COMPUTATION WITH POLYNOMIAL EQUATIONS AND INEQUALITIES ARISING IN COMBINATORIAL OPTIMIZATION

JESUS A. DE LOERA*, PETER N. MALKIN†, AND PABLO A. PARRILO‡

Abstract. This is a survey of a recent methodology to solve systems of polynomial equations and inequalities for problems arising in combinatorial optimization. The techniques we discuss use the algebra of multivariate polynomials with coefficients over a field to create large-scale linear algebra or semidefinite programming relaxations of many kinds of feasibility or optimization questions.

Key words. Polynomial equations and inequalities, combinatorial optimization, Nullstellensatz, Positivstellensatz, graph colorability, max-cut, stable sets, semidefinite programming, large-scale linear algebra, semi-algebraic sets, real algebra.

AMS(MOS) subject classifications. 90C27, 90C22, 68W05.

1. Introduction. A wide variety of problems in optimization can be easily modeled using *systems of polynomial equations and inequalities*. Feasibility and optimization problems translate, either directly or via branching, into the problem of finding a solution of a system of equations and inequalities. In this survey paper, we explain how to manipulate such systems for finding solutions or proving that they do not exist. Although these techniques work in general, we are particularly motivated by problems of combinatorial origin. For example, in the case of graphs, here is how one can think about stable sets, k-colorability and max-cut problems in terms of polynomial (non-linear) constraints:

PROPOSITION 1.1. *Let $G = (V, E)$ be a graph.*

- *For a given positive integer k, consider the following polynomial system:*

$$x_i^2 - x_i = 0 \ \ \forall i \in V, \quad x_i x_j = 0 \ \ \forall (i,j) \in E \ \ \text{and} \ \ \sum_{i \in V} x_i = k.$$

This system is feasible if and only if G has a stable set of size k.

*Department of Mathematics, University of California at Davis, Davis, CA 95616 (deloera@math.ucdavis.edu); partially supported by NSF DMS-0914107 and an IBM OCR award.

†Department of Mathematics, University of California at Davis, Davis, CA 95616 (malkin@math.ucdavis.edu); partially supported by an IBM OCR award.

‡Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 (parrilo@mit.edu); partially supported by AFOSR MURI 2003-07688-1 and NSF FRG DMS-0757207.

J. Lee and S. Leyffer (eds.), *Mixed Integer Nonlinear Programming*, The IMA Volumes in Mathematics and its Applications 154, DOI 10.1007/978-1-4614-1927-3_16,

- *For a positive integer k, consider the following polynomial system of $|V| + |E|$ polynomials equations:*

$$x_i^k - 1 = 0 \ \ \forall i \in V \quad \textit{and} \quad \sum_{s=0}^{k-1} x_i^{k-1-s} x_j^s = 0 \ \ \forall (i,j) \in E.$$

  The graph G is k-colorable if and only if this system has a complex solution. Furthermore, when k is odd, G is k-colorable if and only if this system has a common root over $\overline{\mathbb{F}}_2$, the algebraic closure of the finite field with two elements.
- *We can represent the set of cuts of G (i.e., bipartitions on V) as the 0-1 incidence vectors*

$$SG := \{\chi^F \ : \ F \subseteq E \text{ is contained in a cut of } G\} \subseteq \{0,1\}^E.$$

  Thus, the max cut problem with non-negative weights w_e on the edges $e \in E$ is

$$\max\{\sum_{e \in E} w_e x_e \ : \ x \in SG\}.$$

  The vectors χ^F are the solutions of the polynomial system

$$x_e^2 - x_e = 0 \ \forall \ e \in E, \ \textit{and} \ \prod_{i \in T} x_i = 0 \ \ \forall \ T \text{ an odd cycle in } G.$$

There are many other combinatorial problems that can be modeled concisely by polynomial systems (see [9] and the many references therein). In fact, a given problem can often be modeled non-linearly in many different ways, and in practice choosing a "good" formulation is critical for an efficient solution.

Given a polynomial system encoding a combinatorial question, we explain how to use two famous algebraic identities to derive solution methods. In what follows, let $\mathbb{K}$ denote a field and let $\mathbb{K}[x_1, \ldots, x_n] = \mathbb{K}[x]$ denote the ring of polynomials in n variables with coefficients over $\mathbb{K}$. The situation is slightly different depending on whether only equations are being considered, or if there also inequalities (more precisely, on whether the underlying field $\mathbb{K}$ is formally real):

1. First, suppose that the system contains only the polynomial equations $f_1(x) = 0, f_2(x) = 0, \ldots, f_s(x) = 0$ where $f_1, ..., f_s \in \mathbb{K}[x]$. We explain how to generate a finite sequence of *linear algebra* systems over $\mathbb{K}$ which terminate with either a solution over $\overline{\mathbb{K}}$, the algebraic closure of $\mathbb{K}$, or provide a certificate of infeasibility. Crucially for practical computation, the linear algebra systems are over $\mathbb{K}$, not $\overline{\mathbb{K}}$. The calculations reduce to matrix manipulations over $\mathbb{K}$, mostly rank computations. The techniques we use are a

specialization of prior techniques from computational algebra (see [37, 20, 21, 38]). As it turns out this technique is particularly effective when the number of solutions is finite, when $\mathbb{K}$ is a finite field and when the system has nice combinatorial information (see [9]).

2. Second, several authors (see e.g. [23, 41, 29] and references therein) have considered the solvability (over the reals) of systems of polynomial equations and inequalities. It was shown that in this situation there is a way to set up the feasibility problem

$$\exists x \in \mathbb{R}^n \ s.t. \ f_1(x) = 0, \ldots, f_s(x) = 0, \ g_1(x) \geq 0, \ldots, g_k(x) \geq 0,$$

where $f_1, \ldots, f_s, g_1, \ldots, g_k \in \mathbb{R}[x]$, as a sequence of semidefinite programs terminating with a feasible solution (see [41, 29]). Once more, the combinatorial structure can help in the understanding of the structure of these relaxations, as is well-known from the case of stable sets [32] and max-cut [28]. In recent work, Gouveia et al. [15, 14] considered a sequence of semidefinite relaxations of the convex hull of real solutions of a polynomial system encoding a combinatorial problem. They called these approximations *theta bodies* because, for stable sets of graphs, the first theta body in this hierarchy is exactly Lovász's theta body of a graph [32].

The common central idea to both of the relaxations procedures described above is to use the right *infeasibility certificates* or *theorems of alternative*. Just as Farkas' lemma is a centerpiece for the development of Linear Programming, here the key point is that the infeasibility of polynomial systems can *always* be certified by particular algebraic identities (on non-linear polynomials). To find these infeasibility certificates we rely either on *linear algebra* or *semidefinite programming* (for a quick overview of semidefinite programming see [51]).

We now introduce some necessary notation and algebraic concepts. For a detailed introduction we recommend the books [2, 5, 6, 36]. In the paper $\mathbb{K}$ denotes a field and when the distinction is necessary we denote its algebraic closure by $\overline{\mathbb{K}}$. Let $\mathbb{K}[x_1, \ldots, x_n]$ denote the ring of polynomials in n variables with coefficients over $\mathbb{K}$, which will be abbreviated as $\mathbb{K}[x]$. We denote the monomials in the polynomial ring $\mathbb{K}[x]$ as $x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ for $\alpha \in \mathbb{N}^n$. The degree of x^α is $\deg(x^\alpha) := |\alpha| := \sum_{i=1}^n \alpha_i$. The degree of a polynomial $f = \sum_{\alpha \in \mathbb{N}^n} f_\alpha x^\alpha$, written $\deg(f)$, is the maximum degree of x^α where $f_\alpha \neq 0$ for $\alpha \in \mathbb{N}^n$. Given a set of polynomials $F \subset \mathbb{K}[x]$, we write $\deg(F)$ for the maximum degree of the polynomials in F. Given a set of polynomials $F := \{f_1, \ldots, f_m\} \subseteq \mathbb{K}[x]$, we define the *ideal* generated by F as

$$\mathbf{ideal}(F) := \left\{ \sum_{i=1}^m \beta_i f_i \mid \beta_i \in \mathbb{K}[x] \right\}.$$

To study of solutions of a system over a non-algebraically closed field like $\mathbb{R}$ requires extra structure. Given a set of real polynomials $G := \{g_1, \ldots, g_m\} \subseteq \mathbb{R}[x]$, following page 86 in Section 4.2 of [2], we define the *cone* generated by G as

$$\mathbf{cone}(G) := \left\{ \sum_{\alpha \in \{0,1\}^n} s_\alpha g^\alpha \mid s_\alpha \in \mathbb{R}[x] \text{ is SOS} \right\}$$

where $g^\alpha := \prod_{i=1}^m g_i^{\alpha_i}$ and a polynomial $s(x) \in \mathbb{R}[x]$ is SOS if it can be written as a *sum of squares* of other polynomials, that is, $s(x) = \sum_i q_i^2(x)$ for some $q_i(x) \in \mathbb{R}[x]$. We note that the cone of G is also called a *preordering* generated by G in [36]. If $s(x)$ is SOS, then clearly $s(x) \geq 0$ for all $x \in \mathbb{R}^n$. The sum in the definition of $\mathbf{cone}(G)$ is finite, with a total of 2^m terms, corresponding to the subsets of $\{g_1, \ldots, g_m\}$.

The notions of *ideal* and *cone* are standard in algebraic geometry, but they also have inherent convex geometry: Ideals are affine sets and cones are closed under convex combinations and non-negative scalings, i.e., they are actually cones in the convex geometry sense. Ideals and cones are used for deriving new *valid constraints*, which are logical consequences of the given constraints. For example, notice that by construction, every polynomial in $\mathbf{ideal}(\{f_1, \ldots, f_m\})$ vanishes in the solution set of the system $f_1(x) = 0, \ldots, f_m(x) = 0$ over the algebraic closure of $\mathbb{K}$. Similarly, every element of $\mathbf{cone}(\{g_1, ..., g_m\})$ is clearly non-negative on the feasible set of $g_1(x) \geq 0, \ldots, g_m(x) \geq 0$ over $\mathbb{R}$.

It is well-known that optimization algorithms are intimately tied to the development of infeasibility certificates. For example, the simplex method is closely related to Farkas' lemma. Our starting point is a generalization of this famous principle. We start with a description of two powerful infeasibility certificates for polynomial systems which generalize the classical ones for linear optimization. First, as motivation, recall from elementary linear algebra the "Fredholm alternative theorem" (e.g., see page 28 Corollary 3.1.b in [46]):

THEOREM 1.1 (Fredholm's alternative). *Given a matrix $A \in \mathbb{K}^{m \times n}$ and a vector $b \in \mathbb{K}^m$,*

$$\nexists\, x \in \mathbb{K}^n \text{ s.t. } Ax + b = 0 \;\Leftrightarrow\; \exists\, \mu \in \mathbb{K}^m \text{ s.t. } \mu^T A = 0,\ \mu^T b = 1.$$

It turns out that there are much stronger versions for general polynomials, which unfortunately do not seem to be widely known among optimizers (for more details see e.g., [5]).

THEOREM 1.2 (Hilbert's Nullstellensatz). *Let $F := \{f_1, \ldots, f_m\} \subseteq \mathbb{K}[x]$. Then,*

$$\nexists\, x \in \overline{\mathbb{K}}^n \text{ s.t. } f_1(x) = 0, ..., f_s(x) = 0 \Leftrightarrow 1 \in \mathbf{ideal}(F).$$

Note that $1 \in \mathbf{ideal}(F)$ means that there exist polynomials $\beta_1, \ldots, \beta_m \in \mathbb{K}[x]$ such that $1 = \sum_{i=1}^m \beta_i f_i$, and this polynomial identity is thus a *certificate of infeasibility*. Fredholm's alternative theorem is simply a linear version of Hilbert's Nullstellensatz where all the polynomials are linear and the β_i's are constant.

EXAMPLE 1. *Consider the following set of polynomials in* $\mathbb{R}[x_1, x_2, x_3]$*:*

$$F := \{f_1 := x_1^2 - 1, f_2 := 2x_1x_2 + x_3, f_3 := x_1 + x_2, f_4 := x_1 + x_3\}.$$

By the Nullstellensatz, the system $f_1(x) = 0, f_2(x) = 0, f_3(x) = 0, f_4(x) = 0$ *is infeasible over* $\mathbb{C}$ *if and only if there exist polynomials* $\beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{R}[x_1, x_2, x_3]$ *that satisfy the polynomial identity* $\beta_1 f_1 + \beta_2 f_2 + \beta_3 f_3 + \beta_4 f_4 = 1$*. Here, the system is infeasible, so there exist such polynomials as follows:*

$$\beta_1 = -1 - \frac{2}{3}x_2, \beta_2 = -\frac{2}{3} + \frac{1}{3}x_1, \beta_3 = -\frac{2}{3} + \frac{4}{3}x_1, \beta_4 = \frac{2}{3} - \frac{1}{3}x_1.$$

The resulting identity provides a certificate of infeasibility of the system.

Now, the two theorems above deal only with the case of equations. The inclusion of inequalities in the problem formulation poses additional algebraic challenges because we need to take into account special properties of the reals. Consider first the case of linear inequalities, which is familiar to optimizers, where linear programming duality provides the following characterization:

THEOREM 1.3 (Farkas' lemma). *Let* $A \in \mathbb{R}^{m\times n}$*,* $b \in \mathbb{R}^m$*,* $C \in \mathbb{R}^{k\times n}$*, and* $d \in \mathbb{R}^k$*.*

$$\begin{array}{c} \nexists\, x \in \mathbb{R}^n \ \ s.t.\ \ Ax + b = 0, Cx + d \geq 0 \\ \Updownarrow \\ \exists\, \lambda \in \mathbb{R}^m_+, \exists\, \mu \in \mathbb{R}^k \ \ \ s.t.\ \ \mu^T A + \lambda^T C = 0, \mu^T b + \lambda^T d = -1. \end{array}$$

Again, although not widely known in optimization, it turns out that similar certificates do exist for non-linear systems of polynomial equations and inequalities over the reals. The result essentially appears in this form in [2] and is due to Stengle [49].

THEOREM 1.4 (Positivstellensatz). *Let* $F := \{f_1, \ldots, f_m\} \subset \mathbb{R}[x]$ *and* $G := \{g_1, \ldots, g_k\} \subset \mathbb{R}[x]$*.*

$$\begin{array}{c} \nexists x \in \mathbb{R}^n \ \ s.t.\ \ f_1(x) = 0, \ldots, f_m(x) = 0,\ g_1(x) \geq 0, \ldots, g_k(x) \geq 0 \\ \Updownarrow \\ \exists\, f \in \mathbf{ideal}(F), \exists\, g \in \mathbf{cone}(G)\ \ s.t.\ \ f(x) + g(x) = -1. \end{array}$$

The theorem states that for every infeasible system of polynomial equations and inequalities, there exists a simple polynomial identity of the form $\sum_{i=1}^m \beta_i f_i + \sum_{\alpha\in\{0,1\}^n} s_\alpha g^\alpha = -1$ for some $\beta_i, s_\alpha \in \mathbb{R}[x]$ where s_α are SOS, that directly gives a certificate of infeasibility of real solutions.

EXAMPLE 2. *Consider the polynomial system* $\{f = 0,\, g \geq 0\}$, *where*

$$f := x_2 + x_1^2 + 2 = 0, \qquad g := x_1 - x_2^2 + 3 \geq 0.$$

By the Positivstellensatz, there are no solutions $(x_1, x_2) \in \mathbb{R}^2$ *if and only if there exist polynomials* $\beta, s_1, s_2 \in \mathbb{R}[x_1, x_2]$ *that satisfy*

$$\beta \cdot f + s_1 + s_2 \cdot g = -1 \quad \textit{where } s_1 \textit{ and } s_2 \textit{ are SOS.}$$

Here, the system is infeasible, so there exist such polynomials as follows:

$$s_1 = \tfrac{1}{3} + 2\left(x_2 + \tfrac{3}{2}\right)^2 + 6\left(x_1 - \tfrac{1}{6}\right)^2, s_2 = 2 \textit{ and } \beta = -6.$$

The resulting identity provides a certificate of infeasibility of the system.

Of course, we are very concerned with the effective practical computation of the infeasibility certificates. For the sake of computation and complexity, we must worry about the growth of degrees and thus the growth in the encoding size of the infeasibility certificates. Here, we define the degree of a Nullstellensatz certificate $\sum_{i=1}^m \beta_i f_i = 1$ as $\max_i\{\deg(\beta_i f_i)\}$ and the degree of a Positivstellensatz certificate $\sum_{i=1}^m \beta_i f_i + \sum_{\alpha\in\{0,1\}^n} s_\alpha g^\alpha = -1$ as the larger of $\max_i\{\deg(\beta_i f_i)\}$ and $\max_\alpha\{\deg(s_\alpha g^\alpha)\}$. On the negative side, the degrees of the certificates are expected to grow at least linearly leading to exponential growth in the encoding size of the certificates simply because the NP-hardness of the original combinatorial questions; see e.g. [9]. At the same time, tight exponential upper bounds on the degrees have been derived (see e.g. [22], [16] and references therein). Nevertheless, for many problems of practical interest, it is often the case that it is possible to prove infeasibility using low-degree certificates (see [8, 7]). Even more important is the fact that for a fixed degree of the certificates, the calculations are polynomial time (see Lemma 2.1 and [41]) and can be reduced to either linear algebra or semidefinite programming. We summarize the strong analogies between the case of linear equations and inequalities with high-degree polynomial systems in the following table:

TABLE 1
Infeasibility certificates and their associated computational techniques.

Degree\Field	Arbitrary	Real
Linear	*Fredholm Alternative* Linear Algebra	*Farkas' Lemma* Linear Programming
Polynomial	*Nullstellensatz* Bounded degree Linear Algebra	*Positivstellensatz* Bounded degree SDP

It is important to remark that just as in the classical case of linear programming, the problem of computation of certificates has very natural primal-dual formulations, with the corresponding primal and dual variables playing distinct, but well-defined roles. For example, in the case of

Fredholm's alternative, the primal variables are the variables $x_1, \ldots, x_n$ while there is a dual variable for each equation. For Nullstellensatz and Positivstellensatz there is a similar duality, based on linear duality and semidefinite programming duality, respectively. In what follows, we use the most intuitive or convenient set-up and we leave to the reader the exercise of transferring the results to the corresponding dual version.

The remainder of the paper is divided in two main sections: Section 2 is a study of the Hilbert Nullstellensatz, for general fields, used in the solution of systems of equations. In Section 3, we survey the use of the Positivstellensatz in the context of solving systems of equations and inequalities over the reals. Both sections contain combinatorial applications that show why these techniques can be of interest in this setting. The focus of the combinatorial results is understanding those situations when a constant degree certificate is enough to show infeasibility. These are situations when hard combinatorial problems have polynomial time algorithms and as such provide structural insight. Finally, in Section 4, we describe a methodology, common to both approaches, to recover feasible solutions of the original combinatorial problem from the outcome of these relaxations. In addition, we have included an Appendix A that contains proofs of some the results used in the main body of the paper that are either hard to find or whose original proof, available elsewhere, is not written in the language of this survey.

To conclude the introduction we include some more notation and terminology. The *variety* of F over $\mathbb{K}$, written $\mathcal{V}_{\mathbb{K}}(F)$, is the set of common zeros of polynomials in F in $\mathbb{K}^n$, that is, $\mathcal{V}_{\mathbb{K}}(F) := \{v \in \mathbb{K}^n : f(v) = 0 \; \forall f \in I\}$. Also, $\mathcal{V}_{\overline{\mathbb{K}}}(F)$, the variety of F over $\overline{\mathbb{K}}$, is the set of common zeros of F in $\overline{\mathbb{K}}^n$. Note that in combinatorial problems, the *variety* of a polynomial system typically has finitely many solutions (e.g., colorings, cuts, stable sets, etc.). For an ideal $I \subseteq \mathbb{K}[x]$, when $\mathcal{V}_{\overline{\mathbb{K}}}(I)$ is finite, the ideal is called *zero-dimensional* (this is the case for all of the applications considered here). We say that a system of polynomial equations is a *combinatorial system* when its variety encodes a combinatorial problem (e.g., zeros represent stable sets, colorings, matchings, etc.) and it is zero-dimensional.

An ideal $I \subseteq \mathbb{K}[x]$ is *radical* if $f^k \in I$ for some positive integer k implies $f \in I$. We denote by $\sqrt{I}$ the ideal of all polynomials $f \in \mathbb{K}[x]$ such that $f^k \in I$ for some positive integer k. The ideal $\sqrt{I}$ is necessarily radical and it is called the radical ideal of I. Note that I is radical if and only if $I = \sqrt{I}$. Given a vector space W over a field $\mathbb{K}$, we write $\dim(W)$ for the dimension of W. Given vector spaces $U \subseteq W$, we write W/U as the vector space quotient. Recall that $\dim(W/U) = \dim(W) - \dim(U)$. Given a set $F \subset \mathbb{K}[x]$, $\mathbf{span}(F)$ denotes the vector space generated by F over the field $\mathbb{K}$. Please note the distinction between the vector space $\mathbf{span}(F)$ and the ideal $\mathbf{ideal}(F)$.

2. Solving combinatorial systems of equations. In this section, we wish to solve a given system of polynomial equations $f_1(x) = 0, f_2(x) = 0, \ldots, f_m(x) = 0$ where $f_1, \ldots, f_m \in \mathbb{K}[x]$. The systems we consider have finitely many solutions, each corresponding to a combinatorial object. To simplify our arguments we also assume that $\mathbb{K}$ is algebraically closed, i.e., $\mathbb{K} = \overline{\mathbb{K}}$. We abbreviate this system as $F(x) = 0$ where $F := \{f_1, \ldots, f_m\} \subset \mathbb{K}[x]$. Here, by solving a system, we mean first determining if $F(x) = 0$ is feasible over $\mathbb{K}$, and furthermore finding a solution (or all solutions) of $F(x) = 0$ if feasible. The literature on polynomial solving is very extensive and it continues to be an area of active research (see [50, 6, 10] for an overview and background).

Here we choose to focus on techniques that fit well with traditional optimization methods. The main idea is that solving a polynomial system of equations can be reduced to solving a sequence of linear algebra problems. The foundations of this technique can be traced back to ([37, 20, 21, 38]). The specific approach we take to present this technique is closest to that of Mourrain in [37]. Variants of this technique have been applied to stable sets [9, 35], vertex coloring [8, 35], satisfiability (see e.g., [3]) and cryptography (see for example [4]). This technique is also strongly related to Border basis and Gröbner basis techniques, which can also be viewed in terms of linear algebra computations (see e.g., [20, 21, 38, 50]).

The linear algebra systems of equations have primal and dual representations in the sense of Fredholm's lemma. Specifically, in this survey, the primal approach solves a linear system to find constant multipliers $\mu \in \mathbb{K}^m$ such that $1 = \sum_{i=1}^m \mu_i f_i$ providing a certificate of (non-linear) infeasibility. Then, the dual approach aims to find a vector λ with entries in $\mathbb{K}$ indexed by monomials such that $\sum_\alpha \lambda_{x^\alpha} f_{i,\alpha} = 0$ for all $i = 1, \ldots, m$ and $\lambda_1 = 1$ where $f_i = \sum_\alpha f_{i,\alpha} x^\alpha$ for all i. As we see in Section 2.2, the dual approach amounts to constructing linear relaxations of the set of feasible solutions. In Sections 2.1 and 2.2, we present the primal and dual approaches respectively.

2.1. Linear algebra certificates. Consider the following corollary of Hilbert's Nullstellensatz: If there exist *constants* $\mu \in \mathbb{K}^m$ such that $\sum_{i=1}^m \mu_i f_i = 1$, then the polynomial system $F(x) = 0$ must be infeasible. In other words, if the system $F(x) = 0$ is infeasible, then $1 \in \mathbf{span}(F)$. The crucial point here is that determining whether there exists a $\mu \in \mathbb{K}^m$ such that $\sum_{i=1}^m \mu_i f_i = 1$ is a linear algebra problem over $\mathbb{K}$. The equation $\sum_{i=1}^m \mu_i f_i = 1$ is called a *certificate of infeasibility* of the polynomial system.

EXAMPLE 3. *Consider again the following set of polynomials from Example 1:*

$$F := \{f_1 := x_1^2 - 1, f_2 := 2x_1x_2 + x_3, f_3 := x_1 + x_2, f_4 := x_1 + x_3\}.$$

We can abbreviate the infeasible polynomial system of equations $f_1(x) = 0, f_2(x) = 0, f_3(x) = 0, f_4(x) = 0$ *as* $F(x) = 0$. *We can prove that the system* $F(x) = 0$ *is infeasible if we can find* $\mu \in \mathbb{R}^4$ *satisfying the following:*

$$\begin{aligned} &\mu_1 f_1 + \mu_2 f_2 + \mu_3 f_3 + \mu_4 f_4 = 1 \\ \Leftrightarrow\ &\mu_1(x_1^2 - 1) + \mu_2(2x_1x_2 + x_3) + \mu_3(x_1 + x_2) + \mu_4(x_1 + x_3) = 1 \\ \Leftrightarrow\ &\mu_1 x_1^2 + 2\mu_2 x_1 x_2 + (\mu_2 + \mu_4)x_3 + \mu_3 x_2 + (\mu_3 + \mu_4)x_1 - \mu_1 = 1. \end{aligned}$$

Then, equating coefficients on the left and right hand sides of the equation above gives the following linear system of equations:

$$\begin{aligned} -\mu_1 = 1 \quad (1), \qquad \mu_3 + \mu_4 = 0 \quad (x_1), \qquad \mu_3 = 0 \quad (x_2), \\ \mu_3 + \mu_4 = 0 \quad (x_3), \qquad 2\mu_2 = 0 \quad (x_1x_2), \qquad \mu_1 = 0 \quad (x_1^2). \end{aligned}$$

We abbreviate this system as $\mu^T F = 1$. *Even though* $F(x) = 0$ *is infeasible, the linear system* $\mu^T F = 1$ *is infeasible, and so, we have not found a certificate of infeasibility of* $F(x) = 0$.

More formally, let $f_i = \sum_{\alpha \in \mathbb{N}^n} f_{i,\alpha} x^\alpha$ where only finitely many $f_{i,\alpha}$ are non-zero $i = 1, ..., m$. Then, $\sum_{i=1}^m \mu_i f_i = 1$ if and only if $\sum_{i=1}^m \mu_i f_{i,0} = 1$ and $\sum_{i=1}^m \mu_i f_{i,\alpha} = 0$ for all $\alpha \in \mathbb{N}^n$ where $\alpha \neq 0$. Note that there is one linear equation per monomial appearing in F. We abbreviate this linear system as $\mu^T F = 1$ where we consider F as a matrix whose rows are the coefficient vectors of its polynomials and we consider the constant polynomial 1 as the vector of its coefficients (i.e., a unit vector). The columns of F are indexed by monomials with non-zero coefficients. We remark that in the special case where $F(x) = 0$ is a linear system of equations, then Fredholm's alternative says that $F(x) = 0$ is infeasible if and only if $\mu^T F = 1$ is feasible.

REMARK 2.1. Crucially for computation, when we solve the linear system $\mu^T F = 1$, we can do so over the smallest subfield of $\mathbb{K}$ containing the coefficients of the polynomials in F, which is particularly useful if such a subfield is a finite field.

In general, even if $F(x) = 0$ is infeasible, $\mu^T F = 1$ may not be feasible as in the above example. In order to prove infeasibility, we must add polynomials from $\mathbf{ideal}(F)$ to F and try again to find a μ such that $\mu^T F = 1$. Hilbert's Nullstellensatz guarantees that, if $F(x) = 0$ is infeasible, there exists a finite set of polynomials from $\mathbf{ideal}(F)$ that we can add to F so that the linear system $\mu^T F = 1$ is feasible.

More precisely, it is enough to add polynomials of the form $x^\alpha f$ for x^α a monomial and some polynomial $f \in F$. Why is this? If $F(x) = 0$ is infeasible, then Hilbert's Nullstellensatz says $\sum_{i=1}^m \beta_i f_i = 1$ for some $\beta_1, \ldots, \beta_m \in \mathbb{K}[x]$. Let $d = \max_i\{\deg(\beta_i)\}$. Then, if we add to F all polynomials of the form $x^\alpha f$ where $f \in F$ and $\deg(x^\alpha) \leq d$. Then, the $\mathbb{K}$-linear span of F, that is $\mathbf{span}(F)$, contains $\beta_i f_i$ for all i, and thus,

$1 \in \mathbf{span}(F)$ or equivalently $\mu^T F' = 1$ is feasible (as a linear algebra problem) where F' denotes the larger polynomial system.

EXAMPLE 4. *Consider again the polynomial system $F(x) = 0$ from Example 3. Here, $\mu^T F = 1$ is feasible, so we must thus add redundant polynomial equations to the system $F(x) = 0$. In particular, we add the following redundant polynomial equations: $x_2 f_1(x) = 0$, $x_1 f_2(x) = 0$, $x_1 f_3(x) = 0$, and $x_1 f_4(x) = 0$. Let $F' := \{f_1, f_2, f_3, f_4, x_2 f_1, x_1 f_2, x_1 f_3, x_1 f_4\}$.*

Then, the system $\mu^T F' = 1$ is now as follows:

$$\begin{array}{llll} -\mu_1 = 1 \quad (1), & \mu_3 + \mu_4 = 0 \quad (x_1), & \mu_3 - \mu_5 = 0 \quad (x_2), \\ \mu_2 + \mu_4 = 0 \quad (x_3), & 2\mu_2 + \mu_7 = 0 \quad (x_1 x_2), & \mu_1 + \mu_7 + \mu_8 = 0 \quad (x_1^2), \\ \mu_6 + \mu_8 = 0 \quad (x_1 x_3), & \mu_5 + 2\mu_6 = 0 \quad (x_1^2 x_2). & \end{array}$$

This system is feasible proving that $F(x) = 0$ is infeasible. The solution is $\mu = (-1, -\frac{2}{3}, -\frac{2}{3}, \frac{2}{3}, -\frac{2}{3}, -\frac{1}{3}, \frac{4}{3}, -\frac{1}{3})$, which gives the following certificate of infeasibility as given in Example 1:

$$-f_1 - \frac{2}{3} f_2 - \frac{2}{3} f_3 + \frac{2}{3} f_4 - \frac{2}{3} x_2 f_1 + \frac{1}{3} x_1 f_2 + \frac{4}{3} x_1 f_3 - \frac{1}{3} x_1 f_4 = 1.$$

Next, we present the dual approach to the one in this section.

2.2. Linear algebra relaxations. In optimization, it is quite common to "linearize" non-linear polynomial systems of equations by replacing all monomials in the system with new variables giving a system of linear constraints. Specifically, we can construct a linear algebra relaxation of the solutions of $F(x) = 0$ by replacing every monomial x^α in a polynomial equation in $F(x) = 0$ with a new variable λ_{x^α} thereby giving a system of linear equations in the new λ variables, one variable for each monomial appearing in F. Readers familiar with relaxation procedures such as Sherali-Adams and Lovász-Schrijver (see [27] and references therein) will see a lot of similarities, but here we deal only with *equality* constraints.

EXAMPLE 5. *Consider the following feasible system in $\mathbb{C}[x_1, x_2, x_3]$:*

$$f_1(x) = x_1^2 - 1 = 0, \quad f_2(x) = 2x_1 x_2 + x_3 = 0, \quad f_3(x) = x_1 + x_2 = 0.$$

This system has two solutions $(x_1, x_2, x_3) = (1, -1, 2)$ and $(x_1, x_2, x_3) = (-1, 1, 2)$. Let $F = \{f_1, f_2, f_3\}$. So, we abbreviate the above system as $F(x) = 0$. We can replace the monomials $1, x_1, x_2, x_3, x_1^2, x_1 x_2$ with the variables $\lambda_1, \lambda_{x_1}, \lambda_{x_2}, \lambda_{x_3}, \lambda_{x_1^2}, \lambda_{x_1 x_2}$ respectively. The system $F(x) = 0$ thus gives rise to the following set of linear *equations:*

$$\lambda_{x_1^2} - \lambda_1 = 0, \qquad 2\lambda_{x_1 x_2} + \lambda_{x_3} = 0, \qquad \lambda_{x_1} + \lambda_{x_2} = 0. \tag{2.1}$$

We abbreviate the above system as $F * \lambda = 0$.

Solutions of $F(x) = 0$ give solutions of $F * \lambda = 0$: If x is a solution of $F(x) = 0$ above, then setting $\lambda_1 = 1, \lambda_{x_1} = x_1, \lambda_{x_2} = x_2, \lambda_{x_3} = x_3, \lambda_{x_1^2} =$

$x_1^2, \lambda_{x_1x_2} = x_1x_2$ *gives a solution of* $F*\lambda = 0$. *So, taking* $x = (1, -1, 2)$, *we set* $\lambda_1 = 1$, $\lambda_{x_1} = 1$, $\lambda_{x_2} = -1$, $\lambda_{x_3} = 2$, $\lambda_{x_1^2} = 1$, *and* $\lambda_{x_1x_2} = -1$. *Then, we have* $F * \lambda = 0$. *Thus, the solutions of* $F * \lambda = 0$ *gives a vector space effectively containing all of the solutions of* $F(x) = 0$. *Hence,* $F * \lambda = 0$ *gives a linear relaxation of* $F(x) = 0$.

There are solutions of $F * \lambda = 0$ *that do not correspond to solutions of* $F(x) = 0$ *because the linear system* $F * \lambda = 0$ *does not take into account the non-linear constraints that* $\lambda_1 = 1$, $\lambda_{x_1^2} = \lambda_{x_1}^2$ *and* $\lambda_{x_1x_2} = \lambda_{x_1}\lambda_{x_2}$; *For example,* $\lambda_1 = 1$, $\lambda_{x_1} = 2$, $\lambda_{x_2} = -2$, $\lambda_{x_3} = -2$, $\lambda_{x_1^2} = 1$ *and* $\lambda_{x_1x_2} = 1$ *is a solution of* $F*\lambda = 0$, *but* $x_1 = \lambda_{x_1} = 2$, $x_2 = \lambda_{x_2} = -2$, *and* $x_3 = \lambda_{x_3} = -2$ *is not a solution of* $F(x) = 0$.

We now formalize the above example construction of a linear system. We can consider the polynomial ring $\mathbb{K}[x]$ as an infinite dimensional vector space over $\mathbb{K}$ where the set of all monomials x^α forms a vector space basis of $\mathbb{K}[x]$. In other words, a polynomial $f = \sum_{\alpha\in\mathbb{N}^n} f_\alpha x^\alpha$ can be represented as an infinite sequence $(f_\alpha)_{\alpha\in\mathbb{N}^n}$ where only finitely many f_α are non-zero. We define $\mathbb{K}[[x_1, \ldots, x_n]] = \mathbb{K}[[x]]$ as the ring of formal power series in the variables $x_1, \ldots, x_n$ with coefficients in $\mathbb{K}$. So, the power series $\lambda = \sum_{\alpha\in\mathbb{N}^n} \lambda_\alpha x^\alpha$ can be represented as an infinite sequence $(\lambda_\alpha)_{\alpha\in\mathbb{N}^n}$. Note that we do not require that only finitely many λ_α are non-zero. We define the bilinear form $* : \mathbb{K}[x] \times \mathbb{K}[[x]] \to \mathbb{K}$ as follows: given $f = \sum_{\alpha\in\mathbb{N}^n} f_\alpha x^\alpha \in \mathbb{K}[x]$ and $\lambda = \sum_{\alpha\in\mathbb{N}^n} \lambda_\alpha x^\alpha \in \mathbb{K}[[x]]$, we define $f * \lambda = \sum_{\alpha\in\mathbb{N}^n} f_\alpha\lambda_\alpha$, which is always finite since only finitely many f_α are non-zero. Thus, we define a linear relaxation of $\{x \in \mathbb{K}^n : F(x) = 0\}$, written as $\{\lambda \in \mathbb{K}[[x]] : F * \lambda = 0\}$, as the set of linear equations $f * \lambda = 0$ for all $f \in F$. We denote the set of solutions of the linear system $F * \lambda = 0$ as $F^\circ := \{\lambda \in \mathbb{K}[[x]] : F * \lambda = 0\}$, called the *annihilator* of F, which is a vector subspace of $\mathbb{K}[[x]]$. See Appendix A for further details.

Note that, for any polynomial $f \in \mathbb{K}[x]$ and any point $v \in \mathbb{K}^n$, we have $f(v) = f * \lambda(v)$ where $\lambda(v) = (v^\alpha)_{\alpha\in\mathbb{N}^n}$. Thus, for any $v \in \mathbb{K}^n$, $F(v) = 0$ if and only if $F * \lambda(v) = 0$. So, the system $F * \lambda = 0$ can be considered as a linear relaxation of the system $F(x) = 0$. As mentioned in the above example, there are solutions of $F*\lambda = 0$ that do not correspond to solutions of $F(x) = 0$ because the linear system $F * \lambda = 0$ does not take into account the relationships between the λ variables. Specifically, if λ corresponded to a solution of $F(x) = 0$, then we must have $\lambda_{x^\alpha} = \lambda_{x^\beta}\lambda_{x^\gamma}$ for all monomials $x^\alpha, x^\beta, x^\gamma$ where $x^\alpha = x^\beta x^\gamma$. If we added these non-linear constraints to the linear constraints $F*\lambda = 0$, then we would essentially have the original polynomial system $F(x) = 0$.

The system $F*\lambda = 0$ is always feasible, but the constraint $\lambda_1 = 1$ also holds for any λ that corresponds to a solution x of $F(x) = 0$. Thus, if the inhomogeneous linear system $\{F * \lambda = 0, \lambda_1 = 1\}$ is infeasible, then so is the system of polynomials $F(x) = 0$.

REMARK 2.2. Crucially for computation again, when we solve the linear system $\{F * \lambda = 0, \lambda_1 = 1\}$, we can do so over the smallest subfield of $\mathbb{K}$ containing the coefficients of the polynomials in F.

REMARK 2.3. Importantly, the linear system $\{F * \lambda = 0, \lambda_1 = 1\}$ is dual to the linear system $\mu^T F = 1$ from the previous section by Fredholm's alternative meaning that $\{F * \lambda = 0, \lambda_1 = 1\}$ is infeasible if and only if $\mu^T F = 1$ is feasible.

There is a fundamental observation we wish to make here: *adding redundant polynomial equations can lead to a tighter relaxation.*

EXAMPLE 6. *(Cont.) Add* $x_1 f_3(x) = x_1^2 + x_1 x_2 = 0$ *to the system* $F(x) = 0$ *giving the system* $F'(x) = 0$ *where* $F' := \{f_1, f_2, f_3, x_1 f_3\}$. *The system* $F'(x) = 0$ *has the same solutions as* $F(x) = 0$. *The polynomial equation* $x_1 f_3(x) = 0$ *gives rise to a new linear equation* $\lambda_{x_1^2} + \lambda_{x_1 x_2} = 0$ *giving the following linear system* $F' * \lambda = 0$*:*

$$\lambda_{x_1^2} - \lambda_1 = 0,\ 2\lambda_{x_1 x_2} + \lambda_{x_3} = 0,\ \lambda_{x_1} + \lambda_{x_2} = 0,\ \lambda_{x_1^2} + \lambda_{x_1 x_2} = 0. \quad (2.2)$$

The dimension of the solution space of the original system $F * \lambda = 0$ *is three if we ignore all* λ *variables that do not appear in the linear system, or in other words, if we project the solution space onto the* λ *variables appearing in the system. However, the dimension of the projected solution space of* $F' * \lambda = 0$ *is two; so,* $F' * \lambda = 0$ *is a tighter relaxation of* $F(x) = 0$.

Extending this idea, consider the ideal $I = \mathbf{ideal}(F)$, which is the set of all redundant polynomials given as a polynomial combination of polynomials in F, then I° becomes a finite dimensional vector space where $\dim(I^\circ)$ is precisely the number of solutions of $F(x) = 0$ over $\mathbb{K}$, including multiplicities, assuming that there are finitely many solutions. Note that by linear algebra, I° is isomorphic to the vector space quotient $\mathbb{K}[x]/I$ (see e.g., [50]). Furthermore, if I is radical, then $\dim(I^\circ) = \dim(\mathbb{K}[x]/I)$ is precisely the number of solutions of $F(x) = 0$. So, there is a direct relationship between the number of solutions of a polynomial system and the dimension of the solution space of its linear relaxation (see e.g., [6] for a proof).

THEOREM 2.1. *Let* $I \subseteq \mathbb{K}[x]$ *be a zero-dimensional ideal. Then,* $\dim(I^\circ)$ *is finite and* $\dim(I^\circ)$ *is the number of solutions of polynomial system* $I(x) = 0$ *over* $\mathbb{K}$ *including multiplicities, so* $|\mathcal{V}_{\mathbb{K}}(I)| \leq \dim(I^\circ)$ *with equality when* I *is radical.*

So, if we can compute $\dim(I^\circ)$, then we can determine the feasibility of $I(x) = 0$ over $\mathbb{K}$. Unfortunately, we cannot compute $\dim(I^\circ)$ directly. Instead, under some conditions (see Theorem 2.2), we can compute $\dim(I^\circ)$ by computing the dimension of F° when projected onto the λ_{x^α} variables where $\deg(x^\alpha) \leq \deg(F)$.

2.3. Nullstellensatz Linear Algebra Algorithm (NulLA). We now present an algorithm for determining whether a polynomial system of equations is infeasible using linear relaxations. Let $F \subseteq \mathbb{K}[x]$ and again let

$F(x) = 0$ be the polynomial system $f(x) = 0$ for all $f \in F$. We wish to determine whether $F(x) = 0$ has a solution over $\mathbb{K}$.

The idea behind NulLA [8] is straightforward: we check whether the linear system $\{F * \lambda = 0, \lambda_1 = 1\}$ is infeasible or equivalently whether $\mu^T F = 1$ is feasible (i.e., $1 \in \mathbf{span}(F)$) using linear algebra over $\mathbb{K}$ and if not then we add polynomials from $\mathbf{ideal}(F)$ to F and try again. We add polynomials in the following systematic way: for each polynomial $f \in F$ and for each variable x_i, we add $x_i f$ to F. So, the NulLA algorithm is as follows: if $\{F * \lambda = 0, \lambda_1 = 1\}$ is infeasible, then $F(x) = 0$ is infeasible and stop, otherwise for every variable x_i and every $f \in F$ add $x_i f$ to F and repeat.

In the following, we assume without loss of generality that F is closed under $\mathbb{K}$-linear combinations, that is $F = \mathbf{span}(F)$, and thus, F is a vector space over $\mathbb{K}$. Note that taking the closure of F under $\mathbb{K}$-linear combinations does not change the set of solutions of $F(x) = 0$ and does not change the set of solutions of $F * \lambda = 0$. In practice, we must choose a vector space basis of F for computation, but the point we wish to make is that the choice of basis is irrelevant. Moreover, we find that it is more natural to work with vector spaces and that it leads to a more concise exposition. Recall from above that $\{F * \lambda = 0, \lambda_1 = 1\}$ is infeasible if and only if $1 \in \mathbf{span}(F)$, which when F is a vector space, simplifies to $1 \in F$ since $\mathbf{span}(F) = F$.

For a vector space $F \subset \mathbb{K}[x]$, we define $F^+ := F + \sum_{i=1}^n x_i F$ where $x_i F := \{x_i f : f \in F\}$. Note that F^+ is also a vector subspace of $\mathbb{K}[x]$. Then, F^+ is precisely the linear span of F and $x_i F$ for all $i = 1, \ldots, n$. So, the NulLA algorithm for vector spaces is as follows (see Algorithm 1): if $1 \in F$, then $F(x) = 0$ is infeasible and stop, otherwise set $F \leftarrow F^+$ and repeat. There is an upper bound on the number of times we need to repeat the above step given by the *Nullstellensatz bound* of the system $F(x) = 0$ (see [22]): if $F(x) = 0$ has a Nullstellensatz bound D, then if $F(x) = 0$ is infeasible, there must exist a Nullstellensatz certificate of infeasibility $\sum_i \beta_i f_i = 1$ where $\deg(\beta_i) \leq D$, that is, the degree of the certificate is at most $\deg(F) + D$. After d iterations of NulLA, the set F contains all linear combinations of polynomials of the form $x^\alpha f$ where $|\alpha| \leq d$ and where f was one of the initial polynomials in F, and so, if the system is infeasible, then NulLA will find a certificate of infeasibility in at most the Nullstellensatz bound number of iterations.

While theoretically the Nullstellensatz bound limits the number of iterations, this bound is in general too large to be practically useful (see [8]). Hence, in practice, NulLA is most useful for proving infeasibility (see Section 2.4).

Next, we discuss improving NulLA by adding redundant polynomials to F in such a way so that $\deg(F)$ does not grow unnecessarily. We call this improved algorithm the Fixed-Point Nullstellensatz Linear Algebra (FPNulLA) algorithm. Some variations of FPNulLA appeared, e.g., in

Algorithm 1 NulLA Algorithm [8]

Input: A finite dimensional vector space $F \subseteq \mathbb{K}[x]$ and a Nullstellensatz bound D.

Output: FEASIBLE, if $F(x) = 0$ is feasible over $\mathbb{K}$, else INFEASIBLE.

1: **for** $k = 0, 1, 2, \ldots, D$ **do**
2: If $1 \in F$, then **return** INFEASIBLE.
3: $F \leftarrow F^+$.
4: **end for**
5: **Return** FEASIBLE.

[37, 44, 25]. The basic idea behind the FPNulLA algorithm is that, if $1 \notin F$, then instead of replacing F with F^+ and thereby increasing $\deg(F)$, we check to see whether there are any new polynomials in F^+ with degree at most $\deg(F)$ that were not in F and add them to F, and then check again whether $1 \notin F$. More formally, if $1 \notin F$, then we replace F with $F^+ \cap \mathbb{K}[x]_d$ where $\mathbb{K}[x]_d$ is the set of all polynomials with degree at most $d = \deg(F)$. We keep replacing F with $F^+ \cap \mathbb{K}[x]_d$ until either $1 \in F$ or we reach a *fixed point*, $F = F^+ \cap \mathbb{K}[x]_d$. This process must terminate.

Note that if we find that $1 \in F$ at some stage of FPNulLA this implies that there exists an infeasibility certificate of the form $1 = \sum_{i=1}^{s} \beta_i f_i$ where $\beta_1, ..., \beta_s \in \mathbb{K}[x]$ and the polynomials $f_1, ..., f_s \in \mathbb{K}[x]$ are a vector space basis of the original set F.

Moreover, we can also improve NulLA by proving that the system $F(x) = 0$ is feasible well before reaching the Nullstellensatz bound as follows. When $1 \notin F$ and $F = F^+ \cap \mathbb{K}[x]_d$, then we could set $F \leftarrow F^+$ and $d \leftarrow d + 1$ and repeat the above process. However, when we reach the fixed point $F = F^+ \cap \mathbb{K}[x]_d$, we can use the following theorem to determine if the system is feasible and if so how many solutions it has. First, we introduce some notation. Let $\pi_d : \mathbb{K}[[x]] \to \mathbb{K}[[x]]_d$ be the truncation or projection of a power series onto a polynomial of degree at most d with coefficients in $\mathbb{K}$. Below, we abbreviate $\dim(\pi_d(F^\circ))$ as $\dim_d(F^\circ)$ and similarly $\dim(\pi_{d-1}(F^\circ))$ as $\dim_{d-1}(F^\circ)$.

THEOREM 2.2. *Let $F \subset \mathbb{K}[x]$ be a finite dimensional vector space and let $d = \deg(F)$. If $F = F^+ \cap \mathbb{K}[x]_d$ and $\dim_d(F^\circ) = \dim_{d-1}(F^\circ)$, then $\dim(I^\circ) = \dim_d(F^\circ)$ where $I = \mathbf{ideal}(F)$.*

See the Appendix for a proof of Theorem 2.2 or see original proof in [37]. There are many equivalent forms of the above theorem that appear in the literature (see e.g., [37, 44, 25]).

Recall from Theorem 2.1, that there are $\dim(I^\circ)$ solutions of $F(x) = 0$ over $\mathbb{K}$ including multiplicities where $I = \mathbf{ideal}(F)$ and exactly $\dim(I^\circ)$ solutions when I is radical. Checking the fixed point condition in FPNulLA whether $F \neq F^+ \cap \mathbb{K}[x]_d$ is equivalent to checking whether $\dim(F) \neq \dim(F^+ \cap \mathbb{K}[x]_d)$. Furthermore, to check the condition that $\dim_d(F^\circ) = \dim_{d-1}(F^\circ)$, we need to compute $\dim(F^+ \cap \mathbb{K}[x]_d)$ and $\dim(F \cap \mathbb{K}[x]_{d-1})$

since $\dim(\mathbb{K}[x]_d/F) = \dim_d(F^\circ)$ and also $\dim(\mathbb{K}[x]_{d-1}/(F \cap \mathbb{K}[x]_{d-1})) = \dim_{d-1}(F^\circ)$ (see Lemma A.1). So, in order to check the condition in FPNulLA, we need to compute $\dim(F)$, $\dim(F^+ \cap \mathbb{K}[x]_d)$ and $\dim(F \cap \mathbb{K}[x]_{d-1})$, which amounts to matrix rank calculations over the field of coefficients of a given basis of F.

We can now present the FPNulLA algorithm. See the Appendix or [37, 7] for details. The FPNulLA algorithm always terminates for zero-dimensional polynomials systems, which in particular includes combinatorial systems (see Lemma A.2).

Algorithm 2 FPNulLA Algorithm

Input: A vector space $F \subset \mathbb{K}[x]$.
Output: The number of solutions of $F(x) = 0$ over $\mathbb{K}$ up to multiplicities.

Let $d \leftarrow \deg(F)$.
loop
  if $1 \in F$ **then** Return 0 (infeasible).
  while $F \neq F^+ \cap \mathbb{K}[x]_d$ **do**
    Set $F \leftarrow F^+ \cap \mathbb{K}[x]_d$.
    if $1 \in F$ **then** Return 0 (infeasible).
  end while
  if $\dim_d(F^\circ) = \dim_{d-1}(F^\circ)$ **then** Return $\dim_d(F^\circ)$ (feasible).
  $F \leftarrow F^+$.
  $d \leftarrow d + 1$.
end loop

EXAMPLE 7. *Consider again the system below with polynomials in* $\mathbb{K}[x, y]$ *with* $\mathbb{K} = \overline{\mathbb{F}}_2$. *This system has two solutions.*

$$1 + x + x^2 = 0, \quad 1 + y + y^2 = 0, \quad x^2 + xy + y^2 = 0.$$

Let $F := \mathbf{span}(\{1 + x + x^2, 1 + y + y^2, x^2 + xy + y^2\})$. *Then,* $1 \notin F$ *and* $\deg(F) = 2$. *Now,*

$$\begin{aligned} F^+ &= F + xF + yF \\ &= F + \mathbf{span}(\{x + x^2 + x^3, x + xy + xy^2, x^3 + x^2y + xy^2\}) \\ &\quad + \mathbf{span}(\{y + xy + x^2y, y + y^2 + y^3, x^2y + xy^2 + y^3\}). \end{aligned}$$

Then, $F^+ \cap \mathbb{K}[x]_2 = \mathbf{span}(\{1 + x + x^2, 1 + y + y^2, x^2 + xy + y^2, 1 + x + y\})$. *So,* $F \neq F^+ \cap \mathbb{K}[x]_2$. *Next, let* $F := F^+ \cap \mathbb{K}[x]_2$. *One can check that now* $F = F^+ \cap \mathbb{K}[x]_2$. *Moreover,*

$$\dim_2(F^\circ) = \dim(\mathbb{K}[x]_2/F) = \dim(\mathbb{K}[x]_2) - \dim(F) = 2$$

and

$$\dim_1(F^\circ) = \dim(\mathbb{K}[x]_1/(F \cap \mathbb{K}[x]_1) = \dim(\mathbb{K}[x]_1) - \dim(F \cap \mathbb{K}[x]_1) = 2.$$

Therefore, $\dim_2(F^\circ) = \dim_1(F^\circ)$ *proving that* $F(x) = 0$ *is feasible with at most 2 solutions.*

We refer to the number of iterations (the *for* loop) that NulLA takes to solve a given system of equations as the *NulLA rank* of the system. Note that if an infeasible system $F(x) = 0$ has a NulLA rank of r, then it has a Nullstellensatz certificate of infeasibility of degree $r + \deg(F)$. Similarly to the NulLA rank, we refer to the number of outer iterations (the outer *loop*) that FPNulLA takes to the system as the *FPNulLA rank* of the system. We can consider the NulLA rank and the FPNulLA rank as measures of the "hardness" of proving infeasibility of the system. In section 2.4, we present experimental evidence that the NulLA rank and even more so the FPNulLA are "good" measures of the "hardness" of proving infeasibility of a system (see also [3] for theoretical evidence for FPNulLA).

For a given class of polynomial system of equations, it is interesting to understand the growth of the NulLA rank or FPNulLA rank because of the implications for the complexity of solving the given class of problems. Furthermore, for some fixed rank, it is also interesting to characterize which systems can be solved at that rank since this class of systems are polynomial time solvable by Lemma 2.1 below (see proof in Appendix and a proof for NulLA in [35]). For example, in Section 2.5, we characterize systems encoding 3-colorability with NulLA rank one.

LEMMA 2.1. *Let* $L \in \mathbb{N}$ *be fixed. Let* $F = \mathbf{span}(\{f_1, f_2, \ldots, f_m\}) \subseteq \mathbb{K}[x]$ *be a finite dimensional vector space of* $\mathbb{K}[x]$*. Polynomials are assumed to be encoded as vectors of coefficients indexed by all monomials of degree at most* $\deg(F)$*.*

1. *The first L iterations (the* for *loop) of the NulLA algorithm can be computed in polynomial time in n and the input size of the defining basis of F.*
2. *When* $\mathbb{K}$ *is a finite field, the first L iterations (the outer* loop*) of the FPNulLA algorithm can be computed in polynomial time in n,* $\log_2(|\mathbb{K}|)$ *and the input size of the defining basis of F.*

2.4. Experimental results. In this section, we summarize experimental results for graph 3-coloring from [7], which illustrate the practical performance of the NulLA and FPNulLA algorithms. For further and more detailed results, see [8, 35, 7]. Experimentally, for graph 3-coloring, NulLA and FPNulLA are well-suited to proving infeasibility, that is, that no 3-coloring exists. The system polynomials we use to encode 3-colorability has coefficients on $\mathbb{F}_2$ (see Proposition 1.1) and thus the linear algebra operations are very fast. However, even though in theory NulLA and FPNulLA can determine feasibility, for the experiments described below NulLA and FPNulLA are only suitable for proving infeasibility.

Here, we are interested in the percentage of randomly generated graphs whose polynomial system encoding has a NulLA rank of one, a NulLA rank of two or a FPNulLA rank of one. The $G(n, p)$ model [13] is used

for generating random graphs where n is the number of vertices and p is the probability that an edge is included between any two vertices. Also, without loss of generality, for a slightly smaller polynomial encoding, the color of one of the vertices of each randomly generated graph was fixed.

The experimental results are presented in Figure 1 (taken from [7]), which plots the percentage of 1000 random graphs in $G(100, p)$ that were proven infeasible with a NulLA rank of one, with a NulLA rank of two, with a FPNulLA rank of one, or with an exact method versus the p value. The exact method used was to model graph 3-coloring as a Boolean satisfiability problem [12] and then use the program `zchaff` [52] to solve the satisfiability problem.

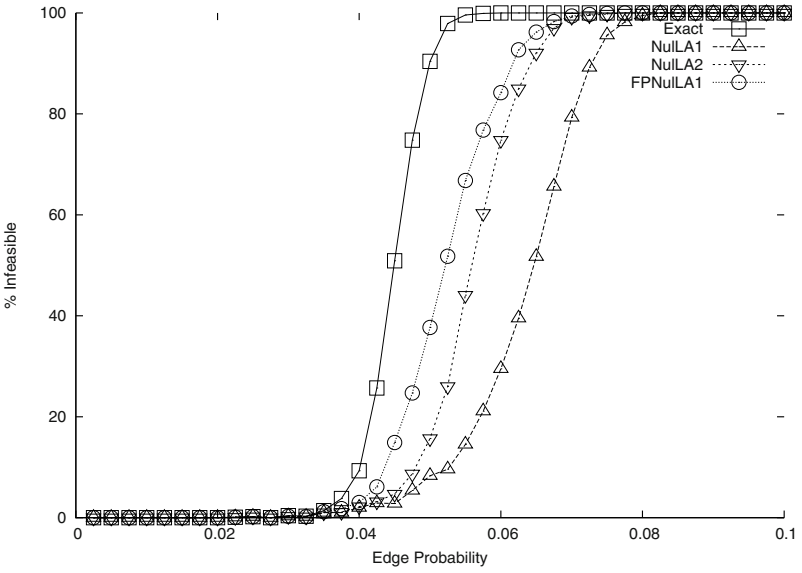

FIG. 1. *Non-3-colorable graphs with NulLA rank 1 and 2 and FPNulLA rank 1.*

It is well-known that there is a distinct phase transition from feasibility to infeasibility for graph 3-coloring, and it is at this phase transition that graphs exists for which it is difficult on average to prove infeasibility or feasibility (see [19]). Observe that the infeasibility curve for NulLA resembles that of the exact infeasibility curve and that the infeasibility curve for FPNulLA also resembles the infeasibility curve and clearly dominates the infeasibility curve for NulLA. These results suggest that the NulLA rank or FPNulLA rank are a reasonable measure of the hardness of proving infeasibility since those graphs that require a high rank are located near the phase transition.

Lastly, we comment on the runnings times of NulLA and FPNulLA and the exact approach using `zchaff` for the experiments on random graphs in

$G(100, p)$ above. The NulLA rank one and FPNulLA rank one approaches ran on average in less than a second for all p values. However, the exact approach using `zchaff` ran in split second times for all p values, but preliminary computational experiments indicate that the gap in running times between the exact approach and the FPNulLA rank one approach closes for larger graphs. The NulLA rank two approach ran on average in less than a second for $p \leq 0.04$ and $p \geq 0.08$, but the average running times peaked at about 24 seconds at $p = 0.65$. Interestingly, for each approach, the average running time peaked at the transition from feasible to infeasible at the p value where about half of the graphs were proven infeasible by the approach.

In order to better understand the practical implications of the NulLA and FPNulLA approaches, there needs to be more detailed computational studies performed to compare this approach with the exact method using satisfiability and other exact approaches such as traditional integer programming techniques. See [8] for some additional experimental data.

2.5. Application: The structure of non-3-colorable graphs. In this section, we state a combinatorial characterization of those graphs whose combinatorial system of equations encoding 3-colorability has a NulLA rank of one thus giving a class of polynomial solvable graphs by Lemma 2.1, and also, we recall bounds for the NulLA rank (see [35]):

THEOREM 2.3. *The NulLA rank for a polynomial encoding over $\mathbb{F}_2$ of the 3-colorability of a graph with n vertices with no 3-coloring is at least one and at most $2n$. Moreover, in the case of a non-3-colorable graph containing an odd-wheel (e.g. a 4-clique) as a subgraph, the NulLA rank is exactly one.*

Now we look at those non-3-colorable graphs that have a NulLA rank of one. Let A denote the set of all possible directed edges or arcs in the graph G. We are interested in two types of substructures of the graph G: oriented partial-3-cycles and oriented chordless 4-cycles (see Figure 2). An **oriented partial-3-cycle** is a set of two arcs of a 3-cycle, that is, a set $\{(i, j), (j, k)\}$ also denoted (i, j, k) where $(i, j), (j, k), (k, i) \in A$. An **oriented chordless 4-cycle** is a set of four arcs $\{(i, j), (j, l), (l, k), (k, i)\}$ also denoted (i, j, k, l) where $(i, j), (j, l), (l, k), (k, i) \in A$ and $(j, k), (i, l) \notin A$.

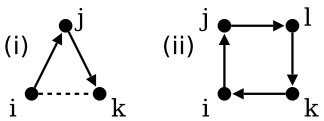

FIG. 2. *(i) oriented partial 3-cycle and (ii) an oriented chordless 4-cycle.*

Now, we can state a sufficient condition for non-3-colorability [7]. This sufficient condition is satisfied if and only if the combinatorial system encoding 3-coloring has a NulLA rank of one, which is proved in [7].

THEOREM 2.4. *The graph G is not 3-colorable if there exists a set C of oriented partial 3-cycles and oriented chordless 4-cycles such that*

1. $|C_{(i,j)}| + |C_{(j,i)}| \equiv 0 \pmod 2$ *for all* $(i,j) \in E$ *and*
2. $\sum_{(i,j)\in A, i<j} |C_{(i,j)}| \equiv 1 \pmod 2$

where $|C_{(i,j)}|$ denotes the number of cycles in C (either 3-cycles or 4-cycles) in which the arc $(i,j) \in A$ appears.

Condition 1 in Theorem 2.4 means that every undirected edge of G is covered by an even number of directed edges from cycles in C (ignoring orientation). Condition 2 in Theorem 2.4 means that, given any orientation of G, the total number of times the arcs in that orientation appear in the cycles of C is odd. The particular orientation we use in Theorem 2.4 is the orientation given by the set of arcs $\{(i,j) \in A : i < j\}$, but the particular orientation we use for Condition 2 is irrelevant (see [7]).

Using Theorem 2.4, proving that graphs containing odd wheels (e.g., 4-cliques) are not 3-colorable (see Theorem 2.3) is straight-forward ([7]):

EXAMPLE 8. *Assume a graph G contains an odd wheel with vertices labelled as in Figure 3 below. Consider the following set of oriented partial 3-cycles: $C := \{(i,1,i+1) : 2 \le i \le n-1\} \cup \{(n,1,2)\}$. The oriented partial 3-cycles of C are shown in Figure 3.*

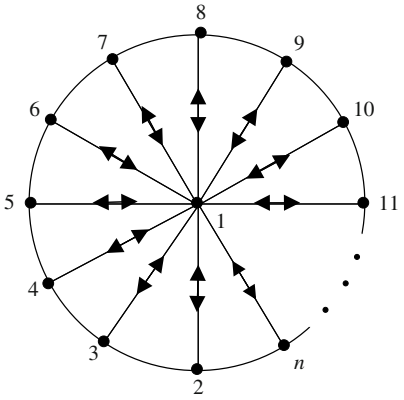

FIG. 3. *Odd wheel.*

The set C satisfies Condition 1 of Theorem 2.4 since each edge is covered by exactly zero or two cycles in C. Also, C satisfies Condition 2 of Theorem 2.4 since each arc $(1,i) \in Arcs(G)$ is covered exactly once by a cycle in C and there are an odd number of arcs $(1,i) \in Arcs(G)$. Thus, G is non-3-colorable by Theorem 2.4.

The Grötzsch graph is a non-trivial example of a non-3-colorable graph with a degree one Nullstellensatz certificate ([7]):

EXAMPLE 9. *Consider the Grötzsch graph (Mycielski 4) in Figure 4, which has no 3-coloring. It contains no 3-cycles. Now, consider the*

following set of oriented chordless 4-cycles, which we show gives a certificate of non-3-colorability by Theorem 2.4.

$$C := \{(1,2,3,7),(2,3,4,8),(3,4,5,9),(4,5,1,10),(1,10,11,7),\\ (2,6,11,8),(3,7,11,9),(4,8,11,10),(5,9,11,6)\}.$$

Figure 4 illustrates the edge directions for the 4-cycles of C. Each undirected edge of the graph is contained in exactly two 4-cycles, so C satisfies Condition 1 of Theorem 2.4. Now,

$$|C_{(6,11)}| = |C_{(7,11)}| = |C_{(8,11)}| = |C_{(9,11)}| = |C_{(10,11)}| = 1,$$

and $|C_{(i,j)}| \equiv 0 \pmod 2$ *for all other arcs* $(i,j) \in A$ *where* $i < j$. *Thus,*

$$\sum_{(i,j)\in A, i<j} |C_{(i,j)}| \equiv 1 \pmod 2,$$

so Condition 2 is satisfied, and therefore, the graph has no 3-coloring.

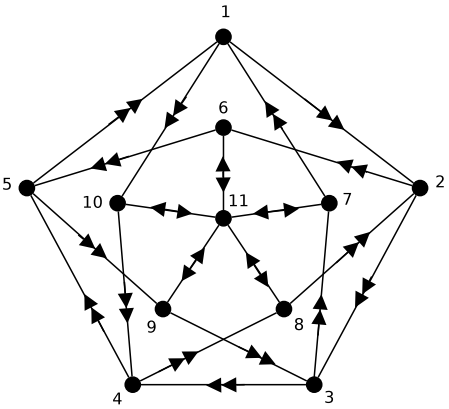

FIG. 4. *Grötzsch graph*

There are no known combinatorial characterizations concerning higher NulLA ranks.

3. Adding polynomial inequalities. Up until this point we have worked over arbitrary fields (with special attention to finite fields due to their fast and exact computation), where the only allowable constraints were equations. Now we turn our attention to the real case (i.e. $\mathbb{K} = \mathbb{R}$), where we have the additional possibility of specifying *inequalities* (more generally, one can work over *ordered* or *formally real* fields). In this case, following the terminology of real algebraic geometry, we call the solution set of a system of polynomial equations and inequalities a *basic semialgebraic set*. Note that convex polyhedra correspond to the particular case where

all the constraint polynomials have degree one. As we have seen earlier in the Positivstellensatz (Theorem 1.4 above), the emptiness of a basic semialgebraic set can be certified through an algebraic identity involving sum of squares of polynomials.

The connection between sum of squares decompositions of polynomials and convex optimization can be traced back to the work of N. Z. Shor [48]. His work went relatively unnoticed for several years, until several authors, including Lasserre, Nesterov, and Parrilo, observed, around the year 2000, that the existence of sum of squares decompositions and the search for infeasibility certificates for a semialgebraic set can be addressed via a sequence of semidefinite programs relaxations [23, 40, 41, 39]. The first part of this section will be a short description of the connections between sums of squares and semidefinite programming, and how the Positivstellensatz allows, in a analogous way to what was presented in Section 2 for the Nullstellensatz, for a systematic way to formulate these semidefinite relaxations.

A very central preoccupation of combinatorial optimizers has been the understanding of the facets that describe the integer hull (normally binary) of a combinatorial problem. As we will see later on, one can recover quite a bit of information about the integer hull of combinatorial problems from a sequence combinatorially controlled SDPs. This kind of approach was pioneered in the lift-and-project method of Balas, Ceria and Cornuéjols [1], the matrix-cut method of Lovász and Schrijver [34] and the linearization technique of Sherali-Adams [47]. Here we try to present more recent developments (see [30] and references therein for a very extensive survey).

3.1. Sums of squares, SDP, and feasibility of semialgebraic sets. Recall that a multivariate polynomial $p(x)$ is a *sum of squares* (SOS for short) if it can be written as a sum of squares of other polynomials, that is, $p(x) = \sum_i q_i^2(x), \quad q_i(x) \in \mathbb{R}[x]$. The condition that a polynomial is a sum of squares is a quite natural sufficient test for polynomial non-negativity. Thus instead of asking whether even degree polynomials are non-negative we ask the easier question whether they are sums of squares. More importantly, as we shall see, the existence of a sum of squares decomposition can be decided via semidefinite programming.

THEOREM 3.1. *A polynomial $p(x)$ is SOS if and only if $p(x) = z^T Q z$, where z is a vector of monomials in the x_i variables, and Q is a symmetric positive semidefinite matrix.*

By the theorem above, every SOS polynomial can be written as a quadratic form in a set of monomials, with the corresponding matrix being positive semidefinite. The vector of monomials z in general depends on the degree and sparsity pattern of $p(x)$. If $p(x)$ has n variables and total degree $2d$, then z can always be chosen as a subset of the set of monomials of degree less than or equal to d, which has cardinality $\binom{n+d}{d}$.

EXAMPLE 10. *The polynomial* $p(x_1, x_2) = x_1^2 - x_1x_2^2 + x_2^4 + 1$ *is SOS. Among infinitely many others,* $p(x_1, x_2)$ *has the following decompositions:*

$$\begin{aligned} p(x_1, x_2) &= \frac{3}{4}(x_1 - x_2^2)^2 + \frac{1}{4}(x_1 + x_2^2)^2 + 1 \\ &= \frac{1}{9}(3 - x_2^2)^2 + \frac{2}{3}x_2^2 + \frac{1}{288}(9x_1 - 16x_2^2)^2 + \frac{23}{32}x_1^2. \end{aligned}$$

The polynomial $p(x_1, x_2)$ *has the following representation:*

$$p(x_1, x_2) = \frac{1}{6}\begin{bmatrix} 1 \\ x_2 \\ x_2^2 \\ x_1 \end{bmatrix}^T \begin{bmatrix} 6 & 0 & -2 & 0 \\ 0 & 4 & 0 & 0 \\ -2 & 0 & 6 & -3 \\ 0 & 0 & -3 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ x_2 \\ x_2^2 \\ x_1 \end{bmatrix}$$

where the matrix in the expression above is positive semidefinite.

In the representation $f(x) = z^TQz$, for the right- and left-hand sides to be identical, all the coefficients of the corresponding polynomials should be equal. Since Q is simultaneously constrained by linear equations and a positive semidefiniteness condition, the problem can be easily seen to be directly equivalent to a semidefinite programming feasibility problem in the standard primal form.

Now we describe an algorithm (originally presented in [40, 41]) and illustrate it with an example, on how we can use SDPs to decide the feasibility of a system of polynomial inequalities. Exactly as we did for the Nullstellensatz case, we can look for the existence of a Positivstellensatz certificate of bounded degree D (see Theorem 1.4). Once we assume that the degree D is fixed we can apply Theorem 3.1 and obtain a reformulation as a semidefinite programming problem. We formalize this description in the following algorithm:

Algorithm 3 Bounded degree Positivstellensatz [40, 41]

Input: A polynomial system $\{f_i(x) = 0, g_i(x) \geq 0\}$ and a Positivstellensatz bound D.

Output: FEASIBLE, if $\{f_i(x) = 0, g_i(x) \geq 0\}$ is feasible over $\mathbb{R}$, else INFEASIBLE.

for $d = 0, 1, 2, \ldots, D$ **do**

If there exist β_i, $s_\alpha \in \mathbb{R}[x]$ such that $-1 = \sum_i \beta_i f_i + \sum_{\alpha \in \{0,1\}^n} s_\alpha g^\alpha$, with s_α SOS, $\deg(\beta_i f_i) \leq d$, $\deg(s_\alpha g^\alpha) \leq d$ then **return** INFEASIBLE.

$d \leftarrow d + 1$.

end for

Return FEASIBLE.

Notice that the membership test in the main loop of the algorithm is, by the results described at the beginning of this section, equivalent to a finite-sized semidefinite program. Similarly to the Nullstellensatz case,

the number of iterations (i.e., the degree of the certificates) serves as a quantitative measure of the hardness in proving infeasibility of the system. As we will describe in more detail in Section 3.4, in several situations one can give further refined characterization on these degrees.

EXAMPLE 11. *Consider the polynomial system* $\{f = 0,\, g \geq 0\}$ *from Example 2, where* $f := x_2 + x_1^2 + 2 = 0$ *and* $g := x_1 - x_2^2 + 3 \geq 0$. *At the d-th iteration of Algorithm 3 applied to the polynomial problem* $\{f = 0,\, g \geq 0\}$, *one asks whether there exist polynomials* $\beta, s_1, s_2 \in \mathbb{K}[x]$ *such that* $\beta f + s_1 + s_2 \cdot g = -1$ *where* s_1, s_2 *are SOS and* $\deg(s_1), \deg(s_2 \cdot g), \deg(\beta \cdot f) \leq d$. *For each fixed positive integer d this can be tested by a (possibly large) semidefinite program.*

Solving this for $d = 2$, *we have* $\deg(s_1) \leq 2$, $\deg(s_2) = 0$ *and* $\deg(\beta) = 0$, *so* s_2 *and* β *are constants and*

$$\begin{aligned} s_1 = z^T Q z &= \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{23} \\ Q_{13} & Q_{23} & Q_{33} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \\ &= Q_{11} + 2Q_{12}x_1 + 2Q_{13}x_2 + Q_{22}x_1^2 + 2Q_{23}x_1x_2 + Q_{33}x_2^2 \end{aligned}$$

where $z = (1, x_1, x_2)^T$ *and* $Q \in \mathbb{R}^{3\times 3}$ *is a symmetric positive semidefinite matrix. Thus, the certificate for* $D = 2$ *is* $\beta f + z^t Q z + s_2 \cdot g = -1$ *where* $Q \succeq 0$ *and* $s_2 \geq 0$. *If we expand the left hand side and equate coefficients on both sides of the equation, we arrive at the following SDP:*

$$\begin{aligned} 2\beta + Q_{11} + 3s_2 &= -1 \quad (1), & 2Q_{12} + s_2 &= 0 \quad (x_1), \\ \beta + 2Q_{13} &= 0 \quad (x_2), & \beta + Q_{22} &= 0 \quad (x_1^2), \\ 2Q_{23} &= 0 \quad (x_1x_2), & Q_{33} - s_2 &= 0 \quad (x_2^2) \end{aligned}$$

where $Q \succeq 0$ *and* $s_2 \geq 0$. *This SDP has a solution as follows:*

$$Q = \begin{bmatrix} 5 & -1 & 3 \\ -1 & 6 & 0 \\ 3 & 0 & 2 \end{bmatrix},\ s_2 = 2 \ \text{ and } \ \beta = -6.$$

The resulting identity, which is the same as the one given in Example 2, proves the inconsistency of the system.

As outlined in the preceding paragraphs, there is a direct connection going from general polynomial optimization problems to SDP, via the Positivstellensatz infeasibility certificates. Even though we have discussed only feasibility problems here, there are obvious straightforward connections with optimization. For instance, by considering the emptiness of the sublevel sets of the objective function, or using representation theorems for positive polynomials, sequences of converging bounds indexed by certificate degree can be directly constructed; see e.g. [40, 23, 42]. These schemes have been implemented in software packages such as SOSTOOLS [43], GloptiPoly [17], and YALMIP [31].

3.2. Semidefinite programming relaxations. In the last section, we have described the search for Positivstellensatz infeasibility certificates formulated as a semidefinite programming problem. We now describe an alternative interpretation, obtained by dualizing the corresponding semidefinite programs. This is the exact analogue of the construction presented in Section 2.2, and is closely related to the approach via truncated moment sequences developed by Lasserre [23].

Recall that in the approach in Section 2.2, the linear relaxations were constructed by replacing every monomial x^α by a new variable λ_{x^α}. Furthermore, new redundant equations were obtained by multiplying an existing constraint $f(x) = 0$ by terms of the form x_i, yielding $x_i f(x) = 0$ (essentially, generating the ideal of valid equations). In the inequality case, and as suggested by the Positivstellensatz, new inequality constraints will be generated by both squarefree multiplication of the original constraints, and by multiplication against sums of squares. That is, if $g_i(x) \geq 0$ and $g_j(x) \geq 0$ are valid inequalities, then so are $g_i(x)g_j(x) \geq 0$ and $g_i(x)s(x) \geq 0$, where $s(x)$ is SOS. After substitution with the extended variables λ, we then obtain a new system of linear equations and inequalities, with the property that the resulting inequality conditions are *semidefinite* conditions. The presence of the semidefinite constraints arises because we do not specify *a priori* what the multipliers $s(x)$ are, but only give their linear span.

EXAMPLE 12. *Consider the polynomial system discussed earlier in Example 2. As described, new linear and semidefinite constraints are obtained by linearizing all the polynomial constraints in the original system. The corresponding relaxation is (for $d = 2$):*

$$\begin{bmatrix} \lambda_1 & \lambda_{x_1} & \lambda_{x_2} \\ \lambda_{x_1} & \lambda_{x_1^2} & \lambda_{x_1x_2} \\ \lambda_{x_2} & \lambda_{x_1x_2} & \lambda_{x_2^2} \end{bmatrix} \succeq 0, \qquad \lambda_{x_2}+\lambda_{x_1^2}+2\lambda_1 = 0, \qquad \lambda_{x_1}-\lambda_{x_2^2}+3\lambda_1 \geq 0,$$

plus the condition $\lambda_1 > 0$ (without loss of generality, we can take $\lambda_1 = 1$). The first semidefinite constraint arises from linearizing the square of an arbitrary degree one polynomial, while the other two constraints are the direct linearization of the original equality and inequality constraints. The resulting problem is a semidefinite program, and in this case, its infeasibility directly shows that the original system of polynomial inequalities does not have a solution.

An appealing geometric interpretation follows from considering the projection of the feasible set of these relaxations in the space of original variables (i.e., λ_{x_i}). For the linear algebra relaxations of Section 2.2, we obtain outer approximations to the *affine hull* of the solution set (an algebraic variety), while the SDP relaxation described here constructs outer approximations to the *convex hull* of the corresponding semialgebraic set. This latter viewpoint will be discussed in Section 3.3, for the case of equations arising from combinatorial problems.

3.3. Theta bodies. Recall that traditional modeling of combinatorial optimization problems often uses 0/1 incidence vectors. The set S of solutions of a combinatorial problem (e.g., the stable sets, traveling salesman tours) is often computed through the (implicit) convex hull of such incidence vectors. Just as in the stable set and max-cut examples in Proposition 1.1, the incidence vectors can be seen at the set of *real* solutions to a system of polynomial equations: $f_1(x) = f_2(x) = \cdots = f_m(x) = 0$, where $f_1, \ldots, f_m \in \mathbb{R}[x] := \mathbb{R}[x_1, \ldots, x_n]$. Over the years there have been well-known attempts to understand the structure of these convex hulls through semidefinite programming relaxations (see [47, 34, 26, 33]) and in fact they are closely related [27, 30]. Here we wish to summarize some recent results that give appealing structural properties, in terms of the associated system of equations (see [15, 14] for details).

Let us start with a historically important example: Given an undirected finite graph $G = (V, E)$, consider the set S_G of characteristic vectors of stable sets of G. The convex hull of S_G, denoted by $\mathrm{STAB}(G)$, is the *stable set polytope*. As we mentioned already the vanishing ideal of S_G is given by $I_G := \langle x_i^2 - x_i \ (\forall\ i \in V),\ x_i x_j \ (\forall\ \{i,j\} \in E)\rangle$ which is a real radical zero-dimensional ideal in $\mathbb{R}[x]$. In [32], Lovász introduced a semidefinite relaxation, $\mathrm{TH}(G)$, of the polytope $\mathrm{STAB}(G)$, called the *theta body* of G. There are multiple descriptions of $\mathrm{TH}(G)$, but the one in [34, Lemma 2.17], for instance, shows that $\mathrm{TH}(G)$ can be defined completely in terms of the polynomial system I_G. It is easy to show that $\mathrm{STAB}(G) \subseteq \mathrm{TH}(G)$, and remarkably, we have that $\mathrm{STAB}(G) = \mathrm{TH}(G)$ if and only if the graph is *perfect*. We will now explain how the case of stable sets can be generalized to construct theta bodies for many other combinatorial problems.

We will construct an approximation of the convex hull of a finite set of points S, denoted $\mathrm{conv}(S)$, by a sequence of convex bodies recovered from "degree truncations" of the defining polynomial systems. In what follows I will be a radical polynomial ideal. A polynomial f is *non-negative* modulo I, written as $f \geq 0 \bmod I$, if $f(s) \geq 0$ for all $s \in \mathcal{V}_{\mathbb{R}}(I)$. More strongly, the polynomial f is a *sum of squares (sos)* mod I if there exists $h_j \in \mathbb{R}[x]$ such that $f \equiv \sum_{j=1}^{t} h_j^2 \bmod I$ for some t, or equivalently, $f - \sum_{j=1}^{t} h_j^2 \in I$. If, in addition, each h_j has degree at most k, then we say that f is *k-sos* mod I. The ideal I is *k-sos* if *every* polynomial that is non-negative mod I is k-sos mod I. If every polynomial of degree at most d that is non-negative mod I is k-sos mod I, we say that I is *(d,k)-sos*.

Note that $\mathrm{conv}(\mathcal{V}_{\mathbb{R}}(I))$, the convex hull of $\mathcal{V}_{\mathbb{R}}(I)$, is described by the linear polynomials f such that $f \geq 0 \bmod I$. A certificate for the non-negativity of f mod I is the existence of a sos-polynomial $\sum_{j=1}^{t} h_j^2$ that is congruent to f mod I. One can now investigate the convex hull of S through the hierarchy of nested closed convex sets defined by the semidefinite programming relaxations of the set of $(1,k)$-sos polynomials.

DEFINITION 3.1. *Let $I \subseteq \mathbb{R}[x]$ be an ideal, and let k be a positive integer. Let $\Sigma_k \subset \mathbb{R}[x]$ be the set of all polynomials that are* **k-sos** *mod I.*

1. *The k-th* theta body *of I is*

$$\mathrm{TH}_k(I) := \{x \in \mathbb{R}^n \,:\, f(x) \geq 0 \text{ for every linear } f \in \Sigma_k\}.$$

2. *The ideal I is* TH_k-exact *if the k-th theta body $\mathrm{TH}_k(I)$ coincides with the closure of* $\mathrm{conv}(\mathcal{V}_{\mathbb{R}}(I))$.
3. *The* theta-rank *of I is the smallest k such that $\mathrm{TH}_k(I)$ coincides with the closure of* $\mathrm{conv}(\mathcal{V}_{\mathbb{R}}(I))$.

EXAMPLE 13. *Consider the ideal $I = \langle x^2y - 1\rangle \subset \mathbb{R}[x,y]$. Then $\mathrm{conv}(\mathcal{V}_{\mathbb{R}}(I)) = \{(p_1,p_2) \in \mathbb{R}^2 \,:\, p_2 > 0\}$, and any linear polynomial that is non-negative over $\mathcal{V}_{\mathbb{R}}(I)$ is of the form $\alpha + \beta y$, where $\alpha, \beta \geq 0$. Since $\alpha y + \beta \equiv (\sqrt{\alpha}xy)^2 + (\sqrt{\beta})^2$ mod I, I is $(1,2)$-sos and TH_2-exact.*

EXAMPLE 14. *For the case of the stable sets of a graph G, one can see that*

$$\mathrm{TH}_1(I_G) = \left\{ y \in \mathbb{R}^n \,:\, \begin{array}{l} \exists\, M \succeq 0, M \in \mathbb{R}^{(n+1)\times(n+1)} \text{ such that} \\ M_{00} = 1, \\ M_{0i} = M_{i0} = M_{ii} = y_i \;\forall\; i \in V \\ M_{ij} = 0 \;\forall\; \{i,j\} \in E \end{array} \right\}.$$

It is known that $\mathrm{TH}_1(I_G)$ is precisely Lovász's theta body of G. *The ideal I_G is TH_1-exact precisely when the graph G is perfect.*

By definition, $\mathrm{TH}_1(I) \supseteq \mathrm{TH}_2(I) \supseteq \cdots \supseteq \mathrm{conv}(\mathcal{V}_{\mathbb{R}}(I))$. As seen in Example 13, $\mathrm{conv}(\mathcal{V}_{\mathbb{R}}(I))$ may not always be closed and so the theta-body sequence of I can converge, if at all, only to the closure of $\mathrm{conv}(\mathcal{V}_{\mathbb{R}}(I))$. But the good news for combinatorial optimization is that there is plenty of good behavior for problems arising with a finite set of possible solutions.

3.4. Application: cuts and exact finite sets. We discuss now a few important combinatorial examples. As we have seen in Section 2.5 for 3-colorability, and in the preceding section for stable sets, in some special cases it is possible to give nice combinatorial characterizations of when low-degree certificates can exactly recognize infeasibility. Here are a few additional results for the real case:

EXAMPLE 15. *For the max-cut problem we saw earlier, the defining vanishing ideal is $I(SG) = \langle x_e^2 - x_e \;\forall\; e \in E,\; x^T \;\forall\; T$ an odd cycle in $G\rangle$. In this case one can prove that the ideal $I(SG)$ is TH_1-exact if and only if G is a bipartite graph. In general the theta-rank of $I(SG)$ is bounded above by the size of the max-cut in G. There is no constant k such that $\mathrm{TH}_k(I(SG)) = \mathrm{conv}(SG)$, for all graphs G. Other formulations of max-cut are studied in [14].*

Recall that when $S \subset \mathbb{R}^n$ is a finite set, its vanishing ideal $I(S)$ is zero-dimensional and real radical (see [36] Section 12.5 for a definition of the real radical). In what follows, we say that a finite set $S \subset \mathbb{R}^n$ is *exact* if its vanishing ideal $I(S) \subseteq \mathbb{R}[x]$ is TH_1-exact.

THEOREM 3.2 ([15]). *For a finite set $S \subset \mathbb{R}^n$, the following are equivalent.*

1. *S is exact.*
2. *There is a finite linear inequality description of* $\mathrm{conv}(S)$ *in which for every inequality* $g(x) \geq 0$*,* g *is* 1*-sos mod* $I(S)$*.*
3. *There is a finite linear inequality description of* $\mathrm{conv}(S)$ *such that for every inequality* $g(x) \geq 0$*, every point in* S *lies either on the hyperplane* $g(x) = 0$ *or on a unique parallel translate of it.*
4. *The polytope* $\mathrm{conv}(S)$ *is affinely equivalent to a compressed lattice polytope (every reverse lexicographic triangulation of the polytope is unimodular with respect to the defining lattice).*

EXAMPLE 16. *The vertices of the following* $0/1$*-polytopes in* $\mathbb{R}^n$ *are exact for every* n*: (1) hypercubes, (2) (regular) cross polytopes, (3) hypersimplices (includes simplices), (4) joins of* 2*-level polytopes, and (5) stable set polytopes of perfect graphs on* n *vertices.*

More strongly one can say the following.

PROPOSITION 3.1. *Suppose* $S \subseteq \mathbb{R}^n$ *is a finite point set such that for each facet* F *of* $\mathrm{conv}(S)$ *there is a hyperplane* H_F *such that* $H_F \cap \mathrm{conv}(S) = F$ *and* S *is contained in at most* $t+1$ *parallel translates of* H_F*. Then* $I(S)$ *is* TH_t*-exact.*

In [15] the authors show that theta bodies can be computed explicitly as projections to the feasible set of a semidefinite program. These SDPs are constructed using the *combinatorial moment matrices* introduced by [29].

4. Recovering solutions in the feasible case. In principle, it is possible to find the actual roots of the system of equations (and thus the colorings, stable sets, or desired combinatorial object) whenever the relaxations are feasible and a few additional conditions are satisfied. Here we outline the linear algebra relaxations case, but the semidefinite case is very similar; see e.g. [18, 25] for this case.

We describe below how, under certain conditions, it is possible to recover the solution of the original polynomial system from the relaxations (linear or semidefinite) described in earlier sections. The main concepts are very similar for both methodologies, and are based on the well-known eigenvalue methods for polynomial equations; see e.g. [6, §2.4]. The key idea for extracting solutions is the fact that from the relaxations one can obtain a finite-dimensional representation of the vector space $\mathbb{K}[x]/I$ and its multiplicative structure, where I is the ideal $\mathbf{ideal}(F)$ (in the case of linear relaxations). In order to do this, we need to compute a basis of the vector space $\mathbb{K}[x]/I$, and construct matrix representations for the multiplication operators $M_{x_i} : \mathbb{K}[x]/I \to \mathbb{K}[x]/I$ where $[f] \mapsto [x_i f]$ for all $[f] \in \mathbb{K}[x]/I$. Then, we can use the eigenvalue/eigenvector methods to compute solutions (see e.g., [10]).

A sufficient condition for the existence of a suitable basis of $\mathbb{K}[x]/I$ is given by Theorem 2.2. Under this condition, multiplication matrices M_{x_i} can be easily computed. In particular, if we have computed a set $F \subset \mathbb{K}[x]$ that satisfies the conditions of Theorem 2.2 by running FPNulLA, then

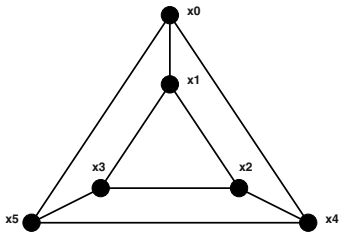

FIG. 5. *Graph for Example 17.*

finding a basis of R/I and computing its multiplicative structure is straightforward using linear algebra (see e.g., [37]). By construction, the matrices M_{x_i} commute pairwise, and to obtain the roots one must diagonalize the corresponding commutative algebra. It is well-known (see, e.g., [6]), that this can be achieved by forming a random linear combination of these matrices. This random matrix will generically have distinct eigenvalues, and the corresponding matrix of eigenvectors will give the needed change of basis. In the case of a finite field, it is enough to choose the random coefficients over an algebraic extension of sufficiently large degree, instead of working over the algebraic closure (alternatively, the more efficient methods in [11] can be used). The entries of the diagonalized matrices directly provide the coordinates of the roots.

REMARK 4.1. The condition in Theorem (2.2) can in general be a strong requirement for recovery of solutions, since it implies that we can obtain *all* solutions of the polynomial system. In some occasions, it may be desirable to obtain just a single solution, in which case weaker conditions may be of interest.

EXAMPLE 17. *Consider the following polynomial system over* $\mathbb{F}_2$*, that corresponds to the 3-colorings of the six-node graph in Figure 5:*

$$x_i^3 + 1 = 0 \quad \forall i \in V, \qquad x_i^2 + x_i x_j + x_j^2 = 0 \quad \forall (i,j) \in E.$$

We add to these equations the symmetry-breaking constraint $x_0 = 1$*. After running NulLA with this system as an input, we obtain multiplication matrices over* $\mathbb{F}_2$*, of dimensions* 4×4*, given by:*

$$M_{x_1} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad M_{x_2} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad M_{x_3} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$M_{x_4} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad M_{x_5} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Diagonalizing the corresponding commutative algebra, we obtain the change of basis matrix given by

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \omega^2 & \omega & \omega & \omega^2 \\ 1 & 1 & \omega^2 & \omega \\ \omega^2 & \omega & 1 & 1 \end{bmatrix},$$

where ω is a primitive root of 1, i.e., it satisfies $w^2 + w + 1 = 0$. It can be easily verified that all the matrices $T^{-1}M_{x_i}T$ are diagonal, and given by:

$$T^{-1}M_{x_1}T = \mathrm{diag}[\omega, \omega^2, \omega, \omega^2] \quad T^{-1}M_{x_2}T = \mathrm{diag}[\omega^2, \omega, 1, 1]$$
$$T^{-1}M_{x_3}T = \mathrm{diag}[1, 1, \omega^2, \omega] \quad T^{-1}M_{x_4}T = \mathrm{diag}[\omega, \omega^2, \omega^2, \omega]$$
$$T^{-1}M_{x_5}T = \mathrm{diag}[\omega^2, \omega, \omega, \omega^2],$$

which correspond to the four possible 3-colorings of the graph. For instance, from the second diagonal entry of each matrix we obtain the feasible coloring $(x_0, x_1, x_2, x_3, x_4, x_5) \to (1, \omega^2, \omega, 1, \omega^2, \omega)$.

Acknowledgements. We are grateful to the two anonymous referees who provided many useful corrections and comments that greatly enhanced the quality of presentation. We are also grateful to Jon Lee, Susan Margulies, Mohamed Omar, and Chris Hillar for their ideas and support.

APPENDIX

A. Proofs. This appendix contains proofs of some the results used in the main body of the paper that are either hard to find or whose original proof, available elsewhere, is not written in the language of this survey.

For the purpose of formally prove Theorem 2.2 we need to formalize further some of the notions of Section 2: The space $\mathbb{K}[[x]]$ is isomorphic to the dual vector space of $\mathbb{K}[x]$ consisting of all linear functionals on $\mathbb{K}[x]$, that is, $\mathbb{K}[[x]] \cong \mathrm{Hom}(\mathbb{K}[x], \mathbb{K})$. We choose to use $\mathbb{K}[[x]]$ instead of as the dual vector space of $\mathbb{K}[x]$ (see e.g., [24]) because using $\mathbb{K}[[x]]$ makes clearer the linearization of a system of polynomial equations. The map $\tau : \mathbb{K}[[x]] \to \mathrm{Hom}(\mathbb{K}[x], \mathbb{K})$ where $(\tau(\lambda))(g) = \sum_{\alpha \in \mathbb{N}^n} \lambda_\alpha g_\alpha = \lambda * g$ for all $\lambda \in \mathbb{K}[[x]]$ and $g \in \mathbb{K}[x]$ is an isomorphism and the inverse map is $\tau^{-1}(\psi) = \sum_{\alpha \in \mathbb{N}^n} \psi(x^\alpha) x^\alpha$ for all $\psi \in \mathrm{Hom}(\mathbb{K}[x], \mathbb{K})$. For a given set $F \subseteq \mathbb{K}[x]$, there is an analogue of the annihilator F° in the context of the dual vector space $\mathrm{Hom}(\mathbb{K}[x], \mathbb{K})$ as follows: $\mathrm{Ann}(\mathbb{K}[x], F) := \{\psi \in \mathrm{Hom}(\mathbb{K}[x], \mathbb{K}) : \psi(f) = 0, \forall f \in F\}$. Note that $\mathbb{F}^\circ \cong \mathrm{Ann}(\mathbb{K}[x], F)$ since $\tau(F^\circ) = \mathrm{Ann}(\mathbb{K}[x], F)$.

LEMMA A.1. *Let $F \subseteq \mathbb{K}[x]$ be a vector subspace and $k \in \mathbb{N}$. Then,* $\dim(\mathbb{K}[x]_k / (F \cap \mathbb{K}[x]_k)) = \dim_k(F^\circ)$.

Proof. We know from Theorem 3.14 in [45], $\mathrm{Ann}(\mathbb{K}[x], F \cap \mathbb{K}[x]_k) = \mathrm{Ann}(\mathbb{K}[x], F) + \mathrm{Ann}(\mathbb{K}[x], \mathbb{K}[x]_k)$; thus, $(F \cap \mathbb{K}[x]_k)^\circ = F^\circ + \mathbb{K}[x]_k^\circ$, and

so, we have $\pi_k((F \cap \mathbb{K}[x]_k)^\circ) = \pi_k(F^\circ) + \pi_k(\mathbb{K}[x]_k^\circ) = \pi_k(F^\circ)$. Moreover, from Theorems 3.12 and 3.15 in [45], we have $\mathrm{Ann}(\mathbb{K}[x]_k, F \cap \mathbb{K}[x]_k) \cong \mathrm{Hom}(\mathbb{K}[x]_k/(F \cap \mathbb{K}[x]_k), \mathbb{K}) \cong \mathbb{K}[x]_k/(F \cap \mathbb{K}[x]_k)$ since $\mathbb{K}[x]_k/(F \cap \mathbb{K}[x]_k)$ is finite dimensional, and finally, $\mathrm{Ann}(\mathbb{K}[x]_k, F \cap \mathbb{K}[x]_k) \cong \pi_k((F \cap \mathbb{K}[x]_k)^\circ)$ since, for the isomorphism $\tau_k : \mathrm{Hom}(\mathbb{K}[x]_k, \mathbb{K}) \to \mathbb{K}[x]_k$ where $\tau(\psi) = \sum_{\alpha \in \mathbb{N}^n : |\alpha| \leq k} \psi(x^\alpha) x^\alpha$, thus $\tau(\mathrm{Ann}(\mathbb{K}[x]_k, F \cap \mathbb{K}[x]_k)) = \pi_k((F \cap \mathbb{K}[x]_k)^\circ)$. ∎

We now present proofs verifying the correctness and efficiency of Algorithm 2. We begin by proving Theorem 2.2.

Proof. [Proof of Theorem 2.2] We will explicitly show that, under the hypothesis of the theorem, one can recover a finite dimensional vector space B such that $B \oplus F = \mathbb{K}[x]_d$ and $B \oplus I = \mathbb{K}[x]$. The result then follows from the equalities $\dim(I^\circ) = \dim(\mathbb{K}[x]/I) = \dim(B) = \dim(\mathbb{K}[x]_d/F) = \dim_d(F^\circ)$. We define the vector space $B \subseteq \mathbb{K}[x]_{d-1}$ such that $B \oplus (F \cap \mathbb{K}[x]_{d-1}) = \mathbb{K}[x]_{d-1}$. By assumption, $\dim_d(F^\circ) = \dim_{d-1}(F^\circ)$ implying $\dim(B) = \dim(\mathbb{K}[x]_{d-1}/F \cap \mathbb{K}[x]_{d-1}) = \dim(\mathbb{K}[x]_d/F)$, and thus, it follows that $B \oplus F = \mathbb{K}[x]_d$. It only remains to show that $B \oplus I = \mathbb{K}[x]$.

Denote $F^{[0]} = F$ and $F^{[k]} = (F^{[k-1]})^+$ for all $k \geq 1$. We show by induction on k that $B \oplus F^{[k]} = \mathbb{K}[x]_{d+k}$ for all $k \geq 0$, and hence $B \oplus I = \mathbb{K}[x]$. We have already established $B \oplus F = \mathbb{K}[x]_d$, so the claim holds for $k = 0$. The claim also holds for $k = 1$ as follows: $\mathbb{K}[x]_{d+1} = (\mathbb{K}[x]_d)^+ = (B \oplus F)^+ = B^+ + F^+ = B + F^+$ since $B^+ \subseteq \mathbb{K}[x]_d = B \oplus F$, and furthermore, the assumption $F^+ \cap \mathbb{K}[x]_d = F$ implies $F^+ \cap B = \emptyset$, and therefore, $B \oplus F^+ = \mathbb{K}[x]_{d+1}$. Now assume that the claim holds for $k \geq 1$ and let us prove it must hold for $k + 1$.

By the assumption that $B \oplus F^{[k]} = \mathbb{K}[x]_{d+k}$, there exists a vector space projection $\rho_k : \mathbb{K}[x]_{d+k} \to \mathbb{K}[x]_{d+k}$ where $\mathrm{im}(\rho_k) = B$, $\rho_k(b) = b$ for all $b \in B$ and $\ker(\rho_k) = F^{[k]}$. We extend the map ρ_k to the map $\rho_{k+1} : \mathbb{K}[x]_{d+k+1} \to \mathbb{K}[x]_{d+k+1}$ by defining $\rho_{k+1}(g) := \rho_k(g_0) + \sum_i \rho_k(x_i \rho_k(g_i))$ where $g = g_0 + \sum_i x_i g_i$ is a representation of g with $g_0, g_1, \ldots, g_n \in \mathbb{K}[x]_{d+k}$. We show below that ρ_{k+1} is well-defined meaning that the value of $\rho_{k+1}(g)$ is independent of the chosen representation of g since there may be multiple possible representations of g. It follows by construction that ρ_{k+1} is $\mathbb{K}$-linear, $\mathrm{im}(\rho_{k+1}) = B$, $\rho_{k+1}(b) = b$ for all $b \in B$ and $\ker(\rho_{k+1}) = F^{[k+1]}$, implying that ρ_{k+1} is a vector space projection and $B \oplus F^{[k+1]} = \mathbb{K}[x]_{d+k+1}$ as required.

We now show that ρ_{k+1} is well-defined. First, consider the special case where $g \in \mathbb{K}[x]_{d+k+1}$ is a monomial, that is, $g = x_i x_j x^\gamma$ for some i, j and some monomial $x^\gamma \in \mathbb{K}[x]_{d+k-1}$, so $\rho_{k+1}(g) = \rho_k(x_i \rho_k(x_j x^\gamma))$ or $\rho_{k+1}(g) = \rho(x_j \rho_k(x_i x^\gamma))$. We thus need to show that $\rho_k(x_i \rho_k(x_j x^\gamma)) = \rho_k(x_j \rho_k(x_i x^\gamma))$. Now, $x^\gamma = b + f$ for some $b \in B$ where $\rho_k(x^\gamma) = b$ and $f \in F^{[k-1]}$ $(k \geq 1)$. Then, $\rho_k(x_i x^\gamma) = \rho_k(x_i b + x_i f) = \rho_k(x_i b) + \rho_k(x_i f) = \rho_k(x_i b)$, and similarly, $\rho_k(x_j x^\gamma) = \rho_k(x_j b)$. Then,

$$\begin{aligned}
&\rho_k(x_i\rho_k(x_jx^\gamma)) - \rho_k(x_j\rho_k(x_ix^\gamma)) \\
&= \rho_k(x_i\rho_k(x_jb)) - \rho_k(x_j\rho_k(x_ib)) \\
&= \rho_k(x_i(x_jb - f_1)) - \rho_k(x_j(x_ib - f_1')) && (f_1, f_1' \in F) \\
&= (x_i(x_jb - f_1) - f_2) - (x_j(x_ib - f_1') - f_2') && (f_2, f_2' \in F) \\
&= x_jf_1' - x_if_1 + f_2' - f_2 \in F^+.
\end{aligned}$$

So, $\rho_k(x_i\rho_k(x^\alpha)) - \rho_k(x_j\rho_k(x^\beta)) \in F^+$. But, $\rho_k(x_i\rho_k(x^\alpha)) \in B$ and $\rho_k(x_j\rho_k(x^\beta)) \in B$ by definition, so $\rho_k(x_i\rho_k(x^\alpha)) - \rho_k(x_j\rho_k(x^\beta)) \in F^+ \cap B = \{0\}$ since $F^+ \cap \mathbb{K}[x]_d = F$. Thus, $\rho_k(x_i\rho_k(x^\alpha)) = \rho_k(x_j\rho_k(x^\beta))$ as required. By the $\mathbb{K}$-linearity of ρ_k, ρ_{k+1} is well-defined on $\mathbb{K}[x]_{d+k+1}$ as required. □

Theorem 2.2 (and its proof) can be seen as an adaptation and simplification of Theorem 4.2 and Algorithm 4.3 in [37], the main difference being that in Mourrain's terminology, we stick to a particular *order ideal* and only need to keep track of vector space dimensions instead of an explicit basis for B.

We now present a proof of termination of the FPNulLA algorithm (see also the comments following Algorithm 4.3 in [37]).

LEMMA A.2. *Let I be a zero-dimensional ideal, then FPNulLA (Algorithm 2) terminates.*

Proof. First, we prove that the inner while loop must terminate. Let $F \subseteq \mathbb{K}[x]_d$ be a vector space. We denote $F^{[0,d]} = F$ and $F^{[k,d]} = (F^{[k-1,d]})^+ \cap \mathbb{K}[x]_d$ for all $k \geq 1$ where $d = \deg(F)$. By construction, $F^{[k,d]} \subseteq F^{[k+1,d]} \subseteq \mathbb{K}[x]_d$ for all k. So, the sequence of vector spaces $F^{[0,d]}, F^{[1,d]}, \ldots, F^{[k,d]}, \ldots$ is an inclusion-wise increasing sequence of vector subspaces of $\mathbb{K}[x]_d$. Since $\mathbb{K}[x]_d$ is finite-dimensional, the sequence must reach a fixed point where $F^{[k,d]} = F^{[k+1,d]}$, which is the terminating condition of the inner loop of FPNulLA (Steps 4-7). Let $F^{[*,d]}$ denote this fixed point.

The outer loop of FPNulLA is essentially the same as NulLA. After k iterations of the outer loop, the vector space F contains at least all linear combinations of polynomials of the form $x^\alpha f$ where the total degree $|\alpha| \leq k$ and where f is one of the initial polynomials in F. Therefore, if the system $F(x) = 0$ is infeasible, Hilbert's Nullstellensatz guarantees that after a finite number of iterations, $1 \in F$ and the algorithm terminates.

It remains to show that the algorithm terminates when the system $F(x) = 0$ is feasible. Let $I = I(F)$. Since I is zero-dimensional, there must exist a finite-dimensional vector space $B \subset \mathbb{K}[x]$ such that $\mathbb{K}[x] = I \oplus B$ (see e.g. [5, 50]). Since the system $F(x) = 0$ is feasible, Hilbert's Nullstellensatz implies $1 \notin I$. Thus, we can choose B such that $1 \in B$. Now after finitely many iterations of the outer loop, any $f \in I$ will eventually be in F. Combined with the fact that B^+ is finite dimensional and $B^+ \subset I \oplus B$, this implies that $B^+ \subset F \oplus B$ after finitely many iterations of the outer loop. Also, since the inner loop has terminated, we know that $F = F^+ \cap \mathbb{K}[x]_d = F^{[1,d]}$. Next, we show that $\mathbb{K}[x]_d = F \oplus B$. Now,

$(F \oplus B)^{[1,d]} = F^{[1,d]} + B^{[1,d]} = F \oplus B$ since $F = F^{[1,d]}$ and $B^+ \subseteq F \oplus B$. Thus, $(F \oplus B)^{[*,d]} = F \oplus B$, and since $B^+ \subseteq F \oplus B$, this implies

$$(B^+)^{[*,d]} \subseteq (F \oplus B)^{[*,d]} = F \oplus B.$$

But $1 \in B$, so $\mathbb{K}[x]_d \subseteq (B^+)^{[*,d]}$ which then implies $\mathbb{K}[x]_d = F \oplus B$. Then, since $B \subseteq \mathbb{K}[x]_{d-1}$, we also have $\mathbb{K}[x]_{d-1} = (F \cap \mathbb{K}[x]_{d-1}) \oplus B$, and thus, $\dim(\mathbb{K}[x]_d/F) = \dim(B) = \dim(\mathbb{K}[x]_{d-1}/F)$, which is the stopping criterion of the outer loop. □

Now, we show that NulLA and FPNulLA algorithms run in polynomial time in the bit-size of the input data when the Nullstellensatz degree is assumed to be fixed. To begin, note that the number of monomials x^α with $\deg(x^\alpha) \leq k$ is $\binom{n+k}{k}$, which is $O(n^k)$.

Proof. (of Lemma 2.1). Let $d = \deg(F)$. First note that by definition (see section 2.1 in [46]) the input size of the defining basis $\{f_1, f_2, \ldots, f_m\}$ of F equals $O(cmn^d)$ where c is the average bit-size of the coefficients in the basis.

For the proof of (1), observe that in the k^{th} iteration of Algorithm 1 (when the F^+ operation has increased the degree of F by k), we solve a system of linear equations $A_k x = b_k$ to find coefficients of the Nullstellensatz certificate in Step 2 of Algorithm 1. The rows of A_k consist of vectors of coefficients of all polynomials of the form $x^\alpha f_i$ where $i = 1, \ldots m$ and $\deg(x^\alpha) \leq k$. Therefore, A_k has $O(mn^k)$ rows and each row has input size $O(cn^{d+k})$. Hence, the input size of A_k is $O(cmn^{d+2k})$. The input size of b_k, which is a vector of zeros and ones, is $O(mn^k)$. Thus, the input size of the linear system $A_k x = b_k$ is $O(cmn^{d+2k})$, which is polynomial in the input size of the basis of F and n, and thus, the system can be solved in polynomial time (see e.g. Theorem 3.3 of [46]). The complexity of the first L iterations is thus bounded by L times the complexity of the Lth iteration, which is polynomial in L, n and the input size of the defining basis of F. This completes the proof of the first part.

We now prove part (2). Denote by F_k the vector space computed at the start of the kth outer loop iteration. Let $\{g_1, \ldots g_{m_k}\}$ be a basis of F_k which was given to us either as an input or from the previous iteration. Observe that the $\deg(F_k) = d + k$, so each basis polynomial of F_k has bit size $O(\log_2(|\mathbb{K}|)n^{d+k})$. Note that $\dim(F_k) = m_k \leq O(n^{d+k})$; therefore the bit size of the entire basis $\{g_1, \ldots g_{m_k}\}$ is $M = O(\log_2(|\mathbb{K}|)n^{2(d+k)})$. Note that M is polynomial size in the input size of the initial basis $f_1, \ldots, f_m$.

Now we proceed to analyze the cost of the k^{th} iteration, meaning steps 3 to 10 in the pseudocode. As in part (1), Step 3, involves solving a linear system of size M; thus it can be done in polynomial time. In Step 4 we check whether $\dim(F_k) = \dim(F_k^+ \cap \mathbb{K}[x]_{d+k})$, which involves computing a basis of $F_k^+ \cap \mathbb{K}[x]_{d+k}$. Note that F_k^+ has bit size $(n+1)M$, and to compute the desired basis we perform Gaussian elimination on a matrix of size $(n+1)M$, which is polynomial time. If $\dim(F_k) \neq \dim(F_k^+ \cap \mathbb{K}[x]_{d+k})$,

then in Step 5, we set $F_k := F_k^+ \cap \mathbb{K}[x]_{d+k}$. We still have $F_k \subseteq \mathbb{K}[x]_d$, and F_k still has bit size M; thus, as above, Step 6 can be computed in polynomial time. The number of iterations of the `while` loop (Steps 4-7) is $O(n^d)$ since the $\dim(F_k)$ is at most $\dim(\mathbb{K}[x]_d) = O(n^d)$ and $\dim(F_k)$ increases each iteration of the loop. So, the loop terminates in polynomial time. Then, Step 8 involves computing a basis for $F_k \cap \mathbb{K}[x]_{d-1}$ using Gaussian elimination, which is polynomial time again. Lastly, Step 9 involves computing a basis of F_k^+ which has bit size $(n+1)M$ and thus polynomial time. The complexity of the first L iterations is thus bounded by L times the complexity of the Lth iteration, which is polynomial in L, n, $\log_2(|\mathbb{K}|)$ and the input size of the defining basis of F, and the result follows. □

REFERENCES

[1] E. Balas, S. Ceria, and G. Cornuéjols, *A lift-and-project cutting plane algorithm for mixed 0-1 programs*, Mathematical Programming, **58** (1993), pp. 295–324.

[2] J. Bochnak, M. Coste, and M.-F. Roy, *Real algebraic geometry*, Springer, 1998.

[3] M. Clegg, J. Edmonds, and R. Impagliazzo, *Using the Groebner basis algorithm to find proofs of unsatisfiability*, in STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, New York, NY, USA, 1996, ACM, pp. 174–183.

[4] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient algorithms for solving overdefined systems of multivariate polynomial equations*, in EUROCRYPT, 2000, pp. 392–407.

[5] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer Verlag, 1992.

[6] ———, *Using Algebraic Geometry*, Vol. **185** of Graduate Texts in Mathematics, Springer, 2nd ed., 2005.

[7] J. De Loera, C. Hillar, P. Malkin, and M. Omar, *Recognizing graph theoretic properties with polynomial ideals*. http://arxiv.org/abs/1002.4435, 2010.

[8] J. De Loera, J. Lee, P. Malkin, and S. Margulies, *Hilbert's Nullstellensatz and an algorithm for proving combinatorial infeasibility*, in Proceedings of the Twenty-first International Symposium on Symbolic and Algebraic Computation (ISSAC 2008), 2008.

[9] J. De Loera, J. Lee, S. Margulies, and S. Onn, *Expressing combinatorial optimization problems by systems of polynomial equations and the nullstellensatz*, to appear in the Journal of Combinatorics, Probability and Computing (2008).

[10] A. Dickenstein and I. Emiris, eds., *Solving Polynomial Equations: Foundations, Algorithms, and Applications*, Vol. **14** of Algorithms and Computation in Mathematics, Springer Verlag, Heidelberg, 2005.

[11] W. Eberly and M. Giesbrecht, *Efficient decomposition of associative algebras over finite fields*, Journal of Symbolic Computation, **29** (2000), pp. 441–458.

[12] A.V. Gelder, *Another look at graph coloring via propositional satisfiability*, Discrete Appl. Math., **156** (2008), pp. 230–243.

[13] E. Gilbert, *Random graphs*, Annals of Mathematical Statistics, **30** (1959), pp. 1141–1144.

[14] J. Gouveia, M. Laurent, P.A. Parrilo, and R.R. Thomas, *A new semidefinite programming relaxation for cycles in binary matroids and cuts in graphs*. http://arxiv.org/abs/0907.4518, 2009.

[15] J. Gouveia, P.A. Parrilo, and R.R. Thomas, *Theta bodies for polynomial ideals*, SIAM Journal on Optimization, **20** (2010), pp. 2097–2118.
[16] D. Grigoriev and N. Vorobjov, *Complexity of Nullstellensatz and Positivstellensatz proofs*, Annals of Pure and Applied Logic, **113** (2002), pp. 153–160.
[17] D. Henrion and J.-B. Lasserre, *GloptiPoly: Global optimization over polynomials with MATLAB and SeDuMi*, ACM Trans. Math. Softw., **29** (2003), pp. 165–194.
[18] ———, *Detecting global optimality and extracting solutions in GloptiPoly*, in Positive polynomials in control, Vol. **312** of Lecture Notes in Control and Inform. Sci., Springer, Berlin, 2005, pp. 293–310.
[19] T. Hogg and C. Williams, *The hardest constraint problems: a double phase transition*, Artif. Intell., **69** (1994), pp. 359–377.
[20] A. Kehrein and M. Kreuzer, *Characterizations of border bases*, Journal of Pure and Applied Algebra, **196** (2005), pp. 251 – 270.
[21] A. Kehrein, M. Kreuzer, and L. Robbiano, *An algebraist's view on border bases*, in Solving Polynomial Equations: Foundations, Algorithms, and Applications, A. Dickenstein and I. Emiris, eds., Vol. **14** of Algorithms and Computation in Mathematics, Springer Verlag, Heidelberg, 2005, ch. 4, pp. 160–202.
[22] J. Kollár, *Sharp effective Nullstellensatz*, Journal of the AMS, 1 (1988), pp. 963–975.
[23] J. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. on Optimization, **11** (2001), pp. 796–817.
[24] J. Lasserre, M. Laurent, and P. Rostalski, *Semidefinite characterization and computation of zero-dimensional real radical ideals*, Found. Comput. Math., **8** (2008), pp. 607–647.
[25] ———, *A unified approach to computing real and complex zeros of zero-dimensional ideals*, in Emerging Applications of Algebraic Geometry, M. Putinar and S. Sullivant, eds., vol. 149 of IMA Volumes in Mathematics and its Applications, Springer, 2009, pp. 125–155.
[26] J.B. Lasserre, *An explicit equivalent positive semidefinite program for nonlinear 0-1 programs*, SIAM J. on Optimization, **12** (2002), pp. 756–769.
[27] M. Laurent, *A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0–1 programming*, Math. Oper. Res., **28** (2003), pp. 470–496.
[28] ———, *Semidefinite relaxations for max-cut*, in The Sharpest Cut: The Impact of Manfred Padberg and His Work, M. Grötschel, ed., Vol. **4** of MPS-SIAM Series in Optimization, SIAM, 2004, pp. 257–290.
[29] ———, *Semidefinite representations for finite varieties*, Mathematical Programming, **109** (2007), pp. 1–26.
[30] ———, *Sums of squares, moment matrices and optimization over polynomials*, in Emerging Applications of Algebraic Geometry, M. Putinar and S. Sullivant, eds., Vol. **149** of IMA Volumes in Mathematics and its Applications, Springer, 2009, pp. 157–270.
[31] J. Löfberg, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.
[32] L. Lovász, *Stable sets and polynomials*, Discrete Math., **124** (1994), pp. 137–153.
[33] ———, *Semidefinite programs and combinatorial optimization*, in Recent advances in algorithms and combinatorics, B. Reed and C. Sales, eds., Vol. **11** of CMS Books in Mathematics, Spring, New York, 2003, pp. 137–194.
[34] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and 0-1 optimization*, SIAM J. Optim., **1** (1991), pp. 166–190.
[35] S. Margulies, *Computer Algebra, Combinatorics, and Complexity: Hilbert's Nullstellensatz and NP-Complete Problems*, PhD thesis, UC Davis, 2008.
[36] M. Marshall, *Positive polynomials and sums of squares.*, Mathematical Surveys and Monographs, **146**. Providence, RI: American Mathematical Society (AMS). xii, p. 187, 2008.

[37] B. Mourrain, *A new criterion for normal form algorithms*, in Proc. AAECC, Vol. **1719** of LNCS, Springer, 1999, pp. 430–443.
[38] B. Mourrain and P. Trébuchet, *Stable normal forms for polynomial system solving*, Theoretical Computer Science, **409** (2008), pp. 229 – 240. Symbolic-Numerical Computations.
[39] Y. Nesterov, *Squared functional systems and optimization problems*, in High Performance Optimization, J.F. et al., eds., ed., Kluwer Academic, 2000, pp. 405–440.
[40] P.A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, PhD thesis, California Institute of Technology, May 2000.
[41] ———, *Semidefinite programming relaxations for semialgebraic problems*, Mathematical Programming, **96** (2003), pp. 293–320.
[42] P.A. Parrilo and B. Sturmfels, *Minimizing polynomial functions*, in Proceedings of the DIMACS Workshop on Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science (March 2001), S. Basu and L. Gonzalez-Vega, eds., American Mathematical Society, Providence RI, 2003, pp. 83–100.
[43] S. Prajna, A. Papachristodoulou, P. Seiler, and P.A. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2004.
[44] G. Reid and L. Zhi, *Solving polynomial systems via symbolic-numeric reduction to geometric involutive form*, Journal of Symbolic Computation, **44** (2009), pp. 280–291.
[45] S. Roman, *Advanced Linear Algebra*, Vol. **135** of Graduate Texts in Mathematics, Springer New York, third ed., 2008.
[46] A. Schrijver, *Theory of linear and integer programming*, Wiley, 1986.
[47] H. Sherali and W. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM Journal on Discrete Mathematics, **3** (1990), pp. 411–430.
[48] N.Z. Shor, *Class of global minimum bounds of polynomial functions*, Cybernetics, **23** (1987), pp. 731–734.
[49] G. Stengle, *A Nullstellensatz and a Positivstellensatz in semialgebraic geometry*, Mathematische Annalen, **207** (1973), pp. 87–97.
[50] H. Stetter, *Numerical Polynomial Algebra*, SIAM, 2004.
[51] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Review, **38** (1996), pp. 49–95.
[52] L. Zhang, *zchaff v2007.3.12*. Available at http://www.princeton.edu/∼chaff/zchaff.html, 2007.