New Lower Bounds for Hopcroft's Problem

Jeff Erickson

Technical Report **A/04/94**

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

Fachbereich 14 – Informatik
Universität des Saarlandes
D-66123 Saarbrücken, Germany
erickson@mpi-sb.mpg.de

November 1994

New Lower Bounds for Hopcroft's Problem*

Jeff Erickson

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

Fachbereich 14 – Informatik
Universität des Saarlandes
D-66123 Saarbrücken, Germany
erickson@mpi-sb.mpg.de

November 22, 1994

Abstract

We establish new lower bounds on the complexity of the following basic geometric problem, attributed to John Hopcroft: Given a set of n points and m hyperplanes in $\mathbb{R}^d$, is any point contained in any hyperplane? We define a general class of *partitioning algorithms*, and show that in the worst case, for all m and n, any such algorithm requires time $\Omega(n \log m + n^{2/3}m^{2/3} + m \log n)$ in two dimensions, or $\Omega(n \log m + n^{5/6}m^{1/2} + n^{1/2}m^{5/6} + m \log n)$ in three or more dimensions. We obtain slightly higher bounds for the counting version of Hopcroft's problem in four or more dimensions. Our planar lower bound is within a factor of $2^{O(\log^*(n+m))}$ of the best known upper bound, due to Matoušek. Previously, the best known lower bound, in any dimension, was $\Omega(n \log m + m \log n)$. We develop our lower bounds in two stages. First we define a combinatorial representation of the relative order type of a set of points and hyperplanes, called a *monochromatic cover*, and derive lower bounds on the complexity of this representation. We then show that the running time of any partitioning algorithm is bounded below by the size of some monochromatic cover.

*This research was partially supported by NSF grant CCR-9058440.

1 Introduction

In the early 1980's, John Hopcroft posed the following problem to several members of the computer science community.

> Given a set of n points and n lines in the plane, does any point lie on a line?

Hopcroft's problem arises as a special case of many other geometric problems, including collision detection, ray shooting, and range searching.

The earliest sub-quadratic algorithm for Hopcroft's problem, due to Chazelle [3], runs in time $O(n^{1.695})$. (Actually, this algorithm counts intersections among a set of n line segments in the plane, but it can easily be modified to count point-line incidences instead.) A very simple algorithm, attributed to Hopcroft and Seidel [8], runs in time $O(n^{3/2}\log^{1/2} n)$. (See [11, p. 350].) Cole *et al.* [8] combined these two algorithms, achieving a running time of $O(n^{1.412})$. Edelsbrunner *et al.* [13] developed a randomized algorithm with expected running time $O(n^{4/3+\epsilon})$.[1] Further research replaced the n^ϵ term in this upper bound with a succession of smaller and smaller polylogarithmic factors. The running time was improved by Edelsbrunner *et al.* [12] to $O(n^{4/3}\log^4 n)$ (expected), then by Agarwal [1] to $O(n^{4/3}\log^{1.78} n)$, then by Chazelle [5] to $O(n^{4/3}\log^{1/3} n)$, and most recently by Matoušek [23] to $n^{4/3}2^{O(\log^* n)}$.[2] This is currently the fastest algorithm known. Matoušek's algorithm can be tuned to detect incidences among n points and m lines in the plane in time $O(n\log m + n^{2/3}m^{2/3}2^{O(\log^*(n+m))} + m\log n)$ [10], or more generally among n points and m hyperplanes in $\mathbb{R}^d$ in time

$$O\left(n\log m + n^{d/(d+1)}m^{d/(d+1)}2^{O(\log^*(n+m))} + m\log n\right).$$

The lower bound history is much shorter. The only previously known lower bound is $\Omega(n\log m + m\log n)$, in the algebraic decision tree and algebraic computation tree models, by reduction from the problem of detecting an intersection between two sets of real numbers [26, 2].

In this paper, we establish new lower bounds on the complexity of Hopcroft's problem. We formally define a general class of *partitioning algorithms*, which includes most (if not all) of the algorithms mentioned above, and show that any such algorithm can be forced to take time $\Omega(n\log m + n^{2/3}m^{2/3} + m\log n)$ in two dimensions, or $\Omega(n\log m + n^{5/6}m^{1/2} + n^{1/2}m^{5/6} + m\log n)$ in three or more dimensions. We improve this lower bound slightly in dimensions four and higher for the *counting* version of Hopcroft's problem, where we want to know the number of incident point-hyperplane pairs.

Some related results deserve to be mentioned here. Erdős constructed a set of n points and n lines in the plane with $\Omega(n^{4/3})$ incident point-line pairs [11]. It follows immediately that any algorithm that reports all incident pairs requires time $\Omega(n^{4/3})$ in the worst case. Of course, we cannot apply this argument to either the decision version or the counting version of Hopcroft's problem, since the output size for these problems is constant. We use Erdős' construction to establish our planar lower bounds.

Chazelle [4] has established lower bounds for the query time required by a simplex range query data structure, given an upper bound on its size, in the Fredman/Yao semigroup arithmetic model [19]. For example, any data structure of size s that supports arbitrary triangular range queries among n points in the plane requires $\Omega(n/\sqrt{s})$ time per query. It follows that solving

[1] In time bounds of this form, ϵ refers to an arbitrary positive constant. For any fixed value of ϵ, the algorithm can be tuned to run within the stated time bound. However, the multiplicative constants hidden in the big-Oh notation depend on ϵ, and typically tend to infinity as ϵ approaches zero.

[2] The iterated logarithm $\log^* n$ is defined to be 1 for $n \leq 2$ and $1 + \log^*(\lg n)$ for all $n \geq 2$.

Hopcroft's problem constructing such a data structure and querying it n times, once for each line, requires time $\Omega(n^{4/3})$ in the worst case. However, this lower bound assumes that the queries are answered online. Chazelle's results say nothing about the complexity of the offline problem, where all the ranges are known in advance. As an immediate corollary to our lower bounds for Hopcroft's problem, we establish the first nontrivial lower bounds on the complexity of several offline range searching problems.

Erickson and Seidel [18] have proven a number of lower bounds on other geometric degeneracy-detection problems, under a model of computation in which only a limited number of geometric queries are allowed. We do not know whether these techniques can be applied directly to Hopcroft's problem. The main difficulty seems to be choosing an appropriate set of primitives. It is quite easy to prove a lower bound of $\Omega(n^2)$ in a decision tree model that allows only queries of the form "On which side of this line does this point lie?" With some effort, we can extend the model to include coordinate comparisons between pairs of points and pairs of lines, and sidedness queries among triples of points and triples of lines. In fact, this lower bound holds for the much easier problem of determining whether all the points lie above all the lines, for which there is a simple (optimal) algorithm that uses only $O(n \log n)$ time. It appears that higher-order queries such as "Is this point to the left or right of the intersection of these two lines?" or "Is the slope of this line larger or smaller than the slope of the line connecting these two points?" are necessary to achieve nontrivial upper bounds. If we allow either of these two primitives, however, it seems unlikely that the techniques developed in [18] can be used to derive nontrivial lower bounds.

The paper is organized as follows. In Section 2, we define a combinatorial representation of the relative order type of a set of points and hyperplanes, called a *monochromatic cover*, and derive lower bounds on the complexity of this representation. In Section 3, we formally define the class of partitioning algorithms, and prove that the running time of such an algorithm that decides Hopcroft's problem is bounded below by the complexity of some monochromatic cover. In Section 4, we discuss a number of related geometric problems for which our techniques give new lower bounds. Finally, in Section 5, we offer our conclusions and suggest directions for further research.

2 Monochromatic Covers

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of points and $H = \{h_1, h_2, \ldots, h_m\}$ a set of hyperplanes in $\mathbb{R}^d$. These two sets induce a *relative orientation matrix* $M(P, H) \in \{+, 0, -\}^{n \times m}$ whose (i, j)'th entry denotes whether the point p_i is above, on, or below the hyperplane h_j. Any minor of the matrix $M(P, H)$ is itself a relative orientation matrix $M(P', H')$, for some $P' \subseteq P$ and $H' \subseteq H$. Hopcroft's problem is to decide, given P and H, whether the matrix $M(P, H)$ contains a zero.

We call a sign matrix *monochromatic* if all its entries are equal. A *minor cover* of a matrix is a set of minors whose union is the entire matrix. If every minor in the cover is monochromatic, we call it a *monochromatic cover*. The *size* of a minor is the number of rows plus the number of columns; the size of a minor cover is the sum of the sizes of the minors in the cover.

A monochromatic cover is a succinct combinatorial representation of the relative order type of a set of points and hyperplanes. In particular, if no point lies on any hyperplane, a monochromatic cover provides a *proof* of this fact.

Monochromatic covers have been previously used to prove lower bounds for various communication complexity problems [21]. Typically, however, these results make use of the *number* of minors in the cover, not the size of the cover as we define it here.[3]

[3]Since any row or column can be split into three or fewer monochromatic minors, any matrix can be covered by

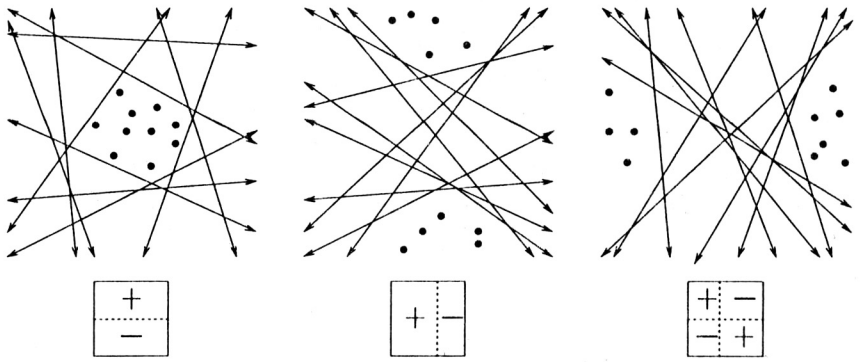

Figure 1. Three collections of points and lines with simple relative orientation matrices.

Let $\mu(P, H)$ denote the minimum size of any monochromatic cover of the relative orientation matrix $M(P, H)$. Let $\mu_d(n, m)$ denote the maximum of $\mu(P, H)$, where P ranges over all sets of n points in $\mathbb{R}^d$, and H ranges over all sets of m hyperplanes in $\mathbb{R}^d$. Let $\mu_d^*(n, m)$ denote the maximum of $\mu(P, H)$ over all sets of points and hyperplanes with no incidences. Clearly, $\mu_d(n, m) \geq \mu_d^*(n, m)$.

We are also interested in the following related quantity. Call any collection of monochromatic minors that covers all (and only) the zero entires in a sign matrix a *zero cover*. Let $\zeta(P, H)$ denote the minimum size of any zero cover of the relative orientation matrix $M(P, H)$, and let $\zeta_d(n, m)$ be the maximum of $\zeta(P, H)$ over all sets of n points and m hyperplanes in $\mathbb{R}^d$. If P and H have no incidences, $\zeta(P, H) = 0$. Since every monochromatic cover must include a zero cover, we have $\mu_d(n, m) \geq \zeta_d(n, m)$.

In the remainder of this section, we develop asymptotic lower bounds for $\mu_d^*(n, m)$ and $\zeta_d(n, m)$, which in turn imply lower bounds for $\mu_d(n, m)$.

2.1 Simple Covers

Relative orientation matrices are defined in terms of a fixed (projective) coordinate system, which determines what it means for a point to be "above" or "below" a hyperplane. More generally, we can assign an absolute orientation to each point and hyperplane, and define relative orientations with respect to these assigned orientations. (See [27].) The orientations of the points and hyperplanes determine which minors of the relative orientation matrix are monochromatic, and therefore determine the minimum monochromatic cover size.

Surprisingly, however, the minimum monochromatic cover size is independent of any choice of absolute orientations, up to a factor of two. We prove an even stronger result. Call a sign matrix *simple* if it can be changed in to a monochromatic matrix by inverting some set of rows and columns. A simple cover is a minor cover in which every minor is simple. A set of points and hyperplanes has a simple relative orientation matrix if and only if every hyperplane partitions the points into the same two subsets, or equivalently, if and only if the points all lie in the same open (projective) cell of the arrangement of hyperplanes. See Figure 1.

Let $\sigma(P, H)$ denote the minimum size of any simple cover of the relative orientation matrix $M(P, H)$. Note that changing the orientations of the points P and hyperplanes H preserves all simple minors. Thus, $\sigma(P, H)$ is strictly independent of the orientations assigned to P and H. We bound $\mu(P, H)$ in terms of $\sigma(P, H)$ as follows.

$3\min(m, n)$ such minors. Furthermore, there are sets of n points and m lines in the plane whose relative orientation matrices require $3\min(m, n)$ monochromatic minors to cover them.

Observation 2.1. $\sigma(P,H) \leq \mu(P,H) \leq 2\sigma(P,H)$

Proof: Every monochromatic minor is also a simple minor. Every simple minor can be partitioned into four monochromatic minors, whose total size is twice that of the original minor. □

2.2 One Dimension

In one dimension, points and hyperplanes are both just real numbers. We can always permute the rows and columns of the relative orientation matrix of two sets of numbers, by sorting the sets, so that the number of $+$ (resp. $-$) entries in successive rows or columns is nonincreasing (resp. nondecreasing).

Theorem 2.2. $\zeta_1(n,m) = n + m$

Proof: Obvious. □

Theorem 2.3. $\mu_1^*(n,m) = \Omega(\min(n + m\log m, n\log n + m))$

Proof: It suffices to consider the special case where $m \leq n$ and both n and m are powers of two. Let N and M be non-intersecting sets of n and m real numbers, respectively, such that n/m elements of N lie strictly between every consecutive pair of elements of M and strictly above the maximum element of M.

The relative orientation matrix of M and N can be divided into two "staircase" matrices, one positive and the other negative. We claim that any collection of minors that covers one of these staircases much have size at least $(m/2)\lg m$.

Fix a minor cover of one of the $n \times m$ staircases. Split the staircase into two smaller $n/2 \times m/2$ staircases and an $n/2 \times m/2$ rectangular submatrix. If a minor in the cover intersects the upper staircase, call it an upper minor; otherwise, call it a lower minor. The upper (resp. lower) minors induce a cover of the upper (resp. lower) staircase, of size at least $(m/4)\lg(m/2)$, by induction. It remains to bound the contribution of the rectangle to the total cover size.

Suppose that some column in the rectangle is completely contained in upper minors. Then those upper minors have (altogether) $m/2$ more rows than we accounted for in the induction step. If there is no such column, then the cover is even bigger. Every row contains an element of a lower minor, and those lower minors have (altogether) $n/2 \geq m/2$ more rows than we accounted for in the induction step.

Thus, the rectangle contributes at least $m/2$ to the total cover size. The total cover size for one triangle is at least $m/2 + (m/2)\lg(m/2) = (m/2)\lg m$, as claimed. Applying the symmetric argument for the case $m > n$ and the trivial lower bound of $n + m$ completes the proof. □

We can directly establish an upper bound that match this lower bound in many cases. In order for our bound to make sense for all values of n and m, we define $\log x = 1 + \lceil \lg x \rceil$.

Theorem 2.4. $\mu_1(n,m) = O(n\log(m^2/n) + m\log(n^2/m))$

Proof: Let N be a multiset of n real numbers, and M another multiset of m real numbers. The relative orientation matrix $M(N,M)$ can be split into a positive staircase, a negative staircase, and a collection of zero minors. The total size of the zero minors is clearly $O(n+m)$. Thus, it suffices to consider the cover size of a staircase with n rows and m columns.

Partition M and N each into two subsets, one containing the elements less than or equal to the median of N, and the other containing the elements larger than the median of N. This induces a partition of the staircase into two smaller staircases and a monochromatic minor. Cover the two smaller staircases recursively. The total size $C(n,m)$ of the cover thus generated follows the recurrence

$$C(n,m) \leq \max_{m_1+m_2=m} \Big(n/2 + m + C(n/2, m_1) + C(n/2, m_2)\Big),$$

whose solution is easily seen to be $O((n+m)\log n)$. If we partition the sets using the median of M instead, we get a cover of size $O((n+m)\log m)$ instead. Thus, we can always construct a cover of size $O(n\log m + m\log n)$, by using the median of the smaller set.

If the two sets have very different sizes, we can do much better. Consider the case $m \leq n$, where n is a multiple of m. Partition the n rows of the staircase into n/m blocks of at most m rows each. Each block can be split into a monochromatic minor and a smaller staircase. Let m_i denote the number of columns in the ith smaller staircase. Clearly, we have $\sum_{i=1}^{n/m} m_i = m$. Now construct a cover for each of the smaller staircases using the earlier algorithm. The total size $C(n,m)$ of the cover generated this way is

$$C(n,m) \leq \sum_{i=1}^{n/m} \Big(2m + O(m_i \log m + m \log m_i)\Big) = 2n + O(m\log m) + O\left(m \cdot \sum_{i=1}^{n/m} \log m_i\right).$$

The final sum is maximized when $m_i = m^2/n$ for all i. In this case, the final sum evaluates to $O(n\log(m^2/n))$, which dominates the other terms. A symmetric argument establishes the upper bound of $O(m\log(n^2/m))$ when $m < n$. □

Our upper and lower bounds for $\mu_1(m,n)$ are asymptotically tight whenever $m = O(n^{1/2})$, $m = \Theta(n)$, or $m = \Omega(n^2)$. For other values of n and m, there is a sublogarithmic gap between the two bounds.

The first half of our upper bound proof is nothing more than an application of quicksort. The connection between the size of the monochromatic cover and the running time of the algorithm is readily apparent in this case. In Section 3, we generalize this idea to higher-dimensional algorithms.

2.3 Two Dimensions

Let $I(P,H)$ denote the number of incident point-hyperplane pairs between a set of points P and a set of hyperplanes H, and let $I_d(n,m)$ denote the maximum of $I(P,H)$ over all sets P of n points and all sets H of m hyperplanes in $\mathbb{R}^d$. To derive lower bounds for $\mu_2^*(n,m)$ and $\zeta_2(n,m)$, we use the following combinatorial result of Erdős. (See [19] or [11, p.112].)

Lemma 2.5 (Erdős). $I_2(n,m) = \Omega(n + n^{2/3}m^{2/3} + m)$

Fredman [19] uses Erdős' construction to prove lower bounds for dynamic range query data structures in the plane.[4] This upper bound is asymptotically tight; the corresponding upper bound was proven by Szemerédi and Trotter [28], and independently by Clarkson *et al.* [7].

Theorem 2.6. $\zeta_2(n,m) = \Omega(n + n^{2/3}m^{2/3} + m)$

[4] Perhaps it is more interesting that Chazelle's static lower bounds [5] do *not* use this construction!

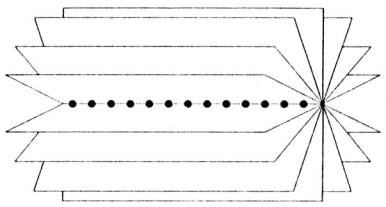

Figure 2. $I_3(n, m) = mn$.

Proof: It is not possible for two distinct points to both be adjacent to two distinct lines; any mutually incident set of points and lines has either exactly one point or exactly one line. It follows that for any set P of points and H of lines in the plane, $\zeta(P, H) \geq I(P, H)$. The theorem now follows from Lemma 2.5. □

Theorem 2.7. $\mu_2^*(n, m) = \Omega(n + n^{2/3}m^{2/3} + m)$

Proof: Consider any configuration of n points and $m/2$ lines with $\Omega(n + n^{2/3}m^{2/3} + m)$ point-line incidences, as given by Lemma 2.5. Replace each line ℓ in this configuration with a pair of lines, parallel to ℓ and at distance ε on either side, where ε is a constant, sufficiently small that all point-line distances in the new configuration are at least ε. The resulting configuration of n points and m lines clearly has no point-line incidences. We call a point-line pair in this configuration *close* if the distance between the point and the line is ε. There are $\Omega(n + n^{2/3}m^{2/3} + m)$ such pairs.

Now consider a single monochromatic minor in the relative orientation matrix of these points and lines. Let P' denote the set of points and H' the set of lines represented in this minor. We claim that the number of close pairs between P' and H' is small.

Without loss of generality, we can assume that all the points are above all the lines. If a point is close to a line, the point must be on the convex hull of P', and the line must support the upper envelope of H'. Thus, we can assume that both P' and H' are in convex position. In particular, we can order the points and lines from left to right.

Either the leftmost point is close to at most one line, or the leftmost line is close to at most one point. It follows inductively that the number of close pairs is at most $|P'| + |H'|$, which is exactly the size of the minor. The theorem follows immediately. □

2.4 Three Dimensions

The technique we used in the plane does not generalize immediately to higher dimensions. Even in three dimensions, there are collections of points and planes where every points is incident to every plane. See Figure 2. Of course, we can cover the relative orientation matrix of such a configuration with a single zero minor. If we apply the plane-doubling trick to such a set to eliminate all incidences, the relative orientation matrix of the resulting set is always simple. Thus, in order to derive a lower bound for either $\zeta_3(m, n)$ or $\mu_3^*(n, m)$, we need a construction of points and planes with many incidences, but without large sets of mutually incident points and planes.

We use the notation $[n]$ to denote the set of integers $\{1, 2, \ldots, n\}$, and $i \perp j$ to mean that i and j are relatively prime. We also use (without proof) a number of simple number-theoretic results concerning the Euler totient function $\phi(n)$, the number of positive integers less than or equal to n that are relatively prime to n. We refer the reader to [20] for relevant background.

Lemma 2.8. *For all n and m such that $\lfloor n^{1/3} \rfloor < m$, there exists a set P of n points and a set H of m planes, such that $I(P,H) = \Omega(n^{5/6}m^{1/2})$ and any three planes in H intersect in at most one point.*

Proof: Fix sufficiently large n and m such that $\lfloor n^{1/3} \rfloor < m$. Let $h(a,b,c;i,j)$ denote the plane passing through the points (a,b,c), $(a+i,b+j,c)$ and $(a+i,b,c+i-j)$. Let $p = \lfloor n^{1/3} \rfloor$ and $q = \lfloor \alpha(m/p)^{1/4} \rfloor$ for some suitable constant $\alpha > 0$. (Note that with n sufficiently large and m in the indicated range, p and q are both positive integers.)

Now consider the points $P = [p]^3 = \{(x,y,z) \mid x,y,z \in [p]\}$ and the hyperplanes

$$H = \left\{ h(a,b,c;i,j) \;\middle|\; i \in [q], j \in [i], i \perp j, a \in [i], b \in [j], c \in \left[\lfloor p/2 \rfloor\right] \right\}$$

The number of planes in H is

$$\left\lfloor \frac{p}{2} \right\rfloor \sum_{i=1}^{q} \sum_{\substack{j=1 \\ j \perp i}}^{i} j = \left\lfloor \frac{p}{2} \right\rfloor \sum_{i=1}^{q} \frac{i^2 \phi(i)}{2} = O(pq^4) = O(m).$$

By choosing the constant α appropriately and possibly adding in $o(m)$ extra planes, we can ensure that H contains exactly m planes. We claim that this collection of points and planes satisfies the lemma.

Consider a single plane $h = h(a,b,c;i,j) \in H$. Since i, j, and $i-j$ are pairwise relatively prime, h intersects exactly one point (x,y,z) such that $x \in [i]$ and $y \in [j]$, namely, the point (a,b,c). Thus, for each fixed i and j we use, the planes $h(a,b,c;i,j) \in H$ are distinct. Since planes with different "slopes" are clearly different, it follows that the planes in H are distinct.

For all $k \in [\lfloor p/2i \rfloor]$, the intersection of $h(a,b,c;i,j) \in H$ with the plane $x = a + ki$ contains at least k points of P. It follows that

$$\left| P \cap h(a,b,c;i,j) \right| \geq \sum_{k=1}^{\lfloor p/2i \rfloor} k > \frac{1}{2} \left\lfloor \frac{p}{2i} \right\rfloor^2 .$$

Thus, the total number of incidences between P and H can be calculated as follows.

$$\begin{aligned}
I(P,H) &\geq \left\lfloor \frac{p}{2} \right\rfloor \sum_{i=1}^{q} i \sum_{\substack{j=1 \\ i \perp j}}^{i} \frac{j}{2} \left\lfloor \frac{p}{2i} \right\rfloor^2 \\
&\geq \left\lfloor \frac{p}{2} \right\rfloor^3 \sum_{i=1}^{q} \sum_{\substack{j=1 \\ i \perp j}}^{i} \frac{j}{2i} \\
&= \left\lfloor \frac{p}{2} \right\rfloor^3 \sum_{i=1}^{q} \frac{\phi(i)}{4} \\
&= \Omega(p^3 q^2) \\
&= \Omega(n^{5/6} m^{1/2})
\end{aligned}$$

Finally, If H contains three planes that intersect in a line, the intersection of those planes with the plane $x = 0$ must consist of three concurrent lines. It suffices to consider only the planes passing through the point $(1,1,1)$, since for any other triple of planes in H there is a parallel triple passing through that point. The intersection of $h(1,1,1;i,j)$ with the plane $x = 0$ is the line through

$(0, 1 - j/i, 1)$ and $(0, 1, j/i)$. Since $i \perp j$, each such plane determines a unique line. Furthermore, since all these lines are tangent to a parabola, no three of them are concurrent. It follows that the intersection of any three planes in H consists of at most one point. □

Edelsbrunner *et al.* [13] prove an upper bound of $O(n \log m + n^{4/5+2\epsilon} m^{3/5-\epsilon} + m)$ on the maximum number of incidences between n points and m planes, where no three planes contain a common line. Using the probabilistic counting techniques of Clarkson *et al.* [7], we can improve this upper bound to $O(n + n^{4/5} m^{3/5} + m)$.

Theorem 2.9. $\zeta_3(n, m) = \Omega(n + n^{5/6} m^{1/2} + n^{1/2} m^{5/6} + m)$

Proof: Consider the case $n^{1/3} < m \leq n$. Fix a set P of n points and a set H of m hyperplanes satisfying Lemma 2.8. Any mutually incident subsets of P and H contain either at most one point or at most two planes. Thus, the number of entries in any zero minor of $M(P, H)$ is at most twice the size of the minor. It follows that any zero cover of $M(P, H)$ must have size $\Omega(I(P, H)) = \Omega(n^{5/6} m^{1/2})$. The dual construction gives us a lower bound of $\Omega(n^{1/2} m^{5/6})$ for all m in the range $n \leq m < n^3$, and the trivial lower bound $\Omega(n + m)$ applies for other values of m. □

Lemma 2.10. *Let P be a set of n points and H a set of m planes in $\mathbb{R}^3$, such that every point in P is either on or above every plane in H, and any three planes in H intersect in at most one point. Then $I(P, H) \leq 3(m + n)$.*

Proof: Call any point (resp. plane) is *lonely* if it is incident to less than three planes (resp. points). Without loss of generality, we can assume that none of the points in P or planes in H is lonely, since each lonely point and plane contributes at most three incidences.

No point in the interior of of the convex hull of P can be incident to a plane in H. Any point in the interior of a facet of the convex hull can be on at most one plane in H. Consider any point $p \in P$ in the interior of an edge of the convex hull. Any plane containing p also contains the two endpoints of the edge. There cannot be more than two such planes in H, so p must be lonely. It follows that every point in P is a vertex of the convex hull of P.

No plane can contain a point unless it touches the upper envelope of H. Any plane that only contains a vertex of the upper envelope must be lonely. For any plane h that contains only an edge of the envelope, two other planes also contain that edge, and any points on h must also be on the other two planes. Then h must be lonely, since any three planes in H intersect in at most one point. It follows that every plane in H spans a facet of the upper envelope of H. Furthermore, every point in P is a vertex of this upper envelope.

Construct a bipartite graph with vertices P and H and edges corresponding to incident pairs. This graph is clearly planar, and thus has at most $3(m + n)$ edges. □

Note that this lemma still holds if we weaken the general position requirement to rule out only mutually incident sets of s points and t planes, where s and t are any fixed constants. We reiterate here that without some sort of general position assumption, we can easily achieve mn incidences.

Theorem 2.11. $\mu_3^*(n, m) = \Omega(n + n^{5/6} m^{1/2} + n^{1/2} m^{5/6} + m)$

Proof: Consider the case $2n^{1/3} < m \leq n$. Fix a set P of n points and a set H of $m/2$ hyperplanes satisfying Lemma 2.8.

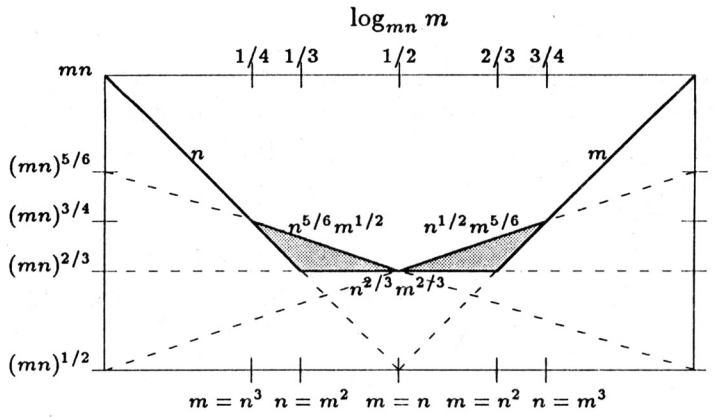

Figure 3. Comparison of lower bounds for $\mu_2^*(n, m)$ and $\mu_3^*(n, m)$

Call a sign matrix *loosely monochromatic* if either none of its entries is $+$ or none of its entries is $-$. For all subsets $P' \subseteq P$ and $H' \subseteq H$, Lemma 2.10 implies that if $M(P', H')$ is loosely monochromatic, then $I(P', H') = O(|P'| + |H'|)$.

Now replace each plane $h \in H$ with a pair of parallel planes at distance ε on either side of h, for some suitably small constant $\varepsilon > 0$. Call the resulting set of m hyperplanes H_ε. We say that a point is *close* to a plane if the distance between them is exactly ε. There are $\Omega(n^{5/6}m^{1/2})$ close pairs between P and H_ε, and no incidences.

For every monochromatic minor of the matrix $M(P, H_\varepsilon)$, there is a corresponding loosely monochromatic minor of $M(P, H)$. Furthermore, there is a one-to-one correspondence between the close pairs in the first minor and the incident pairs in the second. It follows that any monochromatic minor of $M(P, H_\varepsilon)$ orients only a linear number of close pairs. Thus, any monochromatic cover for P and H_ε must have size $\Omega(n^{5/6}m^{1/2})$. Similar arguments apply to other values of m. □

Unfortunately, for the special case $m = \Theta(n)$, this does not improve the previous bound of $\Omega(n^{4/3})$ which we derived earlier for the planar case. For all other values of m between $\Omega(n^{1/3})$ and $O(n^3)$, however, the new bound is an improvement. See Figure 3.

2.5 Higher Dimensions

In order to generalize Lemma 2.8 to arbitrary dimensions, we need the following rather technical lemma. Let us define two series $f_i(t)$ and $F_i(t)$ of polynomials as $f_1(t) = 1$, $f_i(t) = t + i - 1$ for all $i > 1$, and $F_i(t) = \prod_{j=1}^{i-1} f_j(t)$ for all i.

Lemma 2.12. *Let $t_1, t_2, \ldots, t_d$ be distinct real numbers such that $f_j(t_i) \neq 0$ for all $1 \leq i, j \leq d$. The $d \times d$ matrix M, whose (i, j)th entry is $1/f_j(t_i)$, is nonsingular.*

Proof: Let V be the $d \times d$ Vandermonde matrix whose (i, j)th entry is t_i^{j-1}. Since the t_i are distinct, V is nonsingular. We prove the lemma by converting F into V using elementary row and column operations. We transform the matrix inductively, one column at a time. Transforming the first column is trivial.

The inductive step is somewhat easier to understand if we focus on a single row of M, and think of it as a vector of rational functions in some formal variable t, instead of a vector of real numbers. Suppose we have already transformed the first $d - 1$ entries inductively, and we are now ready to

transform the last entry. The first step is to multiply the entire vector by $f_d(t)$; this ensures that every entry in the vector is a polynomial. By induction, the dth entry is now $F_d(t)$, and and for all $j < d$, the jth entry is now $f_d(t) \cdot t^{j-1} = t^j + (d-1)t^{j-1}$. It remains to show that we can transform this vector of polynomials into the vector $(1, t, t^2, \ldots, t^{d-1})$.

Write the coefficients of the polynomials into a $d \times d$ matrix C, whose (i,j)-th entry $c_{i,j}$ is the coefficient of t^{i-1} in the jth polynomial. The only nonzero entries in C are the coefficients of $F_d(t)$ in the last column, $(d-1)$'s along the main diagonal, and ones in the next lower diagonal. For example, when $d = 4$, our vector of polynomials is $(t+3, t^2+3t, t^3+3t^2, t^3+3t^2+2t)$, so we have

$$C = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 1 & 3 & 0 & 2 \\ 0 & 1 & 3 & 3 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Recall that the determinant of C is defined as follows.

$$\det C \triangleq \sum_{\pi \in S_d} \left(\operatorname{sgn}(\pi) \prod_{i=1}^{d} c_{i,\pi(i)} \right)$$

The only permutations that contribute to the determinant are those that start down the main diagonal, jump to the last column, and then finish along the lower diagonal. It follows that $\det C = (-1)^{d-1} F_d(1-d)$. Since $1-d$ is not a root of $F_d(t)$, we conclude that C is nonsingular.

Thus, there is a series of column operations that convert C into the identity matrix. Since each column of C contains the coefficients of a polynomial in the corresponding column of M, the *same* column operations complete the transformation of M into V. □

Lemma 2.13. *For any $\lfloor n^{1/d} \rfloor < m$, there exists a set P of n points and a set H of m hyperplanes in $\mathbb{R}^d$, such that $I(P,H) = \Omega(n^{1-2/d(d+1)} m^{2/(d+1)} + m)$ and any d hyperplanes in H intersect in at most one point.*

Proof (sketch): We sketch the proof for $d = 4$; the generalization to higher dimensions is relatively straightforward. Let $h(a,b,c,d;i,j)$ denote the hyperplane passing through the four points

$$(a,b,c,d) \quad (a+i, b+j, c, d) \quad (a+i, b, c+i+j, d) \quad (a+i, b, c, d+2i+j).$$

Let $p = \lfloor n^{1/4} \rfloor$ and $q = \lfloor \alpha (m/p)^{1/5} \rfloor$ for some suitable constant $\alpha > 0$. Then $P = [p]^4$ and H is the set of hyperplanes $h(a,b,c,d;i,j)$ satisfying the following eight conditions.

$$i \in [q],\ j \in [i],\ j \text{ is odd},\ i \perp j$$
$$a \in [i],\ b \in [j],\ c \in [i+j],\ d \in \left[\lfloor p/2 \rfloor \right]$$

Note that j is odd and relatively prime with i if and only if i, j, $i+j$, and $2i+j$ are pairwise relatively prime. This condition is necessary to establish that the hyperplanes in H are distinct. It follows from relatively straightforward algebraic manipulation that $|H| = O(pq^5) = O(m)$ and $I(P,H) = \Omega(p^4 q^2) = \Omega(n^{9/10} m^{2/5})$.

To establish that no four hyperplanes in H intersect in a common line, we examine the intersection of each hyperplane $h(1,1,1,1;i,j) \in H$ with the hyperplane $x_1 = 0$. This intersection is the plane

$$\frac{1}{i} + \frac{x_2 - 1}{j} + \frac{x_3 - 1}{i+j} + \frac{x_4 - 1}{2i+j} = 0.$$

It follows from Lemma 2.12, by setting $t_k = j_k / i_k$, that no four of these planes are concurrent. □

Note that the lower bound for dimension d only improves the bound for dimension $d-1$ when $n = \Omega(m^{(d-1)/2})$. Again, using probabilistic counting techniques [7], we can prove an upper bound of $I(P,H) = O(n + n^{(2d-2)/(2d-1)}m^{d/(2d-1)} + m)$ if any d hyperplanes in H intersect in at most one point.

Lemma 2.14. *Let P be a set of points and H a set of hyperplanes such that for some constants s and t, no s hyperplanes of H contain t points of P in their intersection. Then $\zeta(P,H) = \Omega(I(P,H))$.*

Proof: Consider any zero minor of $M(P,H)$. If the minor has k entries, then its size is at least $k/(s+t)$. Thus, the total size of the zero minors is at least $I(P,H)/(s+t) = \Omega(I(P,H))$. □

The two previous lemmas immediately gives us the following theorem.

Theorem 2.15. $\zeta_d(n,m) = \Omega\left(\sum_{i=1}^{d}\left(n^{1-2/i(i+1)}m^{2/(i+1)} + n^{2/(i+1)}m^{1-2/i(i+1)}\right)\right)$

Unfortunately, we are unable to generalize Lemma 2.10 even into four dimensions. The best upper bound we can prove on the number of incidences between n points and m hyperplanes in $\mathbb{R}^4$, where every point is above or on every hyperplane, and no four hyperplanes contain a line, is $O(n + n^{2/3}m^{2/3} + m)$. (See [15] for the derivation of a similar upper bound.) No superlinear lower bounds are known in any dimension, so there is some hope for a linear upper bound.

However, we can achieve a super-linear number of incidences in five dimensions, under a weaker combinatorial general position requirement. Unlike in lower dimensions, some sort of *geometric* general position requirement is necessary to keep the number of incidences small. (We do not know of any such requirement that is *sufficient*, except for the trivial requirement that at most $d+1$ hyperplanes contain any point.)

Lemma 2.16. *For all n and m, there exists a set P of n points and a set H of m hyperplanes in $\mathbb{R}^5$, such that every point is on or above every hyperplane, no two hyperplanes in H contain more than one point of P in their intersection, and $I(P,H) = \Omega(n + n^{2/3}m^{2/3} + m)$.*

Proof: Define the function $\sigma : \mathbb{R}^3 \to \mathbb{R}^6$ as follows.

$$\sigma(x,y,z) = (x^2, y^2, z^2, \sqrt{2}\,xy, \sqrt{2}\,yz, \sqrt{2}\,xz)$$

For any $v, w \in \mathbb{R}^3$, we have $\langle \sigma(v), \sigma(w)\rangle = \langle v, w\rangle^2$, where $\langle v, w\rangle$ denotes the usual inner product of v and w. In a more geometric setting, σ maps points and lines in the plane, represented in homogeneous coordinates, to points and hyperplanes in $\mathbb{R}^5$, also represented in homogeneous coordinates [27]. For any point p and line ℓ in the plane, the point $\sigma(p)$ is incident to the hyperplane $\sigma(\ell)$ if and only if p is incident to ℓ; otherwise, $\sigma(p)$ lies above $\sigma(\ell)$. Thus, we can take P and H to be the images under σ of any sets of n points and m lines with $\Omega(n + n^{2/3}m^{2/3} + m)$ incidences, as given by Lemma 2.5. □

Thus, the best lower bound we can derive for $\mu_d^*(n,m)$ for any $d > 3$ derives trivially from Theorem 2.11.

Corollary 2.17. $\mu_d^*(n,m) = \Omega(n + n^{5/6}m^{1/2} + n^{1/2}m^{5/6} + m)$ *for all* $d \geq 3$.

3 Partitioning Algorithms

A *partition graph* is a directed acyclic graph, with one source, called the *root*, and several sinks, or *leaves*. The maximum out-degree of the graph is bounded by some constant Δ. Every non-leaf is either a *primal node* or a *dual node*. Associated with each primal or dual node v is a set $\mathcal{R}_v$ of connected *query regions*, one for each outgoing edge, whose union is $\mathbb{R}^d$. The cardinality of $\mathcal{R}_v$ is at most Δ. We do not require the query regions to be disjoint, convex, simply connected, or even semi-algebraic, nor do we require that each query region have constant descriptional complexity.

Every point $p \in \mathbb{R}^d$ induces a subgraph of a partition graph as follows. We say that a point *reaches* every node in its subgraph and *traverses* every edge in its subgraph. The point p reaches a node v if either v is the root, or p traverses some edge into v. If p reaches a primal node v, then it also traverses every edge corresponding to a query region $R \in \mathcal{R}_v$ that contains p. If p reaches a dual node v, then it also traverses every edge corresponding to a query region that intersects the dual hyperplane p^*. The subgraph induced by p contains all the nodes that p reaches and all the edges that p traverses. Similarly, every hyperplane h induces a subgraph, according to the primal query regions that intersect h and the dual query regions that contain the dual point h^*.

Given a set of points and hyperplanes, a *partitioning algorithm* constructs a partition graph and determines the subgraphs induced by each point and hyperplane. For the purpose of proving lower bounds, we charge unit time whenever a point or hyperplane traverses an edge. In particular, we do not charge for the construction of the partition graph itself, nor for constructing its query regions. We emphasize that partitioning algorithms are *nondeterministic*, since the partition graph and its query regions depend on the input.

A partitioning algorithm decides Hopcroft's problem by reporting an incidence if and only if some leaf in its partition graph is reached by both a point and a hyperplane. It is easy to see that if a point and hyperplane are incident, then there is at least one leaf in every partition graph that is reached by both the point and the hyperplane. Thus, given a set P of points and a set H of hyperplanes, a partition graph in which no leaf is reached by both a point and a hyperplane provides a *proof* that there are no incidences between P and H.

In the remainder of this section, we derive lower bounds for the worst-case running time of partitioning algorithms that solve Hopcroft's problem. With the exception of the basic lower bound of $\Omega(n \log m + m \log n)$, which we prove directly, all of our lower bounds are derived from the minimum monochromatic cover size bounds in Section 2.

3.1 The Basic Lower Bound

Theorem 3.1. *Any partitioning algorithm that decides Hopcroft's problem in any dimension must take time $\Omega(n \log m + m \log n)$ in the worst case.*

Proof: It suffices to consider the following configuration, where n is a multiple of m. P consists of n points on some vertical line in $\mathbb{R}^d$, say the x_d-axis, and H consists of m hyperplanes normal to that line, placed so that n/m points lie between each hyperplane and the next higher hyperplane, or above the top hyperplane. (This is essentially the same configuration used in the proof of Theorem 2.3, lifted to arbitrary dimensions.) For each point, call the hyperplane below it its *partner*. Each hyperplane is a partner of n/m points.

Let G be the partition graph generated by some partitioning algorithm. Let Δ be the maximum out-degree of any node in G. The *level* of any node in G is the length of the shortest path from the root to that node. There are at most Δ^k nodes at level k. We say that a node v *separates* a point-hyperplane pair if both the point and the hyperplane reach v, none of the outgoing edges of

v is traversed by both the point and the hyperplane. Finally, we say that a hyperplane h is *active at level k* if none of the nodes in the first k levels separates h from any of its partners.

Suppose v is a primal node. For each hyperplane h that v separates from one of its partner points p, mark some query region in $\mathcal{R}_v$ that contains p, but misses h. The marked region lies completely above h, but not completely above any hyperplane higher than h. It follows that the same region cannot be marked more than once. Since there are at most Δ regions, at most Δ hyperplanes become inactive. By similar arguments, if v is a dual node, then v separates at most Δ points from their partners.

Thus, the number of hyperplanes that are inactive at level k is less than Δ^{k+2}. In particular, at level $\lfloor \log_\Delta m \rfloor - 3$, at least $m(1-1/\Delta)$ hyperplanes are still active. It follows that at least $n(1-1/\Delta)$ points each traverse at least $\lfloor \log_\Delta m \rfloor - 3$ edges, so the total running time of the algorithm is at least

$$n(1 - 1/\Delta)(\lfloor \log_\Delta m \rfloor - 3) = \Omega(n \log m).$$

Similar arguments establish a lower bound of $\Omega(m \log n)$ when $n < m$. □

3.2 The Lower Bound for the Decision Problem

Let $T_{\mathcal{A}}(P, H)$ denote the running time of an algorithm $\mathcal{A}$ that decides Hopcroft's problem in $\mathbb{R}^d$ for some d, given points P and hyperplanes H as input.

Theorem 3.2. *Let $\mathcal{A}$ be a partitioning algorithm that decides Hopcroft's problem, and let P be a set of points and H a set of hyperplanes such that $I(P, H) = 0$. Then $T_{\mathcal{A}}(P, H) = \Omega(\mu(P, H))$.*

Proof: Recall that the running time $T_{\mathcal{A}}(P, H)$ can be defined in terms of the edges of the partition graph as follows.

$$T_{\mathcal{A}}(P, H) \triangleq \sum_{p \in P} \#\text{edges } p \text{ traverses} + \sum_{h \in H} \#\text{edges } h \text{ traverses}$$

We say that a point or hyperplane *misses* an edge from v to w if it reaches v but does not traverse the edge. (It might still reach w by traversing some other edge.) As before, let Δ denote the maximum out-degree of any node in the partition graph. For every edge that a point or hyperplane traverses, there are at most $\Delta - 1$ edges that it misses.

$$\begin{aligned} \Delta \cdot T_{\mathcal{A}}(P, H) \;\geq\; & \sum_{p \in P} (\#\text{edges } p \text{ traverses} + \#\text{edges } p \text{ misses}) \;+ \\ & \sum_{h \in H} (\#\text{edges } h \text{ traverses} + \#\text{edges } h \text{ misses}) \end{aligned}$$

Call any edge that leaves a primal node a primal edge, and any edge that leaves a dual node a dual edge.

$$\begin{aligned} \Delta \cdot T_{\mathcal{A}}(P, H) \;\geq\; & \sum_{p \in P} (\#\text{primal edges } p \text{ traverses} + \#\text{dual edges } p \text{ misses}) \;+ \\ & \sum_{h \in H} (\#\text{dual edges } h \text{ traverses} + \#\text{primal edges } h \text{ misses}) \\ = \; & \sum_{\text{primal edges } e} (\#\text{points tranversing } e + \#\text{hyperplanes missing } e) \;+ \\ & \sum_{\text{dual edges } e} (\#\text{hyerplanes traversing } e + \#\text{points missing } e) \end{aligned}$$

Consider, for some primal edge e, the set P_e of points that traverse e and the set H_e of hyperplanes that miss e. The edge e is associated with some query region R, such that every point in P_e is contained in R, and every hyperplane in H_e is disjoint from R. It follows immediately that the relative orientation matrix $M(P_e, H_e)$ is simple. Similarly, for any dual edge e, the relative orientation matrix of the set of points that miss e and hyperplanes that traverse e is also simple.

Now consider any point $p \in P$ and hyperplane $h \in H$. Since $\mathcal{A}$ correctly decides Hopcroft's problem, no leaf is reached by both p and h. It follows that some node v separates p and h. If v is a primal node, then h misses the outgoing primal edges that p traverses. If v is a dual node, then p misses the outgoing dual edges that h traverses.

Thus, for each edge in the partition graph, we can associate a simple minor, and this collection of minors covers the relative orientation matrix $M(P, H)$. Furthermore, the size of this simple cover is exactly the lower bound we have for $\Delta \cdot T_{\mathcal{A}}(P, H)$ above. Splitting each simple minor into monochromatic minors at most doubles the size of the cover. Thus, the algorithm induces a monochromatic cover of size at most $2\Delta \cdot T_{\mathcal{A}}(P, H)$. Since this must be at least $\mu(P, H)$, we have the lower bound $T_{\mathcal{A}}(P, H) \geq \mu(P, H)/2\Delta$. □

Corollary 3.3. *The worst-case running time of any partitioning algorithm that solves Hopcroft's problem in* $\mathbb{R}^d$ *is* $\Omega(n \log m + n^{2/3}m^{2/3} + m \log n)$ *for* $d = 2$ *and* $\Omega(n \log m + n^{5/6}m^{1/2} + n^{1/2}m^{5/6} + m \log n)$ *for all* $d \geq 3$.

Proof: Theorems 3.1 and 3.2 together imply that the worst case running time is $\Omega(n \log m + \mu_d^*(n, m) + n \log m)$. Thus, theorem 2.7 gives the planar lower bound, and Corollary 2.17 gives us the lower bound in higher dimensions. □

We emphasize here that the condition that $I(P, H) = 0$ is necessary for the lower bound to hold. If there is an incidence, then the trivial algorithm correctly "detects" it. The partition graph contains one leaf, and since it is reached by every point and hyperplane, the algorithm reports an incidence. If we insist on considering only algorithms that can correctly report an incidence whenever one exists, say by identifying a leaf that is reached by exactly one incident pair, then there is an algorithm that runs in time $O(n + m)$ whenever there is an incidence in the input. The partition graph generated by this algorithm has one primal node, whose query regions are $\{p\}$ and $\mathbb{R}^d \setminus \{p\}$, where p is a point that is incident to some hyperplane.

3.3 The Lower Bound for the Counting Problem

A partitioning algorithm solves the counting version of Hopcroft's problem as follows. The number of incidences associated with a leaf in its partition graph is the number of points that reach it times the number of hyperplanes that reach it. The algorithm returns as its output the sum of these products over all leaves in its partition graph.

Since every incident point-hyperplane pair is guaranteed to reach at least one leaf, it is not possible for a partitioning algorithm to count too few incidences. The only ways the algorithm can go wrong is counting the same incidence more than once, or counting incidences that don't exist. Thus, in order to be correct, the algorithm must ensure that every non-incident point-hyperplane pair is separated.

Theorem 3.4. *Let* $\mathcal{A}$ *be a partitioning algorithm that solves the counting version of Hopcroft's problem, and let* P *be a set of points and* H *a set of hyperplanes. Then* $T_{\mathcal{A}}(P, H) = \Omega(\mu(P, H))$.

Proof: We follow the proof for the decision lower bound almost exactly. We associate a simple minor with every edge just as before. We also associate a monochromatic minor with every leaf, consisting of all points and hyperplanes that reach the leaf. Every non-incident point-hyperplane pair is represented in some edge minor, and every incident pair in exactly one leaf minor. Thus, the minors form a simple cover. The total size of the leaf minors is certainly less than $T_{\mathcal{A}}(P, H)$, since every point and hyperplane that reaches a leaf must traverse one of the leaf's incoming edges. The total size of the edge minors is at most $\Delta \cdot T_{\mathcal{A}}(P, H)$, as established previously. Splitting each edge minor into monochromatic minors at most doubles their size. Thus, we get a monochromatic cover of size at most $(2\Delta + 1)T_{\mathcal{A}}(P, H)$, which implies $T_{\mathcal{A}}(P, H) \geq \mu(P, H)/(2\Delta + 1)$. □

We can prove the following much stronger bound by only paying attention to the minors induced at the leaves. We define an *unbounded partition graph* to be just like a partition graph except that we place no restrictions on the number of query regions associated with each node. Call the resulting class of algorithms *unbounded partitioning algorithms.*

Theorem 3.5. *Let $\mathcal{A}$ be an unbounded partitioning algorithm that solves the counting version of Hopcroft's problem, and let P be a set of points and H a set of hyperplanes. Then $T_{\mathcal{A}}(P, H) = \Omega(\zeta(P, H))$.*

Proof: We associate a zero minor with every leaf, and these minors form a zero cover. The total size of the leaf minors is certainly less than $T_{\mathcal{A}}(P, H)$, since every point and hyperplane that reaches a leaf must traverse one of the leaf's incoming edges. □

The following corollary is now immediate.

Corollary 3.6. *The worst-case running time of any unbounded partitioning algorithm that solves the counting version of Hopcroft's problem in $\mathbb{R}^d$ is*

$$\Omega\left(n \log m + \sum_{i=2}^{d} \left(n^{1-2/i(i+1)} m^{2/(i+1)} + n^{2/(i+1)} m^{1-2/i(i+1)}\right) + m \log n\right).$$

3.4 Containment Shortcuts Don't Help

We might consider adding the following containment shortcut to our model. Suppose a hyperplane h reaches a primal node v, and some query region $R \in \mathcal{R}_v$ is completely contained in h. Then we know immediately that any point in R is incident to h. This means that we could report an incidence (or increment a running counter) whenever a point traverses the edge corresponding to R, and the hyperplane does not need to traverse the edge. We could apply a similar shortcut for points at each dual nodes. In addition to charging for traversing each edge, we now charge unit time each time a hyperplane (either primal or dual) contains a query region.

Clearly, adding this shortcut can only decrease the running time of the algorithm. However, for any algorithm that uses this shortcut, we can derive an equivalent algorithm without shortcuts that is slower by only a small constant factor.

Suppose some query region R is contained in a hyperplane h. Then h must also contain $\text{aff}(R)$, the affine hull of R. We can reverse the containment relation by applying a duality transformation: the point h^* dual to h is contained in the flat $(\text{aff}(R))^*$ dual to $\text{aff}(R)$. Similarly, if a point p is contained in R, then $(\text{aff}(R))^* \in p^*$.

For each node v in the partition graph of the shortcut algorithm, and each query region $R \in \mathcal{R}_v$, we modify the graph as follows. Let e be the edge of the partition graph corresponding to R, and

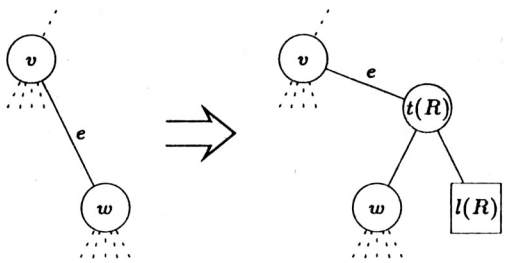

Figure 4. Getting rid of containment shortcuts.

let w be the destination of this edge. We introduce two new nodes, a "test" node $t(R)$ and a leaf $\ell(R)$. If v is a primal node, then $t(R)$ is a dual node, and vice versa. The query subdivision $\mathcal{R}_{t(R)}$ consists of exactly two regions: $(\text{aff}(R))^*$ and $\mathbb{R}^d \setminus (\text{aff}(R))^*$, whose corresponding edges point to $\ell(R)$ and w, respectively. Finally, we redirect e so that it points to $t(R)$. See Figure 4. The new algorithm uses this modified partition graph, without explicitly checking for containments.

The new node $t(R)$ strictly separates the hyperplanes that contain R from the hyperplanes that merely intersect it. Any point contained in R will be redirected to both w and $\ell(R)$. Thus, the new algorithm has the same behavior as the shortcut algorithm; that is, it reports (or counts) exactly the same incidences. We easily verify that the running time of the new algorithm is at most three times the running time of the shortcut algorithm.

3.5 Existing Algorithms

Existing algorithms for Hopcroft's problem all follow roughly the same pattern. The algorithm constructs a number of regions, determines which points and hyperplanes intersect each region, and recursively solve the resuting subproblems. The number of regions used by these algorithms at each level of recursion is typically a small polynomial function of the input size, and not a constant, as required by the definition of the partitioning model. However, we can still model most, if not all, of these algorithms by parittioning algorithms.

In order to determine which points lie in which regions, every existing algorithm constructs a (possibly trivial) point location data structure. Each node in this data structure splits the plane into a constant number of simple regions, and for each region, there is a pointer to another node in the data structure. Each leaf in the data structure corresponds to one of the high-level query regions.

What about locating the hyperplanes? Many algorithms for Hopcroft's problem use the point location data structure to locate hyperplanes as well. Algorithms of this type fit into our model perfectly. Composing all the point location data structures used by the algorithm in all recursive subproblems gives us the algorithm's partition graph. In particular, Matoušek's algorithm [23], which is based on Chazelle's heirarchical cuttings [5] and is the fastest algorithm known, can be modeled this way. Thus, Theorem 3.4 immediately gives us the following theorem.

Theorem 3.7. $\mu_d(n,m) = O\left(m \log n + n^{d/(d+1)} m^{d/(d+1)} 2^{O(\log^*(n+m))} + n \log m\right)$

However, other algorithms do not use the point location data structure to locate hyperplanes, at least not at all levels of recursion. In these algorithms, the query regions are cells in a canonical decomposition of an arrangement of a subset of the hyperplanes. The algorithms determine which

cells a given hyperplane hits by iteratively "inserting" the hyperplane into the arrangement. The Zone Theorem for hyperplane arrangements [14] guarantees that the running time of the "insertion" procedure is proportional to the number of cells the hyperplane hits.

In many cases, modifying the algorithm to directly use the point location data structure instead of the Zone Theorem increases the running time by only a constant factor. If the current data structure locates the hyperplanes too slowly, we may be able to replace it with a different data structure that supports fast hyperplane location, again without increasing the asymptotic running time. We could use, for example, the randomized incremental construction of Seidel [24] in the plane, or the hierarchical cuttings data structure of Chazelle [5] in higher dimensions.

Some algorithms construct a point location data structure for the arrangement of the *entire* set of hyperplanes. Usually, this is done when the number of hyperplanes is much smaller than the number of points. In this case, the algorithm doesn't need to locate the hyperplanes at all! Again, however, we can modify the algorithm so that it uses a point location data strcuture that allows efficient hyperplane location as well, and artificially locates the hyperplanes. If we use an appropriate data structure, the running time will only increase by a constant factor.

These arguments are admittedly rather ad hoc. Extending the partitioning model to *naturally* include algorithms that use the Zone Theorem is an interesting open problem. One possible solution is to allow arbitrarily many query regions at each node, at some additional cost in the running time. Specifically, in addition to the cost for traversing edges, we would charge $\log r$ time for each point and hyperplane that reaches a node with r query regions. Such an extension would allow us to model a large partitioning data structure in a single node of the partition graph, without needing to understand the internal details of the data structure. Unfortunately, it is not clear how to apply our monochromatic cover results in this extended model.

Finally, a few algorithms partition the points *arbitrarily* into subsets, without using geometric information of any kind. In this case, every hyperplane becomes part of every subproblem. We can account for this kind of arbitrary partitioning in one of two ways. The first approach is to simulate the arbitrary paritioning using query regions. We can select any subset of the points by using a spanning tree of the subset as a query region, possibly perturbed so that no other points are included, and possibly with an additional line segment to ensure that the tree intersects every hyperplane. This approach has one unfortunate flaw. Splitting the n points into r subsets using a partition graph requires $\Omega(mr + n \log r)$ time, since each node can only have only a constant number of query regions. However, the actual algorithm need not take any time at all, if the subsets are defined by indices in an array, for example.

The second approach is to strengthen the model. Let us define a *strong partition graph* to be a partition graph with a new type of node that partitions the points or hyperplanes (but not both) into subsets at no extra cost. Call the resulting class of algorithms *strong paritioning algorithms*. Lemmas 3.2 and 3.4 still hold in this stronger model, since the new nodes cannot separate any point from any hyperplane. However, Lemma 3.1 does not hold. In particular, there is a strong partitionging algorithm that decides Hopcroft's problem in $\mathbb{R}^d$ in time $O(n+m^{d+1})$, by "arbitrarily" partitioning the points into subsets, each contained in a single cell of the hyperplane arrangement. We saw a similar effect in the second half of the proof of Theorem 2.4. While we cannot expect any real algorithm to acheive this running time, the nondeterminism of our model allows us to make seemingly unreasonable choices.

4 Related Problems

In this section, we list a number of related problems for which we have new lower bounds in the partitioning model, either by reduction to Hopcroft's problem, or from direct application of our earlier proof techniques. No lower bound bigger than $\Omega(n \log m + m \log n)$ was previously known for any of these problems.

Extreme caution must be taken when applying reduction arguments to partitioning algorithms. It is quite easy to apply a "standard" reduction argument, only to find that the reduction also changes the model.

4.1 Line Segment Intersection

Theorem 4.1. *Any partitioning algorithm that detects bichromatic line segment intersections in the plane requires time $\Omega(n \log m + n^{2/3}m^{2/3} + m \log n)$ in the worst case.*

Theorem 4.2. *Any partitioning algorithm that counts line segment intersections in the plane requires time $\Omega(n^{4/3})$ in the worst case.*

The dual of a line segment is a double wedge, consisting of a contiguous set of lines passing through a single *apex*. Two line segments intersect if and only if each of their double wedges contains the other's apex. If a line segment is just a single point, the dual double wedge is the dual line of the point. Conversely, if the segment is infinitely long, then its dual double wedge covers the entire plane, and its apex is the dual point of the line.

Now consider what happens when we apply a standard reduction argument to reduce Hopcroft's problem to the line segment intersection problem. We consider each point to be a line segment of zero length, and each line to be a line segment of infinite length. The primal nodes partition the points and lines just as we expect. The dual nodes are practically worthless, however, since the dual wedge of every line intersects every dual query region. In applying the reduction argument, we have lost the inherent self-duality of the problem. Since the model is *weaker* than we expect, the lower bounds still hold.

A simple sweep-line algorithm determines the presence of line segment intersections in time $O(n \log n)$. The fastest algorithm for counting line segment intersections, due to de Berg and Schwarzkopf [10], runs in randomized expected time $O(n \log n + n^{2/3}A^{2/3} \log^{1/3} n)$, where A is the number of intersections. Their algorithm can be modified to detect bichromatic intersections in the same amount of time, where A now denotes the number of *monochromatic* intersections.

4.2 Lines in 3-Space

An easy variant of Theorem 3.5 gives us the following lower bound.

Theorem 4.3. *Any (unbounded) partitioning algorithm that counts pairs of intersecting lines in $\mathbb{R}^3$ requires time $\Omega(n^{4/3})$ in the worst case.*

Detecting line intersections in $\mathbb{R}^3$ is at least as hard detecting point-line incidences in the plane, at least in the algebraic decision tree model. However, the following straightforward partitioning algorithm detects line intersections in only $O(n \log n)$ time. If there is an intersection, the trivial algorithm "detects" it. Otherwise, our algorithm arbitrarily splits the lines into two classes of $n/2$ lines each, say "red" lines and "blue" lines. For each class, the algorithm constructs a query region consisting of the union of the lines with enough line segments to connect them. If the connecting

segments are chosen correctly, the red lines do not intersect the blue query region, and the blue lines do not intersect the red query region. Thus, separating the red and blue lines requires only linear time. To separate every pair of lines, the algorithm then proceeds recursively in each class.

Typically, algorithms for problems involving lines in $\mathbb{R}^3$ map the lines to points and hyperplanes in $\mathbb{R}^5$ using Plücker coordinates [27]. Thus, the line intersection problem is just a special case of Hopcroft's problem in $\mathbb{R}^5$. Any partitioning algorithm that uses this approach can be forced to take time $\Omega(n^{4/3})$, even to solve the decision version of this problem. More generally, we could consider algorithms that use a mixture of these approaches, sometimes using Plücker coordinates, and sometimes staying in $\mathbb{R}^3$. Theorem 4.3 also applies to such algorithms.

The fastest known algorithm for detecting or counting line intersections in $\mathbb{R}^3$ runs in time $O(n^{8/5+\epsilon})$ [6].

4.3 Offline Range Searching

Chazelle [4] has developed lower bounds for the query time required by a simplex range query data structure, given an upper bound on its size. However, his results only apply to the range searching problems in which the data structure must be built to handle arbitrary queries online. Previously, no lower bounds larger than $\Omega(n \log n)$ were known for any offline range searching problem, where all the queries are known in advance.

Theorem 4.4. *Any partitioning algorithm that computes, given n points and m halfplanes, the sum over all halfplanes of the number of points contained in each halfplane, requires time $\Omega(n \log m + n^{2/3}m^{2/3} + m \log n)$ in the worst case.*

To reduce the counting version of Hopcroft's problem to the offline halfplane counting problem, we replace each line with the pair of closed halfplanes bounded by that line. The number returned by the halfspace counting algorithm is mn plus the number of incidences.

Efficient partitioning algorithms for the offline halfplane counting problem use the containment shortcut described in Section 3.4. That is, if a query region is completely contained in a halfspace, then the halfplane does not traverse the corresponding edge. Rather, the query region is marked, so that whenever a point is found inside the region, a running total is incremented. Since adding the containment shortcut doesn't significantly speed up algorithms for Hopcroft's problem, our lower bound applies.

Similar arguments apply to other offline range searching problems, such as the following.

Theorem 4.5. *Any partitioning algorithm that determines, given n points and m triangles, whether any triangle contains a point, requires time $\Omega(n \log m + n^{2/3}m^{2/3} + m \log n)$ in the worst case.*

Theorem 4.6. *Any partitioning algorithm that determines, given n line segments and m rays, whether any ray hits a line segment, requires time $\Omega(n \log m + n^{2/3}m^{2/3} + m \log n)$ in the worst case.*

4.4 Circles and Unit Distances

Theorem 4.7. *Any partitioning algorithm that detects incidences between n points and m circles in the plane requires time $\Omega(n \log m + n^{2/3}m^{2/3} + m \log n)$ in the worst case.*

By applying a linear fractional transformation to the plane, we can map the set of points and lines described in Lemma 2.5 to a set of points and circles. Using this approach, Clarkson *et al.* [7] prove a lower bound of $\Omega(n + n^{2/3}m^{2/3} + m)$ on the number of incidences between n points and m circles in the plane. They also prove an upper bound of $O(n + n^{4/5}m^{3/5} + m)$.

Since there are no incidences in the set of points and lines from Lemma 2.5, we can approximate it by a set of points and *unit* circles, where the unit distance is much larger than the distance between any two points. We can further guarantee that no two circle centers are a unit distance apart. This immediately implies the following much stronger lower bound.

Theorem 4.8. *Any partitioning algorithm that detects unit distances among n points in the plane requires time $\Omega(n^{4/3})$ in the worst case.*

This result is somewhat surprising given known bounds on the maximum *number* of unit distances among n points in the plane. Spencer *et al.* [25] and Clarkson *et al.* [7] prove an upper bound of $O(n^{4/3})$. Erdős [16] proves a lower bound of $n^{1+c/\log\log n}$ for some constant c, which is achieved by n points on a regular square lattice, and conjectures that this bound is tight.[5] The fastest known algorithm for detecting unit distances runs in time $O(n^{4/3+\epsilon})$.

4.5 A "Convex" Version of Hopcroft's Problem

The following lower bound derives directly from Lemma 2.14, Lemma 2.16, and Theorem 3.5.

Theorem 4.9. *Any (unbounded) partitioning algorithm that counts incidences between n points and m hyperplanes in $\mathbb{R}^5$, where every point lies on or above every hyperplane, requires time $\Omega(n + n^{2/3}m^{2/3} + m)$ in the worst case.*

In this case, we do not inherit the $\Omega(n\log m + m\log n)$ lower bound from Theorem 3.1, since all the poitns lie on one side of the hyperplanes. In fact, the *decision* version of this problem — Does every point lie strictly above every hyperplane? — can be solved by a partitioning algorithm in time $O(n + m)$, by using the convex hull of the points as a query region.

For the special case $n = m$, the best upper bound known for this problem in both four and five dimensions is $O(n^{4/3+\epsilon})$, using the half-space emptiness query structure of Matoušek [22]. In two and three dimensions, this problem can be solved in linear time in the partitioning model, or in $O(n\log n)$ time in the algebraic decision tree model, using any optimal convex hull algorithm. Proving nontrivial lower bounds for the four-dimensional case remains an open problem.

5 Conclusions and Open Problems

We have proven new lower bounds on the complexity of Hopcroft's problem, in a general model of computation. Our lower bound proofs were developed in two stages. First, we derived lower bounds on the minimum size of a monochromatic cover in the worst case. Second, we showed that the running time of any partitioning algorithm is bounded below by the size of some monochromatic cover.

A number of open problems remain to be solved. The most obvious problem is to improve the lower bounds, in particular for the case $n = m$. The true complexity almost certainly increases with the dimension, but the best lower bound we can achieve in higher dimensions comes trivially from

[5] Erdős has reportedly offered $500 for a proof or disproof of this conjecture.

the two-dimensional case. Does there exist a set of n points and n planes in $\mathbb{R}^3$ whose minimum monochromatic cover size is $\omega(n^{4/3})$?

Our lower bounds are significantly smaller when m and n are close together than when they are far apart. In contrast, known algorithms have complexities that vary evenly for all values of m between $\Omega(n^{1/d})$ and $O(n^d)$. Are the breakpoints in our lower bounds purely an artifact of our proof technique, or is it really easier to detect incidences when the number of points and the number of hyperplanes are nearly equal?

One possible approach is to consider restrictions of the partitioning model. Can we achieve better bounds if we only consider algorithms whose query regions are convex? What if the query regions at every node are distinct? What if the running time depends on the complexity of the query regions?

A related open problem is bounding the monochromatic cover size of *arbitrary* sign matrices, not just those realizable as a relative orientation matrix of points and hyperplanes. The best bounds we have been able to prove for arbitrary square matrices are $\Omega(n^{3/2})$ and $O(n^2/\log n)$.

The class of partitioning algorithms is general enough to directly include many, but not all, existing algorithms for deciding Hopcroft's problem. The model requires that a single data structure be used to determine which points and hyperplanes intersect each query region, but many algorithms use a tree-like structure to locate the points and the Zone Theorem to locate the hyperplanes. We can usually modify such algorithms so that they do fit our model, at the cost of only a constant factor in their running time, but this is a rather ad hoc solution. Any extension of our lower bounds to a more general model, which would explicitly allow different strategies for locating points and hyperplanes, would be interesting. We described one possible approach at the end of Section 3.5.

The partitioning algorithm model is specifically tailored to detect intersections or containments between pairs of objects. There are a number of similar geometric problems for which the partitioning algorithm model simply does not apply. We mention one specific example, the *cyclic overlap problem*. Given a set of line segments in $\mathbb{R}^3$, does any subset form a cycle with respect to the "above" relation? The fastest known algorithm for this problem, due to de Berg *et al.* [9], runs in time $O(n^{4/3+\epsilon})$, using a divide-and-conquer strategy very similar to algorithms for Hopcroft's problem. In the algebraic decision tree model, the cyclic overlap problem is at least as hard as Hopcroft's problem [17]. However, it is not clear that this problem can even be solved by a partitioning algorithm, since the answer might depend on arbitrarily large tuples of segments, arbitrarily far apart. Extending our lower bounds into more traditional models of computation is an important and very difficult open problem.

Acknowledgements. The author gratefully thanks Kurt Mehlhorn and the Max-Planck-Institut für Informatik in Saarbrücken for their generous hospitality, and Raimund Seidel for his continuing support, encouragement, suggestions, and patience.

References

[1] P. K. Agarwal. Partitioning arrangements of lines: II. Applications. *Discrete Comput. Geom.*, 5:533–573, 1990.

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

[3] B. Chazelle. Reporting and counting segment intersections. *J. Comput. Syst. Sci.*, 32:156–182, 1986.

[4] B. Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2:637–666, 1989.

[5] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(2):145–158, 1993.

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

[7] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete Comput. Geom.*, 5:99–160, 1990.

[8] R. Cole, M. Sharir, and C. K. Yap. On k-hulls and related problems. *SIAM J. Comput.*, 16:61–77, 1987.

[9] M. de Berg, M. Overmars, and O. Schwarzkopf. Computing and verifying depth orders. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 138–145, 1992.

[10] M. de Berg and O. Schwarzkopf. Cuttings and applications. Report RUU-CS-92-26, Dept. Comput. Sci., Utrecht Univ., Utrecht, Netherlands, Aug. 1992.

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

[12] H. Edelsbrunner, L. Guibas, J. Hershberger, R. Seidel, M. Sharir, J. Snoeyink, and E. Welzl. Implicitly representing arrangements of lines or segments. *Discrete Comput. Geom.*, 4:433–466, 1989.

[13] H. Edelsbrunner, L. Guibas, and M. Sharir. The complexity of many cells in arrangements of planes and related problems. *Discrete Comput. Geom.*, 5:197–216, 1990.

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

[15] H. Edelsbrunner and M. Sharir. A hyperplane incidence problem with applications to counting distances. In P. Gritzman and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 253–263. AMS Press, 1991.

[16] P. Erdős. On a set of distances of n points. *Amer. Math. Monthly*, 53:248–250, 1946.

[17] J. Erickson. Detecting cycles in depth orders is harder than Hopcroft's problem. Unpublished manuscript, 1994.

[18] J. Erickson and R. Seidel. Better lower bounds on detecting affine and spherical degeneracies. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci. (FOCS 93)*, pages 528–536, 1993.

[19] M. L. Fredman. Lower bounds on the complexity of some optimal data structures. *SIAM J. Comput.*, 10:1–10, 1981.

[20] G. Hardy and E. Wright. *The Theory of Numbers*. Oxford University Press, London, England, 4th edition, 1965.

[21] L. Lovàsz. Communication complexity: A survey. In *Paths, Flows, and VLSI Layout*, volume 9 of *Algorithms and Combinatorics*, pages 235–265. Springer-Verlag, 1990.

[22] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993. The results combined with results of O. Schwarzkopf also appear in *Proc. 6th ACM Sympos. Comput. Geom.*, 1992, pages 16–25.

[23] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10(2):157–182, 1993.

[24] R. Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Comput. Geom. Theory Appl.*, 1:51–64, 1991.

[25] J. Spencer, E. Szemerédi, and W. T. Trotter, Jr. Unit distances in the Euclidean plane. In B. Bollobás, editor, *Graph Theory and Combinatorics: Proceedings of the Cambridge Combinatorial Conference in Honor of Paul Erdős*, pages 293–303. Academic Press, 1984.

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

[27] J. Stolfi. *Oriented Projective Geometry: A Framework for Geometric Computations.* Academic Press, 1991.

[28] E. Szemerédi and W. T. Trotter, Jr. Extremal problems in discrete geometry. *Combinatorica*, 3:381–392, 1983.