RANDOM SAMPLING IN COMPUTATIONAL ALGEBRA: HELLY NUMBERS AND VIOLATOR SPACES

JESÚS A. DE LOERA, SONJA PETROVIĆ, AND DESPINA STASI

Abstract. This paper transfers a randomized algorithm, originally used in geometric optimization, to computational problems in commutative algebra. We show that Clarkson's sampling algorithm can be applied to two problems in computational algebra: solving large-scale polynomial systems and finding small generating sets of graded ideals. The cornerstone of our work is showing that the theory of violator spaces of Gärtner et al. applies to polynomial ideal problems. To show this, one utilizes a Helly-type result for algebraic varieties. The resulting algorithms have expected runtime linear in the number of input polynomials, making the ideas interesting for handling systems with very large numbers of polynomials, but whose rank in the vector space of polynomials is small (e.g., when the number of variables and degree is constant).

Keywords: Violator spaces, ideal generators, solving polynomial systems, randomized algorithm in algebra, large sparse systems.
AMS subject classification: 68W20, 68R05, 12Y05, 13P10, 14Q10, 08A40.

1. Introduction

Many computer algebra systems offer excellent algorithms for manipulation of polynomials. But despite great success in the field, many algebraic problems have bad worst-case complexity. For example, Buchberger's [13, 14, 18] groundbreaking algorithm, key to symbolic computational algebra today, computes a Gröbner basis of any ideal, but it has a worst-case runtime that is doubly exponential in the number of variables [22]. This presents the following problem: *what should one do about computations whose input is a very large, overdetermined system of polynomials? In this paper, we propose to use randomized sampling algorithms to ease the computational cost in such cases.*

One can argue that much of the success in computation with polynomials (of non-trivial size) often relies heavily on finding specialized structures. Examples include Faugère's et al. fast computation of Gröbner bases of zero-dimensional ideals [25, 26, 27, 29, 35], specialized software for computing generating sets of toric ideals [1], several packages in [32] built specifically to handle monomial ideals, and the study of sparse systems of polynomials (i.e., systems with fixed support sets of monomials) and the associated homotopy methods [45]. A more recent example of the need to find good structures is in [16], where Cifuentes and Parrilo began exploiting chordal graph structure in computational commutative algebra, and in particular, for solving polynomial systems. Our paper exploits combinatorial structure implicit in the input polynomials, but this time akin to Helly-type results from convex discrete geometry [36].

At the same time, significant improvements in efficiency have been obtained by algorithms that involve randomization, rather than deterministic ones (e.g. [10, 43]); it is also widely recognized that there exist hard problems for which pathological examples requiring exponential runtimes occur only rarely, implying an obvious advantage of considering average behavior analysis of many algorithms. For example, some forms of the simplex method for solving linear programming problems have worst-case complexity that is exponential, yet [44] has recently shown that in the *smoothed analysis of algorithms* sense, the simplex method is a rather robust and fast algorithm. Smoothed analysis combines the worst-case and average-case algorithmic analyses by measuring the expected performance of algorithms under slight random perturbations of worst-case inputs. Of course, probabilistic analysis, and smoothed analysis in particular, has been used in computational algebraic geometry for some time now, see e.g., the elegant work in [8, 9, 15]. The aim of this paper is to import a randomized sampling framework from geometric optimization to applied computational algebra, and demonstrate its usefulness on two problems.

Our contributions. We apply the theory of violator spaces [31] to polynomial ideals and adapt Clarkson's sampling algorithms [17] to provide efficient randomized algorithms for the following concrete problems:

(1) *solving large (overdetermined) systems of multivariate polynomials equations,*

(2) *finding small, possibly minimal, generating sets of homogeneous ideals.*

Our method is based on using the notion of a *violator space*. Violator spaces were introduced in 2008 by Gärtner, Matoušek, Rüst, and Škovroň [31] in a different context. Our approach allows us to adapt Clarkson's sampling techniques [17] for computation with polynomials. Clarkson-style algorithms rely on computing with small-size subsystems, embedded in an iterative biased sampling scheme. In the end, the local information is used to make a global decision about the entire system. The expected runtime is linear in the number of input elements, which is the number of polynomials in our case (see [12] for a more recent simplified version of Clarkson's algorithm for violator spaces). Violator spaces naturally appear in problems that have a natural linearization and a sampling size given by a combinatorial *Helly number* of the problem. While violator spaces and Clarkson's algorithm have already a huge range of applications, to our knowledge, this is the first time such sampling algorithms are being used in computational algebraic geometry. For an intuitive reformulation of Helly's theorem for algebraic geometers, see Example 2.1. Main ingredients of violator spaces are illustrated through Examples 2.2, 3.2 and 3.6. A typical setup where problem (1) can be difficult and a randomized algorithm appropriate can be found in Example 4.8.

Before stating the main results, let us fix the notation used throughout the paper. We assume the reader is acquainted with the basics of computational algebraic geometry as in the award-winning undergraduate textbook [18]. Denote by K a (algebraically closed) field; the reader may keep $K = \mathbb{C}$ in mind as a running example. Let $f_1 = 0, \dots, f_m = 0$ be a system of m polynomials in $n+1$ variables with coefficients in K. We usually assume that $m \gg n$. As is customary in the algebra literature, we write $f_1, \dots, f_m \in \mathfrak{R} = K[x_0 \dots, x_n]$ and often denote the polynomial ring $\mathfrak{R}$ by a shorthand notation $K[x]$. We will denote by $(f_1, \dots, f_m) \subset \mathfrak{R}$ the ideal generated by these polynomials; that is, the set of all polynomial combinations of the f_i's. Note that if $F = \{f_1, \dots, f_m\}$ is a set of polynomials, the ideal $(f_1, \dots, f_m)$ will equivalently be denoted by (F).

A polynomial is said to be homogeneous if all of its terms are of same degree; an ideal generated by such polynomials is a homogeneous ideal. In this paper, the ideals we consider need not be homogeneous; if they are, that will be explicitly stated. In that case, the set of all homogeneous polynomials of total degree d will be denoted by $[\mathfrak{R}]_d$. Finally, denote by $\mathcal{V}(S)$ the (affine) variety defined by the set of polynomials $S \in \mathfrak{R}$, that is, the Zariski closure of the set of common zeros of the polynomials in the system S. Therefore, the concrete problem (1) stated above simply asks for the explicit description of the variety (solution set) corresponding to an ideal (system of polynomial equations). The concrete problem (2) asks to find a smaller (e.g., minimal with respect to inclusion) set of polynomial equations that generate the same ideal - and thus have the exact same solution set.

Solving large polynomial systems. Suppose we would like to solve a system of m polynomials in $n+1$ variables over the field K, and suppose that m is large. We are interested in the coefficients of the polynomials as a way to linearize the system. To that end, recall first that the d-th Veronese embedding of $\mathbb{P}^n$ is the following map $\nu_d : \mathbb{P}^n \to \mathbb{P}^{\binom{n+d}{d}-1}$:

$$(x_0 : \cdots : x_n) \mapsto (x_0^d : x_0^{d-1}x_1 : \cdots : x_n^d).$$

The map ν_d induces a coefficient-gathering map for homogeneous polynomials in fixed degree d:

$$\begin{aligned} \operatorname{coeff}_d : [\mathfrak{R}]_d &\to K^{\binom{n+d}{d}} \\ \sum_{\alpha:|\alpha|=d} c_\alpha x^\alpha &\mapsto \left[c_{\alpha_1}, \ldots, c_{\alpha_{\binom{n+d}{d}}}\right], \end{aligned}$$

where x^{α_i} corresponds to the i-th coordinate of the d-th Veronese embedding. We follow the usual notation $|\alpha| = \sum_i \alpha_i$. Therefore, if f is a homogeneous polynomial of $\deg(f) = d$, $\operatorname{coeff}(f)$ is a vector in the K-vector space $K^{\binom{n+d}{d}}$. This construction can be extended to non-homogeneous polynomials in the following natural way. Consider all distinct total degrees $d_1, \ldots, d_s$ of monomials that appear in a non-homogenous polynomial f. For each d_i, compute the image under $\operatorname{coeff}_{d_i}$ of all monomials of f of degree d_i. Finally, concatenate all these vectors into the total coefficient vector of f, which we will call $\operatorname{coeff}(f)$ and which is of size $\binom{n+d+1}{n+1}$, the number of monomials in $n+1$ variables of (total) degree ranging from 0 to d. In this way, a system $f_1, \ldots, f_m$ of polynomials in n variables of degree at most d can be represented by its *coefficient matrix* of size $\binom{n+d+1}{n+1} \times m$. Each column of this matrix corresponds to the vector produced by the map coeff_d above. This map allows us to think of polynomials as points in a linear affine space, where Helly's theorem applies.

We utilize this construction to import Clarkson's method [17] for solving linear problems to algebraic geometry and, in particular, we make use of Helly-type theorems for varieties. Helly-type theorems allow one to reduce the problem of solving the system to repeated solution of smaller subsystems, whose size is a Helly number of intersecting linear spaces. As a result, our algorithms achieve *expected linear runtime* in the number of input equations.

Theorem 1.1. *Let $F = \{f_1, \ldots, f_m\} \subset \mathfrak{R}$ be a system of polynomials, and let δ be the dimension of the vector subspace generated by the coefficient vectors of the f_i's, as described above.*

Then there exists a sampling algorithm that outputs $F' = \{f_{i_1}, \ldots, f_{i_\delta}\} \subset F$ such that $\mathcal{V}(F) = \mathcal{V}(F')$ in an expected number $O\left(\delta m + \delta^{O(\delta)}\right)$ of calls to the primitive query that solves a small radical ideal membership problem. F and F' generate the same ideal up to radicals.

It is important to point out that our sampling algorithm will find a small subsystem of the input system that, when solved with whatever tools one has at their disposal, will give the same solution set as the original (input) system. Here by 'small' we mean a system of size δ, where δ is polynomially bounded or constant when the number of variables is constant or when the degree d is small.

That the rank δ of the coefficient matrix of the system of polynomials gives the combinatorial dimension for this problem is shown in Theorem 4.11. There are several interesting special cases of this result. For example, we obtain [21, Corollary 2] as a corollary: if $f_1, \ldots, f_m \in K[x_0, \ldots, x_n]$ are homogeneous and of degree at most d each, then the dimension δ of the vector subspace they generate is at most $\binom{n+d}{d}$ (see Lemma 4.6). Of course, in many situations in practice, this bound is not sharp, as many systems are of low rank. For example, this situation can arise if the monomial support of the system is much smaller than the total number of monomials of degree d. In light of this, Theorem 4.11 gives a better bound for low-rank systems. Note that we measure system complexity by its rank, that is, the vector space dimension δ, and not the usual sparsity considerations such as the structure of the monomial support of the system. Further, our result applies to non-homogeneous systems as well. Its proof is presented in Section 4, along with the proof of Theorem 1.1.

Computing small generating sets of ideals. The problem of finding "nice" generating sets of ideals has numerous applications in statistics, optimization, and other fields of science and engineering. Current methods of calculating minimal generating sets of ideals with an a priori large number of generators are inefficient and rely mostly on Gröbner bases computations, since they usually involve ideal membership tests. Of course there are exceptional special cases, such as ideals of points in projective space [40] or binomial systems [1]. Our second main result shows how to efficiently extract a small or close to minimal generating set for any ideal from a given large generating set and a bound on the size of a minimal generating set.

Theorem 1.2. *Let $I = (H)$ be an ideal generated by a (large) finite set of homogeneous polynomials H, and suppose that γ is a known upper bound for the 0-th total Betti number $\beta(R/I)$.*

Then there exists a randomized algorithm that computes a generating set of I of size γ in expected number of $\mathcal{O}(\gamma|H| + \gamma^\gamma)$ calls to the primitive query that solves a small ideal membership problem.

In particular, if $\gamma = \beta(R/I)$, the algorithm computes a minimal *generating set of I.*

The proof is presented in Section 5.

2. A Warm-Up: Algebraic Helly-type theorems and the size of a meaningful sample

A Helly-type theorem has the following form: Given a family of objects F, a property P, and a *Helly number* δ such that every subfamily of F with δ elements has property P, then the entire family has property P. (See [19, 23, 47, 4].) In the original theorem of E. Helly, F is a finite family of convex sets in $\mathbb{R}^n$, the constant δ is $n+1$, and the property P is to have a non-empty intersection [34]. Here we are looking for non-linear algebraic versions of the same concept, where the objects in F are algebraic varieties (hypersurfaces) or polynomials; the property desired is to have a common point, or to generate the same ideal; and the Helly constant δ will be determined from the structure of the problem at hand. To better understand the algorithms that we present, it is instructive to consider two intuitive easy examples that highlight the fundamental combinatorial framework. The first one is an obvious reformulation of Helly's theorem for algebraic geometers.

Example 2.1. Let $H = \{L_1, L_2, \dots, L_s\}$ be a family of affine linear subspaces in $\mathbb{R}^n$. Consider the case when s is much larger than n. One would like to answer the following question: when do all of the linear varieties have a nonempty intersection? It is enough to check whether each subfamily of H with $n+1$ elements has a non-empty intersection, as that would imply, by Helly's Theorem, that H also has a non-empty intersection. Thus, in practice, one can reduce the task of deciding whether $\cap_{i=1}^{s} L_i \neq \emptyset$ to the collection of smaller queries $\cap_{j=1}^{n+1} L_{i_j}$. However, instead of testing all possible $\binom{s}{n+1}$ many $(n+1)$-tuples, we may choose to randomly sample multiple $(n+1)$-tuples. Each time we sample, we either verify that one more $(n+1)$-tuple has a non-empty intersection thus increasing the certainty that the property holds for all $(n+1)$-tuples, or else find a counterexample, a subfamily without a common point, the existence of which trivially implies that $\cap_{i=1}^{s} L_i = \emptyset$. This simple idea is the foundation of a randomized approach. For now we ask the reader to observe that $n+1$ is the dimension of the vector space of (non-homogeneous) linear polynomials in n variables.

Example 2.2. The next example is just slightly more complicated, but illustrates well some key concepts. Consider next $H = \{f_1(x_1, x_2), f_2(x_1, x_2), \dots, f_s(x_1, x_2)\}$, a large family of affine real plane curves of degree at most d. Imagine that H is huge, with millions of constraints f_i, but the curves are of small degree, say $d = 2$. Nevertheless, suppose that we are in charge of deciding whether the curves in H have a common *real* point. Clearly, if the pair of polynomials $f, g \in H$ intersect, they do so in finitely many points, and, in particular, Bezout's theorem guarantees that no more than d^2 intersections occur. One can observe that if the system H has a solution, it must pass through some of the (at most d^2) points defined by the pair f, g alone. In fact, if we take triples $f, g, h \in H$, the same bound of d^2 holds, as well as the fact that the solutions for the entire H must also be part of the solutions for the triplet f, g, h. Same conclusions hold for quadruples, quintuples, and in general δ-tuples. But how large does an integer δ have to be in order to function as a Helly number? We seek a number δ such that if all δ-tuples of plane curves in H intersect, then all of the curves in H must intersect. The reader can easily find examples where $\delta = d$ does not work, e.g., for $d \geq 2$.

To answer the question posed in Example 2.2, we refer to Theorem 4.11 in Section 4. Without re-stating the theorem here, we state the following Corollary and note that it gives a nice bound on δ. Corollary 2.3 is implied by the observation that there are only $\binom{d+2}{2}$ monomials in two variables of degree $\leq d$ (which says they span a linear subspace of that dimension inside the vector space of all polynomials) and Theorem 4.11.

Corollary 2.3. *Let $H = \{f_1(x,y), f_2(x,y), \ldots, f_s(x,y)\}$ be a family of affine real plane curves of degree at most d. If every $\delta = \binom{d+2}{2}$ of the curves have a real intersection point, then all the curves in H have a real intersection point. If we consider the same problem over the complex numbers, then the same bound holds.*

Thus, it suffices to check all δ-tuples of curves for a common real point of intersection, and if all of those instances do intersect, then we are sure all $|H|$ polynomials must have a common intersection point, too. The result suggests a brute-force process to verify real feasibility of the system, which of course is not a pretty proposition, given that $|H|$ is assumed to be very large. Instead, Section 3 explains how to *sample* the set of δ-tuples in order to obtain a solution to the problem more efficiently. Notice that it is important to find a small Helly number δ, as a way to find the smallest sampling size necessary to detect common intersections. It turns out that in this example and in the case when all f_i are homogeneous, the Helly number is best possible [28].

3. Violator spaces and Clarkson's sampling algorithms

The key observation in the previous section was that the existence of Helly-type theorems indicates that there is a natural notion of sampling size to test for a property of varieties. Our goal is to import to computational algebra an efficient randomized sampling algorithm by Clarkson. To import this algorithm, we use the notion of *violator spaces* which we outline in the remainder of this section. We illustrate the definitions using Example 2.2 as a running example in this section.

In 1992, Sharir and Welzl [41] identified special kinds of geometric optimization problems that lend themselves to solution via repeated sampling of smaller subproblems: they called these *LP-type* problems. Over the years, many other problems were identified as LP-type problems and several abstractions and methods were proposed [2, 3, 11, 33, 37]. A powerful sampling scheme, devised by Clarkson [17] for linear programming, works particularly well for geometric optimization problems in small number of variables. Examples of applications include convex and linear programming, integer linear programming, the problem of computing the minimum-volume ball or ellipsoid enclosing a given point set in $\mathbb{R}^n$, and the problem of finding the distance of two convex polytopes in $\mathbb{R}^n$. In 2008, Gärtner, Matoušek, Rüst and Škovroň [31] invented *violator spaces* and showed they give a much more general framework to work with LP-type problems. In fact, violator spaces include all prior abstractions and were proven in [42] to be the most general framework in which Clarkson's sampling converges to a solution. Let us begin with the key definition of a violator space.

Definition 3.1 ([31])**.** A *violator space* is a pair (H, V), where H is a finite set and V a mapping $2^H \to 2^H$, such that the following two axioms hold:
Consistency: $G \cap \mathsf{V}(G) = \emptyset$ holds for all $G \subseteq H$, and
Locality: $\mathsf{V}(G) = \mathsf{V}(F)$ holds for all $F \subseteq G \subseteq H$ such that $G \cap \mathsf{V}(F) = \emptyset$.

Example 3.2 (Example 2.2, continued)**.** To illustrate our definition, we consider Example 2.2 of s real plane curves $\{f_1, \ldots, f_s\} = H$.

A violator operator for testing the existence of a real point of intersection of a subset $F \subset H$ of the curves should capture the real intersection property. One possible way to define it is the following map $\mathsf{V}_{real} : 2^H \to 2^H$:

$$\mathsf{V}_{real}(F) = \{h \in H : \mathcal{V}_{\mathbb{R}}(F) \supsetneq \mathcal{V}_{\mathbb{R}}(F \cup \{h\})\},$$

where $\mathcal{V}_{\mathbb{R}}(F)$ is the set of common real intersection points of F, in other words, the real algebraic variety of F. Note that, by definition, $\mathsf{V}_{real}(F) = \emptyset$ if the curves in F have no real points of intersection. Before explaining why V_{real} captures the real intersection property correctly (see Example 3.6), let us show that the set (H, V_{real}) is a violator space according to Definition 3.1.

Consistency holds by definition of V_{real}: for any $h \in F$, $\mathcal{V}_{\mathbb{R}}(F) = \mathcal{V}_{\mathbb{R}}(F \cup \{h\})$, and so $h \notin \mathsf{V}_{real}(F)$.

To show locality, we begin by showing the auxiliary fact that $\mathcal{V}_{\mathbb{R}}(F) = \mathcal{V}_{\mathbb{R}}(G)$. The inclusion $\mathcal{V}_{\mathbb{R}}(G) \subseteq \mathcal{V}_{\mathbb{R}}(F)$ is direct. This is because for any $S' \subseteq S$ it is always the case that $\mathcal{V}_{\mathbb{R}}(S') \supseteq \mathcal{V}_{\mathbb{R}}(S)$. We need only show $\mathcal{V}_{\mathbb{R}}(F) \subseteq \mathcal{V}_{\mathbb{R}}(G)$. Recall $F \subseteq G \subseteq H$ and $G \cap \mathsf{V}_{real}(F) = \emptyset$. Consider $g \in G$. By the assumption, g does not violate F, and the proper containment $\mathcal{V}_{\mathbb{R}}(F) \supsetneq \mathcal{V}_{\mathbb{R}}(F \cup \{g\})$ does not hold. Therefore the equality $\mathcal{V}_{\mathbb{R}}(F) = \mathcal{V}_{\mathbb{R}}(F \cup \{g\})$ must hold. By iteratively adding elements of $G \smallsetminus F$ to F and repeating the argument, we conclude that indeed $\mathcal{V}_{\mathbb{R}}(F) = \mathcal{V}_{\mathbb{R}}(G)$.

Finally, we argue that $\mathcal{V}_{\mathbb{R}}(F) = \mathcal{V}_{\mathbb{R}}(G)$ implies $\mathsf{V}_{real}(F) = \mathsf{V}_{real}(G)$. We first show $\mathsf{V}_{real}(F) \subset \mathsf{V}_{real}(G)$. Take $h \in \mathsf{V}_{real}(F)$. Then $\mathcal{V}_{\mathbb{R}}(G) = \mathcal{V}_{\mathbb{R}}(F) \supsetneq \mathcal{V}_{\mathbb{R}}(F \cup \{h\}) = \mathcal{V}_{\mathbb{R}}(F) \cap \mathcal{V}_{\mathbb{R}}(\{h\}) = \mathcal{V}_{\mathbb{R}}(G) \cap \mathcal{V}_{\mathbb{R}}(\{h\}) = \mathcal{V}_{\mathbb{R}}(G \cup \{h\})$. The last and third-to-last equalities follow from the fact that for any two sets S_1, S_2, one always has $\mathcal{V}_{\mathbb{R}}(S_1 \cup S_2) = \mathcal{V}_{\mathbb{R}}(S_1) \cap \mathcal{V}_{\mathbb{R}}(S_2)$. The containment $\mathsf{V}_{real}(F) \supset \mathsf{V}_{real}(G)$ follows in a similar argument. Thus (H, V_{real}) is a violator space.

Every violator space comes equipped with three important components: a notion of *basis*, its *combinatorial dimension*, and a *primitive test* procedure. We begin with the definition of a basis of a violator space, analogous to the definition of a basis of a linear programming problem: a minimal set of constraints that defines a solution space.

Definition 3.3 ([31, Definition 7])**.** Consider a violator space (H, V). $B \subseteq H$ is a *basis* if $B \cap \mathsf{V}(F) \neq \emptyset$ holds for all proper subsets $F \subset B$. For $G \subseteq H$, a basis of G is a minimal subset B of G with $\mathsf{V}(B) = \mathsf{V}(G)$.

It is very important to note that *a violator operator can capture algebraic problems of interest* as long as the basis for that violator space corresponds to a basis of the algebraic object we study. Violator space bases come with a natural combinatorial invariant, related to Helly numbers we discussed earlier.

Definition 3.4 ([31, Definition 19])**.** The size of a largest basis of a violator space (H, V) is called the *combinatorial dimension* of the violator space and denoted by $\delta = \delta(H, V)$.

A crucial property was proved in [31]: knowing the violations $\mathsf{V}(G)$ for all $G \subseteq H$ is enough to compute the largest bases. To do so, one can utilize Clarkson's randomized algorithm to compute a basis of a violator space (H, V) with $m = |H|$. The results about the runtime and the size of the sets involved are summarized below. The primitive operation, used as black box in all stages of the algorithm, is the *violation test primitive*.

Definition 3.5. Given a violator space (H, V), some set $G \subsetneq H$, and some element $h \in H \setminus G$, the *primitive* test decides whether $h \in \mathsf{V}(G)$.

The running example illustrates these three key ingredients.

Example 3.6 (Example 3.2, continued)**.** In the example of s real plane curves, the violator operator we defined detects whether the polynomials have a real point of intersection. Note that a basis would be a (minimal) set of curves $B = \{f_{i_1}, \dots, f_{i_\delta}\}$, for some $\delta < s$, such that either the curves in B have no real point of intersection, or the real points of intersection of the curves in B are the real intersection of all of $H = \{f_1, \ldots, f_s\}$. If the set F has no real intersection point, then $\mathsf{V}_{real}(F) = \emptyset$ by definition, so that set F could be a basis in the sense that it is a certificate of infeasibility for this real-intersection problem. If, on the other hand, F does have a real intersection point, and $\mathsf{V}_{real}(F) = \emptyset$, then this means that F is a basis in the sense that the curves in F capture the intersections of all of H. The combinatorial dimension for general H is provided by Corollary 2.3, and it equals $\delta = \binom{d+2}{2}$. However, special structure of the curves in H may imply a smaller combinatorial dimension.

The primitive query simply checks, given $f_i \in H$ and a candidate subset $G \subseteq H$, whether the set of real points of intersection of $G \cup \{f_i\}$ is smaller than the set of real points of intersection of the curves in G alone. The role of the primitive query is therefore not to find a basis directly, but to check, instead, whether a given candidate subset G can be a basis of H. This can be done by checking whether $f_i \in \mathsf{V}_{real}(G)$ for all $f_i \in H \setminus G$. Clearly, given the primitive test, a basis for H can be found by simply testing all sets of size at most δ, but that would be a waste because the number of times one would need to call the primitive would be $O(|H|^{\delta+1})$.

As we will see, this brute-force approach can be avoided. Namely, in our current example, the randomized algorithm from Theorem 3.7 below will only sample subsets of $\delta = \binom{d+2}{2}$ curves from the set $\{f_1, \ldots, f_s\}$, and find a basis of the violator space of size δ in the sense explained above.

The sampling method in [17] avoids a full brute-force approach. It is presented in two stages, referred to as Clarkson's first and second algorithm. We outline these below.

Clarkson's first algorithm, in the first iteration, draws a small random sample $R \subset G$, calls the second stage to calculate the basis C of R, and returns C if it is already a basis for the larger subset G. If C is not already a basis, but the elements of $G \setminus C$ violating R are few, it adds those elements to a growing set of violators W, and repeats the process with C being calculated as the basis of the set $W \cup R$ for a new randomly chosen small $R \subset G \setminus W$. The crucial point here is that $|R|$ is much smaller than $|G|$ and, consequently, it acts as a Helly number of sorts.

Clarkson's second algorithm (Basis2) iteratively picks a random small ($6\delta^2$ elements) subset R of G, finds a basis C for R by exhaustively testing each possible subset (BruteForce;) taking advantage of the fact that the sample R is very small, and then calculates the violators of $G \setminus C$. At each iteration, elements that appear in bases with small violator sets get a higher probability of being selected.

This idea is very important: we are biasing the sampling process, so that some constraints will be more likely to be chosen. This is accomplished by considering every element h of the set G as having a multiplicity $\mathfrak{m}(h)$; the multiplicity of a set is the sum of the multiplicities of its elements. The process is repeated until a basis of G is found, i.e. until $\mathsf{V}(G \setminus C)$ is empty.

Again, as described above, all one needs is to be able to answer the *Primitive query:* Given $G \subset H$ and $h \in H \setminus G$, decide whether $h \in V(G)$. The runtime is given in terms of the combinatorial dimension $\delta(H, V)$ and the size of H. The key result we will use in the rest of the paper concerns the complexity of finding a basis:

Theorem 3.7. [31, Theorem 27] *Using Clarkson's algorithms, a basis of H of a violator space (H, V) can be found by answering the primitive query an expected $O\left(\delta\,|H| + \delta^{O(\delta)}\right)$ times.*

It is very important to note that, in both stages of Clarkson's method, the query $h \in \mathsf{V}(C)$ is answered via calls to the primitive as a black box. In our algebraic applications, the primitive computation requires solving a small-size subsystem (e.g., via Gröbner bases or numerical algebraic geometry methods), or an ideal membership query applied to the ideal generated by a small subset of the given polynomials. On the other hand, the combinatorial dimension relates to the Helly number of the problem which is usually a number that is problem-dependent and requires non-trivial mathematical results.

Algorithm 1: Clarkson's first algorithm

input : $G \subseteq H$, δ: combinatorial complexity of H
output: $\mathcal{B}$, a basis for G

if $|G| \leq 9\delta^2$ **then**
  return Basis2(G)
else
  $W \leftarrow \emptyset$
  repeat
    $R \leftarrow$ random subset of $G \setminus W$ with $\lfloor \delta\sqrt{|G|} \rfloor$ elements.
    $C \leftarrow$ Basis2$(W \cup R)$
    $V \leftarrow \{h \in G \setminus C \text{ s.t. } h \in \mathsf{V}(C)\}$
    if $|V| \leq 2\sqrt{|G|}$ **then**
      $W \leftarrow W \cup V$
    end
  until $V = \emptyset$
end
return C.

Algorithm 2: Clarkson's second algorithm: Basis2(G)

input : $G \subseteq H$; δ: combinatorial complexity of H.
output: $\mathcal{B}$: a basis of G

if $|G| \leq 6\delta^2$ **then**
  return BruteForce(G)
else
  repeat
    $R \leftarrow$ random subset of G with $6\delta^2$ elements.
    $C \leftarrow$ BruteForce(R)
    $V \leftarrow \{h \in G \setminus C \text{ s.t. } h \in \mathsf{V}(C)\}$
    if $\mathfrak{m}(V) \leq \mathfrak{m}(G)/3\delta$ **then**
      for $h \in V$ **do**
        $\mathfrak{m}(h) \leftarrow 2\mathfrak{m}(h)$
      end
    end
  until $V = \emptyset$
end
return C.

In the two sections that follow we show how violator spaces naturally arise in non-linear algebra of polynomials.

4. A violator space for solving overdetermined systems

We discuss our random sampling approach to solve large-size (non-linear) polynomial systems by applying Clarkson's algorithm. In particular, we prove Theorem 1.1 as a corollary of Theorem 4.11. This result is motivated by, and extends, Helly-type theorems for varieties from [21] and [28], which we use to show that the above algorithms apply to large dense homogeneous systems as well (Corollary 4.7).

First, we define a violator space that captures (in the sense explained in the previous section) solvability of a polynomial system.

Definition 4.1. [Violator Space for solvability of polynomial systems] Let $S \subset H$ be finite subsets of polynomials in $\mathfrak{R}$. Define the violator operator $\mathsf{V}_{solve} : 2^H \to 2^H$ to record the set of polynomials in H which do not vanish on the variety $\mathcal{V}(S)$. Formally,

$$\mathsf{V}_{solve}(S) = \{f \in H : \mathcal{V}(S) \text{ is not contained in } \mathcal{V}(f)\}.$$

Lemma 4.2. *The pair* (H, V_{solve}) *is a violator space.*

Proof. Note that $\mathsf{V}_{solve}(S) \cap S = \emptyset$ by definition of $\mathsf{V}_{solve}(S)$, and thus the operator satisfies the consistency axiom.

To show locality, suppose that $F \subsetneq G \subset H$ and $G \cap \mathsf{V}_{solve}(F) = \emptyset$. Since $F \subsetneq G$ we know that $\mathcal{V}(G) \subseteq \mathcal{V}(F)$. On the other hand, by definition, $G \cap \mathsf{V}_{solve}(F) = \emptyset$ implies that $\mathcal{V}(F) \subseteq \mathcal{V}(g)$ for all $g \in G$. Thus $\mathcal{V}(F)$ is contained in $\bigcap_{g \in G} \mathcal{V}(g) = \mathcal{V}(G)$. But then the two varieties are equal.

To complete the argument we show that $\mathcal{V}(F) = \mathcal{V}(G)$ implies $\mathsf{V}_{solve}(F) = \mathsf{V}_{solve}(G)$. If $h \in \mathsf{V}_{solve}(F)$ then $\mathcal{V}(h)$ cannot contain $\mathcal{V}(F) = \mathcal{V}(G)$, thus $h \in \mathsf{V}_{solve}(G)$ too. The argument is symmetric, hence $\mathsf{V}_{solve}(F) = \mathsf{V}_{solve}(G)$. □

It follows from the definition that the operator V_{solve} gives rise to a violator space for which a basis B of $G \subset H$ is a set of polynomials such that $\mathcal{V}(B) = \mathcal{V}(G)$. Therefore, a basis $B \subset G$ will either be a subset of polynomials that has no solution and as such be a *certificate of infeasibility* of the whole system G, or it will provide a set of polynomials that are sufficient to find *all common solutions* of G, i.e., the variety $\mathcal{V}(G)$.

Next, we need a violation primitive test that decides whether $h \in \mathsf{V}_{solve}(F)$, as in Definition 3.5. By the definition above, this is equivalent to asking whether h vanishes on all irreducible components of the algebraic variety $\mathcal{V}(F)$. As is well known, the points of $\mathcal{V}(F)$ where the polynomial h does not vanish correspond to the variety associated with the saturation ideal $((F) : h^\infty)$. Thus, we may use ideal saturations for the violation primitive. For completeness, we recall the following standard definitions. The *saturation* of the ideal (F) with respect to f, denoted by $((F) : f^\infty)$, is defined to be the ideal of polynomials $g \in \mathfrak{R}$ with $f^m g \in I$ for some $m > 0$. This operation removes from the variety $\mathcal{V}(F)$ the irreducible components on which the polynomial f vanishes. Recall that every variety can be decomposed into irreducible components (cf. [18, Section 4.6] for example). The corresponding algebraic operation is the primary decomposition of the ideal defining this variety.

Lemma 4.3 (e.g. [5, Chapter 4])**.** *Let* $\cap_{i=1}^m Q_i$ *be a minimal primary decomposition for the ideal* I*. The saturation ideal* $(I : f^\infty)$ *equals* $\cap_{f \notin \sqrt{Q_i}} Q_i$*.*

Proof. It is known that $(\cap_{i=1}^m Q_i : f^\infty) = \cap_{i=1}^m (Q_i : f^\infty)$. We observe further that $(Q_i : f^\infty) = Q_i$ if f does not belong to $\sqrt{Q_i}$ and $(Q_i : f^\infty) = (1)$ otherwise. □

This allows us to set up the primitive query for (H, V_{solve}). However we do not need to calculate the decomposition explicitly, but can instead carry it out using elimination ideals via Gröbner bases, as explained for example in [18, Exercise 4.4.9].

Observation 4.4. The primitive query for (H, V_{solve}) is simply *the saturation test* explained above.

Remark 4.5. There is an obvious reformulation of these two ingredients that is worth stating explicitly. Namely, since a basis B for the violator space (H, V_{solve}) is a set of polynomials such that $\mathcal{V}(B) = \mathcal{V}(H)$, the strong Nullstellensatz implies that $\sqrt{(B)} = \sqrt{(H)}$. Thus a basis determines the ideal of the input system up to radicals, and we could have named the violator operator

$\mathsf{V}_{solve} \equiv \mathsf{V}_{radical}$ instead. Furthermore, a polynomial h vanishing on all irreducible components of the algebraic variety $\mathcal{V}(F)$ is equivalent to $h \in \sqrt{(F)}$, i.e., h belonging to the radical of the ideal (F). In particular, the primitive query for V_{solve} can also be stated as the *radical ideal membership test.* This test can be implemented using Gröbner bases, as explained for example in [18, Proposition 4.2.8]: $h \in \sqrt{(F)}$ if and only if $1 \in (F, 1-yh) \subseteq K[x_0, \dots, x_n, y]$. Therefore, computation of one Gröbner basis of the ideal $(F, 1-yh)$ suffices to carry out this test.

Finally, we solve the problem of finding a combinatorial dimension for V_{solve}. For this, consider, as a warm up, the simpler situation where we have a Helly-type theorem for hypersurfaces defined by *homogeneous* polynomials. This was proved by Motzkin [39] and then later reproved by Deza and Frankl [21], and it provides us with a combinatorial dimension for guaranteeing that a large-scale homogeneous system has a solution. Its proof relies on thinking of the polynomial ring $\mathfrak{R}$ as a K-vector space (see also the discussion before Definition 4.9).

Lemma 4.6 ([21], Corollary 2)**.** *Let $f_1, \dots, f_m \subset \mathfrak{R}$ be a system of* homogeneous *polynomials, that is, $f_i \in [\mathfrak{R}]_{d_i}$, and define $d = \max\{d_i\}$. Suppose that every subset of $p = \binom{n+d}{d}$ polynomials $\{f_{i_1}, \dots, f_{i_p}\} \subset \{f_1, \dots, f_m\}$ has a solution. Then the entire system $\{f_1, \dots, f_m\}$ does as well.*

Lemma 4.6 provides the combinatorial dimension that, along with the variety membership primitive from Observation 4.4, allows us to apply Clarkson's algorithms to the violator space (H, V_{solve}).

Corollary 4.7. *Let $(f_1, \dots, f_m) \subset \mathfrak{R}$ be an ideal generated by m homogeneous polynomials in $n+1$ variables of degree at most d; $f_i \in [\mathfrak{R}]_{d_i}$ and $d = \max\{d_i\}$. Let $\delta = \binom{n+d}{d}$. Then there is an adaptation of Clarkson's sampling algorithm that, in an expected $O\left(\delta m + \delta^{O(\delta)}\right)$ number of calls to the primitive query 4.4, computes $\{f_{i_1}, \dots, f_{i_\delta}\}$ such that $\mathcal{V}(f_1, \dots, f_m) = \mathcal{V}(f_{i_1}, \dots, f_{i_\delta})$.*

In particular, this algorithm is linear in the number of input equations m, and a randomized polynomial time algorithm when the number of variables $n+1$ and the largest degree d are fixed. Furthermore, we can extend it to actually *solve* a large system: once a basis $B = \{f_{i_1}, \dots, f_{i_\delta}\}$ for the space $(\{f_1, \dots, f_m\}, \mathsf{V}_{solve})$ is found, then we can use any computer algebra software (e.g. [6, 32, 46]) to solve $f_{i_1} = \cdots = f_{i_\delta} = 0$.

Note that Lemma 4.6 can be thought of as a statement about the complexity of Hilbert's Nullstellensatz. If $(f_1, \dots, f_m) = \mathfrak{R}$ (i.e., $\mathcal{V}(f_1, \dots, f_m) = \emptyset$), then there exists a subset of size $\delta = \binom{n+d}{d}$ polynomials $\{f_{i_1}, \dots, f_{i_\delta}\}$ such that $\mathcal{V}(f_{i_1}, \dots, f_{i_\delta}) = \emptyset$ as well. In particular, there is a Nullstellensatz certificate with that many elements. The dimension $\binom{n+d}{d}$ is, in fact, only an upper bound, attainable only by dense systems. However, in practice, many systems are very large but sparse, and possibly non-homogeneous. Let us highlight again that the notion of 'sparsity' we consider is captured by a low-rank property of the system of polynomial equations, made explicit below in terms of the coefficient matrix. This is crucially different from the usual considerations of monomial supports (Newton polytopes) of the system; instead, we look at the coefficients of the input polynomials - that is, we linearize the problem and consider the related vector spaces, as illustrated in the following example.

Example 4.8. Consider the following system consisting of two types of polynomials: polynomials of the form $x_i^2 - 1$ for $i = 1, \dots, n$, and polynomials of the form $x_i + x_j$ for the pairs $\{i, j : i \not\equiv j \mod 2\}$ along with the additional pair $i = 1, j = 3$. This system has $m = n^2 + n + 1$ equations, and the interesting situation is when the number of variables is a large even number, that is, $n = 2k$ for any large integer k. This system of polynomials generates the 2-coloring ideal of a particular n-vertex non-chordal graph. (See [20] and references therein for our motivation to consider this particular system.)

Consider a concrete graph. Take the n-cycle with all possible even chords, and one extra edge $\{1,3\}$. Thus the pairs $\{i,j\}$ are indexed by edges of the graph G on n nodes where all odd-numbered vertices are connected to all even-numbered vertices, and with one additional edge $\{1,3\}$.

We wish to decide if the system has a solution, but since there are n^2+n+1 many polynomials, we would like to try to avoid computing a Gröbner basis of this ideal. Instead, we search for a subsystem of some specific size that determines the same variety. It turns out that the system actually has no solution. Indeed, a certificate for infeasibility is a random subsystem consisting of the first n quadratic equations, $n-1$ of the edge equations x_i+x_j with $\{i,j : i \not\equiv j \mod 2\}$, and the additional equation x_1+x_3. For example, the first $n-1$ edge polynomials will do to construct a $2n$-sized certificate of this form. Why is the number $n+(n-1)+1=2n$ so special?

To answer this question, let us linearize the problem: to each of the polynomials f associate a coefficient (column) vector $\mathrm{coeff}(f) \in \mathbb{C}^{2n+1}$ whose coordinates are indexed by the monomials appearing in the system $x_1^2,\dots,x_n^2,1,x_1,\dots,x_n$. Putting all these column vectors in one matrix produces the coefficient matrix of the system of the form

$$\begin{bmatrix} I_n & 0 \\ -\mathbf{1} & 0 \\ 0 & E \end{bmatrix},$$

where I_n is the $n\times n$ identity, $-\mathbf{1}$ is the row vector with all entries -1, and E is the vertex-edge incidence matrix of the graph G. Since it is known that the rank of an edge-incidence matrix of an n-vertex connected graph is $n-1$, the rank of this matrix is $\delta = n+(n-1)+1 = 2n$.

Remarkably, the magic size of the infeasibility certificate equals the rank of this coefficient matrix.

This motivating example suggests that the desired Helly-type number of this problem is captured by a natural low-rank property of the system. To define it precisely, let us revisit the extension of the Veronese embedding to non-homogeneous polynomials explained in the Introduction. Here we adopt the notation from [7, Section 2] and consider polynomials in $\mathfrak{R}$ of degree up to d as a K-vector space denoted by $\mathcal{C}_d^{n+1}$. The vector space $\mathcal{C}_d^{n+1}$ has dimension $\binom{d+n+1}{n+1}$, which, of course, equals the number of monomials in $n+1$ variables of (total) degree from 0 to d. In this way, any polynomial $f\in\mathfrak{R}$ is represented by a (column) vector, $\mathrm{coeff}(f)\in\mathcal{C}_d^{n+1}$, whose entries are the *coefficients* of f. Thus, any system $S\subset\mathfrak{R}$ defines a matrix with $|S|$ columns, each of which is an element of $\mathcal{C}_d^{n+1}$.

Definition 4.9. A system $S\subset\mathfrak{R}$ is said to have *rank D* if $\dim_K\langle S\rangle = D$, where $\langle S\rangle$ is the vector subspace of $\mathcal{C}_d^{n+1}$ generated by the coefficients of the polynomials in S.

We need to also make the notion of Helly-type theorems more precise in the setting of varieties.

Definition 4.10 (Adapted from Definition 1.1. in [28])**.** A set $S\subset\mathfrak{R}$ is said to have *the D-Helly property* if for every nonempty subset $S_0\subset S$, one can find $p_1,\dots,p_D\in S_0$ with $\mathcal{V}(S_0)=\mathcal{V}(p_1,\dots,p_D)$.

The following result, which implies Theorem 1.1, is an extension of [28] to non-homogeneous systems. It also implies (the contrapositive of) Lemma 4.6 when restricted to homogeneous systems, in the case when the system has no solution. The proof follows that of [28], although we remove the homogeneity assumption. We include it here for completeness.

Theorem 4.11. *Any polynomial system $S\subset\mathfrak{R}$ of rank D has the D-Helly property.*

In other words, for all subsets $\mathcal{P}\subset S$, there exist $p_1,\dots,p_D\in\mathcal{P}$ such that $\mathcal{V}(\mathcal{P})=\mathcal{V}(p_1,\dots,p_D)$.

Proof. Let $\mathcal{P}\subset S$ be an arbitrary subset of polynomials, and denote by $\langle\mathcal{P}\rangle\subset\mathcal{C}_d^{n+1}$ the vector subspace it generates. Let $d_0=\dim_K\langle\mathcal{P}\rangle$. We need to find polynomials $p_1,\dots,p_D$ such that $\mathcal{V}(p_1,\dots,p_D)=\mathcal{V}(\mathcal{P})$. Note that $d_0\le D$, of course, so it is sufficient to consider the case $\mathcal{P}=S$.

Choose a vector space basis $\langle p_1,\dots,p_D\rangle=\langle\mathcal{P}\rangle=\langle S\rangle$. It suffices to show $\mathcal{V}(p_1,\dots,p_D)\subseteq\mathcal{V}(S)$; indeed, the inclusions $\mathcal{V}(S)\subseteq\mathcal{V}(\mathcal{P})\subseteq\mathcal{V}(p_1,\dots,p_D)$ already hold.

Suppose, on the contrary, there exists $x = (x_0, \dots, x_n) \in \mathbb{C}^{n+1}$ and $p \in S$ such that $p(x) \neq 0$ but $p_i(x) = 0$ for all $i = 1, \dots, D$. Since p_i's generate S as a vector space, there exist constants $\gamma_i \in K$ with $p = \sum_i \gamma_i p_i$, implying that $p(x) = 0$, a contradiction. $\square$

The proof above is constructive: to find a subset $p_1, \dots, p_D \in S_0$, one only needs to compute a vector space basis for $\langle S_0 \rangle$. Thus, linear algebra (i.e., Gaussian elimination) can construct this subset in time $O(|S_0|^3)$. The sampling algorithm based on violator spaces is more efficient.

Proof of Theorem 1.1. From Lemma 4.2, we know that $(\{f_1, \dots, f_m\}, \mathsf{V}_{solve})$ is a violator space. Theorem 4.11 shows that it has a combinatorial dimension, and Observation 4.4 shows that there exists a way to answer the primitive test. Having these ingredients, Theorem 3.7 holds and it is possible for us to apply Clarkson's Algorithm again. $\square$

Remark 4.5 provides the following interpretation of Theorem 4.11:

Corollary 4.12. *Let $I = (f_1, \dots, f_m) \subset \mathfrak{R}$ and let $D = \dim_K \langle f_1, \dots, f_m \rangle$. Then, for all subsets $\mathcal{P}$ of the generators $f_1, \dots, f_m$, there exist $p_1, \dots, p_D \in \mathcal{P}$ such that $\sqrt{\mathcal{P}} = \sqrt{(p_1, \dots, p_D)}$.*

5. A violator space for finding generating sets of small cardinality

In this section, we apply the violator space approach to obtain a version of Clarkson's algorithm for calculating small generating sets of general homogeneous ideals as defined on page 3. As in Section 4, this task rests upon three ingredients: the appropriate violator operator, understanding the combinatorial dimension for this problem, and a suitable primitive query which we will use as a black box. As before, fixing the definition of the violator operator induces the meaning of the word 'basis', as well as the construction of the black-box primitive.

To determine the natural violator space for the ideal generation problem, let $I \subset \mathfrak{R}$ be a *homogeneous* ideal, H some initial generating set of I, and define the operator $\mathsf{V}_{SmallGen}$ as follows.

Definition 5.1. [Violator Space for Homogeneous Ideal Generators] Let $S \subset H$ be finite subsets of $\mathfrak{R}$. We define the operator $\mathsf{V}_{SmallGen} : 2^H \to 2^H$ to record the set of polynomials in H that are not in the ideal generated by the polynomials in S. Formally,

$$\mathsf{V}_{SmallGen}(S) = \{f \in H : (S, f) \supsetneq (S)\}.$$

Equivalently, the operator can be viewed as $\mathsf{V}_{SmallGen}(S) = \{f \in H : f \notin (S)\}$.

Lemma 5.2. *The pair $(H, \mathsf{V}_{SmallGen})$ is a violator space.*

Proof. Note that $\mathsf{V}_{SmallGen}(S) \cap S = \emptyset$ by definition of $\mathsf{V}_{SmallGen}(S)$, and thus the operator satisfies the consistency axiom.

To show locality, suppose that $F \subsetneq G \subset H$ and $G \cap \mathsf{V}_{SmallGen}(F) = \emptyset$. Since $F \subsetneq G$, $(F) \subseteq (G)$. On the other hand $G \cap \mathsf{V}_{SmallGen}(F) = \emptyset$ implies that $G \subseteq (F)$ which in turn implies that $(G) \subseteq (F)$. Then the ideals are equal. Then, because $(G) = (F)$ we can prove that $\mathsf{V}_{SmallGen}(F) = \mathsf{V}_{SmallGen}(G)$. Note first that $(G) = (F)$, holds if and only if $(G, h) = (F, h)$ for all polynomials $h \in H$.

Finally, to show $\mathsf{V}_{SmallGen}(F) = \mathsf{V}_{SmallGen}(G)$, we note that $h \in \mathsf{V}_{SmallGen}(F)$ if and only if $(G, h) = (F, h) \supsetneq (F) = (G)$, and this chain of equations and containment holds if and only if $h \in \mathsf{V}_{SmallGen}(G)$. Therefore, locality holds as well and $\mathsf{V}_{SmallGen}$ is a violator space operator. $\square$

It is clear from the definition that $(H, \mathsf{V}_{SmallGen})$ is a violator space for which the basis of $G \subset H$ is a minimal generating set of the ideal (G).

The next ingredient in this problem is the combinatorial dimension: the size of the largest minimal generating set. This natural combinatorial dimension already exists in commutative algebra, namely, it equals a certain Betti number. (Recall that *Betti numbers* are the ranks $\beta_{i,j}$ of modules in the minimal (graded) free resolution of the ring $\mathfrak{R}/I$; see, for example, [24, Section 1B (pages 5-9)].)

Specifically, the number $\beta_{0,j}$ is defined as the number of elements of degree j required among any set of minimal generators of I. The *(0-th) total Betti number* of $\mathfrak{R}/I$, which we will denote by $\beta(\mathfrak{R}/I)$, simply equals $\sum_j \beta_{0,j}$, and is then the total number of minimal generators of the ideal I. It is well known that while I has many generating sets, every minimal generating set has the same cardinality, namely $\beta(\mathfrak{R}/I)$. In conclusion, it is known that

Observation 5.3. The combinatorial dimension for $(H, \mathsf{V}_{SmallGen})$ is the (0-th) total Betti number of the ideal $I = (H)$; in symbols, $\beta(\mathfrak{R}/I) = \delta(H, \mathsf{V}_{SmallGen})$.

Although it may be difficult to exactly compute $\beta(\mathfrak{R}/I)$ in general, a natural upper bound for $\beta(\mathfrak{R}/I)$ is the Betti number for any of its initial ideals (the standard inequality holds by upper-semicontinuity; see e.g. [38, Theorem 8.29]). In particular, if H is known to contain a Gröbner basis with respect to some monomial order, then the combinatorial dimension can be estimated by computing the minimal generators of an initial ideal of (H), which is a monomial ideal problem and therefore easy. In general, however, we only need $\beta(\mathfrak{R}/I) < |H|$ for the proposed algorithms to be efficient.

The last necessary ingredient is the primitive query for $\mathsf{V}_{SmallGen}$.

Observation 5.4. The primitive query for $\mathsf{V}_{SmallGen}$, deciding if $h \in \mathsf{V}_{SmallGen}(G)$ given $h \in H$ and $G \subset H$, is an ideal membership test.

Of course, the answer to the query is usually Gröbner-based, but, as before, the size of the subsystems G on which we call the primitive query is small: $(O(\delta^2))$. In fact, it is easy to see that many small Gröbner computations for ideal membership cost less than the state-of-the-art, which includes at least one large Gröbner computation.

Proof of Theorem 1.2. From Lemma 5.2 we know $(H, \mathsf{V}_{SmallGen})$ is a violator space and we have shown it has a combinatorial dimension and a way to answer the primitive test. Having these ingredients, Theorem 3.7 holds and it is possible for us to apply Clarkson's Algorithm. □

Remark 5.5. Intuitively, the standard algorithm for finding minimal generators needs to at least compute a Gröbner basis for an ideal generated by $|H|$ polynomials, and in fact it is much worse than that. One can simplify this by skipping the computation of useless S-pairs (e.g. as in [30]), but improvement is not by an order of magnitude, overall. The algorithm remains doubly exponential in the size of H for general input. In contrast, our randomized algorithm distributes the computation into many small Gröbner basis calculations, where "many" means no more than $\mathcal{O}(\beta|H| + \beta^\beta)$, and "small" means the ideal is generated by only $O(\beta^2)$ polynomials.

To conclude, in a forthcoming paper we will study further the structure of our violator spaces and discuss the use of more LP-type methods for the same algorithmic problems. We also intend to present some experimental results for the sampling techniques we discussed here. We expect better performance of the randomized methods versus the traditional deterministic algorithms.

Acknowledgements

We are grateful to Nina Amenta, Robert Ellis, Diane Maclagan, Pablo Soberón, Cynthia Vinzant, and three anonymous referees for many useful comments and references. The first author is grateful to IMA, the Institute of Mathematics and its Applications of the Univ. of Minnesota for the support received when this paper was being written. He was also supported by a UC MEXUS grant. This work was supported by the NSF grant DMS-1522662.

References

[1] 4ti2 team. 4ti2-a software package for algebraic, geometric and combinatorial problems on linear spaces combinatorial problems on linear spaces.

[2] N. Amenta. Bounded boxes, Hausdorff distance, and a new proof of an interesting Helly-type theorem. In *Proceedings of the 10th Annual Symposium on Computational Geometry (SCG)*, pages 340–347. ACM Press, 1994.

[3] N. Amenta. Helly theorems and generalized linear programming. *Discrete and Computational Geometry*, 12:241–261, 1994.

[4] Nina Amenta, Jesús A. De Loera, and Pablo Soberón. Helly's theorem: New variations and applications. Preprint. *arXiv:1508.07606. math.MG (math.CO).*

[5] M. F. Atiyah and I. G. Macdonald. *Introduction to commutative algebra*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1969.

[6] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Bertini: Software for numerical algebraic geometry. Available at bertini.nd.edu with permanent doi: dx.doi.org/10.7274/R0H41PB5.

[7] K. Batselier, P. Dreesen, and B. De Moor. The geometry of multivariate polynomial division and elimination. *SIAM Journal on Matrix Analysis and Applications*, 34(1):102–125, 2013.

[8] C. Beltrán and L.M. Pardo. On Smale's 17th problem: a probabilistic positive solution. *Foundations of Computational Mathematics*, 8(1):1–43, 2008.

[9] C. Beltrán and L.M Pardo. Smale's 17th problem: average polynomial time to compute affine and projective solutions. *J. Amer. Math. Soc.*, 22(2):363–385, 2009.

[10] E. R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24(111):713–735, 1970.

[11] H. Björklund, S. Sandberg, and S. Vorobyov. A discrete subexponential algorithm for parity games. In *Proc. 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 663–674. Springer-Verlag, 2003.

[12] Y. Brise and B. Gärtner. Clarkson's algorithm for violator spaces. In *21st Canadian Conference on Computational Geometry (CCCG2009)*, pages 9–12, 2009.

[13] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, Leopold-Franzens University, 1965.

[14] B. Buchberger. Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal (English translation). *Journal of Symbolic Computation*, 41(3-4):475–511, 2006.

[15] P. Bürgisser and F. Cucker. On a problem posed by Steve Smale. *Annals of Mathematics*, 174(3):1785–1836, 2011.

[16] D. Cifuentes and P. Parrilo. Exploiting chordal structure in polynomial ideals: a Gröbner bases approach. Preprint: *arXiv:1411.1745v1 [cs.SC].*

[17] K. L. Clarkson. Las Vegas algorithms for linear and integer programming. *Journal of the ACM*, 42(2):488–499, 1995.

[18] D. Cox, J.B. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2007.

[19] L. Danzer, B. Grünbaum, and V. Klee. Helly's theorem and its relatives. In *Proc. Sympos. Pure Math., Vol. VII*, pages 101–180. Amer. Math. Soc., Providence, R.I., 1963.

[20] J. A. De Loera, S. Margulies, M. Pernpeintner, E. Riedl, D. Rolnick, G. Spencer, D. Stasi, and J. Swenson. Graph-coloring ideals: Nullstellensatz certificates, Gröbner bases for chordal graphs, and hardness of Gröbner bases. In *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 2015.

[21] M. Deza and P. Frankl. A Helly type theorem for hypersurfaces. *J. Combin. Theory Ser. A*, 45(1):27–30, 1987.

[22] T.W. Dube. The structure of polynomial ideals and Gröbner bases. *SIAM Journal of Computing*, 19(4):750–773, 1990.

[23] J. Eckhoff. Helly, Radon, and Carathéodory type theorems. In *Handbook of convex geometry, Vol. A, B*, pages 389–448. North-Holland, Amsterdam, 1993.

[24] D. Eisenbud. *The geometry of syzygies*, volume 229 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2005. A second course in commutative algebra and algebraic geometry.

[25] J.-C. Faugére, P. Gaudry, L. Huot, and G. Renault. Polynomial systems solving by fast linear algebra. arXiv preprint arXiv:1304.6039, 2013.

[26] J.-C. Faugére, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[27] J.-C. Faugére, P.-J. Spaenlehauer, and J. Svartz. Sparse Gröbner bases: the unmixed case. arXiv preprint arXiv:1402.7205, 2014.

[28] P. Frankl. Helly-type theorems for varieties. *European J. Combin.*, 10(3):243–245, 1989.

[29] S. Gao. *Counting Zeros over Finite Fields Using Gröbner Bases*. PhD thesis, MS Thesis in Logic and Computation, 2009.
[30] S. Gao, V. M Rodrigues, and J. Stroomer. Gröbner basis structure of finite sets of points. Preprint. Available on `http://www.math.clemson.edu/~sgao/papers/GBstr.pdf`. Last accesed: November 2015., 2003.
[31] B. Gärtner, J. Matoušek, L. Rüst, and P. Škovroň. Violator spaces: structure and algorithms. *Discrete Appl. Math.*, 156(11):2124–2141, 2008.
[32] D. R. Grayson and M. E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.
[33] N. Halman. *Discrete and lexicographic Helly theorems and their relations to LP-type problems*. PhD thesis, Tel-Aviv University, 2004.
[34] E. Helly. Über mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 32(175–176), 1923.
[35] Y. N. Lakshman. On the complexity of computing a Gröbner basis for the radical of a zero dimensional ideal. In *Proceedings of the twenty-second annual ACM Symposium on Theory of Computing*, pages 555–563. ACM, 1990.
[36] J. Matoušek. *Lectures on discrete geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002.
[37] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4–5):498–516, 1996.
[38] E. Miller and B. Sturmfels. *Combinatorial commutative algebra*. Graduate Texts in Mathematics. Springer, 2005.
[39] T. S. Motzkin. A proof of Hilbert's Nullstellensatz. *Math. Z.*, 63:341–344, 1955.
[40] I. Ramella. *Algoritmi di Computer Algebra relativi agli ideali di punti dello spazio proiettivo*. PhD thesis, Universita degli studi di Napoli "Federico II", 1990.
[41] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 577 of *Lecture Notes in Computer Science*, pages 569–579. Springer-Verlag, 1992.
[42] P. Škovroň. *Abstract models of optimization problems*. PhD thesis, Charles University, Prague, 2007.
[43] R. Solovay and V. Strasse. A fast Monte-Carlo test for primality. *SIAM Journal of Computing*, 6(1), 1977.
[44] D.A. Spielman and S.H. Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):76–84, 2009.
[45] B. Sturmfels. *Sparse elimination theory*. Mathematical Sciences Institute, Cornell University, 1991.
[46] J. Verschelde. PHCpack: a general-purpose solver for polynomial systems by homotopy continuation. Available at http://homepages.math.uic.edu/∼jan/download.html.
[47] R. Wenger. Helly-type theorems and geometric transversals. In *Handbook of discrete and computational geometry*, CRC Press Ser. Discrete Math. Appl., pages 63–82. CRC, Boca Raton, FL, 1997.

J. De Loera, Department of Mathematics, University of California, Davis.
E-mail address: `deloera@math.ucdavis.edu`

S. Petrović, Applied Mathematics Department, Illinois Institute of Technology.
E-mail address: `Sonja.Petrovic@iit.edu`

D. Stasi, Applied Mathematics Department, Illinois Institute of Technology.
E-mail address: `despina.stasi@gmail.com`