Iterative Algebraic Algorithms for the Recognition of Combinatorial Properties

J.A. De Loera, C. Hillar[*], P.N. Malkin, M. Omar [†]

November 23, 2009

Abstract

Many combinatorial optimization problems can be modeled concisely with a system of polynomial equations. Examples include the detection of k-colorings, stable sets, flows, matchings, and satisfiability (see [12] and the references therein). It follows that solving general systems of polynomial equations is at least NP-hard. For this reason, mathematicians have rarely used nonlinear polynomials for practical computation or to provide complexity bounds (although they can be very useful otherwise [1, 10, 30, 18]).

In this article, we discuss four iterative algorithms tailored to solve *combinatorial* systems of polynomial equations. We explain how these algebraic procedures can be applied to integer hull approximation and also the recognition of combinatorial properties such as k-colorability, unique Hamiltonicity, and automorphism rigidity of graphs. We report on computational complexity bounds, structural results, and computer experiments.

When the field of coefficients is the real numbers our methodology closely resembles other iterative procedures such as Lovász-Schrijver, Sherali-Adams, the Lasserre hierarchy, and others that are used in integer programming and optimization over semialgebraic sets [31, 38, 35, 28]. The algorithms we present are also related to the solvability methods of Laurent, Lasserre and Rostalski [25, 26]. The key difference is that we work over arbitrary fields of coefficients which allows a wider range of modeling.

[*] The second author is partially supported by an NSA Young Investigators Grant and an NSF All-Institutes Postdoctoral Fellowship administered by the Mathematical Sciences Research Institute through its core grant DMS-0441170.

[†] All other authors are partially supported by partially supported by NSF grant DMS-0914107, an IBM OCR award, and a NSERC doctoral fellowship.

1 Introduction

We discuss four iterative algebraic algorithms and explain how they can be used in recognizing combinatorial properties (e.g., k-colorability of graphs) and for the approximation of the integer hull of polyhedra. The general method we propose is as follows. Given a combinatorial question, we associate to it a system of polynomial equations J such that the combinatorial problem is infeasible if and only if system J has no solution. These highly structured systems of equations are then solved using adhoc algebraic tools.

The first algorithm **NulLA** was investigated in [12, 32] and developed further in [11]. NulLA generates a finite sequence of large-scale linear algebra problems, each of which is polynomial-time computable. Hilbert's Nullstellensatz [8] states that a system of polynomial equations $J = \{f_1(x) = 0, \ldots, f_r(x) = 0\}$ with coefficients over a field $\mathbb{K}$ has no solution over its algebraic closure $\overline{\mathbb{K}}$ if and only if there exist polynomial "witnesses" $\alpha_1, \ldots, \alpha_r \in \mathbb{K}[x_1, \ldots, x_n]$ such that $1 = \sum \alpha_i f_i$ (Note that the Nullstellensatz is an extension of Farkas' lemma from Linear Programming to arbitrary *polynomial* systems of equations). Thus, if the system J encodes a combinatorial problem and has no solutions, there exists a *Nullstellensatz certificate* $(\alpha_1, \ldots, \alpha_r)$ that the associated combinatorial problem is infeasible. The maximum degree D of polynomials in such a certificate is called the *Nullstellensatz degree* or *NulLA degree*. Finding a Nullstellensatz certificate of degree D is equivalent to solving a linear system whose variables are bounded by the number of monomials of degree D. For fixed D, the feasibility of this linear system is polynomial-time verifiable.

In practice, D is unknown (although bounded theoretically in [23] and with much smaller bounds for combinatorial systems in [12]), and this lack of control in degree growth is NulLA's main difficulty. Our new contribution is an algebraic adaptation of NulLA called **FPNulLA** which partly remedies this problem. FPNulLA also yields a hierarchy of polynomial-time computable relaxations that terminates with a decision, but unlike NulLA is a primal-dual algorithm. The key new idea is to keep track of a redundant encoding of J, which is adjusted before increasing the degree D. The method is a simplified version of Border bases in polynomial commutative algebra [22]. We revisit the NulLA algorithm and present the improved FPNulLA in Section 2.

The third method we consider is the well-known **Gröbner bases algorithm**, and we show how to use it to characterize uniqueness of certain graph properties algebraically. The theoretical findings can then be combined with algorithms FPNulLA and NulLA to give a relaxation scheme checking for this uniqueness. Finally, the fourth procedure we explore is due to Gouveia, Parrilo and Thomas [17] and is called the **theta body algorithm**. This algorithm uses a generalization of the Lovász theta body for 0/1 polyhedra to generate a sequence of semidefinite programming relaxations computing the integer hull of the zeroes of a set of real polynomials [31, 30]. Here one makes strong use of the hypothesis that the polynomials and their solutions are real.

We study three classical graph problems with these tools. First, in Section 3, we explore k-colorability with NulLA and FPNulLA using a well-known polynomial formulation [2].

Proposition 1.1. *Let $G = (V, E)$ be an undirected simple graph on n vertices. Fix a positive integer k, and let $\mathbb{K}$ be a field with characteristic relatively prime to k. The polynomial system $J = \{x_i^k - 1 = 0,\ x_i^{k-1} + x_i^{k-2}x_j + \cdots + x_j^{k-1} = 0 :\ i \in V,\ (i, j) \in E\}$ has a common zero over $\overline{\mathbb{K}}$ if and only if the graph G is k-colorable.*

If a system has a small NulLA degree certificate, then it is easy to find. In this regard, we characterize in Theorem 3.1 when the (infeasible) combinatorial system for 3-colorability has NulLA degree one. We then present a number of computational experiments using FPNulLA that demonstrate the practical power of these ideas.

Second, as an application of Gröbner bases, we investigate (in Section 4) the detection of Hamiltonian cycles of a digraph G. The following ideals algebraically encode Hamiltonian cycles.

Proposition 1.2. *Let $G = (V, A)$ be a simple digraph on n vertices. Assume that char$(\mathbb{K}) \nmid n$*

and that $\omega \in \mathbb{K}$ is a primitive nth root of unity. Consider the following system in $\mathbb{K}[x_1, \ldots, x_n]$:

$$x_i^n - 1 = 0 \;\; \textit{for} \;\; i \in V \quad \textit{and} \quad \prod_{j \in Adj(i)} (\omega x_i - x_j) = 0 \;\; \textit{for} \;\; i \in V.$$

Here, Adj(i) denotes those vertices j which are connected to i by a directed edge going from i to j. Then G has a Hamiltonian cycle if and only if this system has a solution over $\overline{\mathbb{K}}$.

We can prove a decomposition theorem for the ideal generated by the above polynomials, and based on this structure, we give an algebraic characterization of uniquely Hamiltonian graphs (as was done for k-colorability [19]). Our results also provide a algorithm to decide this property. These findings are related to a well-known theorem of Smith [41] which states that if a 3-regular graph has a Hamiltonian cycle then it has at least three. It is still an open question to decide the complexity of finding a second Hamiltonian cycle knowing that it must exist [3].

Third, we study (in Section 5) the problem of determining the automorphisms of a simple graph G, and in particular, when graphs are rigid (i.e., $Aut(G) = 1$). The complexity of this latter decision problem is still open. Our approach is the point of view of polyhedra, theta bodies, and semidefinite programming, but again, *linearization* plays an important role in our anlysis. As before, the combinatorial object $Aut(G) \subseteq \mathbb{R}^{n \times n}$ is viewed as an algebraic variety.

Proposition 1.3. *Let G be a graph and A_G its adjacency matrix. Then $Aut(G)$ is the real variety determined by the ideal $I(G)$ generated from the equations:*

$$\begin{aligned} (PA_G - A_G P)_{i,j} = 0, \;\; 1 \le i,j \le n; \quad \sum_{i=1}^{n} P_{i,j} = 1, \;\; 1 \le j \le n; \\ \sum_{j=1}^{n} P_{i,j} = 1, \;\; 1 \le i \le n; \quad P_{i,j}^2 - P_{i,j} = 0, \;\; 1 \le i,j \le n. \end{aligned} \tag{1}$$

From Proposition 1.3, the group $Aut(G)$ consists of the integer vertices of the polytope of doubly stochastic matrices commuting with A_G. By replacing the equations $P_{i,j}^2 - P_{i,j} = 0$ in (1) with the linear inequalities $P_{ij} \ge 0$, we obtain a polyhedron P_G which is a linear relaxation of the automorphisms of the graph. We study this polytope and its integer hull. Tinhofer [40] already examined P_G and gave some conditions for it to be integral. Here we uncover more properties of P_G.

First, we prove that it is *quasi-integral*; that is, the graph induced by the integer points in the 1-skeleton of P_G is always connected. Second, from the theory presented in [17], one can use the ideal $I(G)$ to approximate the integer hull of P_G by a sequence of convex bodies, the so-called *theta bodies*, which are expressible as projections of semidefinite programs. These authors also give some applications of their techniques for stable sets [31] and max cut [17]. Our contribution is a study of the theta bodies of the variety of automorphisms of a graph. In particular, we give partial algebraic and combinatorial characterizations of graphs for which the first theta body is already equal to the polytope (in much the same way that perfect graphs have theta body one for their Lovász theta body).

In what follows, we assume the reader is familiar with the basic properties of polynomial ideals and commutative algebra as introduced in the undergraduate-level text [8] (a quick review can be found in Section 2 of [19]). For an introduction to the theory of solving systems of polynomial equations we refer the reader [9, 13]. We denote by R the polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$ with coefficients over a field $\mathbb{K}$. The *monomials* of R are the elements $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, $\alpha \in \mathbb{N}^n$. The degree of a monomial x^α is $|\alpha| := \sum_{i=1}^n \alpha_i$, and the *degree* of a polynomial $f = \sum_{\alpha \in \mathbb{N}^n} f_\alpha x^\alpha$ is the maximum degree of x^α where $f_\alpha \neq 0$ for $\alpha \in \mathbb{N}^n$. We write $\deg(F)$ for the maximum degree of a set of polynomials $F \subseteq R$.

Given a set $F \subseteq R$, the variety of F over $\mathbb{K}$, written $V_{\mathbb{K}}(F)$, is the set of common zeros of polynomials in F that are in $\mathbb{K}^n$; that is, $V_{\mathbb{K}}(F) := \{v \in \mathbb{K}^n : f(v) = 0 \; \forall f \in I\}$. We shall write $\overline{\mathbb{K}}$ for the algebraic closure of $\mathbb{K}$. Given a set of polynomials $F := \{f_1, \ldots, f_m\} \subset R$, we define the *ideal* $I(F) := \langle f_1, \ldots, f_m \rangle_R := \{\sum_{i=1}^m \beta_i f_i \;:\; \beta_1, \ldots, \beta_m \in R\}$. Note that $V_{\mathbb{K}}(F)$ is the same as $V_{\mathbb{K}}(I)$. In all our applications, $\mathbb{K}$ will be the finite field $\mathbb{F}_2$, $\mathbb{R}$, or their algebraic closures.

2 A Primal-Dual Algorithm from Hilbert's Nullstellensatz

In this section, we describe two algorithms based on linear algebra that decide whether a set of *combinatorial* polynomials $F = \{f_1, \ldots, f_m\} \subseteq R$ has a zero. We shall abbreviate the system of equations $\{f(x) = 0 : f \in F\}$ as $F(x) = 0$, and for simplicity, we assume that $\mathbb{K} = \overline{\mathbb{K}}$. The first method discussed is NulLA, which was introduced in [12, 32, 11]. The second is the faster and more practical algorithm FPNulLA, which is our new contribution to this circle of ideas.

Let us remark that we are not the only proponents of the use of *linearization* as a way to solve polynomial systems. Indeed, FPNulLA follows in the spirit of Border bases in commutative algebra [21, 34, 39] and the use of linear algebra speed-ups in the Gröbner bases algorithms of Faugere [15]. Variants of NulLa were applied by authors to problems in logic and complexity [6], cryptography [7], combinatorial optimization [**?**], and recently in mathematical programming to derive semidefinite relaxations for combinatorial optimization problems [29, 24, 27, 35, 36]. To our knowledge, however, we are the first to carry out extensive computational experiments and to derive explicit combinatorial theorems using these techniques.

We begin by explaining the primal-dual relationship between linear and polynomial algebra. The polynomial ring $R = \mathbb{K}[x_1, \ldots, x_n]$ is an infinite dimensional vector space over $\mathbb{K}$ with basis given by all of the monomials of R. This vector space consists of infinite sequences of elements in $\mathbb{K}$ (indexed by monomials) having only finitely many nonzero entries. An ideal $I \subseteq R$ is a vector subspace of R, and the quotient ring R/I is a vector space quotient. Given $F \subset R$, we let $\langle F \rangle_{\mathbb{K}}$ denote the $\mathbb{K}$-vector space generated by F over $\mathbb{K}$. Let $R^* := \mathbb{K}[[x_1, \ldots, x_n]]$ be the ring of formal power series in the variables $x_1, \ldots, x_n$ with coefficients in $\mathbb{K}$. We can consider R^* as the vector space consisting of all infinite sequences in $\mathbb{K}$ indexed by monomials. We now define a bilinear form $* : R \times R^* \to \mathbb{K}$ as follows: given $f = \sum_{\alpha \in \mathbb{N}^n} f_\alpha x^\alpha \in R$ and $\lambda = \sum_{\alpha \in \mathbb{N}^n} \lambda_\alpha x^\alpha \in R^*$, we have $f * \lambda := \sum_{\alpha \in \mathbb{N}^n} f_\alpha \lambda_\alpha$, which is well-defined (only finitely many f_α are nonzero).

A relaxation of $F(x) = 0$ can be defined as the set of linear equations $\{f * \lambda = 0 : f \in F\}$. We abbreviate this system as $F * \lambda = 0$. Note that for any polynomial $f \in R$ and any point $x \in \mathbb{K}^n$, we have $f(x) = f * \lambda(x)$ where $\lambda(x) = (x^\alpha)_{\alpha \in \mathbb{N}^n}$. It follows that for any $x \in \mathbb{K}^n$, we have $F(x) = 0$ if and only if $F * \lambda(x) = 0$. The system $F * \lambda = 0$ is always feasible, but the constraint $\lambda_0 = 1$ also holds for any λ that corresponds to a solution $F(x) = 0$. Therefore, if the inhomogeneous linear system $\{F * \lambda = 0,\ \lambda_0 = 1\}$ is infeasible, then so is the system of polynomials $F(x) = 0$. Generating the linear system $\{F * \lambda = 0, \lambda_0 = 1\}$ from $F(x) = 0$ is often referred to as *linearization*. Of course, some solutions of $\{F * \lambda = 0, \lambda_0 = 1\}$ may not correspond to solutions of $F(x) = 0$ (the multiplicative structure of R is lost); however, it is a basic fact that $F(x) = 0$ is infeasible if and only if $\{I(F) * \lambda = 0,\ \lambda_0 = 1\}$ has no solution. We denote the set of solutions of the linear system $F * \lambda = 0$ as $F^\circ := \{\lambda \in R^* : F * \lambda = 0\}$, called the annihilator of F, which is a vector subspace of R^*.

The dual of the linear system $\{F * \lambda = 0, \lambda_0 = 1\}$ has the following nice interpretation. Consider the following trivial implication of Hilbert's Nullstellensatz: If there are *constants* $\mu \in \mathbb{K}^m$ such that $\sum_{i=1}^m \mu_i f_i = 1$ (i.e., $1 \in \langle F \rangle_{\mathbb{K}}$), then the polynomial system $F(x) = 0$ is not feasible. Determining whether such a μ exists means solving the linear system of equations $\{\sum_{i=1}^m \mu_i f_{i,0} = 1$ and $\sum_{i=1}^m \mu_i f_{i,\alpha} = 0\ \forall \alpha \in \mathbb{N}^n, \alpha \neq 0\}$ for $\mu \in \mathbb{K}^m$. We abbreviate this linear system of equations as $\mu^T F = 1$. Crucially, $\mu^T F = 1$ is the dual linear system to $\{F * \lambda = 0,$ $\lambda_0 = 1\}$. Thus, the infeasibility of $\{F * \lambda = 0,\ \lambda_0 = 1\}$ is the same as the feasibility of $\mu^T F = 1$.

A fundamental observation is that *adding redundant polynomial equations* to $F(x) = 0$ gives a *tighter* linear relaxation, meaning that for sets $F \subseteq \tilde{F} \subseteq I$, we have $\tilde{F}^\circ \subseteq F^\circ$ with $F^\circ = \tilde{F}^\circ$ if and only if $\langle F \rangle_{\mathbb{K}} = \langle \tilde{F} \rangle_{\mathbb{K}}$. In fact, 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 [9]:

Theorem 2.1. *Let $\mathbb{K}$ be an algebraically closed field, and let $I \subseteq R$ be a zero-dimensional ideal. Then,* $\dim(I^\circ)$ *is finite and* $\dim(I^\circ)$ *is the number of solutions of the polynomial system $I(x) = 0$ including multiplicities. In particular, $|V_{\mathbb{K}}(I)| \leq \dim(I^\circ)$ with equality when I is radical.*

Thus, computing $\dim(I^\circ)$ allows us to determine the feasibility of $F(x) = 0$ over $\mathbb{K}$. Unfortunately, we cannot calculate $\dim(I^\circ)$ directly. Instead, under some conditions (see Theorem

2.2), we find $\dim(I^\circ)$ by computing the dimension of the projection of F° onto the variables λ_{x^α} with $\deg(x^\alpha) \leq \deg(F)$.

2.1 Nullstellensatz Linear Algebra Algorithm (NulLA)

We now present an algorithm, NulLA, for determining whether $F(x) = 0$ has a solution over $\mathbb{K}$ using linear relaxations. The idea behind NulLA [11] is straightforward: we check whether $F * \lambda = 0$, $\lambda_1 = 1$ is infeasible or equivalently whether $\mu^T F = 1$ is feasible (i.e., $1 \in \langle F \rangle_{\mathbb{K}}$) using linear algebra over $\mathbb{K}$; if not, we add polynomials from $\langle F \rangle_R$ 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.

In the following, we assume without loss of generality that F is closed under $\mathbb{K}$-linear combinations; that is $F = \langle F \rangle_{\mathbb{K}}$, 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 $F * \lambda = 0$. For computation, we need a vector space basis of F, but the choice of basis is not important, and moreover, we find it more natural and expositionally convenient to use vector spaces. Recall from above that $F * \lambda = 0, \lambda_1 = 1$ is infeasible if and only if $1 \in \langle F \rangle_{\mathbb{K}}$, which when F is a vector space, simplifies to $1 \in F$.

For a vector space $F \subset R$, 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 R: it is the linear span of F and $x_i F$ for all $i = 1, \ldots, n$. The NulLA algorithm for vector spaces works as follows (see Algorithm 1): if $1 \in F$, then $F(x) = 0$ is infeasible and stop, otherwise set $F := F^+$ and repeat. Note that after k iterations of NulLA, the set F contains all linear combinations of polynomials of the form $x^\alpha f$ where the total degree $|\alpha| \leq k$ and where f was one of the initial polynomials in F. 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 $F(x) = 0$. This is an upper bound on the Nullstellensatz degree of the polynomial system (see [23] for worse-case bounds; better bounds exist for combinatorial systems [12]). However, while theoretically the Nullstellensatz bound limits the number of iterations, this bound is in general too large to be practically useful (see [11] and references therein). In practice, NulLA is most useful for proving infeasibility (see Section 3).

Algorithm 1 NulLA Algorithm [11]

```
Input: A finite dimensional vector space $F \subseteq R$ and a Nullstellensatz bound $D$.
Output: FEASIBLE, if $F(x) = 0$ is feasible over $\overline{\mathbb{K}}$, else INFEASIBLE.
  for $d = 0, 1, 2, \ldots, D$ do
    If $1 \in F$, then return INFEASIBLE.
    $F := F^+ \quad (F + \sum_{i=1}^n x_i F)$.
  end for
  return FEASIBLE.
```

2.2 Fixed Point Nullstellensatz Linear Algebra Algorithm (FPNulLA)

Next, we discuss improving NulLA by adding redundant polynomials to F in such a way so that $\deg(F)$ does not grow unnecessarily. The improved algorithm is called the Fixed-Point Nullstellensatz Linear Algebra (FPNulLA) algorithm. 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 if there are any new polynomials in F^+ with degree at most $\deg(F)$ that were not in F. If this is the case, we add them to F, and then check again whether $1 \notin F$. More formally, if $1 \notin F$, we replace F with $F^+ \cap R_d$ where R_d is the set of all polynomials with degree at most $d = \deg(F)$. We keep replacing F with $F^+ \cap R_d$ until either $1 \in F$ or we reach a *fixed point*, $F = F^+ \cap R_d$. This process must terminate.

FPNulLA improves NulLA by proving that the system $F(x) = 0$ is feasible well before reaching the Nullstellensatz's degree bound as follows. When $1 \notin F$ and $F = F^+ \cap R_d$, then we can use the following theorem to determine if $F(x) = 0$ is feasible. First, we introduce some notation. Let $\pi_d : R^* \to R_d$ be the 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 $\dim_d(F^\circ)$.

Theorem 2.2. *Let $F \subset R$ be a finite dimensional vector space and let $d = \deg(F)$. If $F = F^+ \cap R_d$ and $\dim_d(F^\circ) = \dim_{d-1}(F^\circ)$, then $\dim(I^\circ) = \dim_d(F^\circ)$ where $I = \langle F \rangle_R$.*

There are many equivalent forms of Theorem 2.2 that appear in the literature [33, 37, 26]. For example, the above result can also be re-stated as follows: If $F = F^+ \cap R_d$ and $\dim(R_d/F) = \dim(R_{d-1}/F)$, then $\dim(R/I) = \dim(R_d/F)$. This follows since $\dim(I^\circ) = \dim(R/I)$, and $\dim_d(F^\circ) = \dim(R_d/F)$, and $\dim(R_{d-1}/F) = \dim_{d-1}(F^\circ)$ (see for example [39]). Note that the condition $F = F^+ \cap R_d$ is equivalent to $\dim_d(F^\circ) = \dim_d((F^+)^\circ)$ or equivalent to $\dim(R_d/F) = \dim(R_d/F^+)$ since $\dim(R_d/F^+) = \dim_d((F^+)^\circ)$. So, in practice, checking the conditions of Theorem 2.2 means computing the dimensions of vector spaces (i.e., ranks of matrices).

We can now present the FPNulLA algorithm. The FPNulLA algorithm is closely related to Algorithm 4.3 of [33] although it is much simpler. In fact, also Border and Buchberger algorithms for computing a basis of an ideal are very similar to FPNulLA. The main difference is that FPNulLA does not require a term ordering, special order ideals, nor does it keep track of the explicit vector space bases and ranks computed in each iteration. We refer the reader to Stetter [39] for a detailed comparison between Border and Buchberger methods.

Algorithm 2 FPNulLA Algorithm

Input: A vector space $F \subset R$ (typically generated by a finite set of equations).
Output: The number of solutions of $F(x) = 0$ over $\overline{\mathbb{K}}$ up to multiplicities.

```
Let $d = \deg(F)$.
loop
  if $1 \in F$ then Return 0.
  while $F \neq F^+ \cap R_d$ do
    Set $F := F^+ \cap R_d$.
    if $1 \in F$ then return 0.
  end while
  if $\dim_d(F^\circ) = \dim_{d-1}(F^\circ)$ then return $\dim_d(F^\circ)$.
  $F := F^+ \quad (F + \sum_{i=1}^n x_i F)$.
  $d := d + 1$ (Note $d = \deg(F)$ at each iteration).
end loop
```

The proof of the main result and the termination of the algorithm are omitted here. Our result (and its proof) can be seen as an adaptation and simplification of Theorem 4.2 and Algorithm 4.3 result in [33]. The main difference being, in Mourrain's terminology, that we stick to a particular *order ideal* and do not need to keep track explicitly of a basis for the set B in its arguments, only vector space dimensions. From a complexity point of view, it is relevant to know the following:

Corollary 2.3. *Let D be a fixed nonnegative integer. Let $F = \{f_1, \ldots, f_m\}$ be n-variate polynomials in R. Then*

1. *The D-th iteration of NulLA algorithm can be computed in polynomial time in the input size of F and the number of variables.*
2. *Denote by FP_D the vector space generated by F after incrementing the counter index variable d to be $d = D$. If we assume $\mathbb{K}$ is a finite field, then a vector space basis for FP_D is generated in the FPNulLA algorithm in polynomial time in the input data and the size of the field.*

Finally let us comment that FPNulLA only decides when a solution (e.g., such as a 3-coloring) exists, but it does not tell us how to find one! Through further calculations, one can find such solutions explicitly; however, such a discussion is beyond the scope of this paper, and we refer the reader to [9]. Now, we begin our combinatorial applications of NulLA and FPNulLA.

3 Recognizing non-3-colorable graphs

Given a fixed degree, one would like to characterize those graphs which can be proved to have a certain property at that Nullstellensatz degree. These are graphs that can be recognized in polynomial time. In this section, we state a combinatorial characterization of those graphs that have NulLA degree of one. In [32] it was shown that the NulLA degree for a polynomial encoding over $\mathbb{F}_2$ of the 3-colorability of a graph with n vertices with no 3-coloring is between one and $2n$. Moreover, if a non-3-colorable graph contains an odd-wheel or a 4-clique, its NulLA degree is exactly one. We also present evidence that this degree is a good measure of the difficulty of (3-coloring) infeasibility testing.

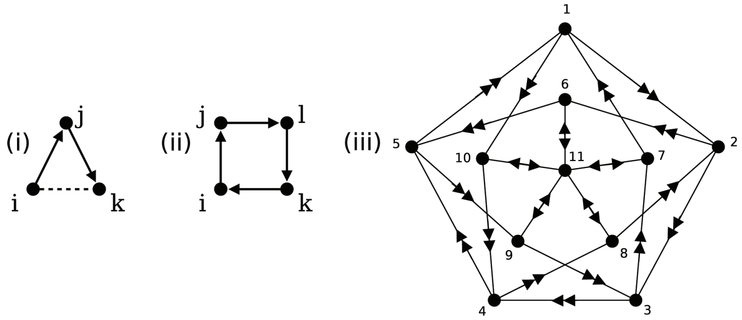

Figure 1: (i) partial 3-cycle, (ii) chordless 4-cycle, and (iii) the Grötzsch graph.

Let A be the set of all possible directed edges (or *arcs*) in an undirected graph G. We are interested in two types of substructures of the graph G (see Figure 1). An *oriented partial-3-cycle* is a set of two arcs of a 3-cycle, and an *oriented chordless 4-cycle* is a set of four arcs $\{(i,j),(j,l),(l,k),(k,i)\}$, denoted (i,j,k,l), with $(j,k),(i,l) \notin A$. The parity conditions in the following theorem are reminiscent of those encountered in proofs of Sperner's Lemma [**?**].

Theorem 3.1. *A graph G has non-3-colorability NulLA degree one if and only 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 in which the arc $(i,j) \in A$ appears. Moreover, such graphs can be recognized in polynomial time.

Condition 1 in Theorem 3.1 means that every undirected edge of G is covered by an even number of directed edges from cycles in C (ignoring orientation). On the other hand, Condition 2 says that given any orientation of G (an assignment of directions to its edges), the total number of times the arcs in that orientation appear in the cycles of C is odd. The particular orientation $\{(i,j) \in A : i < j\}$ we use in the theorem is irrelevant.

There are two possible orientations for every partial 3-cycle and every chordless 4-cycle, but the conditions in Theorem 3.1 are invariant under changing the orientation of any element in the set C. We can thus treat each partial 3-cycle and 4-cycle as unique. Also, note that C gives an edge-covering by 3-cycles and 4-cycles of a non-3-colorable subgraph of G if we include the missing edges of the partial 3-cycles.

Example 3.2. *Consider the Grötzsch graph in Figure 1, which has no 3-coloring and no 3-cycles. The following set of oriented chordless 4-cycles gives a certificate of non-3-colorability by Theorem 3.1:* $C := \{(1,2,3,7), (2,3,4,8), (3,4,5,9), (4,5,1,10), (1,10,11,7), (2,6,11,8), (3,7,11,9), (4,8,11,10), (5,9,11,6)\}$*. Figure 1 (iii) illustrates the edge directions for the 4-cycles of* C*. Each edge of the graph is contained in exactly two 4-cycles, so* C *satisfies Condition 1 of Theorem 3.1. Moreover, one can check that* $\sum_{(i,j)\in A,\ i<j} |C_{(i,j)}| \equiv 1 \pmod 2$*, and so Condition 2 is satisfied. It follows that the graph has no proper 3-coloring.*

We next summarize our experimental results for graph 3-coloring, illustrating the practical performance of the NulLA and FPNulLA algorithms. For more detailed results along these lines, see [11, 32]. Experimentally, for graph 3-coloring, NulLA and FPNulLA are well-suited to proving infeasibility. The polynomial encoding used here is over $\mathbb{F}_2$ (see Proposition 1.1) and thus the linear algebra operations are very fast. However, even though theoretically NulLA and FPNulLA can determine feasibility, in the experiments described below NulLA and FPNulLA were not able to prove feasibility in practice.

We are interested in the percentage of randomly generated graphs whose polynomial system encoding has a NulLA degree of one or a FPNulLA degree of one. The $G(n, p)$ model [16] is used for generating random graphs with n vertices and edges appearing with probability p. Without loss of generality, the color of one of the vertices of each randomly generated graph was fixed giving a slightly smaller polynomial encoding.

Our experimental results are presented in Figure 2, which plots the percentage of 1000 random graphs in $G(100, p)$ that were proven infeasible with a NulLA degree of one, with a FPNulLA degree 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 [42] and then use the program `zchaff` [43] to determine satisfiability.

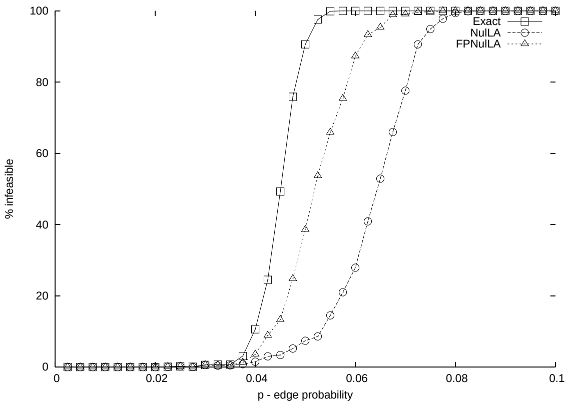

Figure 2: Non-3-colorable graphs with NulLA or FPNulLA degree of 1

It is well-known that there is a distinctive 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 [20]. 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 (but clearly dominates the one for NulLA). These results support the statement that the NulLA degree or FPNulLA degree is a reasonable measure of the hardness of proving infeasibility since those graphs requiring a higher degree than one are located near the phase transition.

4 Recognizing Uniquely Hamiltonian Graphs

Throughout this section we work over an arbitrary algebraically closed field $\mathbb{K}$, although in some cases, we will need to restrict its characteristic. Let us denote by H_G the *Hamiltonian ideal* generated by the polynomials from Proposition 1.2. For a connected digraph G with n vertices, it has a Hamiltonian cycle if and only if the equations defined by H_G have a solution over $\mathbb{K}$ (or, in other words, if and only if $V(H_G) \neq \emptyset$). In a precise sense to be made clear below, the ideal H_G encodes all Hamiltonian cycles of G. However, we need to be somewhat careful about how to count cycles (see Lemma 4.6). In practice ω can be treated as a variable and not as a fixed primitive nth root of unity. For example, if the equation $\omega^n - 1 = 0$, and the set of equations $y_k(\omega^k - 1) = 0$, for all k dividing n, is added to the defining system, then ω simply becomes a variable which can only take on the value of a *primitive* nth root of unity, even if n is not a prime number. Another set of equations ensuring that ω only takes on the value of a *primitive* nth root of unity is the following: $\omega^{k(n-1)} + \omega^{k(n-2)} + \cdots + \omega^k + 1 \ , \quad \text{for } 1 \leq k \leq n$. We can also use the cyclotomic polynomial $\Phi_n(x)$ [14], which is the polynomial whose zeroes are the primitive nth roots of unity.

In this section, we utilize the theory of Gröbner bases to show that H_G has a special (algebraic) decomposition structure in terms of the different Hamiltonian cycles. In the particular case when G has a unique Hamiltonian cycle, we get a specific algebraic criterion which can be algorithmically verified. These results are Hamiltonian analogues to the algebraic k-colorability characterizations of [19]. We first turn our attention more generally to cycle ideals of a simple directed graph G. These will be the basic elements in our decomposition of the Hamiltonian ideal H_G, as they algebraically encode single Hamiltonian cycles C.

When G has the property that each pair of vertices connected by an edge is also connected by an edge in the opposite direction, then we call G *undirected.* Let C be a cycle of length $k > 2$ in G, expressed as a sequence of directed edges, $C = \{(v_1, v_2), (v_2, v_3), \ldots, (v_k, v_1)\}$. We call C an *undirected cycle* if consecutive vertices in the cycle are connected by edges in both directions; otherwise, C is called *directed.* In particular, each cycle in an undirected graph is undirected.

Definition 4.1 (Cycle encodings)**.** *The cycle encoding of an undirected cycle C is the following set of k polynomials in $\mathbb{K}[x_{v_1}, \ldots, x_{v_k}]$:*

$$g_i = \begin{cases} x_{v_i} + \frac{(\omega^{2+i} - \omega^{2-i})}{(\omega^3 - \omega)} x_{v_{k-1}} + \frac{(\omega^{1-i} - \omega^{3+i})}{(\omega^3 - \omega)} x_{v_k} & i = 1, \ldots, k-2, \\ (x_{v_{k-1}} - \omega x_{v_k})(x_{v_{k-1}} - \omega^{-1} x_{v_k}) & i = k-1, \\ x_{v_k}^k - 1 & i = k, \end{cases} \tag{2}$$

in which ω is a fixed primitive kth root of unity and $\mathbb{K}$ has characteristic not dividing k. The cycle encoding of a directed cycle C is the following set of k polynomials:

$$g_i = \begin{cases} x_{v_{k-i}} - \omega^{k-i} x_{v_k} & i = 1, \ldots, k-1, \\ x_{v_k}^k - 1 & i = k. \end{cases} \tag{3}$$

Definition 4.2. *Correspondingly, we define the cycle ideal associated to C to be $I_{G,C} = \langle g_i : i = 1, \ldots, k \rangle \subseteq \mathbb{K}[x_{v_1}, \ldots, x_{v_k}]$, where the g_is are the cycle encoding of C given by (2) or (3).*

The polynomials g_i are computationally useful generators for cycle ideals. (Once again, see Section 2 in [19] for the relevant background on Gröbner bases and term orders.)

Lemma 4.3. *The cycle encoding polynomials $F = \{g_1, \ldots, g_k\}$ are a reduced Gröbner basis for the cycle ideal $I_{G,C}$ with respect to any term order $\prec$ with $x_{v_k} \prec \cdots \prec x_{v_1}$.*

Remark 4.4. *In particular, since reduced Gröbner bases (with respect to a fixed term order) are unique, it follows that cycle encodings can be seen as canonical ways of generating cycle ideals (and thus of representing cycles by Lemma 4.5).*

The naming of these objects is motivated by the following result; in words, it says that the cycle C is encoded as a complete intersection by the ideal $I_{G,C}$.

Lemma 4.5. *The following hold for the ideal $I_{G,C}$.*

1. *$I_{G,C}$ is radical,*
2. *$|V(I_{G,C})| = k$ if C is directed, and $|V(I_{G,C})| = 2k$ if C is undirected.*

Before stating our decomposition theorem, we need to explain how the Hamiltonian ideal encodes all Hamiltonian cycles of the graph G. For each cycle C, we assign a multiplicity $m(C)$, which is $2k$ for undirected cycles and k otherwise. These multiplicities naturally correspond to the symmetries of cycles.

Lemma 4.6. *Let G be a connected directed graph on n vertices. Then, $V(H_G) = \bigcup_C V(I_{G,C})$, where the union is over all Hamiltonian cycles C in G. In particular, $|V(H_G)| = \sum_C m(C)$.*

Combining all of these ideas, we can prove the following result.

Theorem 4.7. *Let G be a connected directed graph with n vertices. Then, $H_G = \bigcap_C I_{G,C}$, where C ranges over all Hamiltonian cycles of the graph G.*

Corollary 4.8. *The graph G is uniquely Hamiltonian if and only if the Hamiltonian ideal H_G is of the form $I_{G,C}$ for some length n cycle C.*

This corollary provides an algorithm to check whether a graph is uniquely Hamiltonian. We simply compute a unique reduced Gröbner basis of H_G and then check that it has the same form as that of an ideal $I_{G,C}$. More importantly, this algebraic characterization can be used in conjunction with NulLA and FPNulLA to detect whether a graph is not uniquely Hamiltonian. One simply searches for a Nullstellensatz certificate using the polynomial generators of the full-cycle ideal $I_{G,C}$.

5 The Integer Hull of $Aut(G)$

We also give a description of the convex hull of the automorphisms of a graph $Aut(G)$. See [4, 5] for background material. Here, the elements of the group $Aut(G)$ are naturally represented as $|V(G)| \times |V(G)|$ permutation matrices; they are the *integer* vertices of the rational polytope P_G from the introduction. We are primarily interested in the integer vertices of the polytope P_G, and we investigate IP_G, the *integer hull* of P_G. In the fortunate case that P_G is already integral ($P_G = IP_G$), we say that the graph G is *compact*, a term coined by Tinhofer [40]. This occurs for example in the special case that G is an independent set on n vertices, P_G is the well-studied Birkhoff polytope, the convex hull of all doubly-stochastic matrices. One can therefore view P_G as a generalization of the Birkhoff polytope to general graphs (and $Aut(G)$ to groups other than S_n). The polytope P_G was first introduced by Tinhofer [40], Unfortunately, the polytope P_G is not always integral. For instance, P_G is not integral when G is the Petersen graph. We can prove:

Theorem 5.1. *The induced subgraph of the integer points of the 1-skeleton of P_G is connected, thus P_G is quasi-integral.*

Of course, we would like to find a tighter description of IP_G in terms of inequalities. We concentrate now on a hierarchy of semidefinite relaxations of $conv(V_{\mathbb{R}}(I(G)))$ that is afforded by an algebraic point of view. When these relaxations are tight, we obtain a description of P_G that allows us to optimize and determine feasibility efficiently via linear programming.

We begin with some preliminary definitions from [17] and motivated by Lovász & Schrijver [31]. Let $I \subset \mathbb{R}[x_1, \ldots, x_n]$ be a *real radical ideal* ($\sum_{i=1}^m f_i^2 \in I \Longrightarrow \sum_{i=1}^m f_i \in I$). A polynomial f is said to be *nonnegative* mod I (written $f \geq 0 \pmod{I}$) if $f(p) \geq 0$ for all $p \in V_{\mathbb{R}}(I)$. Similarly, a polynomial f is said to be a *sum of squares* mod I if there exist $h_1, \ldots, h_m \in \mathbb{R}[x_1, \ldots, x_n]$ such that $f - \sum_{i=1}^m h_i^2 \in I$. If the degrees of the $h_1, \ldots, h_m$ are bounded by some positive integer k, we say f is k-sos mod I. Then the *kth theta body* of I, denoted $TH_k(I)$, is the subset of $\mathbb{R}^n$ that is nonnegative on every function in I that is k-sos mod I. We say that a real variety $V_{\mathbb{R}}(I)$ is theta k-exact if $\overline{conv(V_{\mathbb{R}}(I))} = TH_k(I)$. Theta bodies can be expressed as feasible regions of semidefinite programs (i.e., spectrahedra). For more on this, see [17]. It follows that theta bodies provide a hierarchy of semidefinite relaxations of $\overline{conv(V_{\mathbb{R}}(I))}$:

$$TH_1(I) \supseteq TH_2(I) \supseteq \cdots \supseteq \overline{conv(V_{\mathbb{R}}(I))}.$$

Therefore, when $I = I(G)$ for some graph G and I is theta k-exact, optimization over automorphisms of G can be performed using semidefinite programming. It is interesting to find graphs G such that $I(G)$ is theta k-exact for some k. In this section we pay particular attention to finding graphs G such that $I(G)$ is 1-exact, which we refer to as *exact* from now on. The key to finding such graphs G comes from the following combinatorial characterization found in [17].

Theorem 5.2. *Let $V_{\mathbb{R}}(I) \subset \mathbb{R}^n$ be a finite real variety. Then $V_{\mathbb{R}}(I)$ is exact if and only if there is a finite linear inequality description of $conv(V_{\mathbb{R}}(I))$ such that for every inequality $g(x) \geq 0$, there is a hyperplane $g(x) = \alpha$ such that every point in $V_{\mathbb{R}}(I)$ lies either on the hyperplane $g(x) = 0$ or the hyperplane $g(x) = \alpha$.*

Using this theorem and Sullivant's result [**?**], we can show that any compact graph is exact and that the class of exact graphs properly extends the class of compact graphs. The following theorem extends a result of Tinhofer [40] that says that the union of isomorphic compact graphs is compact.

Theorem 5.3. *If G is a compact graph, then G is also exact. Let $G_1, \ldots, G_m$ be k-regular compact graphs, and let $G = \bigcup_{i=1}^{m} G_i$. Then G is compact if and only if $G_1 \cong \cdots \cong G_m$. Moreover, G is always exact. Thus, the class of exact graphs strictly contains the class of compact graphs.*

As we shall see next exactness can be verified using *toric ideals*. Let A be a permutation group. Consider A as a subset of $\mathbb{Z}^d$ $(d = |A|)$. Let $\mathbb{C}[x] := \mathbb{C}[x_{\sigma_1}, x_{\sigma_2}, \ldots, x_{\sigma_d}]$ be the polynomial ring in d variables indexed by permutations $\sigma_i \in S_n$ corresponding to the permutations in A. Finally, let $\mathbb{C}[t] := \mathbb{C}[t_{11}, t_{12}, \ldots, t_{1n}, t_{21}, \ldots, t_{2n}, \ldots, t_{nn}]$. The semigroup homomorphism $\pi : \mathbb{N}^{n \times n} \to \mathbb{Z}^{n \times n}$ defined by $\pi(\sigma_i) = P_{\sigma_i}$ induces an algebra homomorphism $\pi : \mathbb{C}[x] \to \mathbb{C}[t]$, $x_{\sigma_i} = t^{P_{\sigma_i}}$ whose kernel I_G is an ideal. It is well known that G is exact if and only if for every reverse lexicographic term ordering $\prec$ on $\mathbb{C}[x]$, the Gröbner bases initial ideal $in_{\prec}(I_G)$ is generated by square-free monomials (see, for instance, [**?**]). Now we show a family of groups that are exact: Let $A \leq S_n$ be represented by $n \times n$ permutation matrices. We say that A is *permutation summable* if for any permutations $P_1, \ldots, P_m \in A$ satisfying the inequality $\sum_{i=1}^{m} P_i - I \geq 0$, we have that $\sum_{i=1}^{m} P_i - I$ is also a sum of permutation matrices in A. For example, Birkhoff's Theorem implies S_n is permutation summable. Note in this case P_{S_n} is the Birkhoff polytope which is known to be exact. Nevertheless, we can prove the following:

Theorem 5.4. *Let $A \leq S_n$ be a permutation group.*

(1) If A is permutation summable, then A is exact.

(2) Suppose I_A, the toric ideal associated to A has quadratically generated Gröbner basis with respect to any of the reverse lexicographic orderings $\prec$, then A is permutation summable.

There are many groups that are permutation summable. Moreover, if $A_{n_1}, \ldots, A_{n_m}$ are permutation summable groups with A_{n_i} a subgroup of S_{n_i}, then $A = A_{n_1} \times \cdots \times A_{n_m}$ is permutation summable (simply apply the permutation summable condition on each group G_i and take direct sums). For our purposes, we would like to find permutation summable groups that arise as the automorphism groups of certain classes of graphs. An example of such a class is the class of graphs whose automorphism groups have no non-trivial elements that fix any vertices. We say a graph G is said to be *strongly fixed-point free* if for every $P \in Aut(G) \setminus \{1\}$, we have $Pv \neq v$ for any $v \in V(G)$. We can prove that if G is strongly fixed-point free, then $Aut(G)$ is permutation summable and thus exact. We can see that the group generated by any n cycle in S_n is permutation summable. Another example of strongly fixed-point free permutation group is any dihedral group of order $4n$ that is a subgroup of S_{2n} is permutation summable as well.

References

[1] N. Alon. Combinatorial nullstellensatz. *Combin. Probab. Comput.*, 8(1-2):7–29. Recent trends in combinatorics (Mátraháza, 1995).

[2] D.A. Bayer. *The Division Algorithm and the Hilbert Scheme*. PhD thesis, Harvard University, 1982.

[3] K. Cameron. Thomason's algorithm for finding a second hamiltonian circuit through a given edge in a cubic graph is exponential on krawczyk's graphs. *Discrete Math.*, 235(1-3):69–77. Combinatorics (Prague, 1998).

[4] P.J. Cameron. Automorphisms of graphs. In R.J. Wilson L.W. Beineke, editor, *Topics in Algebraic Graph Theory*, pages 203–221. Cambridge Univ. Press, 2004.

[5] A. Chan and C. Godsil. *Graph Symmetry: Algebraic Methods and Applications*, chapter 4, pages 75–106. Kluwer Academic Publishers, Montréal, QC, Canada., 1997.

[6] M. Clegg, J. Edmonds, and R. Impagliazzo. Using the groebner basis algorithm to find proofs of unsatisfiability. pages 174–183.

[7] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. pages 392–407.

[8] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms.* Undergraduate Texts in Mathematics. Springer, 3 edition. An introduction to computational algebraic geometry and commutative algebra.

[9] D.A. Cox, J. Little, and D. O'Shea. *Using algebraic geometry*, volume 185 of *Graduate Texts in Mathematics.* Springer, 2 edition.

[10] J. A. De Loera. Gröbner bases and graph colorings. *Beiträge Algebra Geom.*, 36(1):89–96.

[11] J.A. De Loera, J. Lee, P.N. 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.

[12] J.A. De Loera, J. Lee, S. Margulies, and S. Onn. Expressing combinatorial problems by systems of polynomial equations and hilbert's nullstellensatz. *Combin. Probab. Comput.*, 18(4):551–582.

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

[14] D.S. Dummit and R. M. Foote. *Abstract algebra.* John Wiley & Sons Inc., 3 edition.

[15] J. C. Faugére. A new efficient algorithm for computing gröbner bases (f_4). *J. Pure Appl. Algebra*, 139(1-3):61–88. Effective methods in algebraic geometry (Saint-Malo, 1998).

[16] E. N. Gilbert. Random graphs. *Ann. Math. Statist.*, 30:1141–1144.

[17] J. Gouveia, P. A. Parrilo, and R. R. Thomas. Theta bodies for polynomial ideals. http://www.arxiv.org:0809.3480, 2008.

[18] C. J. Hillar and L-H. Lim. Most tensor problems are np-hard. *preprint.*

[19] C. J. Hillar and T. Windfeldt. Algebraic characterization of uniquely vertex colorable graphs. *J. Combin. Theory Ser. B*, 98(2):400–414.

[20] T. Hogg and C.P. Williams. The hardest constraint problems: a double phase transition. *Artif. Intell.*, 69(1-2):359–377, 1994.

[21] A. Kehrein and M. Kreuzer. Characterizations of border bases. *J. Pure Appl. Algebra*, 196(2-3):251–270.

[22] A. Kehrein, M. Kreuzer, and L. Robbiano. An algebraist's view on border bases. In A. Dickenstein and I. Emiris, editors, *Solving Polynomial Equations: Foundations, Algorithms, and Applications*, volume 14 of *Algorithms and Computation in Mathematics*, chapter 4, pages 160–202. Springer Verlag, Heidelberg, 2005.

[23] J. Kollár. Sharp effective nullstellensatz. *J. Amer. Math. Soc.*, 1(4):963–975.

[24] J. B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM J. Optim.*, 12(3):756–769 (electronic).

[25] J. B. Lasserre, M. Laurent, and P. Rostalski. Semidefinite characterization and computation of zero-dimensional real radical ideals. *Found. Comput. Math.*, 8(5):607–647.

[26] J.B. Lasserre, Laurent M., and Rostalski P. A unified approach to computing real and complex zeros of zero-dimensional ideals. In M. Putinar and S. Sullivant, editors, *Emerging Applications of Algebraic Geometry*, volume 149 of *IMA Volumes in Mathematics and its Applications*, pages 125–155. Springer, 2009.

[27] M. Laurent. Semidefinite representations for finite varieties. *Math. Program.*, 109(1, Ser. A):1–26.

[28] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming. *Math. Oper. Res.*, 28(3):470–496, 2003.

[29] M. Laurent and F. Rendl. Semidefinite programming & integer programming. In K. Aardal, G. Nemhauser, and R. Weismantel, editors, *Handbook on Discrete Optimization*, pages 393–514. Elsevier B.V., 2005.

[30] L. Lovász. Stable sets and polynomials. *Discrete Math.*, 124(1-3):137–153. Graphs and combinatorics (Qawra, 1990).

[31] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190.

[32] S. Margulies. *Computer Algebra, Combinatorics, and Complexity: Hilbert's Nullstellensatz and NP-Complete Problems.* PhD thesis, UC Davis, 2008.

[33] B. Mourrain. A new criterion for normal form algorithms. pages 430–443.

[34] B. Mourrain and P. Trébuchet. Stable normal forms for polynomial system solving. *Theoret. Comput. Sci.*, 409(2):229–240.

[35] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*, 96(2, Ser. B):293–320. Algebraic and geometric methods in discrete optimization.

[36] P. A. Parrilo. An explicit construction of distinguished representations of polynomials nonnegative over finite sets. IfA AUT02-02, ETH Zürich, 2002.

[37] Greg Reid. Solving polynomial systems via symbolic-numeric reduction to geometric involutive form. *J. Symbolic Comput.*, 44(3):280–291.

[38] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430.

[39] H. J. Stetter. *Numerical polynomial algebra.* Society for Industrial and Applied Mathematics (SIAM).

[40] G. Tinhofer. Graph isomorphism and theorems of birkhoff type. *Computing*, 36:285–300, 1986.

[41] W. T. Tutte. On hamiltonian circuits. *J. London Math. Soc.*, 21:98–101.

[42] A. Van Gelder. Another look at graph coloring via propositional satisfiability. *Discrete Appl. Math.*, 156(2):230–243.

[43] L. Zhang. zchaff v2007.3.12. Available at http://www.princeton.edu/ chaff/zchaff.html, 2007.