The Exact Fitting Problem in Higher Dimensions

Jean-Marc Robert, Leonidas J. Guibas, Mark H. Overmars

RUU-CS-92-37
November 1992

Utrecht University

Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

The Exact Fitting Problem in Higher Dimensions

Jean-Marc Robert, Leonidas J. Guibas, Mark H. Overmars

Technical Report RUU-CS-92-37
November 1992

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN: 0924–3275

The Exact Fitting Problem in Higher Dimensions*

Jean-Marc Robert[†] Leonidas J. Guibas[‡] Mark H. Overmars[§]

Abstract

Let S be a family of n points in E^d. The *exact fitting* problem asks for finding a hyperplane containing the maximum number of points of S. In this paper, we present an $O\left(\min\left\{\frac{n^d}{m^{d-1}}\log\frac{n}{m}, n^d\right\}\right)$ time algorithm where m denoted the number of points the hyperplane. This algorithm is based on upper bounds on the maximum number of incidences between families of points and families of hyperplanes in E^d and on an algorithm to compute these incidences. We also show how the upper bound on the maximum number of incidences between families of points and families of hyperplanes can be used to derive new bounds on some well-known problems in discrete geometry.

1 Introduction

The problem of approximating families of points in E^d by hyperplanes is encountered in fields such as statistical analysis, computer vision, pattern recognition and computer graphics and it is usually referred to as the *linear approximation* or the *linear regression* problem. The problem consists of finding the "best" hyperplane approximating a family of points. There are many possibilities for the optimality criterion used. For example, a hyperplane minimizing the maximum orthogonal Euclidean distance to the points or minimizing the sum of these distances can be used. In [13], [24] and [25], algorithms solving these problems are presented.

In this paper, we consider a variation of this approximation problem: the *exact fitting* problem. This problem asks for finding a hyperplane containing the maximum number of points among a given family of n points in E^d. It can be solved easily

*Research of the third author was partially supported by the Dutch Organization for Scientific Research (N.W.O.) and by the ESPRIT Basic Research Action No. 7141 (project ALCOM II).

[†] Dép. d'informatique et de mathématique, Université du Québec à Chicoutimi, Chicoutimi, Canada.

[‡] Department of Computer Science, Stanford University, Stanford, and DEC Systems Research Center, Palo Alto, USA.

[§] Department of Computer Science, Utrecht University, Utrecht, the Netherlands.

in $O(n^d)$ time by transforming the points into hyperplanes in the dual space [14]. A solution to the exact fitting problem corresponds to a vertex of the hyperplane arrangement incident to the maximum number of hyperplanes. By sweeping every two-dimensional "slice" of the hyperplane arrangement, it is possible to find such a vertex in $O(n^d)$ time as follows: Take any $d-2$ hyperplanes and compute their common intersection π. If π corresponds to a plane, the topological line sweep algorithm of Edelsbrunner and Guibas [15] is used to sweep the line arrangement formed by the intersection of the remaining hyperplanes with π. If π has a dimension greater than two, the corresponding slice of the hyperplane arrangement is simply discarded. This construction has been used in [23] and [2] to solve some other problems. This is the best algorithm known so far to determine whether there exist $(d+1)$ points lying on a common hyperplane; proving its optimality is a well known open problem. On the other hand, it is quite simple to determine in linear time whether there is a hyperplane containing "almost" all the points, i.e., $n-c$ points for any fixed c. Take $c+1$ groups of d linearly independent points. One of these groups should determine the hyperplane containing the $n-c$ points.

We present in this paper a quality-sensitive algorithm solving the exact fitting problem. The algorithm find a solution fast if it contains many points. More precisely, the running time of our algorithm depends inversely on the number of points lying on an optimal hyperplane. Hence, the running time varies from $O(n)$ when a fixed fraction of the points lie on the solution to $O(n^d)$ when a fixed number of the points lie on the solution. This algorithm is a generalization of the algorithm presented in [22] for the planar case.

In the next section, we present upper bounds on the maximum number of incidences between families of points and families of hyperplanes in $\boldsymbol{E}^d$. These results are only sensible when we restrict our attention to so-called restricted sets where any d points span a hyperplane or the intersection of any d hyperplanes has dimension at most 0. We also present an algorithm to compute these incidences. These two results are essential for developing our algorithm solving the exact fitting problem.

In Section 3, we give an algorithm to find all the hyperplanes containing at least m points among a restricted family of n points in $\boldsymbol{E}^d$ in $O\left(\min\left\{\frac{n^d}{m^{d-1}}\log\frac{n}{m}, n^d\right\}\right)$ time. Here the parameter m is given as part of the input. This solution gives an optimal linear time algorithm when m represents a fixed fraction of the points, i.e., $m=\epsilon n$ for some constant $0<\epsilon<1$. This algorithm is then used to solve the exact fitting problem. A hyperplane containing the maximum number of points of the family can be found in $O\left(\min\left\{\frac{n^d}{m^{d-1}}\log\frac{n}{m}, n^d\right\}\right)$ time, where m is this maximal number of points. In this case, only the family of n points is given as input, i.e., m does not need to be known in advance.

Finally, in Section 4, we derive new bounds on some well-known problems in discrete geometry. By using the upper bounds on the maximum number of incidences between families of points and families of hyperplanes, we obtain a new upper bound on the maximum number of pairs of points at unit-distance among S-restricted families of points in $\boldsymbol{E}^d$, i.e., sets in which no $d+1$ points lie on a $(d-1)$-dimensional hypersphere. We also obtain a lower bound on the minimum number of different distances between the points of S-restricted families of points in $\boldsymbol{E}^d$. Finally, we give a new upper bound on the maximum number of furthest neighbors pairs among S-restricted families of points in $\boldsymbol{E}^d$.

2 Incidences between Points and Hyperplanes

Let P be a family of distinct points and let H be a family of distinct hyperplanes in $\boldsymbol{E}^d$. A point p in P is *incident* to a hyperplane h in H if p lies on h. The first problem considered in this section is to find an upper bound on the maximum number of incidences between families of points and families of hyperplanes.

In the plane, tight upper bounds on the maximum number of incidences between families of points and families of lines have been derived. These results are summarized in the following theorem.

Theorem 2.1 ([31], [12]) *Let $I_2(x,y)$ be the maximum number of incidences between x distinct points and y distinct lines in the plane. Then, $I_2(x,y)$ is in $\Theta(x^{\frac{2}{3}}y^{\frac{2}{3}}+x+y)$*[1].

In higher dimensions, the maximum number of incidences between x distinct points and y distinct hyperplanes matches the trivial xy upper bound. Consider a line l in $\boldsymbol{E}^3$ containing x distinct points and lying on y distinct planes. In this case, the number of incidences is exactly xy. This example can be generalized to higher dimensions. Let f be a $(d-2)$-dimensional flat in $\boldsymbol{E}^d$ containing x distinct points and lying on y distinct hyperplanes. Even if any $k+1$ points span a k-dimensional flat, for $0 \leq k \leq d-3$, the number of incidences is still xy . To avoid these trivial cases we define a family of points to be *restricted* iff any d points span a hyperplane. Similar, we define a family of hyperplanes to be *restricted* iff the intersection of any d of them has dimension at most 0. For restricted families of points and hyperplanes we prove upper bounds strictly smaller than the trivial one. Note that restricted does not mean "in general position". A restricted family of points is not necessarily in general position. For example, a set of points in $\boldsymbol{E}^3$ on a plane can be a restricted family.

A $(\frac{1}{r})$-cutting of size k for a family H of hyperplanes in $\boldsymbol{E}^d$ is a collection of k (possibly unbounded) d-dimensional simplices with disjoint interiors covering $\boldsymbol{E}^d$. Furthermore, the interior of each simplex is intersected by at most $O(\frac{|H|}{r})$ hyperplanes. Chazelle and Friedman [8] proved that any family H of hyperplanes has a $(\frac{1}{r})$-cutting of size $O(r^d)$, for any r larger than some constant r_d depending on d. This $(\frac{1}{r})$-cutting is constructed by triangulating the arrangement of $O(r)$ specific hyperplanes of H according to some particular criterion [11].

[1] Throughout this paper, we use the asymptotic notations based on sets (see [6]).

By using such a concept, the following upper bound on the number of incidences between families of points and restricted families of hyperplanes in $\boldsymbol{E}^d$ can be derived.

Theorem 2.2 *Let $I_d(x,y)$ be the maximum number of incidences between x distinct points and a restricted family of y hyperplanes in $\boldsymbol{E}^d$ Then, $I_d(x,y)$ is in $O(x^{\frac{2d-2}{2d-1}}y^{\frac{d}{2d-1}}+x+y)$.*

Proof Let P be any family of x distinct points and let H be any restricted family of y hyperplanes in $\boldsymbol{E}^d$. Let $I_d(P;H)$ denote the number of incidences between the points in P and the hyperplanes in H. The incidences between the points in P and the hyperplanes in H can be encoded with a directed bipartite graph $G_{x,y}=(V_1,V_2,E)$ with x sources and y sinks. Each source in V_1 corresponds to a point in P and each sink in V_2 corresponds to a hyperplane in H. The family of edges E encodes the incidences between the points and the hyperplanes. Since the intersection of any d hyperplanes has dimension at most 0, $G_{x,y}$ does not contain any $K_{2,d}$ subgraph. In such a case, Kövári, Sós and Turán [26] gave two different upper bounds on the cardinality of E. The first one implies that $I_d(P;H)\in O(xy^{\frac{1}{2}}+y)$ and the second one implies that $I_d(P;H)\in O(yx^{\frac{d-1}{d}}+x)$.

Let C be a $(\frac{1}{r})$-cutting of size $O(r^d)$ of H. Let $P_i\subset P$ denote the family of points lying in $int(\Delta_i)$ (i.e. the interior of the simplex Δ_i of C) and let $H_i\subset H$ denote the family of hyperplanes intersecting $int(\Delta_i)$.

Let $p\in P$ be a point lying on a face F of some simplex Δ_i of C. Suppose that the dimension of F is greater than 0. By adding p to P_i, the incidences between p and the hyperplanes intersecting $int(\Delta_i)$ are counted in $I_d(P_i;H_i)$. Thus, only the incidences between p and the hyperplanes supporting Δ_i and containing F are not counted. Since the intersection of any d hyperplanes has dimension at most 0, the number of hyperplanes supporting Δ_i and containing F is at most $d-dim(F)$. Hence, the number of incidences between p and the hyperplanes in H not considered in the subproblems $I_d(P_i;H_i)$ is at most d.

Now, suppose that the dimension of F is equal to 0. In this case, the number of hyperplanes containing p but avoiding $int(\Delta_i)$ is unbounded. Let v be the number of points lying on vertices of the simplices of C. Obviously, $v\in O(\min\{x,r^d\})$. The maximum number of incidences between these v points and all the hyperplanes in H is in $O(yv^{\frac{d-1}{d}}+v)\subset O(y(r^d)^{\frac{d-1}{d}}+x)$.

Therefore, the number of incidences between the points in P and the hyperplanes in H is given by

$$\begin{aligned} I_d(P;H) &\leq \sum_{\Delta_i\in C} I_d(P_i;H_i)+O(y(r^d)^{\frac{d-1}{d}}+x)+dx \\ &\in O\left(x\left(\frac{y}{r}\right)^{\frac{1}{2}}+yr^{d-1}+x\right). \end{aligned}$$

This follows from the fact that $I_d(P_i;H_i) \in O(|P_i||H_i|^{\frac{1}{2}} + |H_i|)$ and the fact that $|H_i| \in O(\frac{y}{r})$ and $\sum |P_i| = x$. By choosing $r = \lfloor x^{\frac{2}{2d-1}}/y^{\frac{1}{2d-1}} \rfloor$, we obtain that $I_d(P;H) \in O(x^{\frac{2d-2}{2d-1}} y^{\frac{d}{2d-1}} + x)$. The choice of r is valid if and only if $\sqrt{r_d^{2d-1} y} \leq x \leq y^d$. Otherwise, the upper bounds of [26] can be used. If $x > y^d$, $I_d(P;H) \in O(x)$ and if $x < \sqrt{r_d^{2d-1} y}$, $I_d(P;H) \in O(y)$. Therefore, the theorem can be proven by combining all the three cases. □

Using the dual transformation between points and hyperplanes [14], the following upper bound on the number of incidences between restricted families of points and families of hyperplanes in $\boldsymbol{E}^d$ can be derived directly from Theorem 2.2.

Corollary 2.3 *Let $I_d^*(x,y)$ be the maximum number of incidences between x distinct hyperplanes and a restricted family of y points in $\boldsymbol{E}^d$. Then, $I_d^*(x,y)$ is in $O(x^{\frac{2d-2}{2d-1}} y^{\frac{d}{2d-1}} + x + y)$.*

This corollary gives an $O(x^{\frac{2}{3}} y^{\frac{2}{3}} + x + y)$ upper bound for $I_2^*(x,y)$ which matches the one given in Theorem 2.1 and an $O(x^{\frac{4}{5}} y^{\frac{3}{5}} + x + y)$ upper bound for $I_3^*(x,y)$ which is better than the $O(x^{4/5+2\delta} y^{3/5-\delta} + y + x \log y)$ upper bound presented in [16].

We now consider the problem of finding an efficient algorithm to compute the incidences between families of points and families of hyperplanes. In the plane, algorithms computing the incidences between families of points and families of lines have been presented. These results are summarized in the following theorem.

Theorem 2.4 ([1], [27]) *The incidences between x distinct points and y distinct lines in the plane can be computed in $O((xy \log y)^{\frac{2}{3}} + (x+y)\log y)$ time.*

We now present an $O((xy \log \min\{x,y\})^{\frac{d}{d+1}} + (x+y) \log \min\{x,y\})$ time algorithm computing the incidences between families of x points and families of y hyperplanes in $\boldsymbol{E}^d$ where one is a restricted family. This algorithm is an extension of the algorithm developed by Agarwal [1] and it is based on the following results presented by Chazelle [7]:

- A $(\frac{1}{r})$-cutting of size $O(r^d)$ for a family of n hyperplanes in $\boldsymbol{E}^d$ can be computed in $O(nr^{d-1})$ time, for any $r_d \leq r \leq n$;
- A family of n hyperplanes in $\boldsymbol{E}^d$ can be preprocessed in $O(n^d)$ time to allow for point-location in $O(\log n)$ time per query.

Let P be a family of x points and H be a family of y hyperplanes in $\boldsymbol{E}^d$. Let us first consider the case where $x \geq \frac{y^d}{\log y}$. Preprocess the hyperplanes in H in $O(y^d)$ time for point-location. By locating the face of the arrangement containing a given point p in $O(\log y)$ time and by enumerating the t hyperplanes incident to that face, the hyperplanes incident to p can be computed in $O(\log y + t)$ time. If the arrangement is represented with the incidence lattice (see [14]), it is possible to enumerate all the hyperplanes incident to a face of the arrangement in a time proportional to the number of hyperplanes incident to that face. Hence, the incidences can be computed in $O(y^d + x\log y + \#\mathit{incidences})$ time. The upper bounds of [26] can be used to prove that $\#\mathit{incidences}$ is in $O(x^{\frac{d-1}{d}}y + x)$ when one of the families is restricted. Therefore, the incidences can be computed in $O(x\log y)$ time.

Now, suppose that $x < \frac{y^d}{\log y}$ and $y < \frac{x^d}{\log x}$. Split the y hyperplanes into g groups of $\lceil \frac{y}{g} \rceil$ elements. For each subproblem, preprocess the hyperplanes in $O((\frac{y}{g})^d)$ time for point-location and find the incidences between the points in P and these hyperplanes. The t hyperplanes incident to a given point can be determined in $O(\log\frac{y}{g} + t)$ time. Thus, the incidences can be computed in $O(g\{(\frac{y}{g})^d + x\log\frac{y}{g}\} + \#\mathit{incidences})$ time. Let g be equal to $\lceil y/(x\log y)^{\frac{1}{d}}\rceil$ (since $x < \frac{y^d}{\log y}$, $g > 1$). Hence, the incidences can be computed in $O(y(x\log y)^{\frac{d-1}{d}} + \#\mathit{incidences})$ time. As we observed earlier, $\#\mathit{incidences}$ is in $O(x^{\frac{d-1}{d}}y + x)$ when one of the families is restricted. Therefore, the incidences can be compute in $O(y(x\log y)^{\frac{d-1}{d}})$ time.

Finally, suppose that $y \geq \frac{x^d}{\log x}$. By reformulating the problem in the dual space, this case corresponds to the first case. Thus, the incidences can be computed in $O(y\log x)$ time.

By combining the three cases, we obtain Algorithm Inc_1, computing the incidences between x points and y hyperplanes in E^d in $O(y(x\log y)^{\frac{d-1}{d}} + x\log y)$ time.

This algorithm can be used as a subroutine to obtain a better solution. Without loss of generality, suppose that the number of points x is smaller than or equal to the number of hyperplanes y. Otherwise, simply reformulate the problem in the dual space. If x is very small compare to y (i.e. if $c\frac{x^d}{\log x} < y$, for any constant c), the point-location technique can be used to compute the incidences in $O(y\log x)$ time. From now on, suppose that $x \leq y \leq \frac{x^d}{\log x}$ (notice that $\log y \in \Theta(\log x)$ in this case). By computing a $(\frac{1}{r})$-cutting C of H, the problem of computing the incidences between the points and the hyperplanes is divided into $O(r^d)$ subproblems: the subproblem S_i associated to the simplex Δ_i of C is composed of the points P_i lying in $int(\Delta_i)$ and the hyperplanes H_i intersecting $int(\Delta_i)$. The points lying on the faces of the simplices are treated differently depending on whether the families of points or the families of hyperplanes are restricted. Let p be a point of S lying in the interior of a face F of some simplex Δ_i. Suppose that any d points in P span a hyperplane. If F is a facet, p is added to P_i and the hyperplanes in H containing the facets of Δ_i are added to H_i. Thus, any hyperplane incident to p is in H_i. Note that at most $(d+1)$ hyperplanes are added to H_i. If F is not a facet, p is put in a family B. Since any d points span a hyperplane, there are at most $(k+1)$ points in P which can lie on a k-dimensional face of a simplex, for $0 \leq k < (d-1)$. Since a d-dimensional simplex has $\binom{d+1}{k+1}$ k-dimensional faces [21], at most $\sum_{k=0}^{d-2}(k+1)\binom{d+1}{k+1}$ points are put in B for each simplex. This implies that $|B|$ is in $O(\min\{x, r^d\})$. Now, suppose that the intersection of any d hyperplanes in H has dimension at most 0. If F is a vertex,

p is put in B. Here again, $|B|$ is in $O(\min\{x, r^d\})$. If F is not a vertex, p is added to P_i and the hyperplanes supporting Δ_i and containing F are added to H_i. Thus, any hyperplane incident to p is in H_i. Since the intersection of any d hyperplanes has dimension at most 0, there are at most $(d-k)$ hyperplanes in H which can support Δ_i and contain F, for $0 < k \leq (d-1)$. Hence, at most $\sum_{k=1}^{d-1}(d-k)\binom{d+1}{k+1}$ hyperplanes are added to H_i.

Therefore, the incidences between the points in P and the hyperplanes in H are given by the incidences between the points in P_i and the hyperplanes in H_i *plus* the incidences between the points in B and all the hyperplanes in H.

The time complexity of this algorithm is determined by the time to construct the subproblems plus the time to solve them. The $(\frac{1}{r})$-cutting C of H can be computed in $O(yr^{d-1})$ time (see [7]). During the construction of the cutting, the hyperplanes are distributed among the simplices. To divide the points among the subproblems, we preprocess the $O(r)$ hyperplanes used to construct C for point-location. This can be done in $O(r^d) \subset O(yr^{d-1})$ time. For each point $p \in P$, locate a cell of the arrangement containing p and then find a simplex in that cell containing p. This can be done in $O(\log r + r^{\lfloor d/2 \rfloor})$ time. The $O(\log r)$-term comes from the point-location query for finding a cell of the arrangement containing p and the $O(r^{\lfloor d/2 \rfloor})$-term comes from the fact that each cell of the arrangement is divided into at most $O(r^{\lfloor d/2 \rfloor})$ simplices. Thus, the overall time needed to distribute the points in P among the subproblems is in $O(x \log r + xr^{\lfloor d/2 \rfloor}) \subset O(x \log y + xr^{d-1})$. Each subproblem S_i is reformulated in the dual space and the incidences between the dual points and the dual hyperplanes are computed with Algorithm Inc_1. Therefore, the subproblem S_i consists of computing the incidences between $O(\frac{y}{r})$ points and x_i hyperplanes. Note that $\sum x_i = x$. Finally, the subproblem consisting of computing the incidences between the v points in B and all the hyperplanes in H can also be solved by using Algorithm Inc_1. Hence,

$$\begin{aligned} T^{(d)}_{Inc_2}(x,y) &\leq \sum_{\Delta_i \in C} T^{(d)}_{Inc_1}(y_i, x_i) + O(yr^{d-1} + x \log y + xr^{d-1}) + T^{(d)}_{Inc_1}(v, y) \\ &\in O\left(x(\frac{y}{r} \log x)^{\frac{d-1}{d}} + r^{d-1}(y \log x) + x \log x\right). \end{aligned}$$

Let r be equal to $\lceil \frac{x^{\frac{d}{d^2-1}}}{(y \log x)^{\frac{1}{d^2-1}}} \rceil$. Since $y < c\frac{x^d}{\log x}$, r is large enough to be sure that a $(\frac{1}{r})$-cutting for the hyperplanes exists. By combining this algorithm and the algorithm based on the point-location when $c\frac{x^d}{\log x} < y$, we obtain Algorithm Inc_2 computing the incidences between the points in P and the hyperplanes in H. The following theorem summarizes the result.

Theorem 2.5 *The incidences between x distinct points and y distinct hyperplanes in $\mathbf{E}^d$, where one of the families is restricted can be computed in $O((xy \log \min\{x,y\})^{\frac{d}{d+1}} + (x+y)\log \min\{x,y\})$ time.*

E.g., assuming $x = y = n$ the result states that for $d = 2$ the incidences can be computed in time $O(n^{4/3} \log^{2/3} n)$ and for $d = 3$ in time $O(n^{3/2} \log^{3/4} n)$. Also, for any dimension the time bound is strictly smaller than $O(n^2)$.

3 The Exact Fitting Algorithm

In this section, we give a quality-sensitive algorithm solving the exact fitting problem for restricted families of points in $\mathbf{E}^d$. Let S be a such family of n points, i.e., any d points in S span a hyperplane. We first show how to find all the hyperplanes containing at least m of the n points in S. Later, this solution will be used to find a hyperplane containing the maximum number of points in S. The following upper bound on the maximum number of distinct hyperplanes containing at least m of the n points in S is derived from Corollary 2.3.

Lemma 3.1 *Let S be a restricted family of n points in $\mathbf{E}^d$ and let $N(S,m)$ be the maximum number of distinct hyperplanes containing at least m of the n points in S. Then, $N(S,m)$ is in $O\left(\max\left\{\frac{n^d}{m^{2d-1}}, \frac{n}{m}\right\}\right)$.*

Proof The number of incidences between the $N(S,m)$ hyperplanes and the n points in S is at least $mN(S,m)$. By Corollary 2.3, the maximum number of incidences between $N(S,m)$ hyperplanes and n points is in $O(n^{\frac{d}{2d-1}} N(S,m)^{\frac{2d-2}{2d-1}} + n + N(S,m))$. By combining these two facts, the result follows immediately. □

We are now ready to present our first algorithm to find all the hyperplanes containing at least m of the points n in S.

Algorithm MinN1

```
Input:  Family $S$ of $n$ points in $\mathbf{E}^d$ such that any $d$ points span a hyperplane and
        an integer $m$ such that $d \leq m \leq n$.
Output: All the hyperplanes containing at least $m$ points.

1. If $m \leq 2d$,
    (a) Dualize the $n$ points.
    (b) For each set of $d - 2$ dual hyperplanes do:
        i. Let $\pi$ be the intersection of these $d - 2$ hyperplanes. If $\pi$ is not a
           plane, go to the next iteration of Step 1b.
        ii. Sweep the line arrangement determined by the intersection of the
            remaining hyperplanes with $\pi$ and output all the vertices incident to
            at least $m$ lines.
2. Otherwise,
    (a) Split $S$ into the two subsets of $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$ points, respectively.
    (b) For each subset, find all the hyperplanes containing at least $\lfloor \frac{m}{2} \rfloor$ points.
```

(c) For each candidate found in Step 2b, determine how many points lie on it.

(d) Output all the candidates containing at least m points.

End of the Algorithm

The first step of this algorithm can be done in $O(n^d)$ time. Each of the $\binom{n}{d-2}$ two-dimensional slices of the hyperplane arrangement can be processed in $O(n^2)$ time with the topological sweep line algorithm [15]. Then, the running time of Algorithm MinN1 can be expressed by the following recurrence

$$\begin{aligned} T_1^{(d)}(n,m) &\in O(n^d) \quad \text{if } m \leq 2d, \\ T_1^{(d)}(n,m) &\leq T_1^{(d)}\left(\left\lfloor \frac{n}{2} \right\rfloor, \left\lfloor \frac{m}{2} \right\rfloor\right) + T_1^{(d)}\left(\left\lceil \frac{n}{2} \right\rceil, \left\lfloor \frac{m}{2} \right\rfloor\right) + V_1^{(d)}(n,m) \quad \text{otherwise.} \end{aligned}$$

The function $V_1^{(d)}(n,m)$ corresponds to the time taken by Step 2c and depends on the number of candidates found in Step 2b and how fast they can be checked with Algorithm Inc_2 developed in the previous section. Let $\#cand(n,m)$ be the total number of candidates. By Lemma 3.1, $\#cand(n,m)$ is in $O\left(\max\left\{\frac{n^d}{m^{2d-1}}, \frac{n}{m}\right\}\right)$. Hence,

$$V_1^{(d)}(n,m) \in O\left(\left(\max\left\{\frac{n^d}{m^{2d-1}}, \frac{n}{m}\right\} n \log \frac{n}{m}\right)^{\frac{d}{d+1}} + \left(\max\left\{\frac{n^d}{m^{2d-1}}, \frac{n}{m}\right\} + n\right) \log \frac{n}{m}\right).$$

To simplify the analysis of the recurrence, suppose that $n = 2^s$ and $m = 2^t$ and let $t_1^{(d)}(s,t)$ denote the value $T_1^{(d)}(2^s, 2^t)$. If $2t \leq s$, $\#cand(n,m)$ is in $O(\frac{2^{ds}}{2^{(2d-1)t}})$ and the recurrence becomes

$$\begin{aligned} t_1^{(d)}(s,1) &\in O(2^{ds}), \\ t_1^{(d)}(s,t) &\leq 2t_1^{(d)}(s-1,t-1) + \\ &\quad O\left(\left(\frac{2^{ds}}{2^{(2d-1)t}} 2^s (s-t)\right)^{\frac{d}{d+1}} + \left(\frac{2^{ds}}{2^{(2d-1)t}} + 2^s\right)(s-t)\right) \\ &\leq 2^k t_1^{(d)}(s-k, t-k) + \\ &\quad O\left(\sum_{i=0}^{k-1} 2^i \left[\frac{2^{d(s-i)}}{2^{\frac{2d^2-d}{d+1}(t-i)}} (s-t)^{\frac{d}{d+1}} + \left(\frac{2^{d(s-i)}}{2^{(2d-1)(t-i)}} + 2^{s-i}\right)(s-t)\right]\right). \end{aligned}$$

By solving this recurrence, $t_1^{(d)}(s,t)$ is in $O(\frac{2^{ds}}{2^{(d-1)t}}(s-t))$. If $2t \geq s$, $\#cand(n,m)$ is in $O(\frac{2^s}{2^t})$ and the recurrence becomes

$$\begin{aligned} t_1^{(d)}(s,t) &\leq 2t_1^{(d)}(s-1,t-1) + O((2^{2s-t}(s-t))^{\frac{d}{d+1}} + 2^s(s-t)) \\ &\leq 2^k t_1^{(d)}(s-k, t-k) + \\ &\quad O\left(\sum_{i=0}^{k-1} 2^i \left[\left(\frac{2^{2(s-i)}}{2^{t-i}}(s-t)\right)^{\frac{d}{d+1}} + 2^{s-i}(s-t)\right]\right). \end{aligned}$$

In this case, the recurrence can be applied as long as $2(t-k) \geq (s-k)$. After $2t-s$ iterations, the value of $t_1^{(d)}(2s-2t, s-t)$ is given by the first part of the analysis. Hence, $t_1^{(d)}(s,t)$ is in $O(\frac{2^{ds}}{2^{(d-1)t}}(s-t)+2^s(s-t)(2t-s)) \subseteq O(\frac{2^{ds}}{2^{(d-1)t}}(s-t)+2^s(s-t)s)$. A solution for the case where m and n are not power of two can also be obtained. For large enough n, the maximum number of candidates $\#cand(n,m)$ is non-decreasing in n and non-increasing in m and the time complexity of Algorithm Inc_2, $T_{Inc_2}^{(d)}(x,y)$, is non-decreasing in both x and y. Hence, $T_1^{(d)}(n,m)$ is non-decreasing in n and non-increasing in m and the general solution is given by

$$T_1^{(d)}(n,m) \in O\left(\frac{n^d}{m^{d-1}} \log \frac{n}{m} + n \log \frac{n}{m} \log n\right).$$

We refer to [6] and [5] for a good treatment of the conditional asymptotic notations. Thus, $T_1^{(d)}$ is in $O\left(\frac{n^d}{m^{d-1}} \log \frac{n}{m}\right)$ when m is smaller than or equal to $\frac{n}{\log^{\frac{1}{d-1}} n}$. We now present another algorithm to deal with the case where m is "close" to n, i.e., when m is larger than $\frac{n}{\log^{\frac{1}{d-1}} n}$.

Algorithm MinN2

Input: Family S of n points in E^d such that any d points span a hyperplane and an integer m such that $d \leq m \leq n$.

Output: All the hyperplanes containing at least m points.

1. Set k to be equal to $\max\{\lfloor \frac{n}{m} \rfloor, d\}$.
2. Split S into $\lceil \frac{n}{k^2} \rceil$ subsets of k^2 points (one subset may have less than k^2 points).
3. For each subset, find all the distinct hyperplanes containing at least k points with Algorithm MinN1.
4. Candidates must appear as answer for at least $\frac{n}{k^4}$ subsets. Determine those hyperplanes.
5. Output all the candidates found in Step 4 containing at least m points of the total set.

End of the Algorithm

The time complexity of this algorithm is given by the following expression

$$T_2^{(d)}(n,m) \leq \left\lceil \frac{n}{k^2} \right\rceil T_1^{(d)}(k^2,k) + V_2^{(d)}(n,m).$$

The function $V_2^{(d)}(n,m)$ corresponds to the time taken by Step 5 and depends on the number of candidates found in Step 4 and how fast they can be checked with Algorithm Inc_2.

Suppose that k is equal to $\lfloor \frac{n}{m} \rfloor$. Thus,

$$\begin{aligned} T_2^{(d)}(n,m) &\leq \left\lceil \frac{n}{\left\lfloor \frac{n}{m} \right\rfloor^2} \right\rceil T_1^{(d)}\left(\left\lfloor \frac{n}{m} \right\rfloor^2, \left\lfloor \frac{n}{m} \right\rfloor \right) + V_2^{(d)}(n,m) \\ &\leq O\left(\frac{n^d}{m^{d-1}} \log \frac{n}{m} \right) + V_2^{(d)}(n,m). \end{aligned}$$

In order to obtain an upper bound on $V_2^{(d)}(n,m)$, first suppose that $n = 2^s$ and $m = 2^t$ and let $\#cand(2^s, 2^t)$ be the maximum number of candidates found in Step 4. By Lemma 3.1, the number of distinct hyperplanes containing at least 2^{s-t} of 2^{2s-2t} points is in $O(2^{s-t})$. Hence, the total number of hyperplanes found in Step 3 is in $O(2^t)$. Now, suppose there are 2^t points lying on a hyperplane h. This hyperplane is defined by the points of at least 2^{4t-3s} subsets. To obtain this lower bound, find the "best" distribution of the 2^t points lying on h among the subsets defined in Step 2 to minimize the number of subsets having at least 2^{s-t} of these points. This distribution is obtained by putting $2^{s-t} - 1$ of the points in each subset and by packing the remaining 2^{2t-s} points in the fewest number of subsets. Hence, the hyperplane h is defined at least $\lfloor 2^{2t-s}/(2^{2s-2t} - 2^{s-t} + 1) \rfloor \geq 2^{4t-3s}$ times implying that $\#cand(2^s, 2^t)$ is in $O(2^{3s-3t})$.

The $O(2^{3s-3t})$ candidates can be determined efficiently by adapting an algorithm finding repeated elements in a multiset. In [28], Misra and Gries showed how to find the $O(c)$ values that occur more than n/c times in a multiset of n elements in $O(n \log c)$ time. Hence, the hyperplanes appearing at least 2^{4t-3s} times among the $O(2^t)$ hyperplanes can be found in $O(2^t(s-t))$ time and can be checked in $O((2^{4s-3t}(s-t))^{\frac{d}{d+1}} + 2^s(s-t))$ time with Algorithm Inc_2. Thus, $V_2^{(d)}(2^s, 2^t)$ is in $O((2^{4s-3t}(s-t))^{\frac{d}{d+1}} + 2^s(s-t))$ implying that $T_2(2^s, 2^t)$ is in $O(2^{2s-t}(s-t))$. A similar result can be obtained when m or n are not powers of two. For large enough values the number of candidates, $\#cand(n,m)$, is non-decreasing in n and non-increasing in m and the time complexity of Algorithm Inc_2, $T_{Inc_2}^{(d)}(x,y)$, is non-decreasing in both x and y. Hence, $V_2^{(d)}(n,m)$ is non-decreasing in n and non-increasing in m which implies that $T_2^{(d)}(n,m) \in O\left(\frac{n^d}{m^{d-1}} \log \frac{n}{m} \right)$.

The case where k is equal to d can be treated similarly. In this case, the candidates correspond to the hyperplanes appearing at least $\frac{n}{d^4}$ time among the list of $O(n)$ hyperplanes produced in Step 3 in $O(n)$ time. These $O(1)$ candidates can be found with the algorithm presented in [28] and checked with the naive brute-force algorithm in $O(n)$ time. Thus, $T_2^{(d)}(n,m) \in O\left(\frac{n^d}{m^{d-1}} \log \frac{n}{m} \right)$.

By choosing the brute-force algorithm based on the topological line sweep algorithm when $m \leq \log n$, Algorithm MinN1 when $m \leq \frac{n}{\log^{\frac{1}{d-1}} n}$ and Algorithm MinN2 otherwise, we obtain the following result

Theorem 3.2 *Let S be a restricted family of n points in E^d. It is possible to determine if there are m points in S lying on a hyperplane in $O\left(\min\left\{\frac{n^d}{m^{d-1}}\log\frac{n}{m}, n^d\right\}\right)$ time.*

We are now ready to present our algorithm to solve the exact fitting problem for restricted families of n points in E^d.

Algorithm EF

Input: Family S of n points in E^d such that any d points span a hyperplane.
Output: A hyperplane containing the maximum number of points.

1. Set i to 1.
2. Find all the hyperplanes containing at least $\frac{n}{2^i}$ points of S.
3. If there is no such hyperplane, increase i by 1 and go to Step 2.
4. If there are such hyperplanes, check all of them and output the hyperplane containing the maximum number of points.

End of the Algorithm

Suppose there are m points in S lying on a hyperplane. Algorithm EF stops when $\frac{n}{2^i}$ is smaller than or equal to m i.e. after $\lceil\log\frac{n}{m}\rceil$ iterations. Hence, the time complexity of this algorithm is given by

$$T_{EF}^{(d)}(n,m) \in O\left(\sum_{i=1}^{\lceil log\frac{n}{m}\rceil} \min\left\{\frac{n^d}{\left\lfloor\frac{n}{2^i}\right\rfloor^{d-1}}\log\frac{n}{\left\lfloor\frac{n}{2^i}\right\rfloor}, n^d\right\}\right).$$

The following corollary summarizes the result

Corollary 3.3 *Let m be the maximum number of points lying on a hyperplane in a restricted family S of n points in E^d. Algorithm EF determines these points in $O\left(\min\left\{\frac{n^d}{m^{d-1}}\log\frac{n}{m}, n^d\right\}\right)$ time.*

4 Unit-Distance and Other Problems

We complete this paper by indicating how to use the combinatorial upper bounds derived in Section 2 to obtain new bounds for some well-known problems in discrete geometry. We start by giving an upper bound on the maximum number of incidences between so-called S-restricted families of points and families of hyperspheres in E^d. A d-dimensional hypersphere in E^d, or more simply a hypersphere, centered at the point c and with a radius r is the locus of points at distance r of c. A $(d-1)$-dimensional hypersphere in E^d corresponds to the intersection of two hyperspheres. By convention, a $(d-2)$-flat represents a degenerate $(d-1)$-dimensional hypersphere in $\boldsymbol{E}^d$. We call a family of points in $\boldsymbol{E}^d$ *S-restricted* iff no $d+1$ points lie on a $(d-1)$-dimensional hypersphere. Let $\mathcal{U}$ be the geometric transformation mapping points in $\boldsymbol{E}^d$ to points on the paraboloid in $\boldsymbol{E}^{d+1}$ defined by the equation $x_{d+1} = \sum_{i=1}^{d} x_i^2$ and hyperspheres in $\boldsymbol{E}^d$ to non-vertical hyperplanes cutting the paraboloid in $\boldsymbol{E}^{d+1}$. For a point $p = (p_1, p_2, ..., p_d)$, $\mathcal{U}(p)$ corresponds to the point $(p_1, p_2, ..., p_d, \sum_{i=1}^{d} p_i^2)$. For a hypersphere σ centered at $c = (c_1, c_2, ..., c_d)$ and with a radius r, $\mathcal{U}(\sigma)$ corresponds to the non-vertical hyperplane defined by the equation $x_{d+1} = \sum_{i=1}^{d} 2c_i x_i + (r^2 - \sum_{i=1}^{d} c_i^2)$. This geometric transformation preserves the incidence relation. Using this geometric transformation, an upper bound on the maximum number of incidences between S-restricted families of points and families of hyperspheres in $\boldsymbol{E}^d$ can be derived directly from Corollary 2.3. The assumption on the families of points implies that any $(d+1)$ points projected on the paraboloid under $\mathcal{U}$ span a hyperplane. Hence, we can apply Corollary 2.3 to obtain the following result.

Corollary 4.1 *Let I_d^s be the maximum number of incidences between x distinct hyperspheres and an S-restricted family of y points in $\boldsymbol{E}^d$. Then $I_d^s(x, y)$ is in $O(x^{\frac{2d}{2d+1}} y^{\frac{d+1}{2d+1}} + x + y)$.*

The $O(x^{\frac{4}{5}} y^{\frac{3}{5}} + x + y)$ upper bound for $I_2^s(x, y)$ matches the one presented in [12]. In the plane, the assumption that no three points lie on a 1-dimensional hypersphere holds if and only if the points are distinct, i.e., all families of distinct points is the plane are S-restricted.

This result can be used to find an upper bound on the maximum number of pairs of points at unit-distance among S-restricted families of n points in $\boldsymbol{E}^d$. This maximum number is denoted $f_d^{ud}(n)$. This problem has been posed by Erdős [18]. He presented an $\Omega(n^{1+o(1)})$ lower bound and an $O(n^{\frac{3}{2}})$ upper bound for $f_2^{ud}(n)$. Since then, the lower bound has not been improved but the upper bound has been reduced to $O(n^{\frac{4}{3}})$ by Spencer, Szemerédi and Trotter [30]. For $f_3^{ud}(n)$, Erdős [19] gave an $\Omega(n^{\frac{4}{3}} \log\log n)$ lower bound and an $O(n^{\frac{5}{3}})$ upper bound. In [12], Clarkson *et al.* reduced the upper bound to $O(n^{\frac{3}{2}} 2^{O(\alpha(n)^2)})$ where $\alpha(n)$ is the functional inverse of Ackermann's function. For $d \geq 4$, an example attributed to Lenz in [19] shows that the lower bound for $f_d^{ud}(n)$ is in $\Omega(n^2)$. On the other hand, Chung [9] presented an $O(n^{2-2/(d+2)})$ upper bound for $f_d^{ud}(n)$ for families of points without three points belonging to more than c hyperspheres, for some absolute constant c.

As observed in [30], the maximum number of pairs of points at unit-distance in families of n points is at most half the maximum number of incidences between families of n points and families of n unit-hyperspheres centered at these n points. Therefore, Corollary 4.1 gives directly the following upper bound on $f_d^{ud}(n)$ for S-restricted families of points.

Corollary 4.2 *For S-restricted families of n points in $\boldsymbol{E}^d$ the maximum number of pairs of points at unit-distance $f_d^{ud}(n)$ is $O(n^{\frac{3d+1}{2d+1}})$.*

Unlike the upper bound presented in [12], our $O(n^{\frac{10}{7}})$ upper bound for $f_3^{ud}(n)$ holds only for families of points without four points lying on a circle (or on a line).

Another problem introduced by Erdös [20] is to determine the minimum sum of the number of different distances from each point in families of n points in $\boldsymbol{E}^d$. Let S be a family of n points in $\boldsymbol{E}^d$ and let $\delta(p,q)$ denote the Euclidean distance between two points. Let $g_d(S,p_i) = |\{\delta(p_i,p_j) \mid p_j \in S\}|$ be the number of different distances from the point p_i in S and let $g_d^{\Sigma}(S) = \sum_i g_d(S,p_i)$ be the sum of the number of the different distances from each point in S. Finally, let $g_d^{\Sigma}(n) = \min\{g_d^{\Sigma}(S) | S \subset \boldsymbol{E}^d$ and $|S| = n\}$ be the minimum sum of the number of the different distances from each point in families of n points. The problem is to find lower and upper bounds for $g_d^{\Sigma}(n)$. In [20], an $O(n^2/\sqrt{\log n})$ upper bound for $g_2^{\Sigma}(n)$ is presented. On the other hand, an $\Omega(n^{\frac{7}{4}})$ lower bound for $g_2^{\Sigma}(n)$ is derived in [12] using the upper bound on the maximum number of incidences between n points and n circles. By using their reduction, a lower bound for $g_d^{\Sigma}(n)$ can be determined for S-restricted families of points. Around each point p_i in S, put $g_d(S,p_i)$ hyperspheres containing all the points. The number of incidences between the n points in S and these $g_d^{\Sigma}(S)$ hyperspheres is exactly $2\binom{n}{2}$. Each pair of points in S determine two incidences. By applying Corollary 4.1, the number of incidences between the points and the hyperspheres is in $O(g_d^{\Sigma}(S)^{\frac{2d}{2d+1}} n^{\frac{d+1}{2d+1}} + g_d^{\Sigma}(S) + n)$. Hence, $g_d^{\Sigma}(S)^{\frac{2d}{2d+1}} n^{\frac{d+1}{2d+1}}$ is in $\Omega(n^2)$, implying that $g_d^{\Sigma}(S) \in \Omega(n^{\frac{d+1}{2d+1}})$ for any S-restricted family of n points. This gives the following result.

Corollary 4.3 *For S-restricted families of n points in $\boldsymbol{E}^d$ the minimum sum of the number of different distances from each point $g_d^{\Sigma}(n)$ is in $\Omega(n^{\frac{3d+1}{2d}})$.*

This corollary gives a lower bound on the minimum number of different distances determined by families of n points in $\boldsymbol{E}^d$. This value is denoted $f_d(n)$. Erdös [18] proved that $\sqrt{n-1}-1 \leq f_2(n) \leq cn/\sqrt{\log n}$. The lower bound has been improved several times. Recently, Chung, Szemerédi and Trotter [10] proved that $f_2(n)$ is in $\Omega(n^{\frac{4}{5}}/\log^c n)$, for some fixed c. Finally, Moser [29] gave a $\Omega(n^{\frac{1}{d}})$ for $f_d(n)$. An easy way to obtain a lower bound on $f_d(n)$ is to observe that $f_d(n)$ is in $\Omega(g_d^{\Sigma}/n)$. For S-restricted families of points, it is possible to obtain the following lower bound for $f_d(n)$ by applying Corollary 4.3.

Corollary 4.4 *For S-restricted families of n points in $\boldsymbol{E}^d$ the minimum number of different distances $f_d(n)$ is in $\Omega(n^{\frac{d+1}{2d}})$.*

Finally, the last problem considered in this section is to determine an upper bound on the maximum number of furthest neighbor pairs among families of n points in $\boldsymbol{E}^d$. This maximum number is denoted $f_d^{fn}(n)$. A point p_j is called a furthest neighbor of the point p_i if $\delta(p_i,p_j) = \max_{p_k} \delta(p_i,p_k)$. Each point has at least one furthest neighbor. Thus, a pair (p_i,p_j) is a furthest neighbor pair if p_j is a furthest neighbor of p_i.

Avis [3] showed that $f_2^{fn}(n)$ is equal to $3n-4$ for even $n \geq 4$ and equal to $3n-4$ or $3n-3$ for odd $n \geq 5$. Edelsbrunner and Skiena [17], with a more precise analysis, proved that $f_2^{fn}(n) = 3n - 4$ for any $n \geq 4$. For $d \geq 3$, an example presented by Avis, Erdős and Pach [4] shows that the lower bound for $f_d^{fn}(n)$ is in $\Omega(n^2)$. On the other hand, an $O(n^{\frac{3}{2}} 2^{O(\alpha(n)^2)})$ upper bound for $f_3^{fn}(n)$ is presented in [12] for families of n points without three collinear points. The proof of this upper bound is quite involved and is based on the structure of the arrangement of spheres in $\boldsymbol{E}^3$.

The problem of finding an upper bound for $f_d^{fn}(n)$ can be reduced to the problem of finding an upper bound on the maximum number of incidences between S-restricted families of n points and families of n hyperspheres. Around each point p_i in the family, put a hypersphere with a radius of $\max_{p_j} \delta(p_i, p_j)$. By applying Corollary 4.1, we obtain the following upper bound on $f_d^{fn}(n)$ for S-restricted families of points.

Corollary 4.5 *For S-restricted families of n points in $\boldsymbol{E}^d$ the maximum number of furthest neighbor pairs $f_d^{fn}(n)$ is in $O(n^{\frac{3d+1}{2d+1}})$.*

This gives an $O(n^{\frac{10}{7}})$ upper bound for $f_3^{fn}(n)$ for families of n points in $\boldsymbol{E}^3$ without four points lying on a circle (or on a line). This bound can not be compared to the $O(n^{\frac{3}{2}} 2^{O(\alpha(n)^2)})$ upper bound for families of points in $\boldsymbol{E}^3$ without three collinear points given in [12]. Both assumptions on the families of points are not comparable.

5 Conclusion

In this paper we presented a bound on the number of incidences between restricted families of point and hyperplanes in $\boldsymbol{E}^d$. From it we derived algorithms for computing such incidences and for solving the exact fitting problem. We also derived some further combinatorial bounds on e.g. the number of unit-distance pairs of points.

A number of open problems remain. The $O\left(\min\left\{\frac{n^d}{m^{d-1}} \log \frac{n}{m}, n^d\right\}\right)$ time algorithm for solving the exact fitting problem for families of n of points in $\boldsymbol{E}^d$ might not be optimal. Any improvement to the algorithm computing the incidences between a family of hyperplanes and a family of points will be reflected in the time complexity of our solution. For example, an optimal $O(x^{2/3}y^{2/3} + x + y)$ time solution for computing the incidence between x distinct points and y distinct lines in the plane would reduce the time complexity of our algorithm to $O\left(\frac{n^2}{m}\right)$.

Another interesting open problem is to extend the exact fitting problem to families of convex objects. For example, suppose we have a family of n line segments in the plane. Find a line *intersecting* at least m of these line segments. This problem can also be solved in $O(n^2)$ time by sweeping the dual arrangement with a topological line (see [15]). But one would prefer an algorithm whose complexity depends on m.

References

[1] P.K. Agarwal. Partitioning arrangements of lines II: applications. *Disc. Comp. Geom.*, 5:533–573, 1990.

[2] E. Anagnostou, L. J. Guibas, and V. Polomenis. Topological sweeping in three dimensions. In *Proc. of the First SIGAL Symp.*, pages 310–317, 1990.

[3] D. Avis. The numbers of furthest neighbour pairs of a finite planar set. *Amer. Math. Monthly*, 91:417–420, 1984.

[4] D. Avis, P. Erdős, and J. Pach. Repeated distances in space. *Graphs and Comb.*, 4:207–217, 1988.

[5] G. Brassard. Crusade for a better notation. *SIGACT News, ACM*, 17(1):60–64, 1985.

[6] G. Brassard and P. Bratley. *Algorithmics: Theory and Practice.* Prentice-Hall, Englewood Cliffs, NJ, 1988.

[7] B. Chazelle. An optimal convex hull algorithm and new results on cutting. In *Proc. of the 32th IEEE Found. of Comp. Science*, 1991.

[8] B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10:229–249, 1990.

[9] F.R.K. Chung. Sphere-and-point incidence in high dimensions with applications to unit distances and furthest-neighbor pair. *Disc. Comp. Geom.*, 4:183–190, 1989.

[10] F.R.K. Chung, E. Szemerédi, and W.T. Trotter. The number of different distances determined by a set points in the Euclidean space. *Disc. Comp. Geom.*, 7:1–11, 1992.

[11] K.L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comp.*, 17:830–847, 1988.

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

[13] N.N. Doroshko, M.M. Korneenko, and N.N. Metelskij. Optimal placement of a line relative to a planar point system. *Institute of Math., Byelorussian Academy of Sciences*, 208:1–19, 1984.

[14] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, 1987.

[15] H. Edelsbrunner and L.J. Guibas. Topologically sweeping an arrangement. *J. Comp. Syst. Sc.*, 38:165–194, 1989.

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

[17] H. Edelsbrunner and S.S. Skiena. On the number of furthest neighbour pairs in a point set. *Amer. Math. Monthly*, 96:614–618, 1989.

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

[19] P. Erdős. On sets of distances of n points. *Magyar Tud. Akad. Kutaló Int.*, 5:165–169, 1960.

[20] P. Erdős. On some problems of elementary and combinatorial geometry. *Ann. Mat. Pura Appl. (IV)*, 103:99–108, 1975.

[21] B. Grünbaum. *Convex Polytopes.* Interscience Publishers, London, 1967.

[22] L. J. Guibas, M. Overmars, and J.-M. Robert. The exact fitting problem for points. In *Proc. of the 3rd Can. Conf. on Comp. Geom.*, pages 171–174, 1991.

[23] M.E. Houle. Algorithms for weak and wide separation of sets. In *Proc. of the Int. Workshop on Disc. Algo. and Comp.*, pages 61–68, 1989.

[24] M.E. Houle, H. Imai, K. Imai, J.-M. Robert, and P. Yamamoto. Orthogonal weighted linear L_1 and L_∞ approximation and applications. To appear in *Disc. App. Math.*

[25] M.E. Houle and G.T. Toussaint. Computing the width of set. *IEEE Trans. Patt. Anal. Mach. Intell.*, 10:761–765, 1988.

[26] T. Kővári, V.T. Sós, and P. Turán. On a problem of K. Zarankiewicz. *Colloq. Math.*, 3:50–57, 1954.

[27] J. Matoušek. Cutting hyperplane arrangements. In *Proc. of the 6th Annual ACM Symp. on Comp. Geom.*, pages 1–9, 1990.

[28] J. Misra and D. Gries. Finding repeated elements. *Sc. Comp. Prog.*, 2:143–152, 1982.

[29] L. Moser. On different distances determined by n points. *Amer. Math. Monthly*, 59:85–91, 1952.

[30] J. Spencer, E. Szemerédi, and W.T. Trotter. Unit distances in the Euclidean plane. In B. Bollobás, editor, *Graph Theory and Combinatorics*, pages 293–303, London, 1984. Academic Press.

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