# Part II

# Divide-and-Conquer Lower Bounds

*He who can properly define and divide is to be considered a god.*

— Plato, c. 400 BC

*Great wits are sure to madness near allied,*
*And thin partitions do their bounds divide.*

— John Dryden, *Absalom and Achitophel*, 1681

# Chapter 6

# Hopcroft's Problem

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 [32], 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 [51], described in [62, p. 350], runs in time $O(n^{3/2} \log^{1/2} n)$. Cole, Sharir, and Yap [51] combined these two algorithms, achieving a running time of $O(n^{1.412})$. Edelsbrunner, Guibas, and Sharir [66] developed a randomized algorithm with expected running time $O(n^{4/3+\varepsilon})$ [1] ; see also [64]. A somewhat simpler algorithm with the same running time was developed by Chazelle, Sharir, and Welzl [43]. Further research replaced the $n^\varepsilon$ term in this upper bound with a succession of smaller and smaller polylogarithmic factors. The running time was improved by Edelsbrunner, Guibas, Hershberger, Seidel, Sharir, Snoeyink, and Welzl [63] 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 [35] to $O(n^{4/3} \log^{1/3} n)$; and most recently by Matoušek [111] to $n^{4/3} 2^{O(\log^* n)}$. This is currently the fastest algorithm known. Matoušek's algorithm can be tuned to detect incidences among

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

$n$ points and $m$ lines in the plane in time $O(n \log m + n^{2/3} m^{2/3} 2^{O(\log^*(n+m))} + m \log n)$ [54], 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 [138, 16].

In this chapter, 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 prove that any partitioning 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.

Informally, a partitioning algorithm covers space with a constant number of (not necessarily disjoint) connected regions, determines which points and hyperplanes intersect each region, and recursively solves each of the resulting subproblems. The algorithm may apply projective duality to reverse the roles of the points and hyperplanes, that is, to partition the input according to which dual hyperplanes and dual points intersect each region. The algorithm is also allowed to merge subproblems arbitrarily before partitioning. For purposes of proving lower bounds, we assume that partitioning the points and hyperplanes requires only linear time, regardless of the complexity of the regions or how they depend on the input. We give a more formal definition of partitioning algorithms in Section 6.3.

To develop lower bounds in this model, we first 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 its worst case complexity. A monochromatic cover is a partition of the sign matrix induced by the relative orientations of the points and hyperplanes into (not necessarily disjoint) minors, such that all entries in each minor are equal. The size of a cover is the total number of rows and columns in the minors. The main result of this chapter (Theorem 6.19) is that the running time of any partitioning algorithm is bounded below by the size of some monochromatic cover of its input.

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 [62, p. 112]. 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. Our planar lower bounds are ultimately based on the Erdős configuration.

Chazelle [33, 38] has established lower bounds for the closely related *simplex range searching* problem: Given a set of points and a set of simplices, how many points are in each simplex? For example, any data structure of size $s$ that supports triangular range queries among $n$ points in the plane requires $\Omega(n/\sqrt{s})$ time per query [33]. It follows that answering $n$ queries over $n$ points requires $\Omega(n^{4/3})$ time in the worst case. For the offline version of the same problem, where all the triangles are known in advance, Chazelle establishes a slightly weaker bound of $\Omega(n^{4/3}/\log^{4/3} n)$ [38], although an $\Omega(n^{4/3})$ lower bound follows immediately from the Erdős construction using Chazelle's methods. In higher dimensions, Chazelle's results imply lower bounds of $\Omega(n^{2d/(d+1)}/\log^{d/(d+1)} n)$ and $\Omega(n^{2d/(d+1)}/\log^{5/2-\gamma} n)$ in the online and offline cases, respectively, where $\gamma > 0$ is a small constant that depends on $d$. All these lower bounds hold in the Fredman/Yao semigroup arithmetic model; see Section 1.3. For related results, see also [23, 44].

Lower bounds in the semigroup model are based on the existence of configurations of points and ranges, such as the planar Erdős configuration, whose incidence graphs have (a subgraph with) no large complete bipartite subgraphs. Our lower bounds have a similar basis. In Section 6.2, we develop point-hyperplane configurations of this type, naturally generalizing the Erdős configuration to arbitrary dimensions. These configurations also allow us to extend Chazelle's offline lower bounds to a counting version of Hopcroft's problem.

## 6.1 Easy Quadratic Lower Bounds

In light of the results of Part I, it is natural to ask whether we can prove lower bounds for Hopcroft's problem in a model of computation that allows only simple queries. The appropriate primitive to consider is the *relative orientation query*: Given a point and a hyperplane, does the point lie above, on, or below the hyperplane? Algebraically, the result of a relative orientation query is given by the inner product of the homogeneous

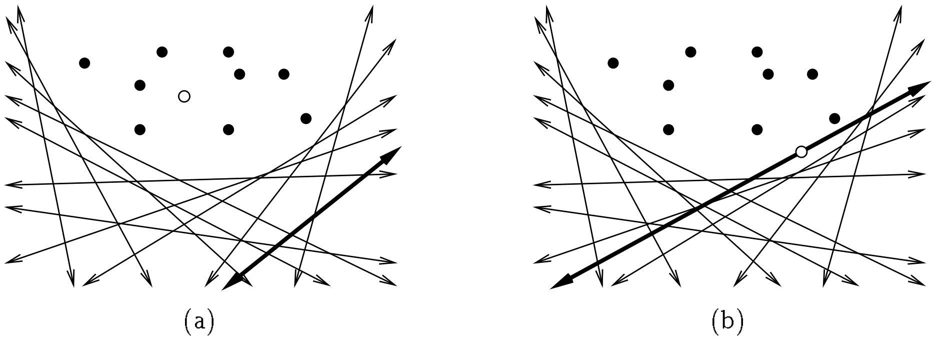


**Figure 6.1.** An easy adversary construction for Hopcroft's problem. (a) The original adversary configuration. (b) A collapsed configuration.

coordinate vectors of the point and the hyperplane [140]. Surprisingly, we can easily establish a *quadratic* lower bound for Hopcroft's problem if this is the only primitive we are allowed.

**Theorem 6.1.** *Any decision tree algorithm that decides Hopcroft's problem in* $\mathbb{R}^d$ *for any* $d \geq 1$*, using only relative orientation queries, must have depth at least* $nm$.

**Proof:** The lower bound follows from a simple adversary argument. The adversary presents the algorithm with a set of $n$ points and $m$ hyperplanes in which every point is above every hyperplane. If the algorithm does not perform a relative orientation query for some point-hyperplane pair, the adversary can move that point onto that hyperplane without changing the relative orientation of any other pair. See Figure 6.1. The algorithm cannot tell the two configurations apart, even though one has an incidence and the other does not. Thus, in order to be correct, the algorithm must perform a relative orientation query for every point-hyperplane pair. □

Obviously, this lower bound is tight.

In dimensions higher than one, we can considerably strengthen the model of computation in which this quadratic lower bound holds. We will explicitly describe only the two-dimensional case; generalization to higher dimensions is straightforward. Our new model of computation is a decision tree with three types of primitives: relative orientation queries, point queries, and line queries. A point query is any decision that is based exclusively on the coordinates the input points. Line queries are defined analogously. We emphasize that point queries can combine information from any number of points, and line

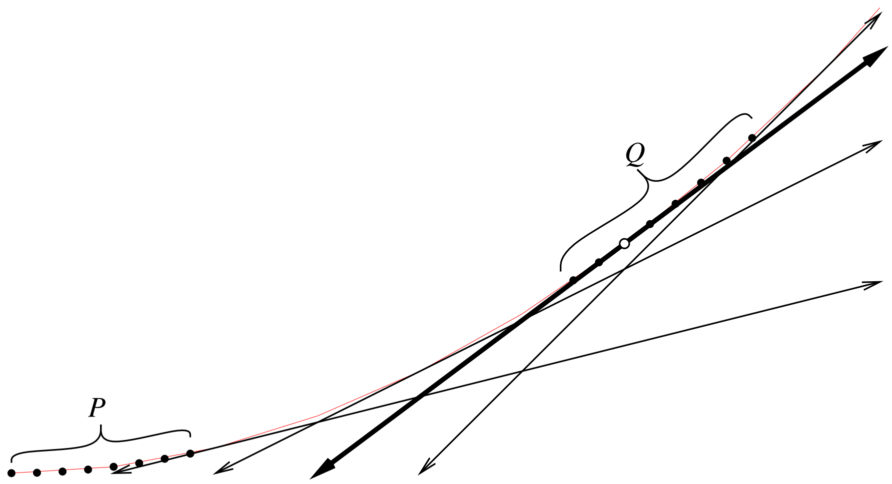


**Figure 6.2.** A harder adversary construction for Hopcroft's problem. The white point in Q is on the dark line; otherwise, every point is above every line.

queries from any number of lines. We call a point query *algebraic* if the result is given by the sign of a multivariate polynomial, not necessarily of bounded degree, evaluated at the point coordinates.

**Theorem 6.2.** *Any decision tree algorithm that solves Hopcroft's problem in the plane, using only relative orientation queries, algebraic point queries, and (arbitrary) line queries, must make at least $nm$ relative orientation queries in the worst case.*

**Proof:** For any real number $x_0$, let $P(x_0)$ denote the set of $n$ points

$$\left\{\left(x_0, x_0^2\right), \left(x_0+1, (x_0+1)^2\right), \ldots, \left(x_0+n-1, (x_0+n-1)^2\right)\right\},$$

and let $L(x_0)$ denote the set of $m$ lines tangent to the unit parabola $y = x^2$ at the points

$$\left\{\left(x_0+n, (x_0+n)^2\right), \left(x_0+2n, (x_0+2n)^2\right), \ldots, \left(x_0+mn, (x_0+mn)^2\right)\right\}.$$

As before, our lower bound follows from an adversary argument. The adversary initially presents the points $P = P(x_0)$ and the lines $L = L(x_0)$, for some real value $x_0$ to be specified later. If the algorithm does not perform a relative orientation query for the $i$th point and the $j$th line, then the adversary replaces the points with the new set $Q = P(x_0 + in - j + 1)$. We easily verify that in the new configuration, the $i$th point and the $j$th line are incident, but otherwise, every point is above every line. See Figure 6.2.

Since the adversary does not change the lines, no line query can distinguish between the two configurations. It remains only to consider the point queries. The result of any algebraic point query in $P(x)$ is given by the sign of a polynomial in the single variable $x$. Let $r_{max}$ be the largest root of all the point query polynomials used by the algorithm, ignoring those that are identically zero. If $x_1$ and $x_2$ are both larger than $r_{max}$, then every point query polynomial has the same sign at both $P(x_1)$ and $P(x_2)$. Thus, if the adversary fixes $x_0 > r_{max}$, then the algorithm cannot distinguish between the original point set $P$ and any of the collapsed point sets $Q$ using point queries.

It follows that the algorithm cannot be correct unless it performs a relative orientation query for every pair. $\square$

Our restriction to algebraic point queries is actually stronger than the result requires; it suffices that for all $x$ in some sufficiently large interval, the result of every point query on the set $P(x)$ is constant. If we rephrase this argument in the dual space, we get a quadratic lower bound in a model that allows arbitrary point queries but requires line queries to be algebraic.

These adversary arguments actually give us a quadratic lower bound for the much easier *halfspace emptiness* problem "Is every point above every hyperplane?". We consider this problem in greater detail in the next chapter. Our arguments can easily be modified to apply to almost any range searching problem, and a wide range of other related problems; see [78]. We leave the details and further generalizations as exercises for the reader.

Of course, none of the sub-quadratic algorithms listed previously follow the models of computation considered in this section. Unlike the degeneracy problems considered in Part I, there does not appear to be a small fixed set of primitives that are used by all known algorithms for Hopcroft's problem. Many algorithms define several levels of higher-order geometric objects, and some of their decisions are based on large fractions of the input.

In light of our results, it is clear that higher-order primitives that involve both points and lines, such as "Is this point to the left or right of the intersection of these two lines?", are necessary to achieve nontrivial upper bounds. If we allow any primitives of this type, however, it seems unlikely that our earlier adversary techniques can be used to derive nontrivial lower bounds, either for Hopcroft's problem or for other range searching problems. We leave the development of such lower bounds as an interesting open problem.

## 6.2 Incidences and Monochromatic Covers

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of points and $H = \{h_1, h_2, \ldots, h_m\}$ be a set of hyperplanes in $\mathbb{R}^d$. These two sets naturally induce a *relative orientation matrix* $M(P, H) \in \{+, 0, -\}^{n \times m}$. Each row of $M(P, H)$ corresponds to one of the points in $P$; each column corresponds to a hyperplane in $H$; and the $(i, j)$th entry denotes whether the point $p_i$ is above $(+)$, on $(0)$, 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 $H' \subseteq H$.

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, and the size of a minor cover is the sum of the sizes of the minors in the cover. Given a set of points and hyperplanes, a monochromatic cover of its relative orientation matrix provides a succinct combinatorial representation of its relative order type.

We similarly define a succinct representation for the incidence structure of a set of points and hyperplanes. A *zero cover* of $P$ and $H$ is a collection of monochromatic minors that covers all (and only) the zeros in the relative orientation matrix $M(P, H)$. A zero cover can also be interpreted as a partition of the incidence graph induced by $P$ and $H$ into (not necessarily disjoint) complete bipartite subgraphs.

Monochromatic covers for 0-1 matrices have been previously used to prove lower bounds for communication complexity problems [106]. 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.[2] Covers of bipartite graphs by complete subgraphs were introduced by Tarján [145] in the context of switching theory. Tuza [149], and independently Chung, Erdős, and Spencer [45], showed that every $n \times m$ bipartite graph has such a cover of size $O(nm/\log(\max(m, n)))$ and that this bound is tight in the worst case, up to constant factors. These results apply immediately to monochromatic covers of arbitrary sign matrices. See also [2] for a geometric application of bipartite clique 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.

[2] Since any row or column can be split into three or fewer monochromatic minors, any sign matrix can be covered by $3\min(m, n)$ 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.

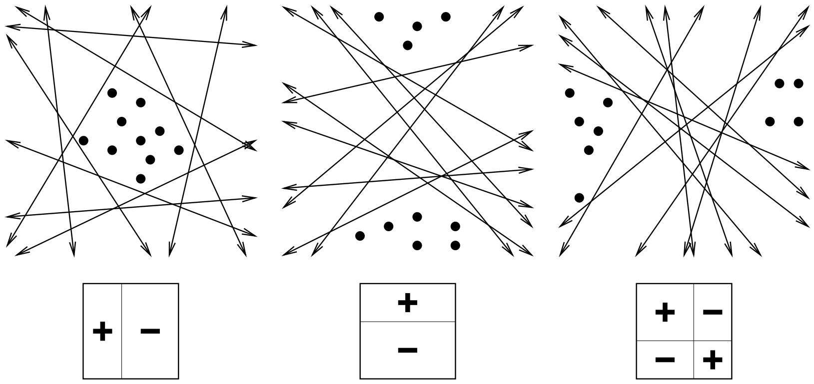

**Figure 6.3.** Collections of points and lines with simple relative orientation matrices.

This coordinate system determines which minors of the relative orientation matrix are monochromatic, and therefore determine the minimum monochromatic cover size. However, we easily observe that the minimum monochromatic cover size is independent of any choice of coordinate system, up to a factor of two, as follows. Call a relative orientation matrix *simple* if it can be changed into a monochromatic matrix by inverting some subset of the rows and columns. Projective transformations preserve simple minors. See Figure 6.3. Every monochromatic minor is simple, and every simple minor can be partitioned into four monochromatic minors, whose total size is twice that of the original minor.

We will use the following notation throughout the rest of the chapter. Given a set $P$ of points and $H$ of hyperplanes, let $I(P, H)$ denote the number of point-hyperplane incidences, $\zeta(P, H)$ the minimum size of their smallest zero cover, and $\mu(P, H)$ the size of their smallest monochromatic cover. 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 $\mathbb{R}^d$, and define $\zeta_d(n, m)$ and $\mu_d(n, m)$ similarly. Finally, let $\mu_d^\circ(n, m)$ denote the maximum of $\mu(P, H)$ over all sets of $n$ points and $m$ hyperplanes in $\mathbb{R}^d$ with no incidences. In the remainder of this section, we develop asymptotic lower bounds for $\mu_d^\circ(n, m)$ and $\zeta_d(n, m)$, which in turn imply lower bounds for $\mu_d(n, m)$.

### 6.2.1 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). The matrix can then be split into two "staircase" matrices, one positive and one negative, and a collection of zero minors of total size at most $n+m$. We immediately observe that $\zeta_1(n,m) = n+m$.

**Theorem 6.3.** $\mu_1^\circ(m,n) = \begin{cases} \Theta(n \log_{n/m} m) & \textit{if } n > m \\ \Theta(n \log n) & \textit{if } n = m \\ \Theta(m \log_{m/n} n) & \textit{if } n < m \end{cases}$

**Proof:** Without loss of generality, we assume $n$ and $m$ are both powers of two. It suffices to bound the cover size of a monochromatic staircase with $n$ rows and $m$ columns.

First consider the simplest case, $n = m$. To prove the upper bound, we construct a cover of an arbitrary $n \times n$ staircase matrix by partitioning the staircase into an $n/2 \times k$ monochromatic minor and two smaller staircases, where $k$ is the number of entries in the $n/2$th row of the original matrix. The total size $C(n,n)$ of this cover is bounded by the recurrence

$$C(n,n) \leq \max_{0 \leq k \leq n} (n/2 + k + C(n/2,k) + C(n/2, n-k))$$

whose solution is $C(n,n) \leq (3n/2)\lg n$.

To prove the matching lower bound, it suffices to consider a triangular matrix, where for all $i$, the $i$th row has $i$ entries. We claim that any cover for this matrix must have size at least $(n/2)\lg n$. Fix a cover. Partition the staircase into an $n/2 \times n/2$ rectangle and two $n/2 \times n/2$ staircases. If a minor in the cover intersects the lower triangle, call it a lower minor; otherwise, call it an upper minor. The upper (resp. lower) minors induce a cover of the upper (resp. lower) triangle, of size at least $(n/4)\lg(n/2)$, by induction. It remains to bound the contribution of the rectangle to the total cover size.

If some row in the rectangle is completely contained in lower minors, then those lower minors have (altogether) $n/2$ more columns than we accounted for in the induction step. Otherwise, every row contains an element of an upper minor, and those upper minors have (altogether) $n/2$ more rows than we accounted for in the induction step. Thus, the rectangle contributes at least $n/2$ to the total cover size. This completes the proof for the case $n = m$.

Now suppose $n > m$. An explicit recursive construction gives us a cover of size $O(n \log_{n/m} m)$ for any $n \times m$ staircase. An inductive counting argument implies that any cover of the $n \times m$ triangular matrix, whose $i$th row has $\lceil im/n \rceil$ entries, must have size at least $(n/2)\log_{n/m} m$. In both arguments, we start by dividing the $n$ rows of the matrix into $n/m$ slabs of $m$ rows each, and cutting each slab into a maximal rectangle and a smaller staircase. We leave the details as an easy exercise.

The final case $n < m$ is handled symmetrically. □

This bound simplifies to $\Theta(n+m)$ when either $n = O(m^k)$ or $m = O(n^k)$ for any constant $k < 1$, and to $\Theta(n \log n)$ when $n = \Theta(m)$. In the special case $n = m$, 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 divide-and-conquer algorithm is readily apparent in this case. In Section 6.3, we generalize this connection to higher dimensions.

### 6.2.2 Two Dimensions

To derive lower bounds for $\mu_2^\circ(n, m)$ and $\zeta_2(n, m)$, we use the following combinatorial result of Erdős. (See [82] or [62, p.112] for proofs.)

**Lemma 6.4 (Erdős).** *For all $n$ and $m$, there is a set of $n$ points and $m$ lines in the plane with $\Omega(n + n^{2/3}m^{2/3} + m)$ incident pairs. Thus, $I_2(n, m) = \Omega(n + n^{2/3}m^{2/3} + m)$.*

Fredman [82] uses Erdős' construction to prove lower bounds for dynamic range query data structures in the plane.[3] This lower bound is asymptotically tight. The corresponding upper bound was first proven by Szemerédi and Trotter [143]. A simpler proof, with better constants, was later given by Clarkson, Edelsbrunner, Guibas, Sharir, and Welzl [46]. A completely elementary proof has recently discovered by Székely [119].

**Theorem 6.5.** $\zeta_2(n, m) = \Omega(n + n^{2/3}m^{2/3} + m)$

**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 6.4. □

**Theorem 6.6.** $\mu_2^\circ(n, m) = \Omega(n + n^{2/3}m^{2/3} + m)$

[3] Perhaps it is more interesting that Chazelle's static lower bounds [33, 38] do *not* use this construction.

**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 6.4. Replace each line $\ell$ in this configuration with a pair of lines, parallel to $\ell$ and at distance $\varepsilon$ on either side, where $\varepsilon$ is chosen sufficiently small that all point-line distances in the new configuration are at least $\varepsilon$. 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 $\varepsilon$. 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 both 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. □

### 6.2.3 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 6.4. In order to derive a lower bound for either $\zeta_3(m, n)$ or $\mu_3^\circ(n, m)$, we need a configuration of points and planes with many incidences, but without large sets of mutually incident points and planes. In the following lemma, we construct such a configuration, naturally generalizing Erdős' planar construction.

We use the notation $[n]$ to denote the set of integers $\{1, 2, \ldots, n\}$; $i \perp j$ to mean that $i$ and $j$ are relatively prime; and $\varphi(n)$ to denote the Euler totient function $\varphi(n)$, the number of positive integers less than or equal to $n$ that are relatively prime to $n$.

Our construction relies on the following lemma, whose (simple) proof we omit. We refer the reader to [98] or [92] for relevant background.

**Lemma 6.7.** $\sum_{i=1}^{n} i^k \varphi(i) = \Theta(n^{k+2})$ *for any nonnegtive integer* $k$.

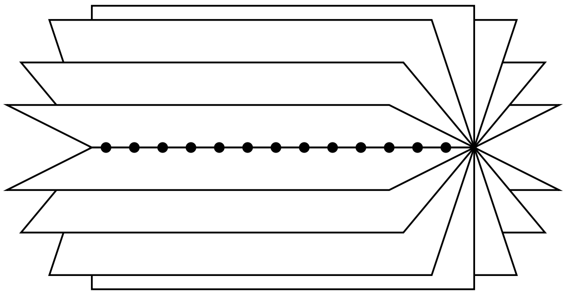

**Figure 6.4.** $I_3(n, m) = mn$. Every point lies on every plane.

**Lemma 6.8.** *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

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

The number of planes in $H$ is

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

By choosing the constant $\alpha$ 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.

Since any rational plane can be represented in the form $h(a, b, c; i, j)$ in an infinite number of different ways, we must first check that the planes in $H$ are actually distinct! 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, $H$ contains $m$ distinct planes.

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.

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

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, Guibas, and Sharir [64] prove an upper bound of $O(n \log m + n^{4/5+2\varepsilon} m^{3/5-\varepsilon} + m)$ on the maximum number of incidences between $n$ points and $m$ planes, where no three planes contain a common line. The probabilistic counting techniques of Clarkson *et al.* [46] imply the better upper bound $O(n + n^{4/5} m^{3/5} + m)$. We omit further details.

**Theorem 6.9.** $\zeta_3(n,m) = \Omega(n + n^{5/6} m^{1/2} + n^{1/2} m^{5/6} + m)$

**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 6.8. 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 6.10.** *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* $I(P, H) \leq 2(m + n)$.

**Proof:** Call any point (resp. plane) *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 two 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 bipartite graph is clearly planar, and thus has at most $2(m + n)$ edges. □

**Theorem 6.11.** $\mu_3^{\circ}(n, m) = \Omega(n + n^{5/6}m^{1/2} + n^{1/2}m^{5/6} + m)$

**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 6.8. Replace each plane $h \in H$ with a pair of parallel planes at distance $\varepsilon$ on either side of $h$, for some suitably small constant $\varepsilon > 0$. Call the resulting

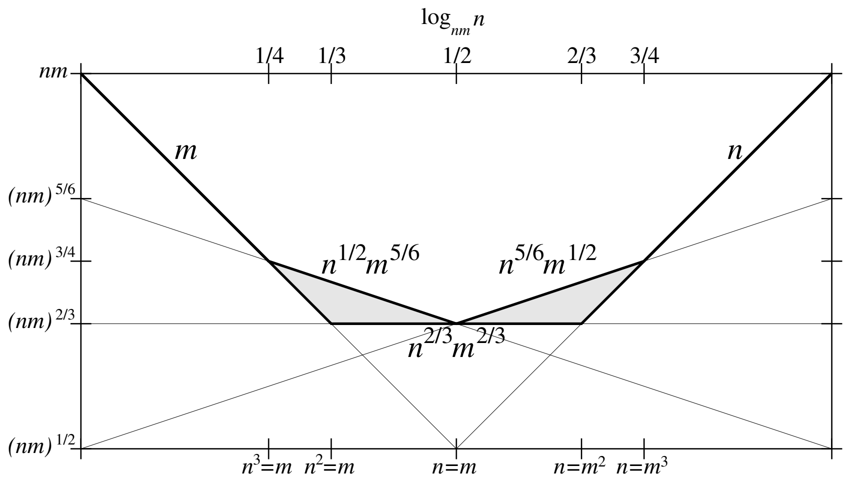


**Figure 6.5.** Comparison of lower bounds for $\mu_2^\circ(n, m)$ and $\mu_3^\circ(n, m)$. See Theorems 6.6 and 6.11.

set of $m$ planes $H_\varepsilon$. We say that a point is *close* to a plane if the distance between them is exactly $\varepsilon$. There are $\Omega(n^{5/6}m^{1/2})$ close pairs between $P$ and $H_\varepsilon$, and no incidences.

Call a sign matrix *loosely monochromatic* if either none of its entries is $+$ or none of its entries is $-$. For any subsets $P' \subseteq P$ and $H' \subseteq H$, Lemma 6.10 implies that if $M(P', H')$ is loosely monochromatic, then $I(P', H') = O(|P'| + |H'|)$.

For every monochromatic minor of the matrix $M(P, H_\varepsilon)$, the corresponding minor of $M(P, H)$ is loosely monochromatic. 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 $H_\varepsilon$ must have size $\Omega(n^{5/6}m^{1/2})$.

Similar arguments apply to other values of $m$. $\square$

For the special case $m = \Theta(n)$, this theorem does not improve the $\Omega(n^{4/3})$ bound 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 6.5.

### 6.2.4 Higher Dimensions

In order to generalize Lemma 6.8 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 $f_1(t) = 1$, $f_i(t) = t + i - 2$ for all $i > 1$, and $F_i(t) = \prod_{j=1}^{i-1} f_j(t)$ for all $i$.

**Lemma 6.12.** *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 $M$ 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 $d$th entry is now $F_d(t)$, and for all $j < d$, the $j$th entry is now $f_d(t) \cdot t^{j-1} = t^j + (d-2)t^{j-1}$. 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 $j$th polynomial. The only nonzero entries in $C$ are the coefficients of $F_d(t)$ in the last column, $(d-2)$'s in rest of the main diagonal, and ones in the next lower diagonal. For example, when $d = 4$, our vector of polynomials is $(t+2, t^2+2t, t^3+2t^2, t^2+t)$, and

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

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 $\det C = (-1)^d F_d(2-d)$. Since $2-d$ 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 $V$. $\square$

**Lemma 6.13.** *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* $I(P, H) = \Omega(n^{1-2/d(d+1)} m^{2/(d+1)})$ *and any* $d$ *hyperplanes in* $H$ *intersect in at most one point.*

**Proof (sketch):** First consider the case $d = 4$. 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 set of conditions.

$$\begin{gathered} i \in [q], \quad j \in [i], \quad j \text{ is odd}, \quad i \perp j \\ a \in [i], \quad b \in [j], \quad c \in [i + j], \quad d \in \Big[ \lfloor p/2 \rfloor \Big] \end{gathered}$$

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 straightforward algebraic manipulation, similar to that used in the proof of Lemma 6.8, 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 6.12, by setting $t_k = j_k / i_k$ for all $k$, that no four of these planes are concurrent.

Now consider the $d$-dimensional case. Let $e_i$ denote the unit vector whose $i$th coordinate is 1 and whose other coordinates are 0. For any point $x \in \mathbb{R}^d$, let $h(x; i, j)$ denote the affine hull of $x$ and the points $x + ie_1 + ((k-2)i + j)e_k$ for all $2 \le k \le d$. Let $p = \lfloor n^{1/d} \rfloor$ and $q = \lfloor \alpha (m/p)^{1/(d+1)} \rfloor$ for some suitable constant $\alpha > 0$. Then $P = [p]^d$; and $H$ is the set of hyperplanes $h(x; i, j)$ such that $i, j, i + j, 2i + j, \ldots, (d-2)i + j$ are pairwise relatively prime; $x_1 \in [i]$; $x_k \in [(k-2)i + j]$ for all $2 \le k \le d$; and $x_d \in [\lfloor p/2 \rfloor]$.

The relative primality conditions on $i$ and $j$ imply that the hyperplanes in $H$ are distinct. The bounds $|H| = O(pq^{d+1}) = O(m)$ and

$$I(P, H) = \Omega(p^d q^2) = \Omega(n^{1-2/d(d+1)} m^{2/(d+1)})$$

follow from straightforward algebraic manipulation. Finally, Lemma 6.12 implies that any $d$ hyperplanes in $H$ intersect in a unique point. We omit further details. □

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 [46], 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.

The previous lemma immediately gives us the following lower bound for $\zeta_d(n, m)$.

**Theorem 6.14.** $\zeta_d(n, m) = \Omega\left(\sum_{i=1}^{d} \left(n^{1-2/i(i+1)} m^{2/(i+1)} + n^{2/(i+1)} m^{1-2/i(i+1)}\right)\right)$

Since our $d$-dimensional lower bound only improves our $(d-1)$-dimensional lower bound for certain values of $n$ and $m$, we have combined the lower bounds from all dimensions $1 \le i \le d$ into a single expression. If the relative growth rates of $n$ and $m$ are fixed, the entire sum can be reduced to a single term.

Unfortunately, we are unable to generalize Lemma 6.10 even into four dimensions. Consequently, the best lower bound we can derive for $\mu_d^\circ(n, m)$ for any $d > 3$ derives trivially from Theorem 6.11. The best upper bound we can prove for 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 [70] 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 superlinear number of incidences in five dimensions, under a weaker combinatorial general position requirement. Thus, unlike in lower dimensions, some sort of *geometric* general position requirement is necessary to keep the number of incidences small.

**Lemma 6.15.** *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 $\langle \cdot, \cdot \rangle$ denotes the usual inner product of vectors. In a more geometric setting, $\sigma$ maps points and lines in the plane, represented in homogeneous coordinates, to points and hyperplanes in $\mathbb{R}^5$, also represented in homogeneous coordinates [140]. For any point $p$ and line $\ell$ in the plane, the point $\sigma(p)$ is incident to the hyperplane $\sigma(\ell)$ if and only if $p$ is incident to $\ell$; otherwise, $\sigma(p)$ lies above $\sigma(\ell)$. Thus, we can take $P$ and $H$ to be the images under $\sigma$ of any sets of $n$ points and $m$ lines with $\Omega(n + n^{2/3}m^{2/3} + m)$ incidences, as given by Lemma 6.4. □

### 6.2.5 A Lower Bound in the Semigroup Model

Our results immediately imply a lower bound for a variant of the counting version of Hopcroft's problem, in the Fredman/Yao semigroup arithmetic model. The lower bound follows from the following result of Chazelle [38, Lemma 3.3]. (Chazelle's lemma only deals with the case $n = m$, but his proof generalizes immediately to the more general case.)

**Lemma 6.16 (Chazelle).** *If $A$ is an $n \times m$ incidence matrix with $I$ ones and no $p \times q$ minor of ones, then the complexity of computing $Ax$ over a semigroup is $\Omega(I/pq - n/p)$.*

**Theorem 6.17.** *Given $n$ weighted points and $m$ hyperplanes in $\mathbb{R}^d$,*

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

*semigroup operations are required to determine the sum of the weights of the points on each hyperplane, in the worst case.*

**Proof:** The lower bound follows immediately from Lemma 6.13. □

As in Theorem 6.14, we have combined the best lower bounds from several dimensions into a single expression. When $m = \Theta(n)$, this bound simplifies to $\Omega(n^{4/3})$, which already follows immediately from Chazelle's lemma and the Erdős construction. For all other values of $m$ between $\Omega(n^{1/d})$ and $O(n^d)$, however, the new bound is an improvement over any previously known lower bounds for this problem. The best known upper bound is given by Matoušek's algorithm [111].

## 6.3 Partitioning Algorithms

A *partition graph* is a directed acyclic graph, with one source, called the *root*, and several sinks, or *leaves*. Associated with each non-leaf node $v$ is a set $\mathcal{R}_v$ of *query regions*, satisfying three conditions.

(1) The cardinality of $\mathcal{R}_v$ is at most some constant $\Delta \geq 2$.

(2) Each region in $\mathcal{R}_v$ is connected.

(3) The union of the regions in $\mathcal{R}_v$ is $\mathbb{R}^d$.

(We do *not* require the query regions to be disjoint, convex, simply connected, semi-algebraic, or of constant combinatorial complexity.) In addition, every non-leaf note $v$ is either a *primal node* or a *dual node*, depending on whether its query regions $\mathcal{R}_v$ should be interpreted as a partition of primal or dual space. Each query region in $\mathcal{R}_v$ corresponds to an outgoing edge of $v$. Thus, the outdegree of the graph is at most $\Delta$.

Given sets $P$ of points and $H$ of hyperplanes as input, a *partitioning algorithm* constructs a partition graph, which can depend arbitrarily on the input, and uses it to drive the following divide-and-conquer process. The algorithm starts at the root and proceeds through the graph in topological order. At every node except the root, points and hyperplanes are passed in along incoming edges from preceding nodes. For each node $v$, let $P_v \subseteq P$ denote the points and $H_v \subseteq H$ the hyperplanes that reach $v$; at the root, we have $P_{\mathrm{root}} = P$ and $H_{\mathrm{root}} = H$. At every non-leaf node $v$, the algorithm partitions the sets $P_v$ and $H_v$ into (not necessarily disjoint) subsets by the query regions $\mathcal{R}_v$ and sends these subsets out along outgoing edges to succeeding nodes. If $v$ is a primal node, then for every query region $R \in \mathcal{R}_v$, the points in $P_v$ that are contained in $R$ and the hyperplanes in $H_v$ that intersect $R$ traverse the outgoing edge corresponding to $R$. If $v$ is a dual node, then for every query region $R \in \mathcal{R}_v$, the points $p \in P_v$ whose dual hyperplanes $p^*$ intersect $R$ and the hyperplanes $h \in H_v$ whose dual points $h^*$ are contained in $R$ traverse the corresponding outgoing edge. Note that a single point or hyperplane may enter or leave a node along several different edges.

For the purpose of proving lower bounds, the entire running time of the algorithm is given by charging unit time whenever a point or hyperplane traverses an edge. In particular, we do not charge for the construction of the partition graph or its query regions, nor for the time that would be required in practice to decide if a point or hyperplane intersects a query region. As a result, partitioning algorithms are effectively nondeterministic. In principle, the algorithm has "time" to compute the optimal partition graph for its input, and even very similar inputs might result in radically different partition graphs.

To solve Hopcroft's problem, the algorithm reports an incidence if and only if some leaf in the partition graph is reached by both a point and a hyperplane. It is easy to see that if some 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, for any set $P$ of points and 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 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 in light of Theorem 6.3 we must prove directly, our lower bounds are derived from the cover size bounds in Section 6.2. In the Section 6.4, we will describe how existing algorithms for Hopcroft's problem fit into our computational framework.

### 6.3.1 The Basic Lower Bound

**Theorem 6.18.** *Any partitioning algorithm that solves 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. (We implicitly used a one-dimensional version of this configuration to prove the lower bound in Theorem 6.3.) 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. Recall that the out-degree of every node in $G$ is at most $\Delta$. The *level* of any node in $G$ is the length of the shortest path from the root to that node. There are at most $\Delta^k$ nodes at level $k$. We say that a node $v$ *splits* a point-hyperplane pair if both the point and the hyperplane reach $v$, and none of the outgoing edges of $v$ is traversed by both the point and the hyperplane. In order for the algorithm to be correct, every point-hyperplane pair must be split. Finally, we say that a hyperplane $h$ is *active at level* $k$ if none of the nodes in the first $k$ levels split $h$ from any of its partners.

Suppose $v$ is a primal node. For each hyperplane $h$ that $v$ splits 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 $\Delta$ regions, at most $\Delta$ hyperplanes become inactive. By similar arguments, if $v$ is a dual node, then $v$ splits at most $\Delta$ points from their partners.

Thus, the number of hyperplanes that are inactive at level $k$ is less than $\Delta^{k+2}$. 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. We conclude that 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 $n < m$. □

### 6.3.2 The Lower Bound for the Decision Problem

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

**Theorem 6.19.** *Let $\mathcal{A}$ be a partitioning algorithm that solves 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 $T_{\mathcal{A}}(P, H)$ is defined in terms of the edges of the partition graph as follows.

$$T_{\mathcal{A}}(P, H) \triangleq \sum_{\text{edge } e} \Big(\#\text{ points traversing } e + \#\text{ hyperplanes traversing } e\Big)$$

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.) For every edge that a point or hyperplane traverses, there are at most $\Delta - 1$ edges that it misses.

$$\Delta \cdot T_{\mathcal{A}}(P, H) \geq \sum_{\text{edge } e} \big(\#\text{ points traversing } e + \#\text{ hyperplanes traversing } e$$
$$+ \#\text{ points missing } e + \#\text{ hyperplanes missing } e\big)$$

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

$$\Delta \cdot T_{\mathcal{A}}(P, H) \geq \sum_{\text{primal edge } e} \Big(\#\text{ points traversing } e + \#\text{ hyperplanes missing } e\Big)$$
$$+ \sum_{\text{dual edge } e} \Big(\#\text{ hyperplanes traversing } e + \#\text{ points missing } e\Big)$$

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 $R$. Since $R$ is connected, 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 solves Hopcroft's problem, no leaf is reached by both $p$ and $h$. It follows that some node $v$ splits $p$ 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, we can associate a simple minor with every edge in the partition graph, 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. Since the size of the resulting monochromatic cover must be at least $\mu(P, H)$, we conclude that $T_{\mathcal{A}}(P, H) \geq \mu(P, H)/2\Delta$. □

**Corollary 6.20.** *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 6.18 and 6.19 imply that the worst case running time is $\Omega(n \log m + \mu_d^{\circ}(n, m) + n \log m)$. Thus, Theorem 6.6 gives the planar lower bound, and Theorem 6.11 gives us the lower bound in higher dimensions. □

We emphasize that the condition $I(P, H) = 0$ is necessary for this lower bound to hold. If there is an incidence, then the trivial partitioning algorithm "detects" it. The partition graph consists of a single leaf, and since that leaf is reached by every point and every hyperplane, the algorithm correctly reports an incidence.

### 6.3.3 The Lower Bound for the Counting Problem

Every partitioning algorithm assumes that a point and hyperplane are incident if they reach the same leaf in its partition graph. Thus, the number of incidences associated with a leaf is the product of the number of points that reach it and the number of hyperplanes that reach it. To solve the counting version of Hopcroft's problem, a partitioning algorithm returns as its output the sum of these products over all leaves in its partition graph. In order for this output to be correct, the algorithm must ensure that every non-incident point-hyperplane pair is separated and that every incident pair reaches exactly one leaf. 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.

**Theorem 6.21.** *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 $T_{\mathcal{A}}(P, H) \geq \mu(P, H)/(2\Delta + 1)$. □

**Corollary 6.22.** *The worst-case running time of any partitioning algorithm that solves the counting version of Hopcroft's problem in* $\mathbb{R}^d$ *is*

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

See the remark after Theorem 6.14.

We can prove the following stronger lower 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*. Note that such an algorithm can solve the decision version of Hopcroft's problem in linear time.

**Theorem 6.23.** *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 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 6.24.** *The worst-case running time of any unbounded partitioning algorithm that solves the counting version of Hopcroft's problem in* $\mathbb{R}^d$ *is*

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

### 6.3.4 Containment Shortcuts Don't Help

We might consider adding the following containment shortcut to our model. Suppose that while partitioning points and hyperplanes at a primal node, the algorithm discovers that a query region $R$ is completely contained in some hyperplane $h$. Then we know immediately that any point contained in $R$ is incident to $h$. Rather than sending $h$ down the edge corresponding to $R$, the algorithm could increment a running counter for each

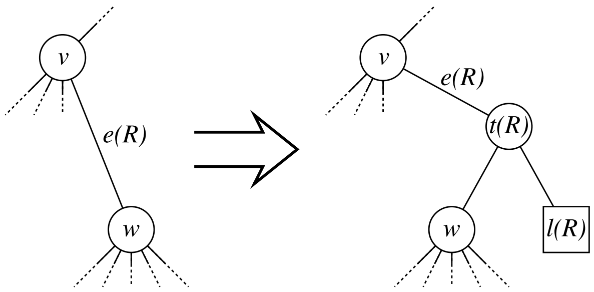


**Figure 6.6.** Eliminating containment shortcuts.

point in R. We can apply a symmetric shortcut at each dual node, potentially reducing the number of points traversing each dual edge. In addition to charging for edge traversals, we now also charge unit time whenever an algorithm discovers that a hyperplane (either primal or dual) contains a query region.

Clearly, adding this shortcut can only decrease the running time of any partitioning 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, as follows.

If a hyperplane $h$ contains a query region $R$, then it must also contain $\mathrm{aff}(R)$, the affine hull of $R$. We can reverse the containment relation by applying a duality transformation — the dual point $h^*$ is contained in the dual flat $(\mathrm{aff}(R))^*$. Similarly, if a point $p$ is contained in $R$, then $(\mathrm{aff}(R))^* \in p^*$.

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 $e(R)$ be the edge of the partition graph corresponding to $R$, and 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: $(\mathrm{aff}(R))^*$ and $\mathbb{R}^d \setminus (\mathrm{aff}(R))^*$, whose corresponding edges point to $\ell(R)$ and $w$, respectively. Finally, we redirect $e(R)$ so that it points to $t(R)$. See Figure 6.6. The new algorithm $\mathcal{A}'$ uses this modified partition graph, without explicitly checking for containments.

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

## 6.4 Real Algorithms for Hopcroft's Problem

Real algorithms for Hopcroft's problem all employ roughly the same divide-and-conquer strategy. Each algorithm divides space into a number of regions, determines which points and hyperplanes intersect each region, and recursively solves the resulting subproblems. In some cases [32, 66, 43], the number of regions used at each level of recursion is a constant, and these algorithms fit naturally into the partitioning algorithm framework.

For most algorithms, however, the number of regions is 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 partitioning algorithms.

In order to determine which points lie in which regions, each of these algorithms constructs a (possibly trivial) point location data structure. Each node in this data structure partitions space 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. Composing all the point location data structures used by the algorithm in all recursive subproblems gives us the algorithm's partition graph. Many of these algorithms alternate between primal and dual spaces at various levels of recursion [35, 111]. The data structures used in primal space give us the primal nodes in the partition graph, and the data structures used in dual space give us the dual nodes.

What about the hyperplanes? Many algorithms also use the point location data structures to determine the regions hit by each hyperplane. Algorithms of this type fit into our model perfectly. In particular, Matoušek's algorithm [111], which is based on Chazelle's hierarchical cuttings [35] and is the fastest algorithm known, can be modeled this way. Matoušek's algorithm and Theorem 6.21 immediately give us the following theorem.

**Theorem 6.25.** $\mu_d(n, m) = O\left(m \log n + n^{d/(d+1)} m^{d/(d+1)} 2^{O(\log^*(n+m))} + n \log m\right)$

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 [63, 1], the query regions form a decomposition of space into cells of constant complexity, typically simplices or trapezoids. The algorithms determine which cells a given hyperplane hits by iteratively "walking" through the cells. At each cell that the hyperplane intersects, the algorithm can determine in constant time which of the neighboring cells are also intersected, by checking each of the boundary facets.

In many cases, modifying such an algorithm to directly use the point location data structure instead of the iterative procedure increases the running time by only a constant factor. If the current point location 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 [135] in the plane, or the hierarchical cuttings data structure of Chazelle [35] in higher dimensions. The modified algorithm can then be described as a partitioning algorithm.

Other algorithms construct a point location data structure for the arrangement of the *entire* set of hyperplanes [51, 66, 35]. Usually, this is done only 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 structure 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 ad hoc. Modifying the partitioning model to *naturally* include algorithms that use different strategies for point and hyperplane location, or strengthening our lower bounds to a similar model that does not require constant-degree partitioning, is an interesting open problem.

Finally, a few algorithms partition the points or the hyperplanes *arbitrarily* into subsets, without using geometric information of any kind [62, p. 350],[51]. In this case, every hyperplane becomes part of every subproblem. In order to take algorithms of this kind into account, we must strengthen our model of computation by adding a new type of node that partitions either the points or the hyperplanes (but not both!) into arbitrary subsets at no cost. Lemmas 6.19 and 6.21 still hold in this stronger model, since the new nodes cannot separate any point from any hyperplane. However, Theorem 6.18 does not hold in this model; for example, we can solve Hopcroft's problem in $\mathbb{R}^d$ in time $O(n+m^{d+1})$ by "arbitrarily" partitioning the points so that each subset is contained in a single cell of the hyperplane arrangement.

## 6.5 Conclusions and Open Problems

We have proven new lower bounds on the complexity of Hopcroft's problem that apply to a broad class of geometric divide-and-conquer algorithms. Our lower bounds 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 of its input.

A number of open problems remain to be solved. The most obvious 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 the two-dimensional case. Is there a configuration of $n$ points and $n$ planes in $\mathbb{R}^3$ whose minimum monochromatic cover size is $\Omega(n^{3/2})$?

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? In the next chapter, we prove slightly better lower bounds for Hopcroft's problem in higher dimensions by restricting the query regions to convex polyhedra with constant complexity.

The class of partitioning algorithms is general enough to directly include many, but not all, existing algorithms for solving 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 an iterative procedure 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.

Our techniques imply lower bounds for several other problems similar to Hopcroft's problem [78]. Unfortunately, the partitioning algorithm model is specifically tailored to detect intersections or containments between pairs of objects, and there are a number of similar geometric problems for which the model simply does not apply. We mention one specific example, the *cyclic overlap problem*. Given a set of nonintersecting 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, Overmars, and Schwarzkopf [53], runs in time $O(n^{4/3+\varepsilon})$, 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 [75]. Apparently, however, this problem cannot 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 remains an important and very difficult open problem.

*"Here's to the new golden age of mathematics," Lord Vickers cried suddenly.*
*There was a chorus of approving remarks.*
*"That was the real thing!"*
*"Plenty of logic."*
*"And so many symbols!"*
*Lord Vickers smiled at me from across the room.*
*"There'll be a place for you at my new institute, Fletcher."*
*I took a glass of sherry.*

— Rudy Rucker, "A New Golden Age", 1981