Finding a Vector Orthogonal to Roughly Half a Collection of Vectors

Pierre Charbit[1], Emmanuel Jeandel[2], Pascal Koiran[3], Sylvain Perifel[3], and Stéphan Thomassé[4]

[1] LIAFA, UMR 7089 CNRS - Université Paris 7
Pierre.Charbit@liafa.jussieu.fr
[2] LIF, UMR 6166 - Universités de la Méditérannée et de Provence
emmanuel.jeandel@lif.univ-mrs.fr
[3] LIP, École Normale Supérieure de Lyon
UMR 5668 ENS Lyon, CNRS, UCBL, INRIA.
[Pascal.Koiran,Sylvain.Perifel]@ens-lyon.fr
[4] LIRMM, UMR 5506 CNRS - Université Montpellier II
Stephan.Thomasse@lirmm.fr

Abstract. Dimitri Grigoriev has shown that for any family of N vectors in the d-dimensional linear space $E = (\mathbb{F}_2)^d$, there exists a vector in E which is orthogonal to at least $N/3$ and at most $2N/3$ vectors of the family. We show that the range $[N/3, 2N/3]$ can be replaced by the much smaller range $[N/2 - \sqrt{N}/2, N/2 + \sqrt{N}/2]$ and we give an efficient, deterministic parallel algorithm which finds a vector achieving this bound. The optimality of the bound is also investigated.
Keywords: algebraic complexity, decision trees, parallel algorithms, derandomization.

1 Introduction

Dimitri Grigoriev [6] has shown that the point location problem[5] in arrangements of m algebraic hypersurfaces of degree D in $\mathbb{R}^n$ can be solved by topological decision trees of depth $O(n \log(mD))$. In topogical decision trees [14,16] nodes are labelled by *arbitrary* polynomials, i.e., the cost of their evaluation is ignored. The key ingredient in his nonconstructive proof is the following combinatorial lemma. Let $\mathbb{F}_2$ be the two-element field. For any family of N vectors in the d-dimensional linear space $E = (\mathbb{F}_2)^d$, there exists a vector in E which is orthogonal to at least $N/3$ and at most $2N/3$ vectors of the family. Orthogonality is defined with respect to the $\mathbb{F}_2$-valued "inner product" $u.v = \sum_{i=1}^{d} u_i v_i$ (strictly speaking, this is of course not a "honest" inner product since for instance a vector can be orthogonal to itself).

In order to explore the constructive aspects of Grigoriev's point location theorem it is useful to have a constructive version of this combinatorial lemma. Here one main goal is to obtain new transfer theorems for algebraic versions of the

[5] It is misleadingly called "range searching problem" in [4] and [6].

P vs. NP problem. It is well known that the point location problem in arrangements of hyperplanes can be solved efficiently by linear decision trees [9,10,11]. This was the main technical tool in the proof that the P vs. NP problem for the real numbers with addition and order is equivalent to the classical problem [4,5]. As suggested in [4] and [7], a better understanding of point location in arrangements of hypersurfaces makes it possible to obtain transfer theorems for a richer model of computation in which multiplication is allowed. Precise statements and proofs can be found in [8]. The goals of the present paper are to improve Grigoriev's lemma and to give a constructive version of it. Namely, we show that the range $[N/3, 2N/3]$ can be replaced by the much smaller range $[N/2 - \sqrt{N}/2, N/2 + \sqrt{N}/2]$ and we give an efficient, deterministic parallel algorithm which finds a vector achieving this bound. Our algorithm is logspace uniform NC, i.e., it can be implemented by a family of logspace uniform boolean circuits of polynomial size and polylogarithmic depth. As explained at the end of the paper, the whole algorithm can also be run in logarithmic space.

Organization of the paper. Grigoriev's lemma is stated in [6] and at the beginning of this introduction in the language of linear algebra. There is an equivalent formulation in a purely set-theoretic language. Namely, we are given a set $\mathcal{F}$ of N distinct subsets of a finite set X. The goal is to find a subset F of X such that roughly $|\mathcal{F}|/2$ elements of $\mathcal{F}$ have an intersection with F of even cardinality. This set-theoretic point of view is developed in Section 2. In Section 2.1 we give a probabilistic proof of the combinatorial lemma which yields the improved range $[N/2 - \sqrt{N}/2, N/2 + \sqrt{N}/2]$. Moreover, we show that a random subset $F \subseteq X$ will fall in the slightly bigger range $[N/2 - \sqrt{N}, N/2 + \sqrt{N}]$ with probability at least $3/4$, so there is a quite simple randomized algorithm for our problem. We then show that a deterministic algorithm can be obtained by derandomizing the probabilistic proof of the combinatorial lemma. In Section 2.2 we give another (graph-theoretic) proof of the lemma which achieves the same bound as the probabilistic proof. This yields another deterministic sequential algorithm given in Section 2.3. The optimality of the bound is then discussed in Section 2.4. We return to the language of linear algebra in Section 3 to describe our parallel algorithm. Note that this algorithm relies on elementary facts about extensions of finite fields. Field extensions seem to be of an intrinsically algebraic nature, so the linear algebraic point of view seems most appropriate to state and prove the results of that section.

It would be interesting to find out whether the probabilistic proof of Section 2.1 can be derandomized to yield not only an efficient sequential algorithm, but also an efficient parallel algorithm (more on this at the end of Section 2.1). We conclude this introduction with a long quote from [12]: "A natural approach towards de-randomizing algorithms is to find a method for searching the associated sample Ω for a good point w with respect to a given input instance I. Given such a point w, the algorithm $\mathcal{A}(I, w)$ is now a deterministic algorithm and it is guaranteed to find a correct solution. The problem faced in searching the sample space is that it is generally exponential in size. The result of Adleman showing that $RP \subseteq P/poly$ implies that the sample space Ω associated with a randomized algorithm always contains a polynomial-sized subspace which has a good point for each possible input instance. However, this result is highly non-constructive and it appears that it cannot be used to actually de-randomize algorithms." Our paper gives an example of a problem for which this "Adlemanian" approach to derandomization is actually feasible. Indeed, our parallel algorithm constructs a polynomial-size list of "candidate vectors" which for any set of N input vectors is guaranteed to contain a vector orthogonal to roughly $N/2$ input vectors. This list is made up of all vectors in a polynomial-size family of "candidate subspaces" of small (logarithmic) dimension. Once the list is constructed we only have to solve an exhaustive search problem, and this can be done quite easily in parallel.

2 The set theoretic point of view

In this section we study the set theoretic formulation of our problem: X is a finite set and $\mathcal{F}$ a set of N nonempty distinct subsets of X. The goal is to find a subset F of X such that the number of elements of $\mathcal{F}$ which have an odd intersection with F is as close as possible to $\frac{|\mathcal{F}|}{2}$.

2.1 A probabilistic proof

The first natural idea for this problem is to take for F a random subset of X.

Theorem 1. *Let X be a finite set and $\mathcal{F}$ be a set of N nonempty subsets of X. There is a subset $F \subseteq X$ such that*

$$-\frac{\sqrt{N}}{2} \leq |\{F_i \in \mathcal{F} : |F \cap F_i| \text{ even}\}| - \frac{N}{2} \leq \frac{\sqrt{N}}{2}. \tag{1}$$

Proof. Call $F_1, \ldots, F_N$ the elements of $\mathcal{F}$. We choose a random subset F of X obtained by selecting or not every element of X with probability $1/2$.

Let Y_i be the random variable defined by:

$$Y_i = 1 \text{ if } |F \cap F_i| \text{ is even, and } Y_i = -1 \text{ otherwise.}$$

Therefore we are interested in the random variable

$$Y = \sum_{i=1}^{N} Y_i = |\{i : |F \cap F_i| \text{ even}\}| - |\{i : |F \cap F_i| \text{ odd}\}| = 2|\{i : |F \cap F_i| \text{ even}\}| - N.$$

We want to show that there exists an F for which $|Y| \leq \sqrt{N}$, i.e. $Y^2 \leq N$.

First, let us prove that $P(Y_i = 1) = 1/2$. This follows immediately from the facts that every subset F occurs with same probability and that there are as many odd as even subsets in each F_i. Thus $E(Y_i) = 0$.

Then we prove that the events $\{Y_i = 1\}$ are *pairwise*[6] independent. For this let us consider two elements F_1 and F_2 of $\mathcal{F}$. We have to prove that

$$P(Y_1 = 1 \cap Y_2 = 1) = P(Y_1 = 1)P(Y_2 = 1) = 1/4. \tag{2}$$

There are three cases:

[6] It can be shown that these events are not always 3-wise independent.

- F_1 and F_2 are disjoint. In this case, it is clear that the events are independent.
- $F_1 \subseteq F_2$. This case can be reduced to the previous one for F_1 and $F_2 \setminus F_1$ and we still have (2).
- The three sets $A = F_1 \setminus F_2$, $B = F_1 \cap F_2$ et $C = F_2 \setminus F_1$ are nonempty. Then $Y_1 = 1$ and $Y_2 = 1$ is equivalent to $|A \cap F| \equiv |B \cap F| \equiv |C \cap F| \bmod 2$. But since these three sets are disjoint, we have a probability $1/8$ to be in the case even-even-even and $1/8$ to be in the case odd-odd-odd. Eventually, we also have (2).

Since the events are pairwise independent we have $E(Y_iY_j) = E(Y_i)E(Y_j) = 0$ if $i \neq j$. Furthermore, $E(Y_i^2) = 1$ so by linearity of the expectation we have

$$E(Y^2) = E(\sum_{i=1}^{N} Y_i^2 + \sum_{i \neq j} Y_iY_j) = N.$$

Hence there exists F for which $Y^2 \leq N$: this is the desired set. □

Remark 1. In the above proof, taking into account the fact that $Y^2 = N^2$ for $F = \emptyset$, we obtain $E(Y^2 | F \neq \emptyset) < N$. Thus there exists a set F for which the inequality is strict, i.e. $Y^2 < N$. In other words, there exists a set F satisfying the stronger inequality:

$$-\frac{\sqrt{N}}{2} < |\{F_i \in \mathcal{F} : |F \cap F_i| \text{ even}\}| - \frac{N}{2} < \frac{\sqrt{N}}{2}.$$

Remark 2. The pairwise independence of the Y_i enables us to evaluate the variance of Y: $Var(Y) = \sum_{i=1}^{N} Var(Y_i) = N$. By Tchebycheff's inequality, we have:

$$P(|Y - E(Y)| > 2\sqrt{N}) = P(|Y| > 2\sqrt{N}) < Var(Y)/(2\sqrt{N})^2 = 1/4.$$

This ensures that at least $3/4$ of the subsets F fall within the range $[N/2 - \sqrt{N}, N/2 + \sqrt{N}]$, and yields a trivial randomized algorithm for finding such a set. The deterministic algorithms of Proposition 1, Section 2.3 and Section 3 achieve however the better range $[N/2 - \sqrt{N}/2, N/2 + \sqrt{N}/2]$ obtained in the theorem.

We now show how to derandomize the proof of Theorem 1 by the method of conditional expectations, in order to obtain a deterministic algorithm. Note that a simpler deterministic algorithm will be presented in Section 2.3.

Proposition 1. *The proof of Theorem 1 can be derandomized using the method of conditional expectations. This yields a polynomial-time deterministic algorithm for finding a set of even intersection with at least $N/2 - \sqrt{N}/2$ and at most $N/2 + \sqrt{N}/2$ of the F_i's.*

Proof. Following the proof of Theorem 1, this amounts to finding a set F for which $Y^2 \leq N$. We build such a set by enumerating the elements of X and deciding in turn for each $x \in X$ whether it must belong to F. Along the way, we keep $E(Y^2)$ bounded above by N, thus giving a guarantee that the final set F will have the expected property.

At the beginning, we know from the proof of Theorem 1 that $E(Y^2) \leq N$. At each subsequent step, we have already determined for some elements whether they belong to F: let us call C this condition (for example, $C \equiv (x_1 \in F) \wedge (x_2 \notin F)$). By induction hypothesis we have $E(Y^2|C) \leq N$. The next step is to determine whether an element $x \in X$ is in F. We have:

$$E(Y^2|C) = 1/2(E(Y^2|C \wedge (x \in F)) + E(Y^2|C \wedge (x \notin F))).$$

Therefore there exists a choice c (either $c \equiv (x \in F)$ or $c \equiv (x \notin F)$) for which $E(Y^2|C \wedge c) \leq E(Y^2|C) \leq N$. We then move on to the next step according to this choice: this will ensure that the induction hypothesis is satisfied at the next step. At the end of the algorithm, i.e., when every element $x \in X$ has been considered, the set F obtained satisfies $E(Y^2|F) \leq N$, and hence the statement of Theorem 1.

The only remaining point to settle is how to compute $E(Y^2|C \wedge (x \in F))$ and $E(Y^2|C \wedge (x \notin F))$: we need these values in order to make our choice. More generally, we want to be able to compute $E(Y^2|C)$ for an arbitrary condition C:

$$C \equiv \bigwedge_{x \in A} (x \in F) \wedge \bigwedge_{x \in B} (x \notin F).$$

Let T_i be the random variable defined by

$$T_i = 1 \text{ if } |(F_i \setminus (A \cup B)) \cap F| \text{ is even, and } T_i = -1 \text{ otherwise.}$$

We have $E(Y_i|C) = (-1)^{|F_i \cap A|} E(T_i)$.

Note that some sets $F_i \setminus (A \cup B)$ can be equal (even if by assumption the F_i's are different), and can even be empty, thus evaluating $E(Y^2|C)$ amounts to computing the expectation of Z^2 where $Z = \sum_i \alpha_i Y_i$ for a set $\{F_1, \ldots, F_k\}$ of (possibly empty) subsets of X together with weights $\alpha_1, \ldots, \alpha_k \in \mathbb{Z}$. As in the proof of Theorem 1, the events $\{Y_i = 1\}$ are pairwise independent. Furthermore, if $F_i = \emptyset$ then of course $E(Y_i) = 1$, otherwise $E(Y_i) = 0$. Finally, $E(Y_i^2) = 1$ for any i. Thus computing $E(Z^2)$ is easy, because

$$E(Z^2) = E(\sum_i \alpha_i^2 Y_i^2 + \sum_{i \neq j} \alpha_i \alpha_j Y_i Y_j) = \sum_i \alpha_i^2 E(Y_i^2) + \sum_{i \neq j} \alpha_i \alpha_j E(Y_i) E(Y_j).$$

This implies that in polynomial time one can compute $E(Y^2|C \wedge (x \in F))$ and $E(Y^2|C \wedge (x \notin F))$, and decide whether x should be taken in F or not. The construction of F thus requires $|X|$ steps, each computable in polynomial time: the overall deterministic algorithm finds a set with the expected property in polynomial time. $\square$

As explained in the introduction, the main goal of Section 3 is to obtain a deterministic parallel algorithm for our problem. It would be interesting to obtain such an algorithm from a different derandomization of Theorem 1. The main derandomization method that yields efficient parallel algorithms is the method of bounded independence, as described for instance in Section 15.2 of [1]. At first sight it looks like this method might be applicable since the proof of Theorem 1 is based on the pairwise independence of the random variables Y_i. Unfortunately, the method is not applicable directly because Y_i is defined only indirectly through the formula

$$Y_i = 1 \text{ if } |F \cap F_i| \text{ is even, and } Y_i = -1 \text{ otherwise.}$$

One must therefore construct a small sample space not for the Y_i but for the random set F. This is achieved in Section 3 through an ad-hoc method.

2.2 A graph-theoretic proof

Here we model the problem as a cut problem in a bipartite graph. We want to find a subset F that minimizes the range between $|\{i : |F \cap F_i| \text{ even}\}|$ and $|\{i : |F \cap F_i| \text{ odd}\}|$. But this means exactly finding F that maximizes the number of pairs $\{F_i, F_j\}$ with $|F \cap F_i| \not\equiv |F \cap F_j| \bmod 2$. Indeed, if t denotes $|\{i : |F \cap F_i| \text{ odd}\}| - \frac{N}{2}$, the number of such pairs is exactly $(N/2-t)(N/2+t) = N^2/4 - t^2$. The crucial fact is that if $F \subseteq X$ and F_i, F_j are two elements of $\mathcal{F}$:

$$|F \cap F_i| \not\equiv |F \cap F_j| \bmod 2 \iff |F \cap (F_i \bigtriangleup F_j)| \equiv 1 \bmod 2.$$

Thus, finding F that minimizes the range between $|\{i : |F \cap F_i| \text{ even}\}|$ and $|\{i : |F \cap F_i| \text{ odd}\}|$, is exactly finding F that maximizes $|\{(i,j) : |F \cap (F_i \bigtriangleup F_j)| \text{ odd}\}|$.

We consider the following bipartite graph (V, E):

- $V = V_1 \cup V_2$ where $V_1 = \{(i,j) : 1 \le i < j \le N\}$, and $V_2 = \mathcal{P}(X)$;
- $((i,j), F) \in E$ iff $|F \cap (F_i \bigtriangleup F_j)|$ odd.

What we are looking for is a vertex of V_2 of maximum degree. Let $N(x)$ denote the set of neighbours of x. We will only need to apply the following lemma for $A = V_2$, as in Lemma 2. However it turns out that we can characterize in Lemma 1 all the subsets $A \subseteq V_2$ for which the proof still holds (see also Remark 3 in Section 3).

Lemma 1. *Let $A \subseteq V_2$ be such that $\emptyset \in A$ and $\forall F, F' \in A$, $(F \bigtriangleup F') \in A$. Assume moreover that $\forall x \in V_1$, $N(x) \cap A \neq \emptyset$. Then*

$$\forall x \in V_1,\ |N(x) \cap A| = \frac{|A|}{2}.$$

Proof. Let $x \in V_1$. By hypothesis, there exists $F \in A$ such that (x, F) is an edge of the graph. And by the other hypothesis the following map is well-defined,

$$\begin{aligned} \phi :\ A &\longrightarrow A \\ F' &\longmapsto (F \bigtriangleup F') \end{aligned}$$

and is a bijection of $N(x) \cap A$ onto $A \setminus N(x)$ which proves the result. $\square$

Lemma 2. *There exists a subset $A \subseteq V_2$ satisfying the hypothesis of Lemma 1.*

Proof. It suffices to take $A = V_2$. $\square$

Corollary 1. *There exists $F \in V_2$ such that $|N(F)| \geq \frac{|V_1|}{2}$*

Proof. By Lemmas 1 and 2, every $x \in V_1$ has $|N(x)| = |V_2|/2$ neighbours. By double counting, there exists an $F \in V_2$ satisfying the hypothesis of the corollary. $\square$

Corollary 2. *There exists $F \subset X$ such that $||\{i : |F \cap F_i| \text{ even}\}| - \frac{N}{2}| \leq \frac{\sqrt{N}}{2}$*

Proof. Let F be given by Corollary 1. Define $t = |\{i : |F \cap F_i| \text{ odd}\}| - \frac{N}{2}$. Then $|N(F)| = (\frac{N}{2} + t)(\frac{N}{2} - t) = \frac{N^2}{4} - t^2$ and by hypothesis on F:

$$\frac{N^2}{4} - t^2 \geq \frac{|V_1|}{2} = \frac{N(N-1)}{4} = \frac{N^2 - N}{4}$$

which implies $|t| \leq \frac{\sqrt{N}}{2}$. $\square$

2.3 A simple deterministic polynomial time algorithm

We now present a very simple polynomial algorithm which finds a subset F achieving inequality (1) from Theorem 1. We work from the point of view described in Section 2.2: given the subsets F_i, we need to find a subset F that has an odd intersection with more than half of the $F_i \triangle F_j$ (taking multiplicities into account). Note that these symmetric differences are all nonempty since the F_i are distinct. The algorithm goes this way.

1. We construct all the sets $F_i \triangle F_j$ and denote by $\mathcal{G}$ the multiset obtained.
2. Let $x \in X$. Let $\mathcal{G}'$ be the multiset of all elements of $\mathcal{G}$ not containing x. Apply recursively the algorithm to $X \setminus \{x\}$ and $\mathcal{G}'$. Thus we get a subset F' of $X \setminus \{x\}$ that has an odd intersection with more than half of the elements of $\mathcal{G}'$. Now there are two cases:
   – F' has an odd intersection with more than half of the elements of $\mathcal{G} \setminus \mathcal{G}'$. In this case $F = F'$ is a solution to our problem.
   – Otherwise, since x belongs to all elements of $\mathcal{G} \setminus \mathcal{G}'$, taking $F = F' \cup \{x\}$ gives a solution.

2.4 Discussion of the bounds

With the help of Theorem 1, we know that it is possible to reach the expected value within a range of order $\sqrt{N}$. One can wonder whether it is possible to ensure a constant range. The following examples prove that this is impossible.

Let us consider a set X with $n = 4k^2 + 1$ elements and $\mathcal{F}$ be the set of all subsets of X of size 2. Let $N = |\mathcal{F}| = n(n-1)/2$. In this context, the problem is to partition X into two parts and count the number of edges through the cut, which are precisely the sets of $\mathcal{F}$ with odd intersection. We want to find $0 \leq a \leq n/2$ such that $a(n-a)$ is as close as possible to $N/2 = k^2(4k^2+1)$. But:

$$(2k^2 - k)(2k^2 + k + 1) = 4k^4 + k^2 - k$$

and

$$(2k^2 - k + 1)(2k^2 + k) = 4k^4 + k^2 + k.$$

The function $a \longmapsto a(n-a)$ being increasing on $[0, n/2]$, this proves that these are the two best values and that the error is at least k, which is of the order of $N^{1/4}$. It is possible to refine this argument further. For instance, the consideration of subsets with 3 elements instead of 2 yields the following result.

Proposition 2. *Let $\mathcal{F}_n$ be the family of subsets of three elements of $\{1, \ldots, n\}$. There exists a constant $c > 0$ such that for infinitely many n, for any subset G of $\{1, \ldots, n\}$,*

$$\left| |\{F \in \mathcal{F}_n : |F \cap G| \ \textit{even}\}| - \frac{|\mathcal{F}_n|}{2} \right| \geq c|\mathcal{F}_n|^{1/3}.$$

Proof. Let $F \subseteq \{1, \ldots, n\}$ be a subset of cardinality a. The number of elements of $\mathcal{F}_n$ whose intersection with F is of odd cardinality is then $a\binom{n-a}{2} + \binom{a}{3}$. Therefore, let

$$f(a) = \frac{a(n-a)(n-a-1)}{2} + \frac{a(a-1)(a-2)}{6} - \frac{n(n-1)(n-2)}{12}$$

be the difference with $|\mathcal{F}_n|/2$. We aim at showing that f is far from zero on integer values, when n is well chosen.

The zeros of f are $n/2$ and $n/2 \pm \sqrt{3n-2}/2$. From the variations of f, we see that the integers i so that $|f(i)|$ is minimal are among the six integers around the zeros. Intuitively, these values should be maximized if the zeros are far from integers (that is, if they are near half-integers). This requires n to be odd and $\sqrt{3n-2}/2$ to be near an integer (i.e. $3n - 2 \simeq 4k^2$ for some k).

These considerations lead to the choice $n = 4k^2/3 + 1$ where $k \equiv 0 \bmod 3$. The integer n is then odd, so

$$f(\lfloor n/2 \rfloor) = f(n/2 - 1/2) = n/4 - 1/4$$

$$f(\lceil n/2 \rceil) = f(n/2 + 1/2) = -n/4 + 1/4$$

Furthermore, if $k \geq 2$ then $\sqrt{3n-2} = \sqrt{4k^2+1}$ is at most $1/8$ away from $2k$, so that

$$\begin{aligned} f(\lfloor n/2 + \sqrt{3n-2}/2 \rfloor) &= f(n/2 - 1/2 + k) \\ &= f(n/2 - 1/2 + \sqrt{3n-3}/2) = -n/2 + O(\sqrt{n}). \end{aligned}$$

Similarly, the other three integers around the zeros have $\Omega(n)$ as image. Since the total number N of subsets of three elements among n is $O(n^3)$, the error is at least $\Omega(N^{1/3})$. □

The same kind of calculations for subsets with 5 elements yields an $\Omega(|\mathcal{F}_n|^{2/5})$ lower bound. The best lower bound that we have obtained is $\Omega(\sqrt{|\mathcal{F}_n|}/(\log|\mathcal{F}_n|)^{1/4})$. As shown below, this almost optimal lower bound is achieved by taking for $\mathcal{F}_n$ the set of all subsets of size $(n-1)/2$.

Theorem 2. *Let $\mathcal{F}_n$ be the family of subsets of $(n-1)/2$ elements of $\{1,\ldots,n\}$, where n is an odd integer. There exists a constant $c>0$ such that for infinitely many n, for any subset G of $\{1,\ldots,n\}$,*

$$\left| |\{F\in\mathcal{F}_n : |F\cap G| \text{ even}\}| - \frac{|\mathcal{F}_n|}{2}\right| \geq c\sqrt{|\mathcal{F}_n|}/(\log|\mathcal{F}_n|)^{1/4}.$$

Proof. Recall the definition of the binomial coefficient: for $x\in\mathbb{R}$ and $k\in\mathbb{N}$,

$$\binom{x}{k} = \frac{\prod_{i=0}^{k-1}(x-i)}{k!}.$$

The special case when x is half an integer will be useful. Namely, for $n<k-1$ we have

$$\binom{n+1/2}{k} = \frac{\prod_{i=0}^{k-1}(n+1/2-i)}{k!} = \frac{(-1)^{k-n+1}(2n+1)!(2k-2n-3)!}{2^{2k-2}n!(k-n-2)!k!}. \quad (3)$$

Now, let us consider a set X with $n=4k+1$ elements and let $\mathcal{F}$ be the set of all subsets of X of size $2k$. The number of sets in $\mathcal{F}$ that a set Y of cardinality j intersects an even number of times is:

$$f(j) = \sum_{p\,\text{even}} \binom{j}{p}\binom{n-j}{2k-p}.$$

The total number of sets is

$$|\mathcal{F}| = \binom{n}{2k} = \sum_p \binom{j}{p}\binom{n-j}{2k-p},$$

so that we are interested in the quantity

$$g(j) = f(j) - \frac{|\mathcal{F}|}{2} = \frac{1}{2}\sum_p (-1)^p \binom{j}{p}\binom{n-j}{2k-p}.$$

Our immediate goal is to prove that

$$g(j) = -4^{2k}\left(\binom{(j-1)/2}{2k+1} - \binom{j/2}{2k+1}\right). \quad (4)$$

We start from the following identity ([15], identity 3.42 or [13]):

$$\sum_p (-1)^p \binom{j}{p}\binom{2m-j}{m-p} = (-4)^m \binom{(j-1)/2}{m}.$$

It is not difficult to check that

$$\binom{j}{p}\binom{4k+1-j}{2k-p}-\binom{j}{p-1}\binom{4k+1-j}{2k-(p-1)} = \binom{j}{p}\binom{4k+2-j}{2k+1-p}-\binom{j+1}{p}\binom{4k+2-(j+1)}{2k+1-p}.$$

As a consequence,

$$\begin{aligned} 2\textstyle\sum_p(-1)^p\binom{j}{p}\binom{4k+1-j}{2k-p} &= \textstyle\sum_p(-1)^p\left[\binom{j}{p}\binom{4k+1-j}{2k-p}-\binom{j}{p-1}\binom{4k+1-j}{2k-(p-1)}\right] \\ &= \textstyle\sum_p(-1)^p\left[\binom{j}{p}\binom{4k+2-j}{2k+1-p}-\binom{j+1}{p}\binom{4k+2-(j+1)}{2k+1-p}\right] \\ &= (-4)^{2k+1}\left(\binom{(j-1)/2}{2k+1}-\binom{j/2}{2k+1}\right), \end{aligned}$$

which proves (4). When j is even, $g(j)$ reduces to

$$g(j) = -(-4)^{2k}\binom{(j-1)/2}{2k+1}.$$

This is a product of half integers. This product is therefore minimal in absolute value when it is centered around 0, that is when $j = 2k$ or $j = 2k+2$. In both cases, we have

$$|g(j)| = 4^{2k}\left|\binom{k-1/2}{2k+1}\right|.$$

When j is odd, $|g(j)|$ reduces to

$$|g(j)| = 4^{2k}\binom{j/2}{2k+1}$$

which is minimal when $j = 2k-1$ or $j = 2k+1$. The minimum is the same as in the even case. By (3), the minimal absolute value that g takes is therefore

$$\mu = 4^{2k}\left|\binom{k-1/2}{2k+1}\right| = \binom{2k-1}{k} \sim \frac{2^{2k-1}}{\sqrt{\pi k}}$$

whereas

$$|\mathcal{F}| = \binom{4k+1}{2k} \sim \frac{2^{4k+1}}{\sqrt{2\pi k}}.$$

Hence $\mu = \Omega(\sqrt{|\mathcal{F}|}/\sqrt[4]{\log|\mathcal{F}|})$.

$\square$

3 The linear algebraic point of view

In this section we are concerned with a parallel algorithm for our problem. More precisely, we shall build a *logspace-uniform* family of circuits of polylogarithmic depth for our problem. In the meantime we are led to exhibit another polynomial-time sequential algorithm, which is a first step towards the parallel one.

We use here techniques of linear algebra, dealing now with 0-1 vectors instead of sets. Let us first formulate Theorem 1 in these terms.

Corollary 3. *Let $u_1, \ldots, u_N \in E = (\mathbb{F}_2)^d$ be distinct nonzero vectors. There exists a vector $v \in E$ such that*

$$-\frac{\sqrt{N}}{2} \leq |\{1 \leq i \leq N : u_i.v = 0\}| - \frac{N}{2} \leq \frac{\sqrt{N}}{2}.$$

In what follows, a vector $v \in E$ as in the corollary is called "good" for $u_1, \ldots, u_N$. We now turn to two algorithms for finding a good vector. As input we have N distinct nonzero vectors $u_1, \ldots, u_N$ of E, given by their coordinates (hence the size of the input is of order Nd). The output will be a good vector for $u_1, \ldots, u_N$. The principle of the algorithms is to restrict the search to a small set V where a suitable vector v is guaranteed to exist. If this "sample space" is small enough, we will then be able to find the vector by exhaustive search.

3.1 Existence of a small sample space

Lemma 3. *Let V be a subspace which is orthogonal to none of the $u_i - u_j$ (i.e. for all $1 \leq i < j \leq N$, there is $v \in V$ so that $v.(u_i - u_j) = 1$) and to none of the u_i. Then there exists a good vector $v \in V$ for $u_1, \ldots, u_N$.*

Proof. Let $v_1, \ldots, v_k$ be a basis of V. The condition that V is orthogonal to none of the $u_i - u_j$ implies that the new vectors u'_i defined by $u'_i = \sum_{l=1}^{k} (u_i.v_l)v_l$ are pairwise distinct. This is because for all $i \neq j$, there exists l such that $u_i.v_l \neq u_j.v_l$. Moreover, the condition that V is orthogonal to none of the u_i implies that none of the u'_i is equal to zero. In geometric terms, u'_i may be thought of as the projection onto V.

We now define on V a new product $\odot : V \times V \to \mathbb{F}_2$ (associated to the basis $(v_1, \ldots, v_k)$) by the formula $(\sum_l \lambda_l v_l) \odot (\sum_l \mu_l v_l) = \sum_l \lambda_l \mu_l$. For this new product (which comes just from a change of basis compared to the original inner-product), Corollary 3 asserts the existence of $w = \sum_l \lambda_l v_l$ which is $\odot$-orthogonal to at least $N/2 - \sqrt{N}/2$ vectors u'_i and at most $N/2 + \sqrt{N}/2$. But $w \odot u'_i = \sum_l \lambda_l v_l \odot u'_i = \sum_l \lambda_l (u_i.v_l) = w.u_i$, and thus w is also suitable for E as a whole (with the usual product on E). □

Remark 3. The above lemma can also be derived from the set theoretic point of view as a consequence of Lemma 1. Note in particular that in Lemma 1, the hypothesis on A of stability under symmetric differences simply means from the linear algebra viewpoint that A is a linear subspace.

We now show that the subspace of Lemma 3 can have small dimension. Recall that E is a vector space over $\mathbb{F}_2$ of dimension d.

Lemma 4. *Let U be a subset of E not containing 0. Then there exists a subspace W of E, of dimension $\geq d - \log(|U| + 1)$, which does not intersect U.*

Proof. By induction on the dimension d of E. For $d = 0$, $|U| = 0$ and the result trivially follows.

Assume $d > 0$. If $|U| = 2^d - 1$, i.e. $U = E \setminus \{0\}$, we can choose $W = \{0\}$. Hence we shall assume that there exists a nonzero vector w_0 in $E \setminus U$. Let W_0 be the subspace (with two elements) generated by w_0. If $|U| \geq 2^{d-1} - 1$, then W_0 suits our needs. Otherwise, E/W_0 is a vector space of dimension $d-1$ and we can apply the induction hypothesis to the set $\bar{U}$ of the classes of elements of U, which are all different from zero. This set satisfies $|\bar{U}| \leq |U|$, hence there exists a subspace $\bar{W}_1$ of E/W_0 of dimension $\geq d - 1 - \log(|U|+1)$, which does not intersect $\bar{U}$.

Call W_1 the subspace of E of dimension $1+\dim(\bar{W}_1)$, consisting of all elements of all classes of $\bar{W}_1$. By definition of E/W_0, W_1 does not intersect U, and is of dimension $\geq d - \log(|U|+1)$. □

We now apply Lemma 4 to $U = \{u_i - u_j : 1 \leq i < j \leq N\} \cup \{u_i : 1 \leq i \leq N\}$: we have $|U| = N(N+1)/2$. Hence there exists a subspace W of E of dimension at least $d - 2\log N$ that does not contain any of the $u_i - u_j$ and of the u_i[7]. The orthogonal space V of W is then of dimension $\leq 2\log N$ and is orthogonal to none of the $u_i - u_j$ and to none of the u_i (because $V^\perp = W^{\perp\perp} = W$, as is easily verified). Note that V contains at most N^2 elements.

This gives a polynomial sequential algorithm for finding a good vector (we only sketch it since we have already described a simpler sequential algorithm in Section 2.3):

1. Find a basis $e_1, \ldots, e_b$ of a subspace W of dimension $\geq d - 2\log N$ which does not contain any of the $u_i - u_j$ and of the u_i (for $1 \leq i < j \leq N$). This is done by induction, taking the quotient space at each step as in the proof of Lemma 4.
2. Find the orthogonal space V of W. This is done by solving the linear system $(e_i.x = 0)_{1 \leq i \leq b}$.
3. Find a good vector v in V. This is done by exhaustive search.

3.2 A parallel algorithm

As in the sequential algorithm sketched above, we plan to perform an exhaustive search in a small sample space. The use of Lemma 4 for finding a sample space is unfortunately intrinsically sequential, since the proof works inductively in a quotient space.

In fact, there is no reason to restrict the search to only one subspace: an exhaustive search can also be performed in polynomially many subspaces of small dimension in parallel. An idea to overcome the difficulty of using Lemma 4 then consists in the following. At the beginning of the algorithm, we build a family of subspaces of large dimension $\mathcal{W} = \{W_1, \ldots, W_k\}$ that is "generic" in the sense that for all subsets $U \subseteq E \setminus \{0\}$ of cardinality $N(N+1)/2$, there exists $W_l \in \mathcal{W}$ for which $U \cap W_l = \emptyset$.

[7] This follows from the inequality $\log(N(N+1)/2+1) \leq 2\log N$, which holds true for $N \geq 2$. There is no loss of generality in assuming that $N \geq 2$ since any vector $v \in E$ will satisfy Corollary 3 for $N = 1$.

For the particular choice $U = \{u_i - u_j \,:\, 1 \le i < j \le N\} \cup \{u_i \,:\, 1 \le i \le N\}$ we see that at least one W_l contains none of the $u_i - u_j$ or of the u_i, so by Lemma 3 $W_l^\perp$ must contain a good vector. If the W_l's are of sufficiently large dimension, $W_l^\perp$ has only polynomially many elements and can be searched efficiently. This yields the following theorem, which is proved in the sequel.

Theorem 3. *There is a parallel algorithm which, given two positive integers N and d with $N \le 2^d$, builds in time $O(\log N + \log d \log\log(dN))$ a family $\mathcal{F}$ of $d^2N^2(N+1)^2$ elements of $(\mathbb{F}_2)^d$ that contains, for any distinct nonzero vectors $u_1, \ldots, u_N \in (\mathbb{F}_2)^d$, a vector v such that*

$$N/2 - \sqrt{N}/2 \le |\{1 \le i \le N \,:\, u_i.v = 0\}| \le N/2 + \sqrt{N}/2.$$

An exhaustive search in this family can therefore be performed in $O(\log(dN))$ parallel time, enabling us to find a good vector v on input $u_1, \ldots, u_N$ in polylogarithmic parallel time $O(\log N + \log d \log\log(dN))$.

In Section 3.3, we show that a generic family $\mathcal{W} = \{W_1, \ldots, W_k\}$ for sets U of size $N(N+1)/2$ indeed exists and can be built efficiently. Our family is of cardinality $k \le 2d|U|$ and each subspace W_l of dimension at least $d-1-\log(d|U|)$. The W_l's will be given as intersection of hyperplanes, so that a spanning family of $W_l^\perp$ is immediately found. In Section 3.3, U denotes an arbitrary subset of $E \setminus \{0\}$. As explained above, a typical choice for U will be $\{u_i - u_j \,:\, 1 \le i < j \le N\} \cup \{u_i \,:\, 1 \le i \le N\}$.

3.3 A generic family of subspaces

To allow more room, we first work in a field extension of $\mathbb{F}_2$. More precisely, we fix an extension K of degree $e > \log((d-1)|U|)$, so that there are more than $(d-1)|U|$ elements in K. Note that for $|U| = N(N+1)/2$, a suitable choice is $e = \lfloor \log(dN(N+1)) \rfloor$. This is the choice which will be made in Section 3.4.

We look at K as $\mathbb{F}_2[X]/(P(X))$ where $P(X)$ is an irreducible polynomial of $\mathbb{F}_2[X]$ of degree e. Thus the elements of K will be viewed as classes of polynomials modulo P. Once the polynomial P is found, it is easy to calculate in K, by manipulating polynomials of degree less than e with coefficients in $\mathbb{F}_2$ (details will be given in Section 3.5).

In K^d, we are able to find $|K|$ hyperplanes so that every set of cardinality $|U|$ has an empty intersection with at least one of them. For every $\theta \in K$, let us indeed consider the hyperplane H_θ of K^d defined by the equation $x_1 + \theta x_2 + \theta^2 x_3 + \cdots + \theta^{d-1} x_d = 0$. There are $|K| > (d-1)|U|$ different hyperplanes in this family $(H_\theta)_{\theta \in K}$, and a point $a \in K^d \setminus \{0\}$ belongs to at most $d-1$ distinct hyperplanes: this is due to the fact that there are at most $d-1$ distinct roots of the polynomial $P(\theta) = a_1 + a_2\theta + \cdots + a_d\theta^{d-1}$. Thus among these hyperplanes, at least one does not intersect U.

To obtain our family over $\mathbb{F}_2$ (instead of K), we now consider the trace of H_θ on $\mathbb{F}_2^d$. For $(x_1, \ldots, x_d) \in \mathbb{F}_2^d$, the equation of the hyperplane H_θ can be rewritten according to the powers of X:

$$x_1 + \theta x_2 + \cdots + \theta^{d-1} x_d \equiv \sum_{i=0}^{e-1} \mu_i(x_1, \ldots, x_d) X^i \pmod{\mathrm{P}}$$

where the μ_i are $\mathbb{F}_2$-linear combinations of the x_j (the coefficient of x_j in μ_i is equal to the X^i-coordinate of θ^{j-1} in the $\mathbb{F}_2$-basis $1, X, \ldots, X^{e-1}$ of K). The intersection $W_\theta = H_\theta \cap \mathbb{F}_2^d$ is then defined by the system of equations $\mu_i(x) = 0$ where i ranges over $\{0, 1, \ldots, e-1\}$. It is therefore a subspace of $E = (\mathbb{F}_2)^d$ of codimension at most e.

This construction yields a family $\mathcal{W} = \{W_1, \ldots, W_k\}$ of $k \leq 2^e$ subspaces with the expected genericity property: for all subsets $U \subseteq E \setminus \{0\}$ of cardinality $N(N+1)/2$, there exists $W_l \in \mathcal{W}$ for which $U \cap W_l = \emptyset$. Since e can be taken $\leq 1 + \log(d|U|))$, we get at most $2d|U|$ subspaces, of dimension at least $d - 1 - \log(d|U|)$ each. As promised, these subspaces are given as intersections of hyperplanes.

3.4 High level description of the algorithm

Let us sum up the main steps of this parallel algorithm. Its implementation and analysis are discussed in the next section. The input is a set $\{u_1, \ldots, u_N\}$ of N distinct nonzero vectors of $E = (\mathbb{F}_2)^d$, and the output is a vector orthogonal to at least $N/2 - \sqrt{N}/2$ and at most $N/2 + \sqrt{N}/2$ of them.

1. Let $e = \lfloor \log(dN(N+1)) \rfloor$. By enumerating in parallel all the polynomials of $\mathbb{F}_2[X]$ of degree e, find an irreducible polynomial P. Let $K = \mathbb{F}_2[X]/(P(X))$.
2. Consider the family $\mathcal{F}$ of hyperplanes in K^d consisting of the $|K| = 2^e$ hyperplanes $(H_\theta)_{\theta \in K}$ described in Section 3.3. Rewrite the equation of each hyperplane of $\mathcal{F}$ as a system of e equations in $\mathbb{F}_2$. This is only a rearrangement of terms. We obtain one subspace W_θ of $(\mathbb{F}_2)^d$ of codimension at most e for each hyperplane H_θ. As a whole, this generic family thus contains at most 2^e subspaces of $(\mathbb{F}_2)^d$.
3. Search in parallel in $W^\perp$, for all W in the generic family. A good vector must exist in at least one of them (note that it is only this third step which actually depends on the input).

As explained in the next section, the execution time of this algorithm is polylogarithmic in the size dN of the input.

3.5 Implementation and analysis

We need now explain how to perform this procedure quickly in parallel. First, in order to find an irreducible polynomial $P \in \mathbb{F}_2[X]$ of degree e, we merely enumerate in parallel all polynomials $A \in \mathbb{F}_2[X]$ of degree e and test their irreducibility. There are $2^e \leq dN(N+1)$ such polynomials. The polynomial A is irreducible if and only if it is not divisible by another non-constant polynomial of degree $\leq e/2$. This yields a straightforward irreducibility test: compute in parallel the division with remainder of A by all non-constant polynomials B of degree $\leq e/2$ and test whether one of the remainders is zero. Finding P therefore takes parallel time $O(e) + T(e)$, where $T(e)$ is the cost of a division in $\mathbb{F}_2[X]$. Hence we only need to use a division algorithm of parallel complexity $O(e)$. Within that generous time bound we may even try in parallel all possible quotients Q, and check whether $A = BQ$. Some parallel division algorithms are of course much faster (but overcomplicated for the problem at hand), see for instance [3].

We now proceed to the second step of the algorithm, which we begin with a preliminary computation. Let P be the irreducible polynomial found at the first step, and let $K = \mathbb{F}_2[X]/(P(X))$ be the field with 2^e elements. We first compute $X^i \bmod P$ for all $i \in [e, 2(e-1)]$. The first element of this sequence is obtained immediately from P, and $X^{i+1} \bmod P$ can be obtained in constant parallel time from $X^e \bmod P$ and $X^i \bmod P$ (basically by a shift of coefficients followed by at most one addition in $\mathbb{F}_2[X]$). The whole sequence can therefore be constructed in time $O(e)$.

At step 2, our main task is to compute θ^i for all $i = 0, \ldots, d-1$ and all $\theta \in K$. By fast exponentiation θ^i can be obtained from θ by $O(\log d)$ multiplications in K, each of *boolean* cost $O(\log e)$. Indeed, to perform such a multiplication we multiply two polynomials of degree $\leq e-1$ with coefficients in $\mathbb{F}_2$ and take the remainder modulo $P(X)$. The cost of the multiplication in $\mathbb{F}_2[X]$ is $O(\log e)$, and yields a polynomial of degree at most $2(e-1)$. At the beginning of step 2 we have precomputed a representation modulo $P(X)$ of all the monomials which can possibly occur in this polynomial. Hence it simply remains to add up at most e polynomials of degree $\leq e-1$. This can be done in parallel time $O(\log e)$. The parallel cost of generating our generic family of 2^e subspaces is therefore $O(\log d \log e)$, which is $O(\log d \log\log dN)$ The orthogonal space of each subspace W_θ contains at most 2^e points since it is of dimension at most e. Altogether, we have at most $(2^e)^2$ points in the union of all orthogonal spaces. Since $2^e \leq dN(N+1)$, this yields the bound $d^2N^2(N+1)^2$ of Theorem 3. The additional cost of the explicit enumeration of all those points is $O(\log e)$ since each point is the sum of at most e spanning vectors of some orthogonal space.

Finally, we can find a good vector among the $d^2N^2(N+1)^2$ candidates in time $O(\log(dN))$ by exhaustive search. First, we compute in parallel the inner products $u_i.v$ for all inputs u_i and all candidate vectors v. This is done in depth $O(\log d)$. Then for fixed v, we have to sum over all u_i to obtain the number of i such that $u_i.v = 1$. It is well known that such an iterated addition can be performed in depth $O(\log N)$ (see for instance [2], proof of Theorem 1.7.2). To that sum we substract $N/2$ and take the absolute value, so that for every candidate v we have computed $||\{1 \leq i \leq N : u_i.v = 1\}| - N/2|$. We now have to find the minimum among the $d^2N^2(N+1)^2$ values; this can be done in depth $O(\log(d^2N^2(N+1)^2)) = O(\log(dN))$ since computing the minimum is an AC^0 problem (see for instance [2], example 6.2.2). Thus the exhaustive search requires parallel time $O(\log(dN))$ as claimed in Theorem 3.

The overall parallel execution time of our algorithm is therefore $O(\log N + \log d \log\log(dN))$, which proves the theorem.

Remark 4. This parallel algorithm can be implemented by a family of logspace uniform boolean circuits of polynomial size and polylogarithmic depth since each of the three steps of the algorithm can (note that there is some redundancy in this statement since a logspace bounded Turing machine can only construct circuit families of polynomial size).

3.6 Logarithmic space

In this section we explain why the problem at hand is also in the complexity class L of problems decided by a Turing machine using $O(\log n)$ work space. The three steps of the algorithm in Section 3.4 can indeed be performed in logarithmic space:

- Step 1 first consists in an enumeration of polynomials of logarithmic degree e and with coefficient in $\mathbb{F}_2$. This takes $O(e)$ work space. Then there is another enumeration of polynomials together with a divisibility test. This still requires $O(e)$ work space.
- Step 2 consists in arithmetic operations in K in order to compute θ^i for i from 0 to d. This amounts to multiplications of polynomials of degree e and reductions modulo $P(X)$: logarithmic space is again enough.
- Step 3 computes the orthogonal space of a vector space W given by its e equations $(w_i.x = 0)_{i \leq e}$. The orthogonal space $W^{\perp}$ is merely the vector space spanned by the w_i's. Enumerating the e coefficients for these vectors therefore suffices to enumerate all the vectors of $W^{\perp}$. This is done in work space $O(e)$. Now, checking whether a vector is good can once again be performed in logarithmic space.

This proves the following result.

Theorem 4. *There is an algorithm working in space $O(\log(dN))$ which, when given a family of N vectors of $(\mathbb{F}_2)^d$, ouputs a good vector v for this family.*

Remark 5. Returning to circuit complexity, we point out that the circuit depth obtained in Theorem 3 is by no means optimal. We have indeed chosen to describe the construction of the list of all $d^2N^2(N+1)^2$ candidate vectors explicitly as a part of our parallel algorithm, but if we work with logspace uniform circuits any precomputation requiring only logarithmic space is allowed. We have just seen that we can construct in logarithmic space the whole list of candidate vectors. After that one simply has to perform an exhaustive search, which can be realized in depth $O(\log dN)$ as explained above. This shows that our problem is in logspace uniform NC^1 (it can be argued, however, that logspace uniformity is not the right uniformity condition for NC^1; see for instance [17], chapter 4).

References

1. N. Alon and J. Spencer. *The Probabilistic Method (second edition).* Wiley Interscience Series in Discrete Mathematics and Optimization. Wiley, 2000.
2. P. Clote and E. Kranakis. *Boolean Functions and Computation Models.* Texts in Theoretical Computer Science (an EATCS Series). Springer, 2002.
3. W. Eberly. Very fast parallel polynomial arithmetic. *SIAM Journal on Computing*, 18(5):955–976, 1989.
4. H. Fournier and P. Koiran. Are lower bounds easier over the reals? In *Proc. 30th ACM Symposium on Theory of Computing*, pages 507–513, 1998.
5. H. Fournier and P. Koiran. Lower bounds are not easier over the reals: Inside PH. In *Proc. 27th International Colloquium on Automata, Languages and Programming*, volume 1853 of *Lecture Notes in Computer Science*, pages 832–843. Springer, 2000.
6. D. Grigoriev. Topological complexity of the range searching. *Journal of Complexity*, 16:50–53, 2000.
7. P. Koiran. Circuits versus trees in algebraic complexity. In *Proc. STACS 2000*, volume 1770 of *Lecture Notes in Computer Science*, pages 35–52. Springer-Verlag, 2000.
8. P. Koiran and S. Perifel. VPSPACE and a transfer theorem over the reals. Available from `http://perso.ens-lyon.fr/pascal.koiran/publications.html`.
9. S. Meiser. Point location in arrangements of hyperplanes. *Information and Computation*, 106(2):286–303, 1993.
10. F. Meyer auf der Heide. A polynomial linear search algorithm for the n-dimensional knapsack problem. *Journal of the ACM*, 31(3):668–676, 1984.
11. F. Meyer auf der Heide. Fast algorithms for n-dimensional restrictions of hard problems. *Journal of the ACM*, 35(3):740–747, 1988.
12. R. Motwani, J. Naor, and M. Naor. The probabilistic method yields deterministic parallel algorithms. *Journal of Computer and System Sciences*, 49:478–516, 1994.
13. John Riordan. *Combinatorial Identities.* Wiley, New York, 1979.
14. S. Smale. On the topology of algorithms. I. *Journal of Complexity*, 3:81–89, 1987.
15. Renzo Sprugnoli. Combinatorial Identities. Technical report, Università di Firenze, 2004.
16. V. A. Vassiliev. On decision trees for orthants. *Information Processing Letters*, 62(5):265–268, 1997.
17. H. Vollmer. *Introduction to Circuit Complexity: a Uniform Approach.* Texts in Theoretical Computer Science (an EATCS Series). Springer, 1999.