Selecting Distances in Arrangements of Hyperplanes Spanned by Points

SERGEI BESPAMYATNIKH, *Department of Computer Science, Duke University, Box 90129, Durham, NC 27708, USA. E-mail: besp@cs.duke.edu*

MICHAEL SEGAL, *Communication Systems Engineering Department, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel. E-mail: segal@cse.bgu.ac.il*

ABSTRACT: In this paper we consider a problem of distance selection in the arrangement of hyperplanes induced by n given points. Given a set of n points in d-dimensional space and a number k, $1 \le k \le \binom{n}{d}$, determine the hyperplane that is spanned by d points and at distance ranked by k from the origin. For the planar case we present an $O(n \log^2 n)$ runtime algorithm using parametric search partly different from the usual approach [21]. We establish a connection between this problem in 3-d and the well-known 3SUM problem using an auxiliary problem of counting the number of vertices in the arrangement of n planes that lie between two sheets of a hyperboloid. We show that the 3-d problem is almost 3SUM-hard and solve it by an $O(n^2 \log^2 n)$ runtime algorithm. We generalize these results to the d-dimensional ($d \ge 4$) space and consider also a problem of enumerating distances.

Keywords: distance selection problem, parametric search, lower bound, counting

1 Introduction

In this paper we consider the following problem:

Hyperplane Distance Selection in $\mathbb{R}^d$. Let S be a set of n distinct points in d-

dimensional space, $d \geq 2$, and let F be the set of $\binom{n}{d}$ hyperplanes, each defined by a d-tuple of points. Let $\alpha_1, \alpha_2, \ldots, \alpha_{\binom{n}{d}}$ be the sorted distances in increasing order between hyperplanes of H and the origin. For a given number $1 \leq k \leq \binom{n}{d}$ determine a hyperplane in H that produces α_k. (We assume, for simplicity, *general position* of the points, so that (1) no $d+1$ points lie in the same hyperplane and (2) any hyperplane that passes through d points is not perpendicular to the hyperplane $x_d = 0$.) For $d = 2$ we call this problem *line distance selection problem.*

The planar version of the problem considered in our paper has a military application. Consider the n input points serving as the military bases. Each pair of bases has a communication connection. The intruder (the disk with the center at the origin) tries either to disturb the normal communication between the bases (closest k connections to his location) or to listen the information transmitted over these communications. It is also known that any communication noise on the line connecting any two bases leads to bad quality communication or even broken communication between these two bases. Thus, the goal of intruder is to touch at least k closest lines to its location.

This version of the problem continues a list of optimization selection problems and very close by its nature to the well-known *slope selection* problem and *distance selection* problem. The slope selection problem, where we are given n points in the plane and an integer k, and we want to find a line passing through two given points with k^{th} ranked slope, received a lot of attention during the past two decades. Cole et al. [9] gave an $O(n \log n)$ time solution, using the parametric searching of Megiddo [21]. Using the duality transform the problem is to find an intersection point between two lines from a collection of n non-vertical lines that has the k-th smallest x-coordinate. The decision algorithm which counts the number of intersection points of lines inside a given slab is based on the counting the number of inversions in the permutation. Another alternative approach which is based on randomization has been proposed by Matoušek [18] and by Dillencourt et al. [10]. Brönnimann and Chazelle consider the problem applying the *cuttings* technique. Katz and Sharir [17] used *expanders* and obtained conceptually simpler than the other deterministic algorithms $O(n \log n)$ time solution.

The solution to the distance selection problem, where we are given n points in the plane and an integer k, and we want to find the k-th smallest distance between a pair of given points, can be obtained using a parametric searching. The decision problem is to compute, for a given real r, the sum $\Sigma_{p \in S} |D_r(p) \cap (S - \{p\})|$, where $D_r(p)$ is the closed disk of radius r centered at p. Agarwal et al. [2] gave an $O(n^{\frac{4}{3}} \log^{\frac{4}{3}} n)$ expected-time randomized algorithm for the decision problem, which yields an $O(n^{\frac{4}{3}} \log^{\frac{8}{3}} n)$ expected-time algorithm for the distance selection problem. Goodrich [15] derandomized this algorithm, at a cost of an additional polylogarithmic factor in the runtime. Katz and Sharir [17] obtained an expander-based $O(n^{4/3} \log^2 n)$-time deterministic algorithm for this problem. By applying a randomized approach Chan [7] was able to obtain an $O(n \log n + n^{2/3} k^{1/3} \log^{5/3} n)$ expected time algorithm for this problem.

The line distance selection problem is also closely related to the problem considered by Efrat et al. [11] where a set of n non-intersecting segments is given in the plane with an integer $k \leq n$ and one wants to find the smallest disk intersecting k segments. They [11] show how to solve this problem in $O(nk \log^2 n)$ (resp. $O(nk \log^2 n \log \frac{n}{k})$) time and $O(nk)$ (resp. $O(n \log n)$) space. Gupta et al. [16] present $O(\log n + k \log^2 n)$ time output-sensitive solution that finds k lines (among the n input lines) that are intersected by the query disk after preprocessing time $O(n^2 \log n)$.

We show that the decision version of the hyperplane distance selection problem is dual to the problem of determining whether the arrangement of n hyperplanes in $\mathbb{R}^d$ contains at most a given number of vertices lying between two sheets of a hyperboloid. We begin with the line distance selection problem. We present an $O(n \log n)$ time solution for the decision problem using the technique of Mount and Netanyahu[19]. For the optimization, we apply Megiddo's [21] parametric search. However, since our decision algorithm is not parallelizable, we had to find an algorithm that solves a completely different problem, but is both parallelizable and enables to generate the optimal solution when the parametric search technique is applied to it. We also apply Cole's technique for speeding up standard parametric searching [8] in order to produce $O(n \log^2 n)$ solution to the line distance selection problem.

Unfortunately, the hyperplane distance selection problem in 3-dimensional space is more difficult than its planar version. In contrary to the planar case it seems that the technique of counting the number of inversions in the permutation cannot be generalized. In fact, we prove that 3-dimensional hyperplane distance selection problem is almost 3SUM-hard (see Section 3 for exact definition). In other words, there is almost no hope to get a subquadratic solution for the 3-dimensional case. In Section 3 we discuss this issue and generalize it to higher dimensions space by reducing a problem which we call dSUM problem to the d-dimensional hyperplane distance selection problem. We dedicate Section 4 to the problem of enumerating k closest line distances. Finally we conclude in Section 5.

2 Planar line distance selection

In this section we present an $O(n \log^2 n)$ algorithm for the planar version of the line distance selection problem. First we show how to obtain $O(n \log n)$ time algorithm for the decision problem. In order to apply the Megiddo's optimization scheme [21] we have to parallelize our decision algorithm. However, the main part of our decision algorithm is not parallelizable, so, as in [1], we come up with an auxiliary problem whose parallel version will generate the optimal solution to our problem.

2.1 The decision algorithm

The decision version of the planar line distance selection problem can be formulated as follows. Given a set S of n points in the plane, an integer $k, 1 \leq k \leq \binom{n}{2}$ and a real value $R > 0$, determine whether a disk D_R centered at origin with radius R is intersected by at least k lines passing through pairs of points in S.

We use the following dual transformation. The point $p \in \mathbb{R}^2$ with coordinates (a, b) in the primal plane maps to the line $y = ax - b$ in the dual plane. The line $y = ax + b$ in the primal plane corresponds to the point q with coordinates $(a, -b)$ in the dual plane. Let D_R^* be the image of the set of all lines intersecting disk D_R. In the dual plane the decision problem is stated as: Given a set S^* of n lines in the plane, an

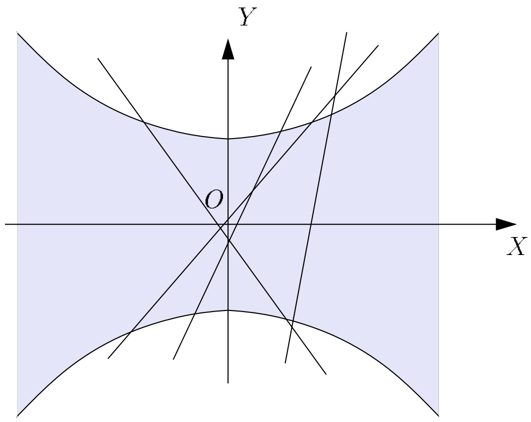

FIG. 1. The shaded region is D_R^*.

integer $k, 1 \leq k \leq \binom{n}{2}$ and a real value $R > 0$, determine whether D_R^* contains at least k vertices of the arrangement of lines in S^*.

LEMMA 2.1
The region D_R^* in the dual plane is bounded by two hyperbola branches $\frac{y^2}{R^2} - x^2 = 1$.

PROOF. Consider a point $(x^*, y^*) \in D_R^*$, see Figure 1. It corresponds to the line $y = x^*x - y^*$ in the primal plane. The distance between origin and this line is $\frac{|y^*|}{\sqrt{(x^*)^2+1}}$. By definition of D_R^*, $\frac{|y^*|}{\sqrt{(x^*)^2+1}} \leq R$. The proof follows. ∎

Our strategy, thus, is to find the number of the vertices of the arrangement of lines in S^* in the hyperbolic region D_R^*. Notice, that there might be lines that either intersect one of the boundaries of D_R^* twice or do not intersect any of them. We apply a counting technique due to Mount and Netanyahu [19] (see also [20]). Their technique works for a closed region with a connected boundary. The lines must satisfy (they [20] have also considered the general pseudolines) the following *boundary intersection properties*:

1. Each line intersects the boundary of this region an even number of times.
2. The number of intersections between a line and the boundary is bounded above by some constant, and

3. The intersections of lines along the the region's boundary can be cyclically sorted in $O(n \log n)$ time.

Mount and Netanyahu [19] show that it is possible to compute the number of intersections between lines that occur within the region in $O(n \log n)$ time and $O(n)$ space. Since, in our case, it may happen that a line may either intersect a boundary only once or not intersect it at all, we can find an axis-parallel rectangle $\bar{D}$ that contains all the intersection points and apply Mount and Netanyahu's algorithm for the region $\bar{D} \cap D_R^*$. This bounding rectangle $\bar{D}$ is defined by the rightmost, leftmost, highest and lowest intersection points which can be computed in $O(n \log n)$ time using the algorithm for the slope selection problem. Thus, we conclude by the theorem.

THEOREM 2.2
Given a set S of n points in the plane, an integer $k, 1 \le k \le \binom{n}{2}$ and a real value $R > 0$, in $O(n \log n)$ time and $O(n)$ space we can determine whether a disk D_R centered at origin with radius R is intersected by at least k lines passing through pairs of points in S.

2.2 *The optimization stage*

Given a set S of n points in the plane, and integer $k, 1 \le k \le \binom{n}{2}$ we need to determine the smallest radius $\tilde{R}$ such that the disk $D_{\tilde{R}}$ centered at origin is intersected by at least k lines passing through pairs of points in S. Our algorithm is based on the parametric search optimization scheme [21]. Let T_s denote the runtime of the sequential decision algorithm, and T_p, resp. P, the time and number of processors of the parallel algorithm for the decision problem; then the optimal solution can be computed in sequential time $O(PT_p + T_s T_p \log P)$ [21].

In order to apply the Megiddo optimization scheme we have to parallelize our decision algorithm. However, the counting algorithm of Mount and Netanyahu proceeds incrementally using a stack, thus making the problem of fast parallelization very difficult. Fortunately, as in [1], we come up with an auxiliary problem whose parallel version will generate the optimal solution to our problem.

The auxiliary problem is described as follows. Assume we have a set S of n points and a fixed radius R. For a point $p_i \in S$ outside D_R we build two tangent lines l_i^1, l_i^2 to D_R passing through p_i. Let T be a set of such lines l_i^1, l_i^2. The cardinality of T is at most $2n$.

The **auxiliary problem** is to find the sorted order of the slopes of lines in T. This can be done in parallel $O(\log n)$ time using $O(n)$ processors.

We now want to apply (generically) this parallel algorithm for finding the optimal radius $\tilde{R}$. First we get an initial interval I_0 where $\tilde{R}$ resides. Clearly, $I_0 = [0, \max_{1 \leq i \leq n} dist(O, p_i)], p_i \in S$. Consider now a single step in the parallel sort (the auxiliary problem). In this step we perform $O(n)$ slope comparisons, each comparison involving a pair of lines. For each such pair of lines we compute *critical values* of R where the sorted order of lines can change. There are two cases: (a) the two compared slopes are defined by the same point, and (b) the two compared slopes are defined by the distinct points. In case (a) we have (at most) two critical values: when one of the rays becomes horizontal (the slope is changing signs). For case (b) let one such comparison involve the points p_i and p_j. In order to resolve this comparison, we must compute the slopes of l_i^h and l_j^k, $h, k \in \{1, 2\}$ and sort them. Of course, we do not know $\tilde{R}$, so we again compute the constant number of critical values: two values defined by the events when one of the lines becomes horizontal, third value is defined when the lines coincide and the last value can be derived from the situation when the lines do not coincide but remain parallel. Now we apply the decision algorithm of the subsection above to perform a binary search over the $O(n)$ critical values that were computed. Thus we find an interval $I \subseteq I_0$ where $\tilde{R}$ resides, resolve all the comparisons of this parallel stage, and proceed to the next parallel stage.

What does resolving mean here? If the crucial value $\tilde{R}$ does not belong to I, then we simply ignore it. Otherwise, the slope ordering of two lines is defined uniquely, because the interval I does not contain any critical value produced at this stage (except maybe endpoints) The closed interval I is always guaranteed to contain $\tilde{R}$ but we need to show that a comparison is made where $R = \tilde{R}$.

CLAIM 2.3

The slope order of the lines changes as R' changes from values slightly smaller than $\tilde{R}$ to values slightly larger than $\tilde{R}$.

PROOF. The value $\tilde{R}$ is defined by some line passing through two points of $p_i, p_j \in S$. One of the critical values obtained by comparison of slopes of the $l_i^1, l_i^2, l_j^1, l_j^2$ is $\tilde{R}$. Since $\tilde{R}$ is a critical value for two lines, the slope order of these lines changes from values slightly smaller than $\tilde{R}$ to values slightly larger than $\tilde{R}$. ∎

Note that at some stage the optimal solution will appear on the boundary of the interval I computed at that stage (it could even appear on the boundary of I_0). However, once it appears, it will remain one of the endpoints of all subsequently computed intervals. At the end, we run the decision algorithm for the left endpoint of the final interval. If the answer is positive, then this endpoint is $\tilde{R}$, otherwise $\tilde{R}$ is the right endpoint of the final interval.

Plugging the sequential and parallel algorithm into a parametric search machinery we obtain an $O(n \log^3 n)$ time algorithm for the optimization problem. However, we can apply Cole's technique [8] in order to speed up Megiddo's parametric search. Since our parallel algorithm is based on sorting, we can use the sorting algorithm based on AKS network [3] in order to shave one logarithm from the running time for the optimization problem. Thus, we conclude by

THEOREM 2.4

The planar line distance problem can be solved in $O(n \log^2 n)$ time using $O(n)$ space.

3 Lower bound for $d \geq 3$

For simplicity we demonstrate a lower bound proof for the hyperplane distance selection problem in the 3-dimensional space and then show how to extend it to higher dimensions. In fact we establish a lower bound for the decision version of the hyperplane distance problem.

Gajentaan and Overmars [14] defined 3SUM-hard class of the problems. The main characteristics of these problems is the existence of $O(n^2)$ barrier in the complexity of these problems. Namely, the best algorithms for these problems take time $O(n^2)$, while no non-trivial lower bounds are known.

We cite the definitions and notations from [14].

DEFINITION 3.1

Given two problems PR1 and PR2 we say that PR1 is $f(n)$-solvable using PR2 iff every instance of PR1 of size n can be solved using a constant number of instances of PR2 (of at most linear size) and $O(f(n))$ additional time. We denote this as PR1 $<_{f(n)}$ PR2.

LEMMA 3.2 ([14])

Let PR1 $<_{f(n)}$ PR2. Let $f(n)$ and $g(n)$ be polynomials. If PR2 can be solved in $O(g(n))$ time and $f(n) = O(g(n))$ then PR1 can be solved in $O(g(n))$ time. Hence, if $\Omega(g(n))$ is a lower bound for PR1 and $f(n) = o(g(n))$ then $\Omega(g(n))$ is also a lower bound for PR2.

The base problem considered in [14] is the following

3SUM Problem: Given a set S of n integers, are there $a, b, c \in S$ with $a + b + c = 0$?

DEFINITION 3.3

We call a problem PR 3SUM-hard if and only if 3SUM is $f(n)$-solvable using PR, where $f(n) = o(n^2)$.

Gajentaan and Overmars [14] have proved that the following problem is also 3SUM-hard.

3SUM' Problem: Given three sets of integers A, B and C of total size $O(n)$, are there $a \in A$, $b \in B$ and $c \in C$ with $a + b = c$.

We generalize the 3SUM-hardness definition to the dSUM-hardness definition.

***d*SUM Problem**: Given a set S of n integers, are there $x_1, x_2, \ldots, x_d \in S$ with $\Sigma_{i=1}^{d} x_i = 0$? Erickson [13] shows how to solve the dSUM Problem ($d \geq 2$) in $T_d(n)$ time, where $T_d(n) = O(n^{\frac{d}{2}} \log n)$ for even d, and $O(n^{\frac{d+1}{2}})$ for odd values of d.

DEFINITION 3.4

We call a problem PR dSUM-hard if and only if dSUM is $f(n)$-solvable using PR, where $f(n) = o(n^{\frac{d}{2}} \log n)$ for even d and $f(n) = o(n^{\frac{d+1}{2}})$ for odd d.

Similarly to Erickson [12] definition we introduce the notion of *almost* hardness.

DEFINITION 3.5

Given two problems PR1 and PR2 of complexities $T_1(n)$ and $T_2(n)$ respectively, we say that PR1 is almost PR2-hard if $T_2(n) = O(T_1(n)\log n)$.

3.1 $d = 3$

Now we focus on the 3-dimensional case. We apply the following dual transformation: the point $p \in \mathbb{R}^3$ with coordinates (a, b, c) in the primal space maps to the plane $ax + by + z + c = 0$ in the dual space and the plane $AX + BY + CZ + D = 0$ in the primal space corresponds to the point q with coordinates $(A/C, B/C, D/C)$ in the dual space ($C \neq 0$ because the points of S are in general position). Similarly to the analysis in Section 2, we let D^*_R be the image of the set of all planes intersecting ball D_R. In the dual space the decision problem for the hyperplane distance problem is: Given a set S^* of n planes $\mathbb{R}^3$ in , an integer $k, 1 \leq k \leq \binom{n}{3}$ and a real value $R > 0$, determine whether D^*_R contains at least k vertices of the arrangement of hyperplanes in S^*.

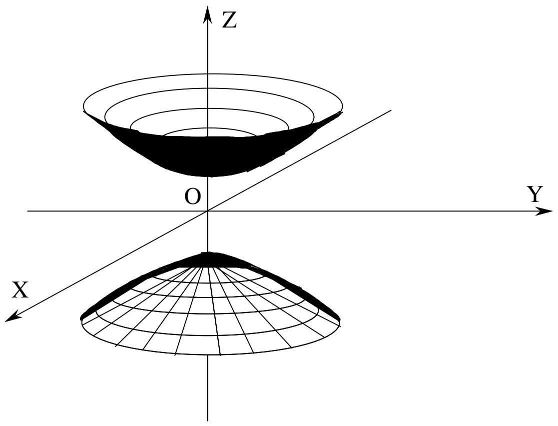

FIG. 2. Two-sheeted circular hyperboloid.

LEMMA 3.6

The region D^*_R in the dual space is bounded by two sheets of circular hyperboloid oriented along the z-axis $x^2 + y^2 - \frac{z^2}{R^2} = -1$.

PROOF. See Figure 2. Consider a point $(x^*, y^*, z^*) \in D_R^*$. It corresponds to the plane $x^*x + y^*y + z + z^* = 0$ in the primal space. The distance between origin and this plane is $\frac{|z^*|}{\sqrt{(x^*)^2+(y^*)^2+1}}$. By definition of D_R^*, $\frac{|z^*|}{\sqrt{(x^*)^2+(y^*)^2+1}} \le R$. The proof follows. ∎

Let us define the following two problems.

Hyperboloid Counting Problem (HCP) in $\mathbb{R}^3$: Given a collection of n planes in $\mathbb{R}^3$ and the hyperboloid $P_a = \{x^2 + y^2 - \frac{z^2}{a^2} = -1\}, a > 0$ determine the number of the vertices of the arrangement of the planes between two sheets of P_a.

Hyperboloid Rank Problem (HRP) in $\mathbb{R}^3$: Given a collection of n planes in $\mathbb{R}^3$, an integer k, $1 \le k \le \binom{n}{3}$ and the hyperboloid $P_a = \{x^2 + y^2 - \frac{z^2}{a^2} = -1\}, a > 0$ determine whether the number of the vertices of the arrangement of the planes between two sheets of P_a is at least k.

Notice that the HRP problem is the decision version of the HCP problem. We can solve the HCP problem by a binary search over the $\binom{n}{3}$ possible values for k using a HRP algorithm at each step at the search. By Lemma 3.6 the decision version of the hyperplane distance selection problem in $\mathbb{R}^3$ is equivalent to the HRP problem in $\mathbb{R}^3$ (for $a = R$). Therefore, the decision version of the hyperplane distance selection problem in $\mathbb{R}^3$ is almost HCP-hard.

COMMENT 3.7

PROOF. We use the following dual transformation. The point $p_i \in S$ with coordinates (a_i, b_i, c_i) in the primal space maps to the plane $z = a_i x + b_i y + c_i$ in the dual space. The plane $ax + by + cz + d = 0$ in the primal space corresponds to the point q with coordinates $(a/c, b/c, -d/c)$ in the dual space. Recall that $H = \{h_1, h_2, \ldots, h_{\binom{n}{3}}\}$ is the set of planes, each defined by a triple of points of S and $\alpha_1, \alpha_2, \ldots, \alpha_{\binom{n}{3}}$ are the corresponding sorted angles between planes of H and plane $z = 0$. Assume that the plane $ax + by + cz + d = 0$ defines the solution of the slope selection problem for a given value of k, i.e. the angle formed by $ax + by + cz + d = 0$ and $z = 0$ is equal to α_k. This angle α_k is defined as $\cos\alpha_k = \frac{c}{\sqrt{a^2+b^2+c^2}}$. Since the points are in general position, $\cos\alpha_k = \frac{1}{\sqrt{(\frac{a}{c})^2+(\frac{b}{c})^2+1}}$. In other words, $\cos\alpha_k = \frac{1}{\sqrt{x(q)^2+y(q)^2+1}}$, where q_x and q_y are the coordinates of the point p. Let Q be the vertices in the dual space

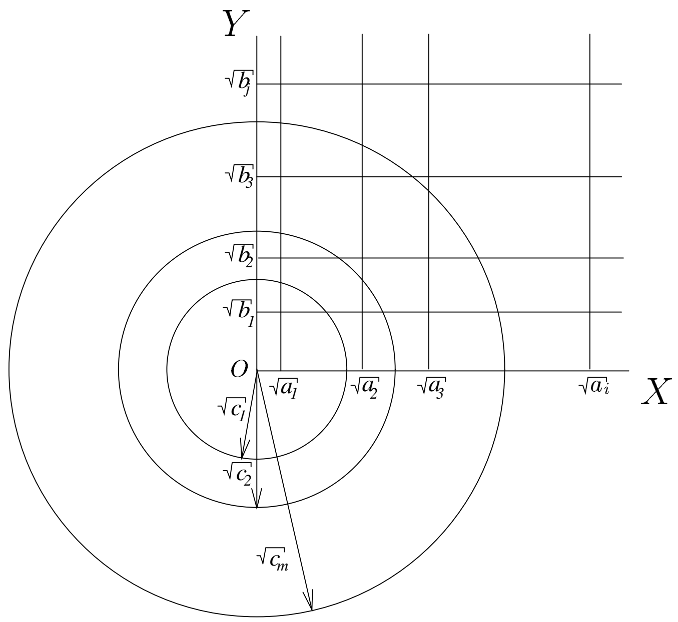

FIG. 3. Arrangement of planes with hyperboloid viewed from $z = +\infty$.

corresponding to the planes of H and let $q_i \in Q$ be the image of $h_i \in H$. Since q_k has coordinates $(a/c, b/c, -d/c)$ in the dual space, $\cos\alpha_k = 1/\sqrt{x(q_k)^2 + y(q_k)^2 + 1}$. Taking into account that $\cos x$ is the decreasing function in $[0, \frac{\pi}{2}]$ we obtain that the sequence $\{x^2(q_i) + y^2(q_i)\}_{i=1}^{\binom{n}{3}}$ is increasing. Hence, there are $\binom{n}{3} - k + 1$ of Q above paraboloid P containing q_k on its boundary.

∎

Next, we prove the main result of this subsection.

THEOREM 3.8

The hyperboloid counting problem in $\mathbb{R}^3$ is 3SUM-hard.

PROOF. We show the reduction from 3SUM' problem to the hyperboloid counting problem. Given an instance A,B and C, $|A| + |B| + |C| = n$ of the 3SUM' problem, we can assume that all the integers in the sets A, B and C are positive; otherwise we can add the same large number L to the elements of A and B and add $2L$ to the elements of C. We define an instance of the hyperboloid counting problem by taking hyperboloid P_R with $R = 1$ and defining n planes as follows : $x = \sqrt{a_i}, a_i \in A, 1 \le i \le |A|$, $y = \sqrt{b_j}, b_j \in B, 1 \le i \le |B|$ and $z = \pm\sqrt{c_m + 1}, c_m \in C, 1 \le m \le |C|$. Notice that the circles obtained by intersection the planes $z = \pm\sqrt{c_m + 1}$ with hyperboloid P_1 have radius $\sqrt{c_m}$. It is clear that a vertex of the arrangement of these n planes lies on P_1 if and only if exist $a \in A, b \in B, c \in C$, such that $\sqrt{(a)^2} + \sqrt{(b)^2} = \sqrt{(c)^2}$ holds (Pythagoras' Theorem). In other words, if and only if $a + b = c$ holds and the 3SUM' problem has a solution. See Figure 3.

In order to detect whether such a vertex exists we apply the following strategy. We apply the HCP algorithm and determine the number of vertices v_1 of the arrangement of the planes between two sheets of P_1. A vertex of the arrangement of planes lies on P_1 if and only if for sufficiently small $\varepsilon > 0$ the number of vertices of the arrangement of the planes between two sheets of $P_{1+\varepsilon}$ is less than v_1. So, by applying the HCP algorithm again we can answer the question. The only problem is the finding a sufficiently small $\varepsilon > 0$. One can imagine a situation when no vertex of the arrangement of planes lies on P_1 but for some values of ε the number of vertices of the arrangement of the planes between two sheets of $P_{1+\varepsilon}$ is less than v_1.

We show that by taking

$$\varepsilon = \sqrt{1 + \frac{1}{\max_{a_i \in A} a_i + \max_{b_j \in B} b_j + 2}} - 1 \tag{3.1}$$

we guarantee that there are no vertices of the arrangement of planes lying on the hyperboloid P_R for any $1 < R < 1 + \varepsilon$. A point $p(x, y, z) \in P_R$ satisfies the equation $R^2(x^2 + y^2 + 1) = z^2$. Let p be arbitrary vertex of the arrangement of planes with coordinates $x = \sqrt{a_i}$, $y = \sqrt{b_j}$, $z = \sqrt{c_m + 1}$ for some i, j, m. By the equation (1) $1 < R^2 < 1 + \frac{1}{\max_{a_i \in A} a_i + \max_{b_j \in B} b_j + 2}$ for any $1 < R < 1 + \varepsilon$. Assuming that p lies on P_R we have: $R^2 = \frac{c_m + 1}{a_i + b_j + 1}$. Thus $R^2 - 1 = \frac{c_m - a_i - b_j}{a_i + b_j + 1} > 0$ and $c_m - a_i - b_j > 0$. It follows that $c_m - a_i - b_j \ge 1$ and

$$R^2 - 1 \ge \frac{1}{a_i + b_j + 1} > \frac{1}{\max_{a_i \in A} a_i + \max_{b_j \in B} b_j + 2}.$$

It contradicts $R < 1 + \varepsilon$. ∎

COROLLARY 3.9

The decision version of the hyperplane distance selection problem in $\mathbb{R}^3$ is almost

3SUM-hard.

We can obtain an algorithm with $O(n^2 \log n)$ runtime performance for the decision version of the hyperplane distance selection problem in $\mathbb{R}^3$. Following duality above we consider the HRP problem. The HRP problem can be solved by counting the number of vertices of the arrangement in a plane for each of n planes separately. Let us call these problems $\text{HRP}_1, \text{HRP}_2, \ldots, \text{HRP}_n$ problems. To avoid multiple counting of the same vertex we apply a search for the planes in a lexicographic order, i.e. for the first plane we count the vertices obtained by all planes, for the second plane we count the vertices obtained by all planes except the first, etc. For each plane we apply a planar algorithm similarly to one described in Section 2.

In order to solve the hyperplane distance selection problem in $\mathbb{R}^3$ we apply approach similar to the planar case. We define an auxiliary problem as follows. Assume we have a set S of n points in $\mathbb{R}^3$ and a ball B of fixed radius R centered at the origin. Fix a point $p_i \in S$ outside B. For each point $p \in S$ build two tangent planes to B passing through p and p_i. Let T_i be a set of such tangent planes. The auxiliary problem is to sort all the planes in T_i by their slopes (angle formed by a plane with OXY axis), separately for each i. We use $O(n^2)$ of processors (by assigning $O(n)$ processors to each point $p_i \in S$) in the parallel algorithm in order to solve this problem in $O(\log n)$ time. The solution for the optimization problem can be obtained by combining this parallel algorithm with the sequential algorithm for the HRP problem. At each parallel step we perform $O(n)$ comparisons for each point p_i yielding in total $O(n^2)$ comparisons. The $O(n)$ comparisons for each point p_i are resolved using the decision algorithm of the HRP_i problem, similarly to the optimization stage described in Section 2. One can show that the hyperplane distance selection problem in $\mathbb{R}^3$ can be solved in $O(n^2 \log^2 n)$ time.

3.2 $d > 3$

Similarly to the 3-dimensional case we prove that the hyperboloid counting problem in $\mathbb{R}^d$ is dSUM-hard. The dSUM' problem is defined as: Given d sets of integers

$A_1, A_2, \ldots, A_d$ of total size $O(n)$, are there $a_1 \in A_1$, $a_2 \in A_2$, ..., $a_d \in A_d$ with $\Sigma_{i=1}^{d-1} a_i = a_d$. It is easy to see that the dSUM' problem is dSUM-hard.

The idea is to reduce the dSUM' problem to the hyperboloid counting problem in $\mathbb{R}^d$ by taking hyperboloid $P = \{\Sigma_{i=1}^{d-1} x_i^2 = x_d^2 - 1\}$ and defining $O(n)$ hyperplanes $x_1 = \sqrt{a_{1,i}}$, $a_{1,i} \in A_1, 1 \leq i \leq |A_1|$, $x_2 = \sqrt{a_{2,i}}$, $a_{2,i} \in A_2, 1 \leq i \leq |A_2|$, ..., $x_d = \pm\sqrt{a_{d,i} + 1}$, $a_{d,i} \in A_d, 1 \leq i \leq |A_d|$. A vertex of the arrangement of these $O(n)$ hyperplanes lies on P if and only if there is a solution to the corresponding dSUM' problem. By counting the number of vertices of the arrangement of hyperplanes between two sheets of P and comparing with the number of vertices of the arrangement of hyperplanes between two sheets of $P' = \{\Sigma_{i=1}^{d-1} x_i^2 = \frac{x_d^2}{(1+\varepsilon)^2} - 1\}$, where

$$\varepsilon = \sqrt{1 + \frac{1}{\Sigma_{i=1}^{d-1} \max_{a_j \in A_i} a_j + 2}} - 1$$

we can detect whether such a vertex exists.

4 Enumerating k line distances

Given a set S of n points in the plane, an integer $k, 1 \leq k \leq \binom{n}{2}$, we want to enumerate (in sorted order) the k smallest distances between the origin and lines passing through pairs of points in S. We explain the idea behind the algorithm using the kinetic framework [4, 5]. We assume that we have a disk D centered at origin with radius growing from 0 to infinity. Our goal is report lines passing through pairs of points and intersecting disk D. The algorithm stops after reporting k such lines. Notice that at the current moment of time points of S lying inside of D will not participate in future events. For a point $p_i \in S$ outside D we build two tangent lines l_i^1, l_i^2 to D passing through p_i. Let L be a set of such lines l_i^1, l_i^2. The cardinality of L is at most $2n$. Our events are when any two lines in T become of the same slope during the process of growing D. Thus, we maintain the following data structure: a binary search tree T maintaining the sorted order of slopes of moving lines in L and an event queue Q of sorted events (in increasing order) defined by the adjacent lines in the sorted order maintained in T. We process the current event defined by lines, e.g. l_i^1 and l_j^2 by checking whether they tangent to D at the same point. Only if the answer is positive, we report a line passing through points p_i and $_j$. In both cases we delete this event from Q, swap lines l_i^1 and l_j^2 in sorted order maintained by T and produce two new events defined by l_i^1 and l_j^2 and their new neighbors. Clearly, the whole process takes $O((n+k)\log n)$ time since we can carry out each event in $O(\log n)$ time.

One may wonder whether the additional $O(n \log n)$ factor is really necessary in the running time. As a matter of fact, a lower bound of $\Omega(n \log n)$ can be established even for the case $k = 1$ using the *set disjointness problem* [22]. Let $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$ be two sets of real nonnegative numbers. To test whether A and B do not share any elements requires $\Omega(n \log n)$ comparisons. We can transform set disjointness to our problem with $k = 1$ by mapping A and B to the first and the third quadrant of the unit circle C in the plane as follows: a_j is mapped to the intersection of C with the line $y = a_j x$ in the first quadrant, while b_j is mapped to the analogous intersection in the third quadrant. Let S be the set of these $2n$ intersections. Definitely, the line defining the closest distance to the origin passes through origin if and only if $A \cap B \neq \emptyset$. Thus, the selection of 1^{st} (k^{th}) distance between lines and origin requires $\Omega(n \log n)$ operations in the algebraic computational-tree model.

5 Conclusions

It would be interesting to obtain an optimal $O(n \log n)$ (maybe expected) runtime algorithm for the planar version of the line distance selection problem. One of the possible approaches is by applying different dual transformation: the point with coordinates (a, b) maps to the line $ax + by = 1$. Then the obtained search region is a circle. Another possible way is by applying randomized optimization techniques, like randomized halving in order to obtain better results.

Acknowledgments. We express our thanks to Daniel Berend for helpful discussions regarding this paper and to anonymous referees whose remarks improved the presentation of the paper.

References

[1] P. Agarwal and M. Sharir. "Planar geometric location problems", *Algorithmica*, 11, pp. 185–195, 1994.

[2] P. Agarwal, B. Aronov, M. Sharir, S. Suri, "Selecting distances in the plane", *Algorithmica*, 9, pp. 495–514, 1993.

[3] M. Ajtai, J. Komlos and E. Szemeredi, "An $(n \log n)$ sorting network", *Combinatorica*, 3, pp. 1–19, 1983.

[4] J. Basch, "Kinetic Data Structures", Ph.D. thesis, Stanford University, USA, 1999.

[5] J. Basch, L. Guibas and J. Hershberger "Data structures for mobile data", In *Proc. 8th ACM-SIAM Sympos. Descrete Algorithms*, pp. 747–756, 1997.

[6] H. Brönnimann and B. Chazelle, "Optimal Slope Selection Via Cuttings", *Comput. Geom. Theory Appl*, 10(1), pp. 23–29, 1998.

[7] T. Chan "On enumerating and selecting distances", In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pp. 279–286, 1998.

[8] R. Cole, "Slowing down sorting networks to obtain faster sorting algorithms", *J. ACM*, 34, pp. 200–208, 1987.

[9] R. Cole and J. Salowe and W. Steiger and E. Szemerédi, "An optimal-time algorithm for slope selection", *SIAM J. Comput.*, 18(4), pp. 792–810, 1989.

[10] M. B. Dillencourt and D. M. Mount and N. S. Netanyahu, "A randomized algorithm for slope selection", *Internat. J. Comput. Geom. Appl.*, 2, pp. 1–27, 1992.

[11] A. Efrat, M. Sharir and A. Ziv, "Computing the smallest k-enclosing circle and related problems", *Computational Geometry: Theory and Applications* 4, pp. 119-136, 1994.

[12] J. Erickson, "On the relative complexities of some geometric problems", in *Proceedings of the 7th Canadian Conference on Computational Geometry*, pp. 85–90, 1995.

[13] J. Erickson, "Lower bounds for fundamental geometric problems", Ph.D thesis, University of California at Berkeley, 1996.

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

[15] M. Goodrich, "Geometric partitioning made easier, even in parallel", *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pp. 73-82, 1993.

[16] P. Gupta and R. Janardan and M. Smid, "Algorithms for some intersection searching problems involving curved objects", in *International Journal of Mathematical Algorithms*, Vol. 1, pp. 35–52, 1999.

[17] M. J. Katz and M. Sharir, "Optimal slope selection via expanders", *Inform. Process. Lett.*, 47, pp. 115–122, 1993.

[18] J. Matoušek, "Randomized optimal algorithm for slope selection", *Inform. Process. Lett.*, 39, pp. 183–187, 1991.

[19] D. Mount and N. Netanyahu, "Efficient algorithms for robust circular arc estimators", in *Proc. 5th Canad. Conf. Comput. Geom.*, pp. 79–84, 1993.

[20] D. Mount and N. Netanyahu, "Efficient randomized algorithms for robust estimation of circular arcs and aligned ellipses", Comput. Geom. Theory Appl., 19, pp. 1–33, 2001.

[21] N. Megiddo "Applying parallel computation algorithm in the design of serial algorithms", *Journal of ACM*, 30, pp. 852–865, 1983.

[22] F. Preparata and M. Shamos "Computational Geometry: An Introduction", Springer-Verlag, New York, NY, 1985.

Received 15 September, 2000