Lower Bounds for Linear Satisfiability Problems*

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

Abstract

We prove an $\Omega(n^{\lceil r/2 \rceil})$ lower bound for the following problem: For some fixed linear equation in r variables, given a set of n real numbers, do any r of them satisfy the equation? Our lower bound holds in a restricted linear decision tree model, in which each decision is based on the sign of an arbitrary affine combination of r or fewer inputs. In this model, out lower bound is as large as possible. Previously, this lower bound was known only for even r, and only for one special case. We also apply reduction arguments to achieve new lower bounds on a number of higher-dimensional geometric decision problems.

Our lower bounds follow from a relatively simple adversary argument. We use a theorem of Tarski to show that if we can construct a hard input containing infinitesimals, then for every decision tree algorithm, there exists a corresponding set of real numbers which is hard for that algorithm. Furthermore, we argue that it suffices to find a single input with a large number of "collapsible tuples", even if that input is highly degenerate, *i.e.*, there are several subsets that satisfy the equation.

*Portions of this research were done at the NSF Regional Geometry Institute at Smith College, Northampton, MA, July 1993. This research was partially supported by NSF Presidential Young Investigator grant CCR-9058440. An extended abstract of this paper appears in *Proc. 6th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 388–395, 1995.

1 Introduction

Many computational decision problems, particularly in computational geometry, can be reduced to questions of the following form: For some fixed multivariate polynomial ϕ, given a set of n real numbers, is any subset in the zero-set of ϕ? In this paper, we develop general techniques for proving lower bounds on the complexity of deciding problems of this type. In particular, we examine *linear satisfiability problems*, in which the polynomial ϕ is linear. Any r-variable linear satisfiability problem can be decided in $O(n^{(r+1)/2})$ time when r is odd, or $O(n^{r/2} \log n)$ time when r is even. The algorithms that achieve these time bounds are extraordinarily simple; even so, these are the best known upper bounds.

We consider these problems under two models of computation, both restrictions of the linear decision tree model. In the *direct query* model, each decision is based on the sign of an assignment to ϕ by r of the input variables. In the *r-linear decision tree* model, each decision is based on the sign of an arbitrary affine combination of at most r input variables. We show that in these models, any algorithm that decides an r-variable linear satisfiability problem must perform $\Omega(n^{\lceil r/2 \rceil})$ direct queries in the worst case. This matches known upper bounds when r is odd, and is within a logarithmic factor when r is even. Moreover, results of Fredman [16] establish the existence of *nonuniform* algorithms whose running times match our lower bounds exactly.

Our lower bounds are based on a relatively straightforward adversary argument. Our approach is to derive, for each algorithm in a given class, an input configuration with a large number of "collapsible tuples". If an algorithm does not perform a direct query for every collapsible tuple, our adversary creates a new "collapsed" configuration, so that the algorithm cannot distinguish between the original configuration and the collapsed one, even though the two configurations should produce different answers.

We derive these adversary configurations by applying two new tricks. First, we allow our adversary configurations to contain formal infinitesimals, instead of just real numbers. Tarski's Transfer Principle implies that for any algorithm, if there is a hard configuration with infinitesimals, then a corresponding real configuration exists with the same properties. Previously, Dietzfelbinger and Maass [7, 6] used a similar technique to prove lower bounds, using "inaccessible" numbers, or numbers having "different orders of magnitude". Unlike their technique, using infinitesimals makes it possible, and indeed sufficient, to derive a *single* adversary configuration for any problem, rather than explicitly constructing a different configuration for every algorithm.

Second, we allow our adversary configurations to be degenerate. That is, both the original configuration and the collapsed configuration contain tuples in the zero-set of ϕ. We show that such a configuration can always be perturbed into general position, so that the new configuration has just as many collapsible tuples as the original. This idea was used earlier by Erickson and Seidel [15] to derive lower bounds on some geometric degeneracy problems.

The first lower bound of this type is due to Fredman [16], who demonstrated an $\Omega(n^2)$ lower bound on the number of simple comparisons required to sort any set of the form $X + Y$. This result was generalized by Dietzfelbinger [6], who derived an $\Omega(n^{r/2})$ lower bound on the depth of any comparison tree algorithm that determines, given a set of n reals, whether any two subsets of size $r/2$ have the same sum. In our terminology, he proves a lower bound for the specific r-variable linear satisfiability problem with

$$\phi = \sum_{i=1}^{r/2} x_i - \sum_{i=1}^{r/2} x_{i+r/2}$$

in the direct query model, for all even r. Dietzfelbinger claims that his lower bound holds in the

r-linear decision tree model as well. More recent techniques of Erickson and Seidel [15] can be used to prove an $\Omega(n^2)$ lower bound for many 3-variable linear satisfiability problems in the direct query model, but there are still a number of such problems for which these techniques appear to be inadequate. With these few exceptions, the only lower bounds known for any problem of this type are $\Omega(n \log n)$ in the linear decision tree model [8, 2], in the algebraic decision tree model [20, 23], and in the algebraic computation tree model [1, 22, 23].

We also derive new lower bounds for some geometric degeneracy problems, essentially by reducing them to linear satisfiability problems. Our higher-dimensional lower bounds hold in what we call the projected r-linear decision tree model, for some constant r determined by the problem. In this model, each decision is based on a polynomial that can be written as an affine combination of r or fewer of the (say) x_1-coordinates of the input points, with coefficients that are *arbitrary* functions of the other coordinates.

Our lower bounds should be compared with the following result of Meyer auf der Heide [18]: For any fixed n, there exists a linear decision tree of depth $O(n^4 \log n)$ that decides the n-dimensional knapsack problem. This nonuniform algorithm can be adapted to solve any of the linear satisfiability problems we consider, in the same amount of time [7]. Thus, there is no hope of proving lower bounds bigger than $\Omega(n^4 \log n)$ for any of these problems in the linear decision tree model. We reiterate that our lower bounds apply only to linear decision trees where the number of terms in any query is bounded by a constant.

In Section 2, we provide some background information. In Section 3, we prove our new lower bound for linear satisfiability problems. We describe our higher-dimensional results in Section 4. Finally, in Section 5, we offer our conclusions and suggest directions for further research.

2 Background and Overview

An *algebraic decision tree* is a ternary tree in which each interior node v in the tree is labeled with a multivariate *query polynomial* $q_v \in \mathbb{R}[t_1, \ldots, t_n]$ and its branches labeled -1, 0, and $+1$. Each leaf is labelled with some value — for our purposes, these values are all either "true" or "false". Computation with such a tree works as follows. Given an input $X \in \mathbb{R}^n$, the sign of $q_v(X)$ is computed, where v is the root of the tree, and the computation proceeds recursively in the appropriate subtree. When a leaf is reached, its label is returned as the algorithm's output. (Compare [20].) A *linear* decision tree is an algebraic decision tree, each of whose query polynomials is linear. An *r-linear* decision tree is a linear decision tree, each of whose query polynomials has at most r terms. We refer to the space $\mathbb{R}^n$ of possible inputs as *configuration space*, and its individual points as *configurations*.

A *formally real field* is a field in which there are no nontrivial solutions to the equation $\sum_i a_i^2 = 0$. Formally real fields are also known as *ordered fields*, since the elements of any formally real field can be given a strict linear ordering. A *real closed field* is an ordered field, no proper algebraic extension of which is also an ordered field. The *real closure* $\widetilde{K}$ of an ordered field K is the smallest real closed field that contains it. We refer the interested reader to [3] or [19] for further details and more formal definitions, and to [4, 5] for previous algorithmic applications of real closed fields.

An *elementary formula*[1] is a finite quantified boolean formula, each of whose clauses is a multivariate polynomial inequality with real coefficients. An elementary formula *holds in* an ordered field K if and only if the formula has no free variables, and the formula is true if we interpret each

[1] More formally, this is called a *formula in the first-order language of ordered fields with parameters in* $\mathbb{R}$ [3].

variable as an element of K and addition and multiplication as field operations in K.

The following principle was originally proven by Tarski [21], in a slightly different form.

The Transfer Principle: *Let $\widetilde{K}$ and $\widetilde{K}'$ be two real closed fields. An elementary formula holds in $\widetilde{K}$ if and only if it holds in $\widetilde{K}'$.*

In particular, this implies that if an elementary formula holds in any real closed field $\widetilde{K}$, then it also holds in $\mathbb{R}$.

For any ordered field K, we let $K(\varepsilon)$ denote the ordered field of rational functions in ε with coefficients in K, where ε is positive but less than every positive element of K. In this case, we say that ε is *infinitesimal in K*. We use towers of such field extensions. In such an extension, the order of the infinitesimals is specified by the description of the field. For example, in the ordered field $\mathbb{R}(\Delta, \delta, \varepsilon)$, Δ is infinitesimal in the reals, δ is infinitesimal in $\mathbb{R}(\Delta)$, and ε is infinitesimal in $\mathbb{R}(\Delta, \delta)$. Infinitesimals have been used extensively in perturbation techniques [11, 14, 24] and in algorithms dealing with real semi-algebraic sets [4, 5].

3 The Main Theorem

In this section, we prove the following theorem.

Theorem 1. *Any r-linear decision tree that decides an r-variable linear satisfiability problem must have depth $\Omega(n^{\lceil r/2 \rceil})$.*

3.1 The Adversary Configuration

Throughout this section, let ϕ denote a fixed linear expression in r variables. We say that an r-tuple is *degenerate* if in the zero-set of ϕ, and that a configuration X is degenerate if it contains any degenerate r-tuples. For any configuration X, we call an r-tuple of elements of X *collapsible* if the following properties are satisfied.

(1) The tuple is nondegenerate.

(2) There exists another *collapsed* configuration $\hat{X}$, such that the corresponding tuple in $\hat{X}$ is degenerate, but the sign of every other linear combination of r or fewer elements is the same for both configurations.

In other words, the only way for an r-linear decision tree to distinguish between X and $\hat{X}$ is to perform a direct query on the tuple.

To prove a lower bound, it suffices to prove the existence of a nondegenerate input configuration with lots of collapsible tuples. If an r-linear decision tree algorithm does not perform a direct query for each collapsible tuple, given this configuration as input, then an adversary can collapse one of the tuples. The algorithm would be unable to distinguish between the original configuration and the collapsed configuration, even though one is degenerate and the other is not. Thus, the number of collapsible tuples is a lower bound on the running time of the algorithm.

Unfortunately, this approach seems to be doomed from the start. For *any* two sets of real numbers X and $\hat{X}$, there are an infinite number of query polynomials that are positive at X and negative at $\hat{X}$. It follows that real configurations cannot have any collapsible tuples. Moreover, for any set of real numbers X, there is an algorithm which requires only $|X|$ queries to decide whether X satisfies any fixed linear satisfiability problem. Thus, it is impossible to find a single adversary set of real numbers that is hard for every algorithm.

To get around this problem, we introduce the use of infinitesimals. That is, we allow our adversary's configuration to contain elements of an ordered field of the form $K = \mathbb{R}(\varepsilon_1, \ldots, \varepsilon_m)$, instead of the reals. Since the field K is ordered, and since any real polynomial can be thought of as a function from K to K, it is reasonable to talk about the behavior of any algebraic decision tree given such a configuration as input.

Allowing the adversary to use infinitesimals allows us to construct a configuration with several collapsible tuples, even though such configurations are impossible if we restrict ourselves to the reals. Our construction relies on a special matrix M satisfying the following lemma.

Lemma 2. *There exists an $r \times \lfloor r/2 \rfloor$ matrix M satisfying the following two conditions.*

(1) There are $\Omega(n^{\lceil r/2 \rceil})$ vectors $v \in \{1, 2, \ldots, n\}^r$ such that $M^\top v = 0$.

(2) Every set of $\lfloor r/2 \rfloor$ rows of M forms a nonsingular matrix.

Proof: Let $M = (m_{ij})$ be the $r \times \lfloor r/2 \rfloor$ integer matrix whose first $\lceil r/2 \rceil$ rows form a Vandermonde matrix with $m_{ij} = i^{j-1}$, and whose last $\lfloor r/2 \rfloor$ rows form a negative identity matrix. We claim that this matrix satisfies conditions (1) and (2).

We construct a vector $v = (v_1, v_2, \ldots, v_d) \in \{1, 2, \ldots, n\}^r$ such that $M^\top v = 0$ as follows. Let $m_{\max}$ denote the largest element in M; that is, $m_{\max} = \lceil r/2 \rceil^{\lfloor r/2 \rfloor}$. Fix the first $\lceil r/2 \rceil$ coordinates of v arbitrarily in the range

$$1 \le v_i \le \frac{n}{\lceil r/2 \rceil m_{\max}}.$$

Now assign the following values to the remaining $\lfloor r/2 \rfloor$ coordinates:

$$v_j = \sum_{i=1}^{\lceil r/2 \rceil} m_{i, j - \lceil r/2 \rceil} v_i.$$

Since each m_{ij} is a positive integer, the v_j are all positive integers in the range $\lceil r/2 \rceil \le v_j \le n$. We easily verify that $M^\top v = 0$. There are

$$\left(\frac{n}{\lceil r/2 \rceil m_{\max}} \right)^{\lceil r/2 \rceil} \ge \frac{n^{\lceil r/2 \rceil}}{\lceil r/2 \rceil^{\lceil r/2 \rceil^2}} = \Omega(n^{\lceil r/2 \rceil}).$$

different ways to choose the vector v. Thus, M satisfies condition (1).

Let M' be a matrix consisting of $\lfloor r/2 \rfloor$ arbitrary rows of M. Using elementary row and column operations, we can write

$$M' = W \begin{pmatrix} V & 0 \\ 0 & -I \end{pmatrix},$$

where W is a matrix with determinant ± 1, V is a square minor of a Vandermonde matrix, and I is an identity matrix. Since W, V', and I are all nonsingular, so is M'. Thus, M satisfies condition (2). $\square$

Lemma 3. *There exists a configuration $X \in K^n$ with $\Omega(n^{\lceil r/2 \rceil})$ collapsible tuples, for some ordered field K.*

Proof: We explicitly construct a configuration $X \in \mathbb{R}(\Delta_1, \ldots, \Delta_{r-1}, \delta_1, \ldots, \delta_{\lfloor r/2 \rfloor}, \varepsilon_1, \ldots, \varepsilon_r)$ that satisfies the lemma. We assume without loss of generality that n is a multiple of r.

Write $\phi = \sum_{i=1}^r a_i t_i$ with real coefficients a_i and formal variables t_i. Let the matrix $M = (m_{ij})$ be given by the previous lemma. Our configuration X is the union of r smaller sets X_i, each containing n/r elements x_{ij} defined as follows.

$$x_{ij} = \frac{1}{a_i}\left((-1)^i(\Delta_{i-1} + \Delta_i) + \sum_{k=1}^{\lfloor r/2 \rfloor} m_{ik}\delta_k j\right) + \varepsilon_i j^2$$

For notational convenience, we define $\Delta_0 = \Delta_d = 0$.

For example, in the simplest nontrivial case $r = 3$, our configuration X lies in the field $\mathbb{R}(\Delta_1, \Delta_2, \delta_1, \varepsilon_1, \varepsilon_2, \varepsilon_3)$. If we take $M = (1, 1, -1)^\top$, then X contains the following elements:

$$\begin{aligned}
x_{1j} &= \tfrac{1}{a_1}(-\Delta_1 \qquad\quad + \delta_1 j) + \varepsilon_1 j^2 && [1 \le j \le n/3]\\
x_{2j} &= \tfrac{1}{a_2}(\quad \Delta_1 + \Delta_2 + \delta_1 j) + \varepsilon_2 j^2 && [1 \le j \le n/3]\\
x_{3j} &= \tfrac{1}{a_3}(\qquad\quad - \Delta_2 + \delta_1 j) + \varepsilon_3 j^2 && [1 \le j \le n/3]
\end{aligned}$$

We claim that any tuple $(x_{1p_1}, \ldots, x_{rp_r})$ satisfying the equation $M^\top(p_1, \ldots, p_r) = 0$ is collapsible. By condition (1), there are $\Omega((n/r)^{\lceil r/2 \rceil}) = \Omega(n^{\lceil r/2 \rceil})$ such tuples. The adversary collapses the tuple by replacing X with $\hat{X}$, with elements $\hat{x}_{ij} = x_{ij} + \varepsilon_i(p_i^2 - 2jp_i)$. So, for example, when $r = 3$, we have $\hat{X} = \hat{X}_1 \cup \hat{X}_2 \cup \hat{X}_3$, where

$$\begin{aligned}
\hat{x}_{1j} &= \tfrac{1}{a_1}(-\Delta_1 \qquad\quad + \delta_1 j) + \varepsilon_1 (j - p_1)^2 && [1 \le j \le n/3]\\
\hat{x}_{2j} &= \tfrac{1}{a_2}(\quad \Delta_1 + \Delta_2 + \delta_1 j) + \varepsilon_2 (j - p_2)^2 && [1 \le j \le n/3]\\
\hat{x}_{3j} &= \tfrac{1}{a_3}(\qquad\quad - \Delta_2 + \delta_1 j) + \varepsilon_3 (j - p_3)^2 && [1 \le j \le n/3]
\end{aligned}$$

We easily confirm that the collapsed tuple is degenerate. It remains to show that no other r-linear query expression changes sign.

Consider the query expression $Q = \sum_{i=1}^r Q_i$, where for each i,

$$Q_i = a_i \sum_{j=1}^{n/r} \alpha_{ij} x_{ij},$$

and at most r of the coefficients α_{ij} are not zero. We refer to x_{ij} as a *query variable* if its coefficient α_{ij} is not zero. We define A_i and J_i as follows.

$$A_i = \sum_{j=1}^{n/r} \alpha_{ij} \qquad\qquad J_i = \sum_{j=1}^{n/r} \alpha_{ij} j$$

Rewrite Q as a linear combination of the infinitesimals:

$$Q = \sum_{i=1}^{r-1} D_i \Delta_i + \sum_{i=1}^{\lfloor r/2 \rfloor} d_i \delta_i + \sum_{i=1}^{r} e_i \varepsilon_i$$

From the definitions above, we have $D_i = (-1)^i(A_i - A_{i+1})$ for all i. If we let $\hat{Q}$ be the corresponding query expression for the collapsed configuration $\hat{X}$, we can write

$$\hat{Q} = \sum_{i=1}^{r-1} D_i \Delta_i + \sum_{i=1}^{\lfloor r/2 \rfloor} d_i \delta_i + \sum_{i=1}^{r} \hat{e}_i \varepsilon_i,$$

where $\hat{e}_i = e_i - 2p_i A_i + p_i^2 J_i$ for all i.

To prove the lemma, it suffices to consider only queries for which $D_i = 0$ and $d_i = 0$ for all i. Note that in this case, all the A_i's are equal. There are three cases to consider.

Case 1. Suppose no subset X_i contains exactly one of the query variables. (This includes the case where all query variables belong to the same subset.) Then at most $\lfloor r/2 \rfloor$ of the Q_i's are not identically zero. It follows that $A_i = 0$ for all i. The vector J of nontrivial J_i's must satisfy the matrix equation $(M')^\top J = 0$, where M' is a square minor of the matrix M. By condition (2) above, M' is nonsingular, so the J_i's must be zero. It follows that $\hat{e}_i = e_i$ for all i, which implies that $Q = \hat{Q}$.

Case 2. Suppose some subset X_k contains exactly one query variable x_{kj} and some other subset X_l contains none. Then $A_k = \alpha_{kj}$ and $A_l = 0$. Since all the A_i are equal, it follows that $\alpha_{kj} = 0$. This contradicts the assumption that x_{kj} is a query variable. Thus, this case never happens.

Case 3. Finally, suppose each query variable comes from a different subset. Recall that all the A_i are equal. Since we are only interested in the sign of the query, we can conclude without loss of generality that $A_i = \alpha_{ij} = 1$ for each query variable x_{ij}. Thus, each of the e_i's is positive, which implies that Q is positive. Furthermore, unless the query variables are exactly x_{ip_i} for all i, each of the $\hat{e}_i$'s is also positive, which means $\hat{Q}$ is also positive.

This completes the proof of Lemma 3. □

3.2 Moving Back to the Reals

Intuitively, the use of infintesimals in our adversary configuration makes it unsable for proving lower bounds. After all, the algorithms we consider are only required to behave correctly when they are given real input. Therefore, we must somehow get rid of the infintesimals before applying our adversary argument. Since we know that no single real adversary configuration exists, we instead derive a different adversary configuration for each algorithm.

Fix an r-linear decision tree A, and let Q_A denote the set of query polynomials used by A. (We assume, without loss of generality, that Q_A includes all $\Theta(n^r)$ direct queries, since otherwise the algorithm cannot correctly detect all possible degenerate tuples.) For any input configuration X, we call an r-tuple of elements in X *relatively collapsible* if the following properties are satisfied.

(1) The tuple is nondegenerate.

(2) There exists another *collapsed* configuration $\hat{X}$, such that the corresponding tuple in $\hat{X}$ is degenerate, but the sign of every other polynomial in Q_A is the same for both configurations.

Clearly, any collapsible tuple is also relatively collapsible. To prove a lower bound, it suffices to prove, for each r-linear decision tree A, the existence of a corresponding nondegenerate input configuration with lots of relatively collapsible tuples.

Lemma 4. *For any r-linear decision tree A, there exists a real configuration $X_A \in \mathbb{R}^n$ with $\Omega(n^{\lceil r/2 \rceil})$ relatively collapsible tuples.*

Proof: Fix A, and let Q_A denote the set of query polynomials used by A. Each of the collapsible tuples in X is also relatively collapsible. Each relatively collapsible tuple Y in X corresponds to a polynomial ϕ_Y, such that $\phi_Y(X) = \phi(Y)$. Call the set of these polynomials Φ.

It follows directly from the definitions that the following elementary formula holds in K.

$$\exists X \bigwedge_{\phi_Y \in \Phi} \left(\phi_Y(X) \neq 0 \;\wedge\; \exists \hat{X} \left(\phi_Y(\hat{X}) = 0 \;\wedge \bigwedge_{q \in Q_A \setminus \{\phi_Y\}} \operatorname{sgn} q(X) = \operatorname{sgn} q(\hat{X}) \right) \right)$$

This is just a convenient shorthand for the actual formula. Each reference to $\phi_Y(X)$ or $q(X)$ should be expanded into an explicit polynomial in X, and the equation $\operatorname{sgn} a = \operatorname{sgn} b$ into the boolean formula $((ab > 0) \vee (a = 0 \wedge b = 0))$. Since the sets Φ and Q_A are finite, the expanded formula is also finite and therefore elementary.

Since K is a subset of its real closure $\widetilde{K}$, and the formula is only existentially quantified, the formula holds in $\widetilde{K}$. Thus, by the Transfer Principle, it also holds in $\mathbb{R}$. The lemma follows immediately. □

With a little more care, we can show that the real configurations are derived by replacing the infinitesimals by sufficiently small and sufficiently well-separated real values.

3.3 Removing Degeneracies

One final problem remains. The adversary configurations we construct (and by implication, the real configurations we get by invoking the previous lemma) are degenerate, which makes them unsuitable for proving lower bounds. In simple cases, we can construct nondegenerate adversary configurations, but this becomes considerably more difficult as we consider larger values of r. Thus, instead of giving an explicit construction, we show that one can perturb the existing degenerate configurations into general position. This technique was previously used by Erickson and Seidel [15] to prove lower bounds on some geometric degeneracy problems.

Lemma 5. *For any r-linear decision tree A, there exists a nondegenerate real configuration $X_A^* \in \mathbb{R}^n$ with $\Omega(n^{\lceil r/2 \rceil})$ relatively collapsible tuples.*

Proof: As before, let Q_A denote the set of query polynomials used by A. The set Q_A induces a finite hyperplane arrangement in the configuration space $\mathbb{R}^n$. For notational convenience, we color each hyperplane "red" if it corresponds to a direct query, and "green" otherwise. Each configuration corresponds to a point in some cell C in this arrangement, and each collapsible tuple corresponds to a boundary facet of C that is uniquely spanned by a red hyperplane.

Let C be any cell in the arrangement, and let C' be any cell in the boundary of C. Then any hyperplane that uniquely spans a boundary facet of C' also uniquely spans a boundary facets of C. Since the cell containing X has k red boundary facets, it follows by induction that there is a full-dimensional cell with k red boundary facets.[2] We can choose X_A^* anywhere in this cell. □

This completes the proof of Theorem 1.

3.4 Nonuniform Upper Bounds

Our lower bound matches known upper bounds when r is odd, but is a logarithmic factor away when r is even and greater than 2. We use the following result of Fredman [16] to show that our lower bounds cannot be improved in this case.

[2]This conclusion is stronger than the lemma requires. It suffices that some cell, contained in only green hyperplanes, has k red boundary facets.

Lemma 6 (Fredman [16]). *Let Γ be a subset of the $n!$ orderings of $\{1, \ldots, n\}$ for some fixed n. There exists a comparison tree of depth at most $\log_2(|\Gamma|) + 2n$ that sorts any sequence of n numbers with order type in Γ.*

Theorem 7. *Let Π be an r-variable linear satisfiability problem with n inputs, for some fixed n and $r > 2$. Then there exists an r-linear decision tree with depth $O(n^{\lceil r/2 \rceil})$ that decides Π.*

Proof: It suffices to consider the case when r is even, since for any odd r there is a simple uniform algorithm with running time $O(n^{(r+1)/2})$. Suppose we are trying to satisfy the equation $\sum_{i=1}^{r} a_i t_i = 0$ for some fixed coefficients $a_i \in \mathbb{R}$. Given a configuration $X = (x_1, \ldots, x_n) \in \mathbb{R}^n$, we (implicitly) construct sets Y and Z of $n^{\underline{r/2}}$ real numbers each, as follows:

$$Y = \left\{ \sum_{i=1}^{r/2} a_i y_i \;\middle|\; \{y_1, \ldots, y_{r/2}\} \subset \{x_1, \ldots, x_n\} \right\}$$

$$Z = \left\{ \sum_{i=1}^{r/2} -a_{i+r/2} z_i \;\middle|\; \{z_1, \ldots, z_{r/2}\} \subset \{x_1, \ldots, x_n\} \right\}$$

Then X is degenerate if and only if the sets Y and Z share an element. We can detect this condition by sorting $Y \cup Z$ using Fredman's "comparison" tree, which is really a r-linear decision tree.

Every pair of elements of $Y \cup Z$ induces a hyperplane in the configuration space $\mathbb{R}^n$. There is a one-to-one correspondence between the cells in the resulting hyperplane arrangement and the possible orderings of $Y \cup Z$. Since an arrangement of N hyperplanes in $\mathbb{R}^D$ has at most $O(N^D)$ cells [9], there are at most $O((2n^{\underline{r/2}})^{2n}) = O((2n)^{rn})$ possible orderings. It follows that the depth of Fredman's decision tree is at most $4n^{\underline{r/2}} + O(rn \log n) = O(n^{r/2})$. □

Of course, this result does not imply the existence of a single $O(n^{\lceil r/2 \rceil})$-time algorithm that works for *all* values of n. Closing the logarithmic gap between these upper and lower bounds, even for the special case of sorting $X + Y$, is a long-standing and very difficult open problem.

4 Higher-Dimensional Problems

In this section, we extend the results of the previous section to geometric degeneracy problems in higher dimensions.

We say that a multivariate polynomial which maps a tuple of points in $\mathbb{R}^d$ to the reals is *r-linear in the x_i-coordinate* if it can be expressed as a linear combination of the x_i-coordinates of r of its arguments, with coefficients that are polynomials in the remaining coordinates. Let $\phi : (\mathbb{R}^d)^s \to \mathbb{R}$ be a fixed polynomial that is r-linear in some coordinate. The associated *projected r-linear satisfiability problem* asks, given a set of n points in $\mathbb{R}^d$, whether any s-tuple of these points is in the zero-set of ϕ. (Note that $s = r$ in most natural applications, but $s > r$ is certainly possible.) If all of the query polynomials in an algebraic decision tree are r-linear in some coordinate, we call it a *projected r-linear decision tree.*

For example, the *affine degeneracy problem* asks, given a set of n points in $\mathbb{R}^d$, whether any $d + 1$ of them are on the same hyperplane [15]. This is equivalent to asking whether any $d + 1$

points satisfy the polynomial equation

$$\begin{vmatrix} 1 & p_{01} & p_{02} & \cdots & p_{0d} \\ 1 & p_{11} & p_{12} & \cdots & p_{1d} \\ & & \vdots & & \\ 1 & p_{d1} & p_{d2} & \cdots & p_{dd} \end{vmatrix} = 0.$$

This determinant is $d+1$-linear in every coordinate, since for all j, we can rewrite it as a linear combination of the p_{ij}'s. Thus, the affine degeneracy problem is a projected $d+1$-linear satisfiability problem.

Let $\phi : (\mathbb{R}^d)^s \to \mathbb{R}$ be a fixed polynomial that is r-linear in some coordinate, and let X be a configuration of n points in $\mathbb{R}^d$. We easily generalize the definitions of degeneracy, collapsible tuples, and relatively collapsible tuples to configurations of points in higher dimensions. The *sign* of an s-tuple Y of points in X is just the sign of $\phi(Y)$. Given a projected r-linear decision tree A with query polynomials Q_A, we say that a s-tuple Y of points in X is *relatively reversible* if the following conditions hold.

(1) The tuple is not degenerate.

(2) One can smoothly deform X into another configuration $\hat{X}$, changing the sign of the tuple, but not changing the sign of any other query polynomial in Q_A at any time.

Clearly, any relatively reversible tuple is also relatively collapsible. The converse is not true in general, but is true in the special case where both ϕ and all the polynomials in Q_A are actually linear in *all* their arguments, as in the previous section.

We will use the following generalization of Lemma 5.

Lemma 8. *Suppose there exists a configuration $X \in \mathbb{R}^{dn}$ with k relatively reversible tuples. Then there also exists a nondegenerate configuration $X^* \in \mathbb{R}^{dn}$ with k relatively reversible tuples.*

Proof: The set Q_A of query polynomials induces a set of algebraic surfaces in the configuration space $\mathbb{R}^{dn}$. Color each surface red if it induced by a potential assignment to ϕ, and green otherwise. These surfaces define a cellular decomposition. We say that a surface σ *nicely bounds* a cell C in this decomposition if there is a continuous path from a point strictly inside C to a point strictly outside C that is either entirely contained in or disjoint from every surface except σ.

Let C be any cell, and let C' be a cell in the boundary of C. Then any surface that nicely bounds C' also nicely bounds C. Since the cell containing X is nicely bounded by k red surfaces, it follows by induction that there is a full-dimensional cell that is nicely bounded by at least k surfaces. We can take X^* to be any point in this cell. □

Theorem 9. *Any projected r-linear decision tree that decides a projected r-linear satisfiability problem has depth $\Omega(n^{\lceil r/2 \rceil})$.*

Proof: Let $\phi : (\mathbb{R}^d)^s \to \mathbb{R}$ be a fixed polynomial that is r-linear in some coordinate, and let A be a fixed projected r-linear decision tree with query variables Q_A that decides the corresponding projected linear satisfiability problem, given a set of n points in $\mathbb{R}^d$. The previous lemma implies that to prove the lower bound, it suffices to prove the existence of a (possibly degenerate) configuration with $\Omega(n^{\lceil r/2 \rceil})$ relatively reversible tuples. For the sake of readability, we will explicitly consider only the particular case $s = r$; our proof requires only minor modifications to handle the general case.

We start by constructing a configuration $X \in (K^d)^n$ containing $\Omega(n^{\lceil r/2 \rceil})$ collapsible tuples, where K is an extension of the reals by infinitesimals. The set X is the union of r smaller subsets $X_1 \cup \cdots \cup X_r$. Each subset contains n/r elements on some real vertical line; that is, every point in a subset has the same set of 1st through $(d-1)$th coordinates, all of which are real numbers.

An r-tuple $(p_1, \ldots, p_r)$, where $p_i \in X_i$ for each i, is degenerate if and only if the x_d-coordinates of the p_i's satisfy the linear equation

$$\sum_{i=1}^{r} a_i p_{id} = 0,$$

where the (real) coefficients a_i are determined by the positions of the vertical lines. The particular choice of coordinate values for each subset is not important, except that none of the a_i's should be zero. It is always possible to choose appropriate values, since the set of "bad" coordinate values form an algebraic variety of codimension 1 in $\mathbb{R}^{(d-1)r}$.

We fill in the x_1-coordinates of the points with the elements of the adversary configuration for the corresponding linear satisfiability problem, as described in the proof of Lemma 3. This gives us an adversary configuration X for the present problem consisting of n points in K^d, where $K = \mathbb{R}(\Delta_1, \ldots, \Delta_{r-1}, \delta_1, \ldots, \delta_{\lfloor r/2 \rfloor}, \varepsilon_1, \ldots, \varepsilon_r)$.

This derived configuration inherits all $\Omega(n^{\lceil r/2 \rceil})$ collapsible tuples from the original configuration. To collapse any such r-tuple, it suffices to change only the x_d-coordinates of the points in X, as described in the proof of Lemma 3. Unfortunately, since our query expressions are in general not linear, we do not get the lower bound immediately.

We can easily generalize Lemma 4 to prove that there exists a real configuration $X_A \in \mathbb{R}^{dn}$ with $\Omega(n^{\lceil r/2 \rceil})$ relatively collapsible tuples. Furthermore, by suitably modifying the elementary formula, we can ensure that the x_1- through x_{d-1}-coordinates of the points in X_A are the same as those in the original configuration X, and that collapsing tuples does not change these coordinates. (Recall that these coordinates were already real numbers.)

Thus, both the original configuration X_A and any collapsed configuration $\hat{X}_A$ satisfy a set of $(d-1)n$ linear constraints, each of which induces an axis-normal hyperplane in the configuration space $\mathbb{R}^{dn}$. The intersection of these hyperplanes is an n-flat $S \subset \mathbb{R}^{dn}$. Furthermore, for any polynomial p that is linear in x_d, the intersection of S and the algebraic surface induced by p is an $(n-1)$-flat, which we can think of as a hyperplane in the space S.

Thus, the query polynomials Q_A induce a finite hyperplane arrangement in S. It follows easily that every relatively collapsible r-tuple in X is also relatively reversible. The theorem immediately follows from Lemma 8 and the usual adversary argument. $\square$

This theorem immediately implies the following lower bounds.

Corollary 10. *In the worst case, $\Omega(n^{\lfloor d/2 \rfloor + 1})$ steps are required to solve the affine degeneracy problem, in the projected $(d+1)$-linear decision tree model.*

The best upper bound for the affine degeneracy problem is $O(n^d)$, and the algorithms that achieve this bound follow the direct query model, in which every decision is based on the sign of a sidedness determinant [12, 13, 10]. Previously, Erickson and Seidel [15] established a lower bound of $\Omega(n^d)$ in the direct query model, using a version of Lemma 8. While our new lower bounds are smaller for all $d > 2$, they hold in a much richer model of computation.

In the two-dimensional case, Erickson and Seidel proved that the $\Omega(n^2)$ lower bound still holds if the algorithm is allowed to compare coordinates of points or slopes of lines, in addition to making sidedness queries. Since comparing slopes of lines is not a projected 3-linear query, our new lower bound for this problem is incomparable with the bound established in [15].

Corollary 11. *In the worst case, given n points in the plane, $\Omega(n^2)$ steps are required to decide if any pair of connecting lines is parallel, in the projected 4-linear decision tree model.*

The best known upper bound for this problem is $O(n^2 \log n)$, following from the obvious slope-sorting algorithm. Lemma 6 implies the existence of a quadratic nonuniform algorithm, so our lower bound cannot be improved.

5 Conclusions and Open Problems

We have developed a new general technique for proving lower bounds in decision tree models of computation. We show that it suffices to construct a single input configuration, possibly degenerate and possibly containing infinitesimals, containing lots of collapsible tuples. Using this technique, we have proven $\Omega(n^{\lceil r/2 \rceil})$ lower bounds on the depth of any r-linear decision tree that decides an r-variable linear satisfiability problem. This is the best possible lower bound in this model. We have also generalized our technique to prove new lower bounds on some geometric degeneracy-detection problems.

Gajentaan and Overmars [17] describe a large class of so-called "3SUM-hard" problems in computational geometry. All such problems can be reduced, in sub-quadratic time, to the following problem: Given a set of n real numbers, is the average of any two distinct elements also in the set? In particular, an $\Omega(n^2)$ lower bound on the complexity of average detection in the algebraic decision tree model would imply similar bounds for a large number of separation, covering, visibility, motion planning, and other geometric problems. Unfortunately, the model in which our new lower bounds hold is generally too weak to apply to these more general problems.

An obvious open problem is to improve our lower bounds to stronger models of computation. Even seemingly small improvements would lead to significant new results. For example, an $\Omega(n^2)$ lower bound on average detection in the 6-linear decision tree model would immediately imply the first $\Omega(n^2)$ lower bound for the problem of finding the minimum area triangle among n points in the plane. Unfortunately, a lower bound even the 4-linear decision tree model seems to be completely out of reach at present.

Ultimately, we would like to prove a lower bound larger than $\Omega(n \log n)$ for *any* non-NP-hard polynomial satisfiability problem, in some general model of computation such as linear decision trees, algebraic decision trees, or even algebraic computation trees. Linear satisfiability problems, in particular the problem of average detection, seem to be good candidates for study.

Acknowledgements. The author gratefully acknowledges Raimund Seidel for several helpful discussions.

References

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

[2] A. Björner, L. Lovász, and A. C. C. Yao. Linear decision trees: volume estimates and topological bounds. In *Proc. 24th Annu. ACM Sympos. Theory. Comput.*, pages 170–177, 1992.

[3] J. Bochnak, M. Coste, and M.-F. Roy. *Géométrie algébraique réelle*. Number 12 in Ergebnisse der Mathematik und ihrer Grenzgebeite 3. Springer-Verlag, 1987.

[4] J. Canny. Some algebraic and geometric configurations in PSPACE. In *Proc. 20th Annu. ACM Sympos. Theory Comput.*, pages 460–467, 1988.

[5] J. Canny. Computing roadmaps of semi-algebriac sets. In *Proc. 9th Annu. Sympos. Algebraic Algorithms and Error Corr. Codes*, pages 94–107, 1991.

[6] M. Dietzfelbinger. Lower bounds for sorting of sums. *Theoret. Comput. Sci.*, 66:137–155, 1989.

[7] M. Dietzfelbinger and W. Maass. Lower bound arguments with "inaccessible" numbers. *J. Comput. Syst. Sci.*, 36:313–335, 1988.

[8] D. P. Dobkin and R. J. Lipton. On the complexity of computations under varying sets of primitives. *J. Comput. Syst. Sci.*, 18:86–91, 1979.

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

[10] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, 38:165–194, 1989. Corrigendum in 42 (1991), 249–251.

[11] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9:66–104, 1990.

[12] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, 15:341–363, 1986.

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

[14] I. Emiris and J. Canny. A general approach to removing degeneracies. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 405–413, 1991.

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

[16] M. L. Fredman. How good is the information theory bound in sorting? *Theoret. Comput. Sci.*, 1:355–361, 1976.

[17] A. Gajentaan and M. H. Overmars. n^2-hard problems in computational geometry. Report RUU-CS-93-15, Dept. Comput. Sci., Utrecht Univ., Utrecht, Netherlands, Apr. 1993.

[18] F. Meyer auf der Heide. A polynomial linear search algorithm for the n-dimensional knapsack problem. *J. ACM*, 31:668–676, 1984.

[19] A. Prestel. *Lectures on Formally Real Fields*, volume 1093 of *Lecture Notes in Mathematics*. Springer-Verlag, 1984.

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

[21] A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* University of California Press, 1951. Prepared for publication by J. C. C. McKinsey.

[22] A. C.-C. Yao. Algebraic decision trees and Euler characteristics. In *Proc. 33rd Annu. IEEE. Sypos. Found. Comput. Sci.*, pages 268–277, 1992.

[23] A. C.-C. Yao. Decision tree complexity and Betti numbers. In *Proc. 25th Annu. ACM Sympos. Theory Comput.*, 1994.

[24] C. K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40:2–18, 1990.