Better Lower Bounds on Detecting Affine and Spherical Degeneracies*

Jeff Erickson
Computer Science Division
University of California
Berkeley, CA 94720, USA
jeffe@cs.berkeley.edu

Raimund Seidel
Computer Science Division
University of California
Berkeley, CA 94720, USA
seidel@cs.berkeley.edu

Abstract

We show that in the worst case, $\Omega(n^d)$ sidedness queries are required to determine whether a set of n points in $\mathbb{R}^d$ is affinely degenerate, *i.e.*, whether it contains $d+1$ points on a common hyperplane. This matches known upper bounds. We give a straightforward adversary argument, based on the explicit construction of a point set containing $\Omega(n^d)$ "collapsible" simplices, any one of which can be made degenerate without changing the orientation of any other simplex. As an immediate corollary, we have an $\Omega(n^d)$ lower bound on the number of sidedness queries required to determine the order type of a set of n points in $\mathbb{R}^d$. Using similar techniques, we also show that $\Omega(n^{d+1})$ in-sphere queries are required to decide the existence of spherical degeneracies in a set of n points in $\mathbb{R}^d$.

1 Introduction

A fundamental problem in computational geometry is determining whether a given set of points is in "general position." A simple example of this type of problem is determining, given a set of points in the plane, whether any three of them are colinear. In 1983, van Leeuwen [15] asked for an algorithm to solve this problem in time $o(n^2 \log n)$. Chazelle, Guibas, and Lee [2] and Edelsbrunner, O'Rourke, and Seidel [6] independently discovered an algorithm that runs in time and space $O(n^2)$ by constructing the arrangement of lines dual to the input points. Edelsbrunner *et al.* [6] also solved the higher-dimensional version of this problem, which we call the *affine degeneracy problem.* Their algorithm, given n points in $\mathbb{R}^d$, determines

*An earlier version of this paper was presented at the 34th Annual IEEE Symposium on Foundations of Computer Science [8]. This research has been supported by NSF Presidential Young Investigator grant CCR-9058440.

whether $d+1$ of them lie on the same hyperplane, in time and space $O(n^d)$ (but see also [7]). Edelsbrunner and Guibas [5] later improved the space bound to $O(n)$ in all dimensions.

A basic primitive used by all of these algorithms is the *sidedness query*: Given $d+1$ points $p_0, \ldots, p_d$, does the point p_0 lie "above", on, or "below" the oriented hyperplane $\mathrm{aff}(p_1, \ldots, p_d)$? These are also sometimes called orientation queries, simplex queries, or (in the plane) triangle queries. The result of a sidedness query is given by the sign of the following determinant.

$$\begin{vmatrix} 1 & p_{01} & p_{02} & \cdots & p_{0d} \\ 1 & p_{11} & p_{12} & \cdots & p_{1d} \\ & & \vdots & & \\ 1 & p_{d1} & p_{d2} & \cdots & p_{dd} \end{vmatrix}$$

The *value* of this determinant is $d!$ times the signed volume of the simplex spanned by $p_0, \ldots, p_d$.

In the algebraic decision tree and algebraic computation tree models, there is a somewhat trivial lower bound of $\Omega(n \log n)$ on finding affine degeneracies in any dimension, since it takes $\Omega(n \log n)$ time just to determine whether all the points are distinct [14, 1]. Previously, no better lower bound was known under any model of computation.

In this paper, we derive a lower bound of $\Omega(n^d)$ on the number of sidedness queries required to determine whether set of n points in $\mathbb{R}^d$ is affinely degenerate. This matches the known upper bounds. Our lower bound holds in a decision tree model of computation in which every decision is based on the result of a sidedness query. We are not allowed, for example, to compare the values of different sidedness determinants. This is not quite as unreasonable a restriction as it may appear at first glance; all known algorithms for determining degeneracy or order type rely (or can be made to rely) exclusively on sidedness queries [2, 5, 6]. Our lower bound implies that there is no hope of improving these algorithms unless other primitives are used.

Two sets of labeled points are said to have the same *order type* if corresponding simplices have the same orientation. The order type of a set of points can be represented by the face lattice of its dual hyperplane arrangement or by its lambda-matrix [11], both representations requiring space $\Omega(n^d)$. Arbitrary order types can also be represented by canonical point sets with integer coordinates, but this representation can require exponential storage in the worst case [13]. The fastest known algorithm for determining the order type of a set of points constructs its dual hyperplane arrangement in time and space $O(n^d)$ [6]. Even though all known representations of order type require space $\Omega(n^d)$, there is some hope of a smaller representation, and thus, a faster algorithm, since it is known that there are only $(n/d)^{\Theta(d^2 n)} = 2^{\Theta(n \log n)}$ order types [12]. As an immediate corollary to our degeneracy lower bound, we show that $\Omega(n^d)$ sidedness queries are required to determine the order type of a set of n points in $\mathbb{R}^d$. Previously, the information-theoretic lower bound of $\Omega(n \log n)$ was the only lower bound known for this problem.

We prove our lower bounds with a straightforward adversary argument. We describe a nondegenerate set of points that contains a large number of independent "collapsible" simplices, any one of which the adversary can make degenerate without changing the orientation of any other simplex. If an algorithm fails to perform a sidedness query for every collapsible simplex, the adversary can move the points so that the perturbed set is degenerate, and the algorithm will be unable to distinguish between the original set and the perturbed set.

In an analogous development, we show that determining the existence of *spherical degeneracies*, *i.e.*, deciding whether a set of n points in $\mathbb{R}^d$ contains $d+2$ points on a common sphere, requires $\Omega(n^{d+1})$ in-sphere queries. Again this matches known upper bounds, which are typically achieved by reducing the spherical degeneracy problem in $\mathbb{R}^d$ to an affine degeneracy problem in $\mathbb{R}^{d+1}$ (see [4]). The typically used in-sphere query decides whether a point q lies "inside", on, or "outside" the oriented sphere defined by $d+1$ points $p_0, \ldots, p_d$. Algebraically, the result of this query is given by the sign of the following determinant.

$$\begin{vmatrix} 1 & p_{01} & p_{02} & \cdots & p_{0d} & \sum_i p_{0i}^2 \\ 1 & p_{11} & p_{12} & \cdots & p_{1d} & \sum_i p_{1i}^2 \\ & & & \vdots & & \\ 1 & p_{d1} & p_{d2} & \cdots & p_{dd} & \sum_i p_{di}^2 \\ 1 & q_1 & q_2 & \cdots & q_d & \sum_i q_i^2 \end{vmatrix}$$

In Section 2, we derive lower bounds on a simplified version of the affine degeneracy problem, in which vertical hyperplanes are ignored. We extend our results to the general problem in Section 3. In Section 4, we strengthen the model of computation somewhat by allowing some additional tests such as comparisons between coordinates. Section 5 addresses the spherical degeneracy problem. In Section 6, we describe an algorithm for another simplified version of the degeneracy problem, whose running time is less than our lower bound. Finally, we discuss some of the implications of our results.

2 Lower Bounds for a Simplified Affine Degeneracy Problem

The *non-vertical affine degeneracy problem* asks, given a set of n points in $\mathbb{R}^d$, whether there is a non-vertical hyperplane passing through $d+1$ of them. In this section, we prove the following theorem.

Theorem 2.1. *Any decision tree algorithm that solves the non-vertical affine degeneracy problem in $\mathbb{R}^d$, using only sidedness queries, must have depth $\Omega(n^d)$.*

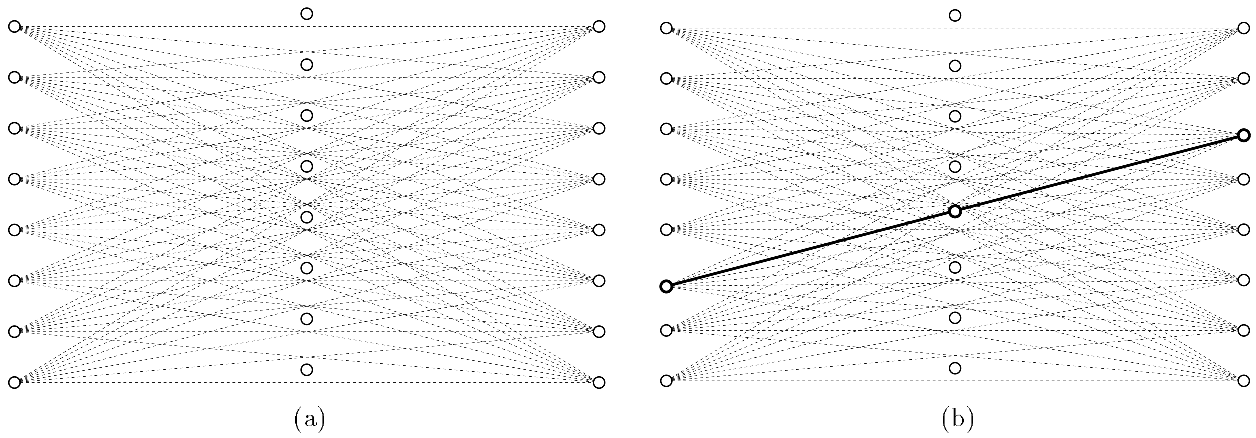

Figure 1. Planar lower bound construction. (a) Initial set. (b) Perturbed set, showing collapsed triangle.

In order to give a more intuitive picture, we first consider the planar case, and then generalize to arbitrary dimensions.

2.1 The Planar Lower Bound

Without loss of generality, we assume n is a multiple of 3. The adversary presents the following set of points:

$$S \triangleq \bigcup_{i=1}^{n/3} \{(-1, 4i), (0, 4i+1), (1, 4i)\}.$$

The set S consists of three smaller sets of points, evenly spaced along vertical line segments. See Figure 1(a). If we pick points p and r from the left and right segments, respectively, there is a unique point q in the middle segment such that the vertical distance from q to $\overleftrightarrow{pr}$ is exactly one. We shall refer to each such triple $\{p, q, r\}$ as a *collapsible triangle*, for the following reason. Without loss of generality, let q lie below $\overleftrightarrow{pr}$. If we perturb the set by moving p and r down by $1/2$ and moving q up by $1/2$, then the three points become colinear. See Figure 1(b). No other degeneracies are introduced by this perturbation. Moreover, no other triangle changes orientation.

The adversary's point set contains $n^2/9 = \Omega(n^2)$ collapsible triangles. If the algorithm does not check the orientation of every collapsible triangle, the adversary perturbs the set so that some unchecked triangle becomes degenerate. The algorithm cannot distinguish between the original point set and the perturbed point set. This completes the proof in the planar case.

It may be helpful to see what this construction looks like in the dual setting. Here we are given n lines in the plane and asked if any three of them have a common

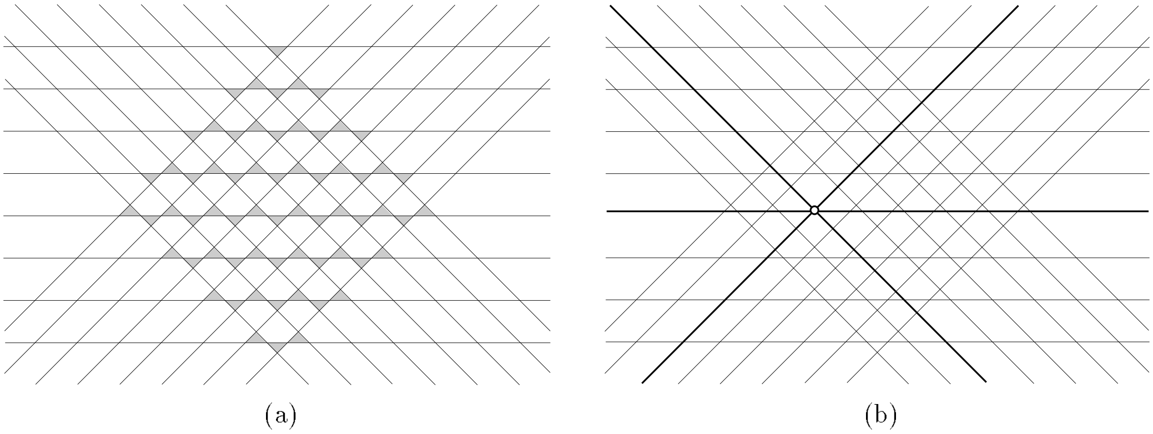

Figure 2. Dual version of the planar lower bound construction. (a) Initial set, with collapsible triangles shaded. (b) Perturbed set, showing collapsed triangle.

intersection.[1] The dual of the adversary's point set consists of three bundles of parallel lines. Two of the bundles meet in a mesh of squares, and the third cuts through the squares at a 45° angle, so that each square in the mesh has a small triangle cut off one corner. See Figure 2(a). Each of the small triangles in the mesh corresponds to a collapsible triangle in the primal point set. To collapse a triangle, the adversary simply moves its three bounding lines so that they intersect at the triangle's centroid. See Figure 2(b).

2.2 Higher Dimensions

For the d-dimensional problem, the adversary's point set consists of $d+1$ smaller sets. The points in each smaller set are evenly spaced along vertical line segments $l_0, l_1, \ldots, l_d$. These line segments intersect any horizontal hyperplane at the centroid and vertices of a regular $(d-1)$-simplex.

Without loss of generality, we assume n is a multiple of $3d$. Each of the outer segments $l_1, \ldots, l_d$ contains $2n/3d$ points, and l_0 contains the remaining $n/3$ points. The x_d coordinates of the outer points are multiples of $2d$ between 0 and $4n/3 - 2d$. Thus, any hyperplane defined by d points, one from each outer segment, intersects the x_d-axis at an even integer coordinate between 0 and $4n/3 - 2d$. The points in the inner set lie at alternate odd integer coordinates between 1 and $4n/3 + 1$. This gives us $\lfloor (d-1)/2 \rfloor$ "wasted" points at the top of the inner segment, which we can ignore.

Suppose we pick one point from each of the outer sets. These points define a hyperplane h. The vertical distance between h and the unique point in the inner

[1]The restriction to non-vertical colinearities in the primal setting is reflected in the dual by ignoring the intersection points "at infinity" between parallel lines.

set that is closest to h is exactly 1. We refer to each such set of $d+1$ points as a *collapsible simplex*. The adversary can make any collapsible simplex degenerate by simultaneously moving the inner point up and the outer points down (or vice versa) a distance of 1/2. Clearly, no other simplex changes orientation because of this perturbation. There are $(2n/3d)^d = \Omega(n^d)$ collapsible simplices in the adversary's point set, each of which must be checked by the algorithm.

This completes the proof of Theorem 2.1.

2.3 Lower Bounds for Determining Order Type

Two sets of labeled points have the same order type if and only if corresponding sidedness queries return the same value. Since collapsing a simplex changes the order type of the set, we immediately have the following corollary.

Corollary 2.2. *Any decision tree algorithm that determines the order type of a set of n points in $\mathbb{R}^d$, using only sidedness queries, must have depth $\Omega(n^d)$.*

3 Lower Bounds for the General Problem

In this section, we modify the arguments of the previous section to derive a lower bound for the *general affine degeneracy problem*: Given n points in $\mathbb{R}^d$, is there a hyperplane that passes through more than d of them?

In the plane, we can explicitly modify the set S given in the previous section. Specifically, the adversary presents a set of points closely approximating S, except that the vertical line segments are replaced by very steep parabolas.

$$S^* \triangleq \bigcup_{i=1}^{n/3} \left\{ \left(\frac{i^2}{n^2} - 1, 4i\right), \left(\frac{i^2}{n^2}, 4i+1\right), \left(\frac{i^2}{n^2} + 1, 4i\right) \right\}$$

Any line through two points on the same parabola misses the points on the other two parabolas. Since no three points on a parabola can be colinear, the set is clearly nondegenerate. As in the original construction, there are $n^2/9$ collapsible triangles. (Actually, there are slightly more, but we only care about the collapsible triangles with one point in each parabola.) Each point in S^* is at most 1/9 away from the corresponding point in S. Thus, in each collapsible triangle, the vertical distance from the inner vertex to the line connecting the outer vertices is between 7/9 and 11/9. To collapse a triangle, the adversary slides its vertices along the corresponding parabolas. This ensures that no other triangle collapses or changes orientation.

The situation in higher dimensions is not so simple, since there are many more possibilities for degeneracies than in the plane. We would like to simply specify moment curves that approximate the original vertical line segments, and place points on these curves at regular intervals. In the plane, if the approximating curves are steep enough, we only have to ensure that no three points from the same curve segment can be colinear. Even in three dimensions, the situation is much more complicated. Simply by choosing moment curves, we ensure that no four points from the same segment are coplanar, but there are other ways of choosing four points — three from one segment and one from another, two each from two segments, and so forth — and we must guarantee that none of these choices is coplanar, even after we collapse an arbitrary collapsible simplex. With sufficient patience, one can construct an appropriate three-dimensional point set, but as the dimension increases, we quickly face a combinatorial explosion, and the problem becomes intractable.

Because of these difficulties, rather than giving an explicit construction, we content ourselves with a pure existence proof. Given a set of points S and a nondegenerate simplex σ with vertices in S, we say σ is *reversible* if S can be continuously deformed to a new set in which the orientation of σ is reversed, without changing the orientation of any other simplex at any time. Clearly, every reversible simplex is collapsible. Furthermore, we easily verify that each of the collapsible simplices in the adversary sets described in section 2 is reversible.

Lemma 3.1. *There exists a non-degenerate set of n points in $\mathbb{R}^d$ containing $\Omega(n^d)$ reversible simplices.*

Proof: We can think of any set of n points in $\mathbb{R}^d$ as a single point in the "configuration space" $\mathbb{R}^{dn}$ [12]. Each of the $\binom{n}{d+1}$ possible sidedness determinants implicitly define an algebraic surface in $\mathbb{R}^{dn}$, and these surfaces define a cellular decomposition. Non-degenerate sets correspond to points in dn-dimensional cells in this decomposition; degenerate sets correspond to points in cells of lower dimension.

Each configuration of points in $\mathbb{R}^d$ corresponds to a point in some cell C in this decomposition. As long as the order type of the set does not change, moving points in the set corresponds to moving the configuration point within C. Reversing a non-degenerate simplex moves the configuration point through the corresponding surface σ. If no other simplex changes orientation, then the configuration point neither meets nor crosses any other surface, except those surfaces containing the entire path of the configuration point. In this case, we say that σ *nicely bounds* C. More formally, σ uniquely spans a boundary facet of C and cuts the intersection of the surfaces that contain C at some point on that boundary facet.[2] Thus, there is a one-to-one correspondence between reversible simplices and nice bounding surfaces. To prove the lemma, it suffices to show that there is a dn-dimensional cell that is nicely bounded by $\Omega(n^d)$ distinct surfaces.

Let C be an arbitrary cell, and let C' be one of the cells in its boundary. Then every surface σ that nicely bounds C' also nicely bounds C. Thus, if there is a

[2] Let $N_\varepsilon(p)$ denote the open ball of radius ε centered at the point p. Suppose an algebraic surface σ intersects an algebraic variety V at a point p. We say that σ cuts V at p if for all sufficiently small ε, V intersects both components of $N_\varepsilon(p) \setminus \sigma$.

cell of *any* dimension nicely bounded by $\Omega(n^d)$ surfaces, then by induction, there must be a dn-dimensional cell bounded by $\Omega(n^d)$ surfaces. Since reversible simplices correspond to nice bounding surfaces, it suffices to show that there exists a (possibly degenerate) set of points containing $\Omega(n^d)$ reversible simplices. We described such a set in the proof of Theorem 2.1. □

Our adversary argument now immediately gives us the following two theorems.

Theorem 3.2. *Any decision tree algorithm that solves the general affine degeneracy problem in* $\mathbb{R}^d$*, using only sidedness queries, must have depth* $\Omega(n^d)$*.*

Theorem 3.3. *Any decision tree algorithm that determines the order type of a non-degenerate set of* n *points in* $\mathbb{R}^d$*, using only sidedness queries, must have depth* $\Omega(n^d)$*.*

4 Strengthening the Model of Computation

If we modify our planar construction slightly, we can also allow comparisons between coordinates of points and between slopes of lines defined by pairs of points, thus strengthening the model under which our lower bound holds. In fact, even with these extra tests, $\Omega(n^2)$ sidedness queries are still required in the worst case.

It is somewhat easier to explain the modification in dual space. We use the duality transformation of Edelsbrunner *et al.* [6], which takes each point (a, b) to the line $y = ax + b$, and vice versa. We modify our original collection of lines, illustrated in Figure 2(a), so that any algorithm can compare coordinates of intersection points, slopes of lines, and intercepts of lines with either coordinate axis.

We can already allow comparisons between slopes, since the adversary's perturbation doesn't change the slope of any line. We can translate the set of lines so that all intersection points lie in the same quadrant, even after the adversary collapses any triangle. The translated set has the property that the order in which the lines intersect either coordinate axis is not changed when a triangle is collapsed.

To allow comparisons between coordinates of intersection points, we modify the set as follows. First, we shrink all the "down" triangles in the mesh to points. Second, we rotate the entire construction slightly, so that every intersection point has a unique x- and y-coordinate. Finally, we expand the triple intersection points back into very small triangles, all the same size. If the intersection points are ordered by either coordinate, the vertices of any of the small triangles are adjacent in the ordering. See Figure 3.

Each of the $\lceil n^2/18 \rceil$ small triangles is reversible, and thus collapsible. If the adversary collapses one of these triangles, the only place the coordinate orders change is at the vertices of that triangle. However, since none of our lines are horizontal or

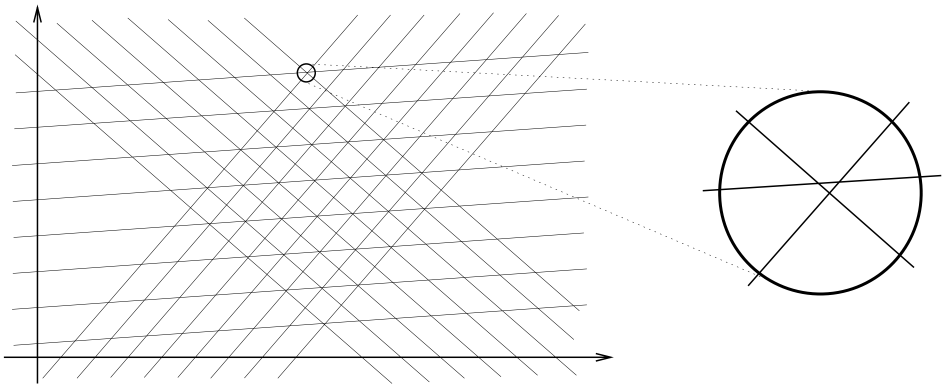

Figure 3. Modified dual construction. Each triple intersection point is actually a very small collapsible triangle.

vertical, comparing the coordinates of two intersection points of a triangle is algebraically the same as performing a sidedness query. So for each collapsible triangle, any algorithm must perform a sidedness query on its three lines.

Theorem 4.1. *Any decision tree algorithm that solves the non-vertical affine degeneracy problem in the plane, using only sidedness queries, coordinate comparisons, and slope comparisons, must make* $\Omega(n^2)$ *sidedness queries in the worst case.*

We can apply the technique of Section 3 to achieve lower bounds on the general problem.

Theorem 4.2. *Any decision tree algorithm that solves the general affine degeneracy problem in the plane, using only sidedness queries, coordinate comparisons, and slope comparisons, must make* $\Omega(n^2)$ *sidedness queries in the worst case.*

Proof: Each sidedness query, slope comparison, and coordinate comparison implicitly defines an algebraic surface in the configuration space $\mathbb{R}^{2n}$. (We ignore any slope comparison between two sides of the same triangle, since such a test is algebraically equivalent to a sidedness query.) For notational convenience, we color each surface "red" if it is defined by a sidedness determinant, and "green" otherwise. These surfaces define a cellular decomposition of $\mathbb{R}^{2n}$. There is a one-to-one correspondence between the reversible triangles in any set of points in the plane and the *red* surfaces that nicely bound the cell containing the corresponding configuration point.

The set of points we described above corresponds to a configuration point in a low-dimensional cell nicely bounded by $\Omega(n^2)$ red surfaces. Every cell in the decomposition is nicely bounded by at least as many red surfaces as any cell in its boundary. By induction, therefore, there must be a $2n$-dimensional cell nicely bounded by $\Omega(n^2)$ red surfaces. The theorem follows immediately.[3] □

Corollary 4.3. *Any decision tree algorithm that determines the order type of a non-degenerate set of n points in the plane, using only sidedness queries, coordinate comparisons, and slope comparisons, must make $\Omega(n^2)$ sidedness queries in the worst case.*

5 Lower Bounds for Finding Spherical Degeneracies

The *spherical degeneracy problem* asks, given n points in $\mathbb{R}^d$, if any $d+2$ lie on the same sphere. This problem can be transformed into the affine degeneracy problem in $\mathbb{R}^{d+1}$ by projecting the input vertically onto the paraboloid $x_{d+1} = x_1^2 + \cdots + x_d^2$. The images of cospherical points in $\mathbb{R}^d$ under this projection are coplanar points in $\mathbb{R}^{d+1}$. Furthermore, if the point q lies inside (resp. outside) the sphere defined by $d+1$ points $p_0, \ldots, p_d$ in $\mathbb{R}^d$, then the image of q lies below (resp. above) the hyperplane in $\mathbb{R}^{d+1}$ defined by the images of $p_0, \ldots, p_d$ [4]. Sidedness queries on this $(d+1)$-dimensional point set are thus equivalent to *in-sphere queries* in the original d-dimensional point set.

In this section, we prove a lower bound on the number of in-sphere queries required to solve the spherical degeneracy problem, using the techniques in Sections 2 and 3. As before, we first consider only a simplified version of the problem, in which spheres with infinite radius (*i.e.*, hyperplanes) are ignored. We refer to any set of $d+2$ points lying on a common sphere of finite radius as a *proper* spherical degeneracy.

We give a simple construction that deals with the planar case, followed by a more complicated construction that deals with the general d-dimensional case.

5.1 The Planar Lower Bound

The adversary presents the following set of points:

$$S \triangleq \bigcup_{i=1}^{n/6} \left\{(2^{5i+2}, 0), (2^{5i-2}, 0), (0, 2^{5i})\right\} \cup \bigcup_{i=1}^{n/2} \left\{(0, 2^{5(i-n/6)+1})\right\}$$

The set consists of four subsets of points, two contained in each positive coordinate axis. We easily verify that this set contains no proper circular degeneracies, using the fact that four points $(a, 0)$, $(b, 0)$, $(0, c)$, and $(0, d)$ are cocircular if and only if $ab = cd$.

[3] Actually, this statement is stronger than the theorem requires. It suffices that some cell, contained in only green surfaces, is nicely bounded by $\Omega(n^2)$ red surfaces.

Theorem 5.1. *Any decision tree algorithm that solves the proper circular degeneracy problem, using only in-circle queries, must have depth $\Omega(n^3)$.*

Proof: For each $1 \le i, j, k \le n/6$, the following points are "almost" cocircular.

$$(2^{5i+2}, 0), (2^{5j-2}, 0), (0, 2^{5k}), (0, 2^{5(i+j-k)+1})$$

Each such set of points is a *reversible* 4*-tuple*. The adversary can collapse any such tuple by changing the four points to the following.

$$(2^{5i+3/2}, 0), (2^{5j-3/2}, 0), (0, 2^{5k-1/2}), (0, 2^{5(i+j-k)+1/2})$$

We easily verify that this change does not introduce any other new circular degeneracies or change the result of any other in-circle query. There are $n^3/216 = \Omega(n^3)$ collapsible 4-tuples, each of which must be checked by the algorithm. □

Once again, we can extend this result to the general problem using the technique of Section 3, since each of the collapsible 4-tuples in S is actually reversible.

Theorem 5.2. *Any decision tree algorithm that solves the general circular degeneracy problem, using only in-circle queries, must have depth $\Omega(n^3)$.*

The adversary's perturbation does not change the relative coordinate orders of any pair of points or the orientation of any triangle. Thus, using the techniques of Section 4, we can strengthen the model under which our lower bound holds.

Theorem 5.3. *Any decision tree algorithm that solves the general circular degeneracy problem, using only in-circle queries and sidedness queries, must make $\Omega(n^3)$ sidedness queries in the worst case.*

5.2 Higher Dimensions

In order to extend this lower bound to the d-dimensional case, we exhibit a set S of $O(n)$ points in $\mathbb{R}^d$ that contains $\Omega(n^{d+1})$ reversible $(d+2)$-tuples, *i.e.*, sets of $d+2$ non-cospherical points in S that can be moved so that they become cospherical, without changing the result of any other in-sphere query.

The point set S in question is the union of $d+1$ smaller sets, $S_1 \cup \cdots \cup S_d \cup D$, where each S_i consists of $n/2$ even integer points on the positive x_i-axis, and D consists of about $(d+1)n/2$ "odd" points on the main diagonal $(t, \ldots, t)$. At the risk of confusing the reader, we let each subscripted variable t_i refer simultaneously to a point on the x_i-axis and that point's non-zero coordinate. Similarly, each unsubscripted variable t refers simultaneously to a point on the main diagonal and the value of all its coordinates.

To make our construction precise, the sets S_i include points t_i such that t_i is even and

$$a_i < t_i \leq a_i + n,$$

where $a_1 = 0$, a_i is large for all $1 < i < d$ (say $a_i = n^3 + in$), and a_d is huge (say $a_d = 2^n$). The set D includes points t such that dt is odd and

$$A < dt \leq A + (d+1)n,$$

where $A = (a_1 + \cdots + a_d)$.

Lemma 5.4. *The set S contains no proper spherical degeneracies.*

Proof: For all $1 \leq i \leq d$, let t_i and t_i' be two distinct points in S_i, and let t and t' be two distinct points in D. Note that with our choice of values for a_i we have the following bounds.

$$\begin{array}{rcccl} -(d+1)n & < & t_1 + \cdots + t_d - dt & < & dn \\ 1/n & < & \dfrac{1}{t_1} + \cdots + \dfrac{1}{t_d} - \dfrac{1}{t} & < & 1 \\ \dfrac{d-2}{n^3} - o\left(\dfrac{1}{n^3}\right) & < & \dfrac{1}{t_2} + \cdots + \dfrac{1}{t_d} - \dfrac{1}{t} & < & \dfrac{d-2}{n^3} \end{array}$$

By examining the appropriate in-sphere determinants, we find that the cosphericity of any set of $d+2$ points from S is expressed by the vanishing of one of the following algebraic expressions.

- Two points from the x_i-axis, one from each of the other axes, and one from the main diagonal:

$$t_1 + \cdots + t_d - dt + t_i t_i' \left(\frac{1}{t_1} + \cdots + \frac{1}{t_d} - \frac{1}{t}\right) \tag{1}$$

- Two points from the main diagonal, and one from each axis:

$$t_1 + \cdots + t_d - dt + dtt' \left(\frac{1}{t_1} + \cdots + \frac{1}{t_d} - \frac{1}{t}\right) \tag{2}$$

- Two points from the x_i axis, two points from the x_j axis, and $d-2$ points elsewhere:

$$t_i t_i' - t_j t_j' \tag{3}$$

- Two points from the x_i axis, two points from the main diagonal, and $d-2$ points elsewhere:

$$t_i t_i' - dtt' \tag{4}$$

With t_i, t_i', t, t' chosen in the indicated ranges and with the indicated parities, expression (1) never vanishes, since the last term dominates when $i > 1$, and the whole expression differs from an odd integer by less than d/n when $i = 1$. Expression (2) never vanishes, since the last term always dominates. Expression (3) never vanishes, since the x_i-range and the x_j-range are disjoint. Finally, expression (4) never vanishes, since the second term dominates when $i < d$, and the first term dominates when $i = d$. □

Lemma 5.5. *The set S contains $\Omega(n^{d+1})$ reversible $(d+2)$-tuples.*

Proof: For any choice of two distinct points t_1, t_1' from S_1 and one point t_i from each of the other S_i, we can choose the point from D with all coordinates equal to $(t_1 + \cdots + t_d + t_1' - 1)/d$, so that these points form a reversible $(d+2)$-tuple. To reverse the tuple, one can decrease the non-zero coordinates of the axis points by $1/(2d+2)$ and increase each coordinate of the main diagonal point by $1/2d + 1/n$. □

Appealing to the previous method of perturbing S into non-degenerate position, we conclude:

Theorem 5.6. *Any decision tree algorithm that solves the general spherical degeneracy problem in $\mathbb{R}^d$, using only in-sphere queries, must have depth $\Omega(n^{d+1})$.*

The correspondence between affine degeneracies in $\mathbb{R}^d$ and spherical degeneracies in $\mathbb{R}^{d-1}$ immediately implies the following stronger version of Theorem 3.2.

Corollary 5.7. *For all $d > 2$, any decision tree algorithm that solves the general affine degeneracy problem in $\mathbb{R}^d$, using only sidedness queries, must have depth $\Omega(n^d)$, even if the input is restricted to sets of points in convex position.*

Given a set of points in convex position in the plane, we can easily determine whether any three are colinear in $O(n \log n)$ time.

6 Beating the Lower Bound

In this section, we describe two simple algorithms which beat our lower bound for another simplified version of the affine degeneracy problem. Our algorithms do not use only sidedness queries, but also compute the signs of certain linear forms. In addition to providing a pedagogical example of the importance of choosing the right model of computation, these algorithms indicate that a new approach will be required to extend our lower bounds into more general models of computation, at least in higher dimensions.

Consider the following problem. Suppose we have $d+1$ sets $S_0, \ldots, S_d \subset \mathbb{R}^d$, each containing n points, such that each set S_i is contained in a vertical line l_i. Given such a collection of points, is there a non-vertical hyperplane containing $d+1$ of them? We call this the *restricted* affine degeneracy problem. Clearly, the results in Section 2 show that $\Omega(n^d)$ sidedness queries are required to solve this problem in the worst case.

Since all the points lie on $d+1$ vertical lines, the only possible non-vertical degeneracies consist of one point from each line. For each input, there exist constants α_i, determined by the positions of the lines l_i, such that any collection of points $p_0 \in l_0, \ldots, p_d \in l_d$ contained in a non-vertical hyperplane satisfies the equation

$$\sum_{i=0}^{d} \alpha_i p_{id} = 0.$$

We describe two algorithms, one for even dimensions and one for odd dimensions. Our algorithms compare the weighted sums of k-tuples of x_d-coordinates of points, where the weight of each point is determined by the set from which it is taken. We call such a query a *k-tuple comparison.* Both algorithms work in two phases, a sorting phase and a scanning phase. In the sorting phase, both algorithms perform $\lceil d/2 \rceil$-tuple comparisons. In the scanning phase, both algorithms perform sidedness queries. However, in the odd-dimensional case, the sidedness queries we perform are actually $(d+1)/2$-tuple comparisons.

In the discussion that follows, p_i always refers to a point in the set S_i.

If d is even, we sort all possible values of the expressions

$$\sum_{i=0}^{d/2-1} \alpha_i p_{id} \quad \text{and} \quad \sum_{i=d/2}^{d-1} \alpha_i p_{id}.$$

Then for each point $p_d \in S_d$, we scan through the two lists, looking for a pair of elements whose sum is $-\alpha_d p_{dd}$. This algorithm runs in $O(n^{d/2+1})$ time.

If d is odd, we sort all possible values of the expressions

$$\sum_{i=0}^{\lfloor d/2 \rfloor} \alpha_i p_{id} \quad \text{and} \quad \sum_{i=\lceil d/2 \rceil}^{d} -\alpha_i p_{id},$$

and then simultaneously scan through the two lists for duplicate elements. This algorithm runs in $O(n^{(d+1)/2} \log n)$ time. A simple variant of this algorithm can be used to solve a slightly more general problem, in which the points are only constrained to lie on two vertical $(d+1)/2$-flats, which necessarily intersect at a vertical line l. Instead of sorting weighted sums, we sort the possible positions at which the affine hulls of $(d+1)/2$-tuples of points from the same $(d+1)/2$-flat intersect l. This algorithm also runs in $O(n^{(d+1)/2} \log n)$ time.

In the odd-dimensional case, we can show that our original algorithm is within a logarithmic factor of being optimal when only $(d+1)/2$-tuple comparisons are allowed. Note that the odd-dimensional restricted affine degeneracy problem is a special case of the following problem: Given a set of n real numbers, decide whether any two subsets of size $(d+1)/2$ have the same sum. Dietzfelbinger [3] has derived lower bounds for this more general problem, and his techniques immediately imply the following lower bound.

Theorem 6.1. *Any decision tree algorithm that solves the restricted affine degeneracy problem in $\mathbb{R}^d$, using only $(d+1)/2$-tuple comparisons, must have depth $\Omega(n^{(d+1)/2})$, for all odd d.*

Results of Fredman [9] imply the existence of a nonuniform family of decision trees that solve this problem, using only $(d+1)/2$-tuple comparisons, with depth $O(n^{(d+1)/2})$. Thus, despite the fact that the fastest known uniform algorithm is slower by a logarithmic factor, this lower bound cannot be improved. Closing this logarithmic gap, even in the simple case $d=3$, is a long-standing and very difficult open problem.

We do not know whether our even-dimensional algorithm is optimal in this model, except in the case $d=2$, where optimality follows from Theorem 4.2. It seems unlikely that the odd-dimensional lower bound can be extended to more general versions of the affine degeneracy problem, since the model itself breaks down unless the points are in special position.

7 Implications

We have presented lower bounds for a number of degeneracy detection problems, under fairly natural models of computation. Our results are based on two fundamental techniques. First, we consider a simplified version of the problem, and we present an explicit nondegenerate input containing several independent "collapsible" configurations. We argue that any algorithm must explicitly check each collapsible configuration; otherwise, an adversary could perturb some unchecked configuration, resulting in a degenerate input which the algorithm could not distinguish from the original input. Second, by invoking a simple property of cellular decompositions, we argue that lower bounds on the simplified problem hold for the general problem as well.

A problem similar to finding degeneracies is finding the minimum measure simplex. Unfortunately, our results are *not* sufficient to improve the $\Omega(n \log n)$ lower bound on this problem. Any algorithm that finds the minimum measure simplex must at least be able to compare the values of the sidedness determinants. Such comparisons are not allowed in our model of computation. This difference may be best understood by looking at the one-dimensional case: $\Omega(n \log n)$ simple comparisons are required to sort a list of n numbers, but a stronger model is required to say anything about finding the closest pair.

Gajentaan and Overmars [10] discuss a wide range of so-called n^2-hard problems in computational geometry. They give strong reductions from the non-vertical planar affine degeneracy problem to a number of separation, covering, visibility, and motion planning problems. Theorem 2.1 implies $\Omega(n^2)$ lower bounds for a very small number of these problems. Even in the few cases where the lower bound carries over, the models of computation under which they hold are extremely weak.

For most of the problems discussed in [10], the reduction from planar affine degeneracy uses primitives that our model of computation does not allow. In these cases, one may still be able to achieve quadratic lower bounds by directly applying the techniques in this paper. For example, consider the following problem, which Gajentaan and Overmars call SEPARATOR2: Given a set of n non-intersecting line segments in the plane, is there a non-intersecting line that separates the set into two non-empty subsets? Using the techniques in this paper, one can derive a quadratic lower bound for this problem, under a model that allows sidedness queries, coordinate comparisons, and slope comparisons, using the endpoints of the segments as input.

Even so, many of the problems in [10] cannot even be solved in the models under which our techniques apply, as is the case with the minimum-measure simplex problem. Finally, many n^2-hard problems already have $O(n^2)$ solutions that use primitives outside our model.

In light of these shortcomings, an obvious question is whether our lower bound also holds in models where other decisions are allowed. Several possibilities suggest themselves. A natural model would allow the signed volumes of simplices to be compared. In the plane, Jiří Matoušek and Leo Guibas [personal communication] have independently suggested looking at "second order" queries of the following form: Given points p_1, p_2, q_1, q_2, r, is the point r closer to $\overleftrightarrow{p_1 p_2}$ or to $\overleftrightarrow{q_1 q_2}$? These second order queries and slope comparisons are both special cases of signed area comparisons. Ultimately, of course, we would like a lower bound that holds in the algebraic decision tree model.

Acknowledgements The authors gratefully acknowledge the anonymous referees for pointing out errors in an earlier version of this paper.

References

[1] M. Ben-Or. Lower bounds for algebraic computation trees. In *Proc. 15th Annu. ACM Sympos. Theory Comput.*, pages 80–86, 1983.

[2] B. Chazelle, L. J. Guibas, and D. T. Lee. The power of geometric duality. *BIT*, 25:76–90, 1985.

[3] M. Dietzfelbinger. Lower bounds for sorting of sums. *Theoret. Comput. Sci.*, 66:137–155, 1989.

[4] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science.* Springer-Verlag, Heidelberg, West Germany, 1987.

[5] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, 38:165–194, 1989. Corrigendum in 42 (1991), 249–251.

[6] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, 15:341–363, 1986.

[7] H. Edelsbrunner, R. Seidel, and M. Sharir. On the zone theorem for hyperplane arrangements. *SIAM J. Comput.*, 22(2):418–429, 1993.

[8] J. Erickson and R. Seidel. Better lower bounds on detecting affine and spherical degeneracies. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 528–536, 1993.

[9] M. L. Fredman. How good is the information theory bound in sorting? *Theoret. Comput. Sci.*, 1:355–361, 1976.

[10] A. Gajentaan and M. H. Overmars. n^2-hard problems in computational geometry. Report RUU-CS-93-15, Dept. Comput. Sci., Utrecht Univ., Utrecht, Netherlands, Apr. 1993.

[11] J. E. Goodman and R. Pollack. Multidimensional sorting. *SIAM J. Comput.*, 12:484–507, 1983.

[12] J. E. Goodman and R. Pollack. Allowable sequences and order types in discrete and computational geometry. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, pages 103–134. Springer-Verlag, 1993.

[13] J. E. Goodman, R. Pollack, and B. Sturmfels. The intrinsic spread of a configuration in $\mathbb{R}^d$. *J. Amer. Math. Soc.*, 3:639–651, 1990.

[14] J. M. Steele and A. C. Yao. Lower bounds for algebraic decision trees. *J. Algorithms*, 3:1–8, 1982.

[15] J. van Leeuwen. Problem P20. *Bull. EATCS*, 19:150, 1983.