Decision Problem (D):** Given a set $I \subseteq \mathbf{R}^m$ of n points, decide if there exists a point $p \in \mathbf{R}^\ell$ such that for all $c \in I$, $P(p, c) \le 0$.

We say $P(\mathbf{x},\mathbf{y})$ has an *order k linearization* if there exist $2k+1$ polynomials, $X_i = X_i(\mathbf{x})$ $(i = 1, \ldots, k)$ and $Y_i = Y_i(\mathbf{y})$ (for $i = 0, \ldots, k$), such that

$$P(\mathbf{x},\mathbf{y}) = Y_0 + \sum_{i=1}^{k} X_i Y_i.$$

Theorem 3
(i) If $P(\mathbf{x},\mathbf{y})$ has an order k linearization, the decision problem (D) can be solved in time $O(n^{\lfloor k/2 \rfloor})$ in the algebraic model.
(ii) In the bit model, if each input coordinate has L bits, the problem (D) can be solved in time $O(\mu(L)n^{\lfloor k/2 \rfloor})$.

In our application, we need to transform our original problem into some suitable versions of Problem (D) and give efficient linearizations.

(IV) Finally, in view of considerable interest in implementation of these algorithms [Ya], we report on some experimental work using a heuristic approach – a "local" numerical optimization technique, implemented in `C`. Our heuristic seems to be quite effective, as verified against exact answers. The exact answers come from a MAPLE implementation of a simple algorithm to enumerate all cylinders with fixed radius through 4 of the given points. It should be noted that the size of our input data (see section 4) lies on the edge of what could reasonably be handled by MAPLE.

1.3 Subproblems

In order to get approximation algorithms, we may consider *restricted* versions of (P1) with fewer degrees of freedom, and discretize the remaining parameters by a grid, with step-size depending on ε. Suppose we eliminate the *rotational degrees of freedom* and ask for a cylinder with fixed axis direction. This problem reduces to the well-known *smallest enclosing circle* problem in the plane – which is solvable in linear time. An intrinsically different situation arises when we eliminate the *translational degrees of freedom*:

> **Smallest Anchored Cylinder Problem (P2):** Let I be a given set of n points in 3-space. Find a line ℓ through the origin which minimizes $\max\{\ d(\ell, c)\ :\ c \in I\ \}$.

As this problem is non-convex (section 2.3), usual approaches to obtain subquadratic solutions fail. But it is noteworthy that two interesting subcases can be solved in subquadratic time: when the input points are relatively far from the origin, and when we ask for an optimal location of a ray instead of a line.

1.4 Related Work

Problem (P1) belongs to a class of problems that have been considered from a complexity–theoretic viewpoint in [GK]. Although problem (P1) is routinely solved in engineering applications using numerical optimization techniques, few complexity theoretic results have been published. A more general version of this problem has been shown to be polynomial time solvable by Faigle, Kern and Streng [FKS], and studied from the viewpoint of nonlinear optimization theory by Streng [S]. Concrete geometrical properties have first been investigated in [Pa], with focus on the *decision problem* to determine if there exists a cylinder with radius $r = 1$ (a *unit cylinder*) which encloses the input points.

Proposition 1 ([Pa])
(a) If there exists a unit cylinder that encloses all input points, then there also exists a unit enclosing cylinder which touches 4 of the input points, or whose axis is parallel to an edge of the convex hull of I.
(b) There is only a finite number of cylinders with radius 1 that touch 4 non-collinear points in 3-space.

With these (geometrically non-trivial) results, the decision problem for fixed radius can be solved by enumerating all cylinders through choices of 4 points, and by checking if one of these encloses the input points. This algorithm has complexity $O(n^5)$. It is not hard to see that the optimization problem can be solved in time $O(n^5 \log n)$ by binary search. (It is also possible to use a straightforward application of parametric search.)

The linearization technique appears to have been used first by Yao and Yao [YY]. More recently, Agarwal and Matoušek [AM] use linearization in the context of range searching with semialgebraic sets. They also gave a simple procedure for finding the optimal linearization for a given polynomial.

The maximin counterpart of (P2) has nice applications in robotics. It has been solved by parametric search in time $O(n \log^4 n)$ [Fo]. The variant of (P2) where we ask for an enclosing silo instead of a cylinder was solved in the same paper in time $O(n \log^3 n \log\log n)$.

Recently, by posing the problem as a problem of finding line transversals of balls, Agarwal *et al.* [AAS] have established a bound of $O(n^{3+\epsilon})$ on the combinatorial complexity of the set of cylinders of a given radius enclosing a set of n points, as well as an $\Omega(n^3)$ lower bound. Here ϵ is any positive constant. They give an $O(n^{3+\epsilon})$ time algorithm for solving (P1) and a slight generalization thereof, as well as an approximation algorithm for finding a cylinder whose radius is at most $1+\delta$ times the optimum in time $O(n/\delta^2)$. These algorithms operate in an algebraic model of computation, and the first algorithm requires the use of parametric search.

1.5 Parametric Search vs. Exact Approximation

Parametric search is an ingenious technique to design optimization algorithms in the algebraic model of computing. Introduced in [Me], it has been applied to numerous optimization problems.

But while low–dimensional problems like (P2) often possess simple algebraic characterizations, the algebraic structure of problem (P1) is much more involved. Its solution requires the calculation of roots of polynomials with high degree. In the algebraic model, this calculation is regarded as a constant time operation. Even worse, the results of such a computation may be used in a parametric search strategy as coefficients of a polynomial in a subsequent step, potentially increasing substancially the bit complexity of the numbers involved. This is a major reason to consider bit-complexity.

On the other hand, the decision scheme that underlies a parametric search solution immediately provides an approximation algorithm that guarantees an error of ε (with our assumptions, absolute error) by adding just a factor of $\log \varepsilon^{-1}$ to the running time. Finally, in a bit model, this approximation can be made *exact* (in the sense of providing a combinatorial solution) by exploiting techniques (esp., root bounds) from the theory of exact computation.

While parametric search provides a clean dependency of running time on the number n of input points, the exact approach is more suitable if accuracy is the main goal to achieve. This gets increasingly important as the algebraic source of complexity comes into play.

1.6 Algebraic and Bit Complexity Models

Most geometric algorithms are developed within one of two distinct computational frameworks. In the *algebraic framework*, the complexity of an algorithm is measured by the number of algebraic operations on real-valued variables, assuming exact computations. The input size corresponds to the number n of input values. In the *bit framework*, the complexity is measured by the number of bitwise boolean operations on binary strings. The input generally consists of integers, and the parameter n is supplemented by an additional parameter L that bounds the maximal bit-size of any input value.

While the size of the input is measured differently in the algebraic and in the bit model, the output can often be treated in a uniform way by asking for a *combinatorial solution* to the problem. In the case of (P1), we may assume the required output to be a list of those input points that specify the optimal cylinder(s).

Another way to define the output of optimization algorithms is to consider the *approximation problem*, in our case to find an enclosing cylinder with radius r such that $|r - r^*| \leq \varepsilon$, where r^* denotes the sought optimum and ε the required *absolute error*. Traditionally, approximation algorithms are treated in an algebraic model of computing. However, we note that the bit model is also a reasonable choice, especially since the size of input numbers can have influence on the approximation error ε.

One of our basic assumptions is that – in the algebraic model – each input point $c \in I$ is enclosed in the unit sphere (i.e., $\|c\| < 1$), and that – in the bit model – the coordinates of each $c \in I$ are given as homogeneous rational numbers of bit-size L.

1.7 Overview

Section 2 contains mathematical preliminaries, and serves to clarify basic properties of the problem. In particular, subsection 2.1 describes the technical framework that underlies our MAPLE implementation, and subsection 2.2 treats major aspects of the bit-complexity analysis which is necessary to make ε-approximation algorithms exact.

Section 3 is devoted to our optimization technique, and to the proofs of theorems 2, 1 and 3. We also present results for the restricted problem (P2) in this section.

Experimental results and a discussion conclude the paper in sections 4 and 5.

2 Preliminaries

2.1 Algebraic Formulation

A cylinder C in 3-space is specified by 5 real parameters, its axis line ℓ and its radius r. We follow the approach suggested by Proposition 1, and first specify the set $C(c_1, \ldots, c_4)$ of cylinders that touch 4 given points $c_1, \ldots, c_4 \in I$.

By translation of the coordinate system, we can assume $c_1 = (0, 0, 0)$. Let $u \in \mathbf{R}^3$ be any direction vector of ℓ. Let E be the plane passing through the origin and orthogonal to u, and let $c_1^*, \ldots, c_4^*$ be the orthogonal projection of the input points $c_1, \ldots, c_4$ onto E. Then the cylinder C passes through $c_1, \ldots, c_4$ if and only if $c_1^*, \ldots, c_4^*$ are cocircular.

The first problem that we face in the algebraic computation of solutions is to find a suitable parametrization for the direction vector u. We will treat the case when u is not parallel to the plane containing c_2, c_3, c_4. (Otherwise, we have a simpler subproblem.) We may likewise assume that $c_1, \ldots, c_4$ do not lie in a plane. Let

$$u = xc_2 + yc_3 + zc_4.$$

Note that we may choose u to lie in the plane of c_2, c_3, c_4, by setting $z = 1 - x - y$. The parameters x, y, z are also called the *barycentric coordinates* of u with respect to c_2, c_3, c_4.

Now, let $R_1(x, y, z)$ be the squared radius of the circumcircle of c_1^*, c_2^*, c_3^* in E, and $R_2(x, y, z)$ the squared radius of the circumcircle of c_1^*, c_3^*, c_4^*. Then the set $C(c_1, \ldots, c_4)$ can be interpreted as a 2-dimensional surface in 3-space, defined by $R_1(x, y, z) = R_2(x, y, z)$.

Lemma 1 *The condition $R_1(x, y, z) = R_2(x, y, z)$ is equivalent to $P(x, y, z) = 0$, with*

$$\begin{aligned} P(x, y, z) = {} & \Delta_{1,2,4}(xz^2 + x^2z) \\ & + \Delta_{1,3,4}(yz^2 + y^2z) \\ & + \Delta_{1,2,3}(xy^2 + x^2y) \\ & + (\Delta_{1,2,4} + \Delta_{1,3,4} + \Delta_{1,2,3} - \Delta_{2,3,4})(xyz), \end{aligned}$$

where $\Delta_{i,j,k}$ is equal (respectively, proportional) to the squared area of the triangle with vertices c_i, c_j, c_k.

Proof. Let $u = xc_2 + yc_3 + zc_4$ be the direction of projection and v and w two vectors which supplement $\frac{u}{|u|}$ to an orthonormal system. The four projected points c_i^* are cocircular iff

$$\det \begin{bmatrix} 1 & v^T c_1 & w^T c_1 & (v^T c_1)^2 + (w^T c_1)^2 \\ 1 & v^T c_2 & w^T c_2 & (v^T c_2)^2 + (w^T c_2)^2 \\ 1 & v^T c_3 & w^T c_3 & (v^T c_3)^2 + (w^T c_3)^2 \\ 1 & v^T c_4 & w^T c_4 & (v^T c_4)^2 + (w^T c_4)^2 \end{bmatrix} = 0$$

Since $c_1^T = (0, 0, 0)$ this is equivalent to

$$\det \begin{bmatrix} v^T c_2 & w^T c_2 & {c_2^*}^2 \\ v^T c_3 & w^T c_3 & {c_3^*}^2 \\ v^T c_4 & w^T c_4 & {c_4^*}^2 \end{bmatrix} = 0$$

where ${c_i^*}^2 = (v^T c_i)^2 + (w^T c_i)^2 = c_i^2 - \frac{(u^T c_i)^2}{u^2}$. Using $v \times w = \frac{u}{|u|}$ the expansion of this determinant yields

$${c_2^*}^2 u^T (c_3 \times c_4) + {c_3^*}^2 u^T (c_4 \times c_2) + {c_4^*}^2 u^T (c_2 \times c_3) = 0$$

Substituting $u = xc_2 + yc_3 + zc_4$ we get (provided that $\det(c_2, c_3, c_4) \neq 0$):

$$x{c_2^*}^2 + y{c_3^*}^2 + z{c_4^*}^2 = 0$$

Multiplying this equation with u^2 yields

$$(c_2 \times c_3)^2 (xy^2 + yx^2) + (c_3 \times c_4)^2 (yz^2 + zy^2) + (c_4 \times c_2)^2 (zx^2 + xz^2)$$
$$- 2((c_2 \times c_3)^T (c_4 \times c_2) + (c_2 \times c_3)^T (c_3 \times c_4) + (c_3 \times c_4)^T (c_4 \times c_2))(xyz) = 0$$

since $u^2 {c_i^*}^2 = (c_i \times u)^2$.

If we set $\Delta_{i,j,k} = (c_i \times c_j + c_j \times c_k + c_k \times c_i)^2$ we get the desired result. □

With $z = 1 - x - y$, P can also be interpreted as a polynomial in the 2 variables x and y, or as a 1-dimensional curve in the x-y-plane. We note that the total degree of P is 3, and the degree in each variable is 2.

In order to compute the cylinders with fixed radius r in the set $C(c_1, \ldots, c_4)$, the additional condition $R_1(x, y, z) = r$ has to be satisfied. Unfortunately, this leads to a significantly more complicated polynomial equation $Q(x, y) = 0$, with total degree 6.

Let $C_f(c_1, \ldots, c_4, r)$ be the set of all cylinders with radius r that pass through $c_1, \ldots, c_4$ and whose axis line is not parallel to the plane through c_2, c_3, c_4. Then C_f is given by the set of solutions of the system $\{\ Q(x, y) = 0, P(x, y) = 0\ \}$, and can be obtained algebraically by computing the roots of the resultants $F_x = Res(P, Q, y)$ and $F_y = Res(P, Q, x)$. These resultants have degree 12.

Lemma 2 *If $c_1, \ldots, c_4$ are not collinear, the set $C_f(c_1, \ldots, c_4, r)$ contains at most 12 cylinders. Assuming that the c_i are rational points, each cylinder is specified uniquely by algebraic numbers of degree at most 12.*

In this lemma, we assume a cylinder is specified by the direction vector u introduced above.

2.2 Bit Complexity

Proposition 1 provides the framework for an approximation algorithm for (P1). By exploiting ideas from the theory of exact computation, we can make such an approximation algorithm "exact" in the sense that – given rational input points with coordinates of bit-size $\leq L$ – it is possible to find the input points that define the smallest enclosing cylinder(s) in time depending polynomially on n and L.

In the following, it is useful to consider the optimization function (the radius r of a smallest enclosing cylinder) as a function of the axis direction, and thus as a surface in $\mathbf{R}^3$. This surface is given by 2-dimensional surface patches (corresponding to cylinders that touch 3 points), 1-dimensional ridges (corresponding to cylinders that touch 4 points), and vertices (defined by tuples of 5 points).

Now assume that r_i°, $i = 1, 2$, denotes a local minimum of a surface patch, a local minimum of a ridge, or the "height" of a vertex. Further, let δ be a *separation gap* between any two values r_1° and r_2° that are not equal, i.e., $|r_1^\circ - r_2^\circ| \geq \delta$ for all $r_1^\circ \neq r_2^\circ$. Then the combinatorial solution of (P1) can easily be derived from a δ–approximate solution of (P1).

The computation of the gap δ is non-trivial, requiring an algebraic characterization of the local minima above, and the application of multi-variate root bounds. In the sequel, we shall focus on the computation of δ for the most complicated case, when r_1° and r_2° are the local minima of ridges.

Let $c_1, \ldots, c_4$ be an arbitrary choice of input points. Our goal is to compute a discrete set of values which contains r_1°, a local minimum value with respect to $c_1, \ldots, c_4$. Following subsection 2.1, let $R_1(x, y)$ be the squared radius of the circumcircle of c_1^*, c_2^*, c_3^*, and $P_1(x, y)$ the polynomial which defines the cylinder with direction parameters (x, y) passing through $c_1, \ldots, c_4$. Then the candidates for r_1° are the local minimum values of $R_1(x, y)$ under the side condition $P_1(x, y) = 0$. By the rule of Lagrange, there exists a parameter λ such that the following two conditions hold at the minima:

$$(1)\;\; \frac{\partial R_1}{\partial x} + \lambda \frac{\partial P_1}{\partial x} = 0, \quad (2)\;\; \frac{\partial R_1}{\partial y} + \lambda \frac{\partial P_1}{\partial y} = 0.$$

Eliminating λ in these equations, let $Q_1(x, y)$ be the numerator of the expression

$$\frac{\partial R_1}{\partial x} - \frac{\partial R_1}{\partial y} \frac{\partial P_1}{\partial x} \left(\frac{\partial P_1}{\partial y} \right)^{-1}.$$

Then $r_1^\circ = \sqrt{R_1(x_1^\circ, y_1^\circ)}$, where (x_1°, y_1°) is a solution of the system $\{\ P_1(x, y) = 0, Q_1(x, y) = 0\ \}$.

Analogously, let r_2° be a minimum candidate for a different choice of input points, and P_2, Q_2, R_2 the corresponding defining formulas. Then the needed separation gap can be obtained as a lower bound for $|\delta|$ in the system of equations

$$\begin{aligned}
&(1)\;\; P_1(x_1, y_1) = 0,\\
&(2)\;\; Q_1(x_1, y_1) = 0,\\
&(3)\;\; P_2(x_2, y_2) = 0,\\
&(4)\;\; Q_2(x_2, y_2) = 0,\\
&(5)\;\; \sqrt{R_1(x_1, y_1)} - \sqrt{R_2(x_2, y_2)} = \delta.
\end{aligned}$$

By repeated squaring, formula (5) can be transformed into a polynomial equation $R(x_1, y_1, x_2, y_2, \delta) = 0$ such that the set of solutions is only increased by a finite number of new candidates. Now, a bound for δ can be obtained from the gap-theorem of Canny [Ca].

Proposition 2 ([Ca]) *Let $f_1, \ldots, f_n$ be n polynomials in n variables, with degree $\leq d$ and coefficient magnitude $\leq c$. Assume that the system $\{f_1 = 0, \ldots, f_n = 0\}$ has only a finite number of solutions when homogenized. If $(\alpha_1, \ldots, \alpha_n)$ is a solution with $\alpha_i \neq 0$, then $|\alpha_i| > (3dc)^{-nd^n}$.*

With $c = 2^L$, $d = const$ and $n = 5$, we get $|\delta| = 2^{-O(L)}$. This gives us:

Proposition 3 *Let C be a smallest enclosing cylinder for input set I, with radius r^*. Then any cylinder $C' \neq C$ that touches a different set of points than C has radius $r = r^*$ or $r \geq r^* + 2^{-cL}$, for a suitable constant c.*

To conclude, $O(L)$ iterations of the decision algorithm are sufficient to determine the combinatorial solution.

Remark 1 *The use of the general gap-theorem 2 gives constants that are far beyond from being practical. It would be desirable to derive sharper bounds for special cases of this theorem.*

2.3 Combinatorial Complexity

The goal of this subsection is to provide some intuition on the combinatorial complexity of the considered problems – with focus on lower bounds. In particular, our constructions show that the problems are far from being convex or LP-type.

We start with an example that provides a lower bound on the number of *global optima* in both (P1) and (P2). Consider an even number n of points in a plane, arranged as a regular n-gon. In this setting, there are exactly $n/2$ smallest enclosing cylinders, corresponding to the smallest enclosing slabs in the plane. Each of these cylinders is a locally smallest enclosing cylinder of 4 of the input points. Note that, with a similar example, it is also possible to have two global minima which lie arbitrarily close together, i.e., whose axis lines can be brought to coincidence by an arbitrarily small rotation.

To give a lower bound on the possible number of *local optima* is slightly more complicated, and we first turn our attention to the restricted problem (P2). Consider an even number n of points that are arranged on the unit sphere S^2, $n/2$ on the circle C_1 (C_2) of intersection with the plane $z = 0$ ($y = 0$). Further, we assume that the points on each circle are uniformly stepped and diametrically opposed. Let each line through the origin be parameterized by its intersection with the sphere S^2. Now let us ask for the set of cylinders with distance $\geq 1-\varepsilon$ to one input point c. This set corresponds to a thin stripe on S^2, and describes the forbidden cylinders with respect to c. The set of enclosing cylinders with radius $\leq 1 - \varepsilon$ is the complement of the union of the stripes for all $c \in I$. For ε sufficiently small, this set has quadratic complexity, and any connected component must correspond to a local minimum.

Finally, let us consider the general setting (P1) for the same input set I above. In this case, it is easy to see that – due to symmetry – a necessary condition for a line ℓ to be the axis of a locally smallest enclosing cylinder is that the line ℓ passes through the origin. Hence, the $\Omega(n^2)$ local minima in (P2) stay local minima even if we add the remaining translational degrees of freedom.

Proposition 4 *For n given input points, there can be $\Omega(n)$ globally smallest and $\Omega(n^2)$ locally smallest enclosing cylinders.*

Remark 2 *Agarwal* et al. *[AAS] have a similar result. They show that the set of cylinders of a given radius enclosing a set of n points can consist of $\Omega(n^2)$ connected components.*
It remains open if there exist examples for (P1) with more than a quadratic number of local or of global minima.

3 Optimization Algorithms

3.1 Linearization

In order to illustrate the basic idea of our optimization technique, we first consider the anchored problem (P2). Our focus is the fixed-parameter problem to decide whether there exists an anchored cylinder of given radius r that encloses all input points.

Let ℓ_{ab} be the line through the points $a, b \in \mathbf{R}^3$. We fix a at the origin and w.l.o.g. require b to lie on the plane $z = 1$:

$$a = (0, 0, 0), \quad b = (b_x, b_y, 1).$$

Further, let $c = (c_x, c_y, c_z)$ be an arbitrary input point. We call ℓ_{ab} *admissible* with respect to c if

$$d(\ell_{ab}, c)^2 \leq r^2, \tag{1}$$

with

$$\begin{aligned} d(\ell_{ab}, c)^2 = &((c_y^2 + c_z^2)b_x^2 + (c_x^2 + c_z^2)b_y^2 \\ &-2c_xc_yb_xb_y - 2c_xc_zb_x - 2c_yc_zb_y \\ &+(c_x^2 + c_y^2)) \ / \ (b_x^2 + b_y^2 + 1). \end{aligned}$$

We *embed* our problem into a higher-dimensional space by setting

$$X_1 = b_x, X_2 = b_y, X_3 = b_x^2, X_4 = b_y^2, X_5 = b_xb_y. \tag{2}$$

Now, equation (1) is true if and only if

$$P_c(X_1, \ldots, X_5) \leq 0, \tag{3}$$

where P_c is the linear equation

$$\begin{aligned} &P_c(X_1, \ldots, X_5) \\ &= (-2c_xc_z)X_1 + (-2c_yc_z)X_2 \\ &\quad +(c_y^2 + c_z^2 - r^2)X_3 + (c_x^2 + c_z^2 - r^2)X_4 \\ &\quad +(-2c_xc_y)X_5 + (c_x^2 + c_y^2 - r^2). \end{aligned}$$

According to this equation, P_c defines a hyperplane in $\mathbf{R}^5$, and inequality (3) a halfspace H_c. The set of equations (2) defines a 2-dimensional manifold which can be written as

$$M = \{ \ (X_1, \ldots, X_5) : \ Q(X_1, \ldots, X_5) = 0 \ \}$$

with

$$\begin{aligned} &Q(X_1, \ldots, X_5) \\ &= (X_1^2 - X_3)^2 + (X_2^2 - X_4)^2 + (X_5 - X_1X_2)^2. \end{aligned}$$

For the set I of input points, the fixed-parameter problem has a solution if and only if there exists a line ℓ_{ab} which is admissible with respect to each $c \in I$. This is equivalent to the existence of a common intersection of the halfspaces H_c and the manifold M. The intersection

$$H = \bigcap_{c \in I} H_c$$

is a convex polytope of complexity $O(n^2)$, and can be constructed in the same time bound by Chazelle's result [Ch]. In order to intersect H with M, we triangulate H into $O(n^2)$ simplices. Each of these simplices can be tested for intersection with M separately in constant time if we assume an algebraic model of computing, and in time $O(\mu(L))$ if we assume a bit model [Re]. (Note here that M is a semi-algebraic set and the above test corresponds to deciding the satisfiability for a system of polynomial equations and inequalities.) We have shown:

Lemma 3 *The fixed radius version of problem (P2) can be solved in time $O(n^2)$ in the algebraic model, and in time $O(n^2\mu(L))$ in the bit model.*

This argument generalizes in a straightforward way to proving the general theorem 3.

Remark 3 *Due to the possible quadratic number of local minima, this result may be optimal. It is an open question whether problem (P2) belongs to the class of n^2-hard problems introduced in [GO].*

The above technique extends to problem (P1). In this case, we consider – w.l.o.g. – axis lines that are not parallel to the plane $z = 0$. Let ℓ_{ab} be the line through the points $a, b \in \mathbf{R}^3$, with

$$a = (a_x, a_y, 0), \quad b = (a_x + b_x, a_y + b_y, 1).$$

Then ℓ_{ab} is admissible with respect to $c = (c_x, c_y, c_z)$ and given radius r iff

$$P_c(a_x, a_y, b_x, b_y) \leq 0,$$

with

$$\begin{aligned}
&P_c(a_x, a_y, b_x, b_y) \\
&= c_x^2(b_y^2 + 1) + c_y^2(b_x^2 + 1) + c_z^2(b_x^2 + b_y^2) \\
&\quad + c_x c_y(-2b_x b_y) + c_x c_z(-2b_x) + c_y c_z(-2b_y) \\
&\quad + c_z(2b_y a_y + 2b_x a_x) \\
&\quad + c_x(-2a_x - 2a_x b_y^2 + 2b_x b_y a_y) \\
&\quad + c_y(-2a_y - 2a_y b_x^2 + 2b_x a_x b_y) \\
&\quad + (a_x^2 b_y^2 + a_y^2 b_x^2 + a_x^2 + a_y^2 \\
&\qquad - r^2(b_x^2 + b_y^2) - 2b_x a_x b_y a_y) \\
&\quad - r^2.
\end{aligned}$$

At first glance, P_c has an order 10 linearization. However, we can save one variable by grouping the terms with factors c_x^2, c_y^2 and c_z^2 differently:

$$\begin{aligned}
&c_x^2(b_y^2 + 1) + c_y^2(b_x^2 + 1) + c_z^2(b_x^2 + b_y^2) \\
&= (c_y^2 + c_z^2)b_x^2 + (c_x^2 + c_z^2)b_y^2 + (c_x^2 + c_y^2),
\end{aligned}$$

and setting

$$\begin{aligned}
&X_1 = b_x, X_2 = b_y, X_3 = b_x^2, X_4 = b_y^2, X_5 = b_x b_y, \\
&X_6 = b_y a_y + b_x a_x, \\
&X_7 = -a_x - a_x b_y^2 + b_x b_y a_y, \\
&X_8 = -a_y - a_y b_x^2 + b_x a_x b_y, \\
&X_9 = a_x^2(b_y^2 + 1) + a_y^2(b_x^2 + 1) - r^2(b_x^2 + b_y^2) \\
&\qquad - 2b_x a_x b_y a_y.
\end{aligned}$$

Applying theorem 3, we conclude:

Lemma 4 *The fixed radius version of problem (P1) can be solved in time $O(n^4)$ in the algebraic model, and in time $O(n^4\mu(L))$ in the bit model.*

Remark 4 *If P_c has an order 8 linearization, this fact would not improve the asymptotic complexity of the problem. But it means we could use some of the $O(n^{\lceil k/2 \rceil})$ convex hull algorithms to achieve the same complexity bounds.*

3.2 Parametric Search and Exact Approximation

In this subsection we shall apply parametric search and exact approximation to problem (P1), based on the decision algorithm from the previous subsection. Note that the presented techniques apply as well to the restricted setting (P2).

We shall use the parametric search paradigm in its general form (see eg. [AST] for a detailed description). Let T_s denote the running time of a sequential decision algorithm for the fixed-parameter problem, and T_p (resp., P) the time (resp., number of processors) of a parallel decision algorithm, then the optimal value (here, r^*) can be computed in sequential time $O(PT_p + T_sT_p \log P)$. It remains to give a parallel version of the decision algorithm. Here we exploit the new parallel algorithm for convex hulls of [AGR]. For dimension $d \geq 4$, there is an algorithm with time $O(\log n)$ and work $O(n^{\lfloor d/2 \rfloor} \log^{c(\lceil d/2 \rceil - \lfloor d/2 \rfloor)} n)$, for some constant $c > 0$. Further, with $O(n^{\lfloor d/2 \rfloor})$ processors, the test for intersection of H with M can be done in constant time in an algebraic model (resp., a real RAM, see [Re]). Plugging this into the parametric search paradigm, and observing that – in an algebraic model – the combinatorial solution of (P1) can easily be constructed from the computed optimum value r^*, we obtain:

Lemma 5 *A combinatorial solution of (P1) can be computed by parametric search in time $O(n^4 \log^k n)$, for a fixed constant $k > 0$.*

Turning our attention to the bit model, as shown in subsection 2.2, the combinatorial solution of (P1) can be obtained from an ε-approximate solution for r^* if $\varepsilon = 2^{-O(L)}$. To compute this approximate solution, it suffices to run the decision algorithm for the fixed-parameter problem $O(L)$ times, with radii of bit-size $O(L)$ as input. This yields:

Lemma 6 *A combinatorial solution of (P1) can be computed in the bit model in time $O(L\mu(L)n^4)$.*

3.3 ε-Approximation and Trade-off

Subsection 3.1 describes a decision algorithm for the fixed-parameter problem for (P1). In an algebraic model of computing, and with our assumption $\|c\| = O(1)$ for $c \in I$, this algorithm turns into an ε–approximation algorithm by using binary search for $r \in [0, 1]$:

Lemma 7 *An ε-approximate solution of problem (P1) can be computed in time $O(n^4 \log \varepsilon^{-1})$.*

As a trivial application of *discretization*, we may also exploit that elimination of the “rotational freedom” reduces (P1) to the problem of finding a smallest enclosing circle for a set of points in a plane – which can be solved in time linear in n.

Any change of the axis direction by an angle $\alpha \leq \varepsilon$ can change the location of an arbitrary, projected point c^* (see section 2) by at most $O(\varepsilon)$. Thus, we get an ε-approximation if we discretize the directions of the axis by a uniform grid on S^2. Finally, this yields a quadratic dependency of running time on $1/\varepsilon$, but only a linear dependency on n:

Lemma 8 *An ε-approximate solution of problem (P1) can be computed in time $O(n\varepsilon^{-2}\log\varepsilon^{-1})$.*

The rest of this subsection is devoted to the interesting problem of how to fill the gap between lemma 7 and lemma 8. Again, we shall use the linearization technique.

Let us consider the line ℓ_{ab} through the points

$$a = (a_x, a_y, 0), \quad b = (a_x + b_x, a_y + sb_x, 1).$$

The point b lies on a line in the plane $z = 1$, with origin (a_x, a_y) and "slope" s. Again, for given s and r, ℓ_{ab} is admissible with respect to $c = (c_x, c_y, c_z) \in I$ iff

$$P_c(a_x, a_y, b_x) \leq 0,$$

with P_c the numerator of

$$d(\ell_{ab}, c)^2 - r^2.$$

Following the same strategy to group variables in P_c as above, we can write P_c as

$$\begin{aligned}
&P_c(a_x, a_y, b_x)\\
&= b_x^2(s^2c_x^2 + c_y^2 + (1+s^2)c_z^2 - 2sc_xc_y)\\
&\quad +b_x(-2sc_yc_z - 2c_xc_z)\\
&\quad +(-2a_x + 2sb_x^2a_y - 2s^2a_xb_x^2)c_x\\
&\quad +(-2a_y - 2a_yb_x^2 + 2sa_xb_x^2)c_y\\
&\quad +(2a_xb_x + 2sa_yb_x)c_z\\
&\quad +(a_y^2b_x^2 - r^2b_x^2 + a_x^2 + a_y^2\\
&\qquad - 2sa_xa_yb_x^2 + s^2a_x^2b_x^2 - r^2s^2b_x^2)\\
&\quad -r^2.
\end{aligned}$$

Hence, we obtain a linearization of P_c with 6 variables $X_1, \ldots, X_6$. Proceeding as in 3.1, we get – for any fixed s – an algebraic decision algorithm with running time $O(n^3)$.

We now discretize (P1) by choosing lines with uniformly stepped *slope angle* for the position of b. Further, we consider choices of a and b in planes parallel to $x = 0$, $y = 0$ and $z = 0$. It is again easy to see that independent optimization for each of these instances will yield an ε-approximate solution of (P1).

Lemma 9 *An ε-approximate solution of problem (P1) can be computed in time $O(n^3\varepsilon^{-1}\log\varepsilon^{-1})$.*

Remark 5 *The presented discretizations rely on the assumption that all input points are enclosed in the unit sphere, and that we are only interested in absolute errors. In a bit model where input numbers are bounded by 2^L, it would be interesting to remove these assumptions. As mentioned earlier, in the algebraic model, Agarwal* et al. *[AAS], recently gave an approximation algorithm for finding a cylinder whose radius is at most $1+\delta$ times the optimum in time $O(n/\delta^2)$. This is accomplished by first computing a crude estimate of the direction for the optimum cylinder axis, and applying a procedure similar to that of Lemma 8.*

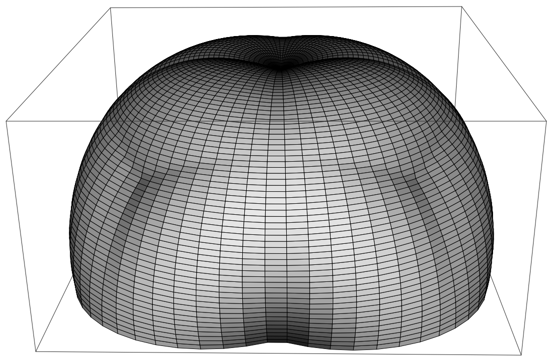

Figure 1: Polar plot of the radius of the smallest enclosing cylinder as a function of its axis orientation. The set of points is the vertices of a cube.

4 Experimental Results

In this section we describe a simple optimization method and evaluate this method by comparing its results to "exact" results that we obtained with MAPLE.

To implement a fast optimization technique, we use our usual representation of a smallest enclosing cylinder by specifying only its axial direction. By this reduction, the optimization problem can be viewed as a search for the minimum on a 2-dimensional surface in 3-space. Each point of this surface can be obtained as the result of a convex optimization problem. (An example of this surface can be seen in figure 1.) Thus, we seek the minimum of a composed function $f \circ g$. We choose an optimization technique which only requires function *evaluations* but not to compute derivatives. This technique – the standard downhill simplex algorithm as described in [PTVF] – tries to follow the direction of steepest descent. It is applied in two layers, to compute the minimum of f and (recursively) that of g.

For a given start axis, the optimization method converges to some local minimum. To locate a global minimum, one can choose a 2-dimensional grid of start values. However, our experiments indicate that there may be a better choice for optimization start values: the set of directions of edges in the convex hull of I (note the special meaning of these directions in Proposition 1).

In the sequel, we shall report on some experimental results with this special set of start values. We first computed smallest enclosing cylinders for randomly generated tetrahedra. In a sequence of 100 tests, at least one of the 6 considered start values (the edge directions of the tetrahedron) led to the optimum. The number of tests in which k start values succeeded is listed below:

k	6	5	4	3	2	1	0
successes	46	16	15	13	9	1	0

In two additional test sequences, we tested 50 sets of 5 random points, and 10 sets of 8 random points (the coordinates have been chosen in a way to guarantee convex position). Again, in each test the downhill simplex algorithm converged to the minimum for at least one starting value, and generally for many.

The most complex examples which we tried consisted of 12 points. The MAPLE implementation did run several days on these sets to find the optimum. The numerical optimization converged within seconds for each starting value. To stimulate further research, we include the data as benchmarks:

Example 1: The 12 input points are arranged near to two circles of radius 10 in the two parallel planes $z = -10$ and $z = 20$:

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}
x	5	−5	−11	−5	5	10	4	−6	−11	−5	5	9
y	8	9	0	−8	−9	0	9	9	1	−9	−8	−1
z	−10	−9	−10	−11	−10	−9	21	20	19	20	21	18

According to MAPLE, the smallest enclosing cylinder passes through the 5 points $c_3, c_5, c_6, c_9, c_{10}$, and has radius 10.5003 ± 10^{-4}. The second-smallest enclosing cylinder through 5 points touches $c_3, c_5, c_6, c_8, c_{10}$, and has radius ≈ 10.5009.
The downhill simplex algorithm converged to the minimum radius ≈ 10.5003 for the starting values (c_7, c_4), (c_9, c_3), (c_{10}, c_4), (c_{12}, c_5) and (c_{12}, c_1).

Example 2: The 12 input points are arranged near the 12 vertices of an icosahedron with center at the origin:

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}
x	−12	−12	9	10	23	10	−10	−24	12	12	−10	−8
y	−19	1	−13	12	−1	−21	21	1	19	−1	−12	12
z	−6	−20	−18	−17	−2	5	−6	2	6	20	17	18

The optimal solution has been computed by MAPLE as the cylinder through the 5 points $c_1, c_3, c_7, c_9, c_{11}$, with radius ≈ 21.0309. All but 2 points lie close to the surface of this cylinder. (For an exact model of the icosahedron, the optimal cylinders would pass through 10 points.) The downhill simplex algorithm obtained this solution for the starting values (c_3, c_1), (c_3, c_2), (c_4, c_2), (c_5, c_1), (c_8, c_5) and (c_9, c_8).

To conclude this section, we observe that the proposed downhill algorithm behaves amazingly well, and did not fail for the examples we tried.

5 Final Remarks

As the field of geometric optimization matures, it treats problems of increasingly non-trivial algebraic complexity. The traditional neglect of bit complexity is no longer justified. The smallest cylinder problem is one of these problems. By combining the general linearization technique with parametric search, we developed efficient algorithms in both models. These results, as well as the exact results of Agarwal *et al.*, seem mainly of theoretical interest.

The ε-approximation schemes have possibly greater practical applicability. But even here, our numerical experiments suggest that these may not be competitive with some heuristic numerical approaches. For specific applications, like metrology, the extra effort of using an exact approach can be justified. Here, the model of computing has to be chosen carefully, and the bit model offers advantages when compared to highly involved algorithms that operate in the algebraic model. For other applications, like assembly planning, computational geometry will probably fail in an attempt to substitute numerical approaches by exact and "efficient" but highly complex algorithms. Here, theoretical contributions can increase the understanding of numerical techniques.

References

[AAS] P. K. Agarwal, B. Aronov, M. Sharir, “Line Transversals of Balls and Smallest Enclosing Cylinders in Three Dimensions”, *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, to appear, 1997.

[AGR] N. Amato, M. Goodrich, E. Ramos, “Parallel algorithms for higher-dimensional convex hulls”, *IEEE FOCS*, 1994, pp. 683–694.

[AM] P. K. Agarwal, J. Matoušek, “On range searching with semialgebraic sets”, *Discrete Comput. Geom.*, 11, 1994, pp. 393-418.

[AST] P. Agarwal, M. Sharir, S. Toledo, “Applications of parametric searching in geometric optimization”, *Journal of Algorithms*, 17, 1994, pp. 292-318.

[Asme] ASME Y14.5M-1994, *Dimensioning and tolerancing*, American Society of Mechanical Engineers, New York, NY, 1994.

[Ca] J. Canny, *The Complexity of Robot Motion Planning*, MIT Press, 1987.

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

[Fo] F. Follert, “Maxmin location of an anchored ray in 3-space and related problems”, *7th Canadian Conf. on Comp. Geom.*, Quebec, 1995, pp. 7–12.

[FSS] F. Follert, E. Schömer, J. Sellen, “Subquadratic algorithms for the weighted maximin facility location problem”, *7th Canadian Conf. on Comp. Geom.*, Quebec, 1995, pp. 1–6.

[GO] A. Gajentaan, M. Overmars, “On a class of $O(n^2)$ problems in computational geometry”, *Comput. Geom. Theory Appl.*, 5, 1995, pp. 165–185.

[GK] P. Gritzmann, V. Klee, ”Computational complexity of inner and outer j-radii of polytopes in finite-dimensional normed spaces”, *Math. Program.*, Vol. 59, 1993, pp. 163-213.

[FKS] U. Faigle, W. Kern, M. Streng, ”Note on the computational complexity of j-radii of polytopes in $\mathbf{R}^n$”, Electronic Colloquium on Computational Complexity Report TR95-014, 1995.

[Me] N. Megiddo. “Applying parallel computation algorithms in the design of serial algorithms”, *Journal of the ACM*, 30, 1983, pp. 852-865.

[Pa] J. Pach, unpublished notes.

[PTVF] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1988.

[Re] J. Renegar, “Recent progress on the complexity of the decision problem for the reals”, *DIMACS Series in Discr. Math. and Th. Comp. Sc.*, Vol. 6, 1991, pp. 287-308.

[ST] E. Schömer, Ch. Thiel, “Efficient collision detection for moving polyhedra”, *11th ACM Symposium on Comp. Geom.*, Vancouver, 1995, pp. 51-60.

[SV] V. Srinivasan, H. B. Voelcker. *Dimensional Tolerancing and Metrology*, The American Society of Mechanical Engineers, 345 East 47th Street, New York, NY 10017, CRTD-Vol. 27, 1993.

[S] M. Streng, *Chebyshev approximation of finite sets by curves and linear manifolds*, Ph.D. thesis, University of Twente, Netherlands, 1993.

[YY] A.C. Yao, F.F. Yao, "A general approach to D-dimensional geometric queries", *Proc. 17th Annu. ACM Sympos. Theory Comput.*, 1985, pp. 163-168.

[Ya] C. Yap, "Exact Computational Geometry and Tolerancing Metrology", in *Snapshots of Computational and Discrete Geometry, Vol.3*, eds. David Avis and Jit Bose, McGill School of Comp.Sci, Tech.Rep. No.SOCS-94.50, A Volume Dedicated to Godfried Toussaint, 1994, pp. 34–48.