OPERATIONS RESEARCH CENTER

Working Paper

A New Algebraic Geometry Algorithm for Integer Programming

by

D. Bertsimas

G. Perakis

S. Tayur

OR 329-98 June 1998

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

A New Algebraic Geometry Algorithm for Integer Programming

by

D. Bertsimas
G. Perakis
S. Tayur

OR 329-98 June 1998

A New Algebraic Geometry Algorithm for Integer Programming

Dimitris Bertsimas* Georgia Perakis † Sridhar Tayur ‡

March 1998

Abstract

We propose a new algorithm for solving integer programming (IP) problems that is based on ideas from algebraic geometry. The method provides a natural generalization of Farkas lemma for IP, leads to a way of performing sensitivity analysis, offers a systematic enumeration of all feasible solutions, and gives structural information of the feasible set of a given IP. We provide several examples that offer insights on the algorithm and its properties.

1 Introduction

In this paper we introduce a new approach for solving integer programming problems (IPs). Our results are inspired from the observation that we can view any 0-1 IP as a system of quadratic equalities. We apply ideas from algebraic geometry to provide an algorithm for the problem that has several implications. Conti and Traverso (1991) introduced a very different approach for solving IPs that was also based on ideas from algebraic geometry. Nevertheless,

*Boeing Professor of Operations Research, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Mass. 02139. Research supported by NSF grant DMI-9610486.

†Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, Mass. 02139. Research supported by NSF grant DMI-9634736.

‡Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213.

unlike the approach in Conti and Traverso (1991) (see also Tayur, Thomas and Natraj (1995) or Thomas (1995)), our approach uses a specific right hand side b, and is directly motivated by 0-1 integer programming problems. We believe that our approach is computationally promising especially as a proof technique for infeasible IPs. In particular, our algorithm may be viewed as a generalization of Farkas lemma as well as a way of performing sensitivity analysis for IPs. Moreover, preliminary computational results indicate that our algorithm shows promise for problems that are either infeasible or have a small number of feasible solutions.

To make our results more accessible to the reader we will first focus on the 0-1 feasibility integer programming problem. Later in the paper we will illustrate how to extend our results for solving 0-1 as well as general integer optimization problems.

Definition 1 *Given a $m \times n$ matrix A, and a m-vector b, the* `0-1 feasibility integer programming problem` *is the problem of deciding whether there is a n-vector x with 0-1 coordinates such that*

$$Ax = b, \tag{1}$$

$$x \in \{0,1\}^n.$$

This problem can be rewritten equivalently as the following system of equations,

$$Ax = b,$$

$$x_j^2 = x_j, \quad j = 1, \ldots, n.$$

The contributions of this paper are as follows:

1. We provide an algorithm for the 0-1 IP feasibility problem that systematically enumerates all feasible solutions or shows that none exists .

2. We extend the algorithm for solving general IP optimization problems.

3. We establish that the algorithm leads to a strong duality theory for IPs, in the sense that it provides a certificate for infeasibility.

4. We address the reverse problem, i.e., given a set of integer points in $\{0,1\}^n$ we provide an underlying IP.

5. We show that our method leads to a way of performing sensitivity analysis.

6. We report examples that illustrate that our method is computationally promising.

The paper is structured as follows. In Section 2, we review definitions and basic results from algebraic geometry to make the paper self contained. In Section 3, we present our algorithm for the 0-1 IP feasibility problem, and illustrate through examples several of its properties. In Section 4, we extend our approach to the feasibility and optimization problems of general IPs. Section 5 illustrates the application of our algorithm for sensitivity analysis for IPs. A technical result is included in the appendix.

2 Preliminaries

In order to make this paper self contained, we review in this section some basic definitions and results from an introductory text on computational algebraic geometry by Cox, Little and O'Shea (1997). The interested reader may consult this book for further details.

In this paper we work over an algebraically closed field k, which is the field of complex numbers $\mathcal{C}$. The polynomial ring over this field is represented by $k[x_1,\ldots,x_n]$.

Definition 2 *Given polynomials $f_1,\ldots,f_s \in k[x_1,\ldots,x_n]$, the set $V(f_1,\ldots,f_s)$ such that*

$$V(f_1,\ldots,f_s) := \Big\{(a_1,\ldots,a_n) \in k^n : f_i(a_1,\ldots,a_n) = 0 \ \forall i\Big\},$$

is an `affine variety` *defined by $f_1,\ldots,f_s$.*

The notion of ideals is closely connected to the notion of affine varieties.

Definition 3 *A subset I of $k[x_1,\ldots,x_n]$, is an* `ideal` *if*
(i) $0 \in I$;
(ii) if $f,g \in I$, then $f+g \in I$;
(iii) if $f \in I$, and $h \in k[x_1,\ldots,x_n]$, then $hf \in I$.

Remark:

Given polynomials $f_1, \dots, f_s$, we define $< f_1, \dots, f_s >$ as the set that consists of all polynomials that are obtained by $\sum_{i=1}^{s} h_i f_i$, with $h_i \in k[x_1, \dots, x_n]$. It is not difficult to see that $< f_1, \dots, f_s >$ is an ideal. We call this "the ideal generated by $f_1, \dots, f_s$."

Definition 4 *Given a* `term order` *(that is, a total order, such as lexicographic) on the monomials in $k[x_1, \dots, x_n]$, we define as the* `leading monomial` *of a given polynomial to be the monomial with the highest term order.*

We are now ready to introduce the notion of a Groebner basis.

Definition 5 *A set of polynomials $Q := \{q_1, \dots, q_t\}$ is a* `Groebner basis` *generating an ideal $F :=< f_1, \dots, f_s >$, if it has the property that all the leading monomials of F can be generated by the leading monomials of the polynomials in Q.*

One of the central results in the area of algebraic geometry due to Hilbert is the following. The book by Cox, Little and O'Shea (1997) provides a proof and further details.

Theorem A (`Hilbert Basis Theorem`)

Every ideal has a Groebner basis which has a finite number of elements.

Given an ideal F generated by polynomials $f_1, \dots, f_s$ and a term order, we can compute the Groebner basis of F using Buchberger's algorithm (see Cox et. al. (1997) for further details).

We use $I(V)$ to denote the ideal that contains all polynomials that vanish on a given variety V and $V(I)$ to denote the variety $V(r_1, \dots, r_s)$ where $R := \{r_1, \dots, r_s\}$, is a Groebner basis of the ideal I.

Definition 6 *Given $I :=< f_1, \dots, f_s >\subset k[x_1, \dots, x_n]$, the lth* `elimination ideal` *I_l is the ideal in $k[x_{l+1}, \dots, x_n]$ defined by $I \cap k[x_{l+1}, \dots, x_n]$.*

In the remainder of this paper we will use the following results from Cox, Little and O'Shea (1997).

Lemma A

If $f_1, \ldots, f_s \in k[x_1, \ldots, x_n]$, *then* $< f_1, \ldots, f_s > \subset I(V(f_1, \ldots, f_s))$, *although equality need not occur.*

Theorem B (`The Elimination Theorem`)

Let $I \subset k[x_1, \ldots, x_n]$ *be an ideal and let* G *be a Groebner basis of* I *with respect to lex order where* $x_1 > x_2 > \ldots > x_n$. *Then, for every* $0 \leq l \leq n$, *the set*

$$G_l = G \cap k[x_{l+1}, \ldots, x_n]$$

is a Groebner basis of the l*th elimination ideal* I_l.

Theorem C (`The Extension Theorem`)

Let $I =< f_1, \ldots, f_s >\subset k[x_1, \ldots, x_n]$ *and let* I_1 *be the first elimination ideal of* I. *For each* $1 \leq i \leq s$, *we can write* f_i *in the form*

$$f_i = g_i(x_2, \ldots, x_n)x_1^{N_i} + \textit{terms in which } x_1 \textit{ has degree smaller than } N_i,$$

where $N_i \geq 0$ *and* $g_i \in k[x_2, \ldots, x_n]$ *is nonzero. Suppose we have a partial solution* $(a_2, \ldots, a_n) \in V(I_1)$. *If* $(a_2, \ldots, a_n) \notin V(g_1, \ldots, g_s)$, *then there exists* $a_1 \in \mathcal{C}$ *such that* $(a_1, \ldots, a_n) \in V(I)$.

Corollary D

Let $V(f_1, \ldots, f_s) \subset \mathcal{C}^n$, *and assume that for some* i, f_i *is of the form*

$$f_i = cx_1^N + \textit{terms in which } x_1 \textit{ has degree smaller than } N_i,$$

where $c \in \mathcal{C}$ *is nonzero and* $N > 0$. *If* I_1 *is the first elimination ideal, then in* $\mathcal{C}^{n-1}$

$$\pi_1(V) = V(I_1),$$

where π_1 *is the projection on the last* $n-1$ *components.*

Lemma E

Let $V = V(f_1, \ldots, f_s)$ *and* $\pi_l : \mathcal{C}^n \to \mathcal{C}^{n-l}$ *be a projection map which sends* $(a_1, \ldots, a_n)$ *to*

$(a_{l+1}, \ldots, a_n)$. *Let* $I_l =< f_1, \ldots, f_s > \cap k[x_{l+1} \ldots, x_n]$ *be the* l*th elimination ideal. Then, in* $\mathcal{C}^{n-l}$, *we have*

$$\pi_l(V) \subset V(I_l).$$

3 An algorithm for the 0-1 feasibility IP

In the previous section we laid the foundation for presenting an algorithm for solving the 0-1 feasibility IP (1). We consider the following polynomials in $k[x_1, \ldots, x_n]$

$$\begin{aligned} f_i &= \sum_{j=1}^{n} a_{ij} x_j - b_i, \quad i = 1, \ldots, m \\ g_j &= x_j^2 - x_j, \quad j = 1, \ldots, n. \end{aligned}$$

We let $\tilde{V} := V(f_1, \ldots f_m, g_1, \ldots, g_n)$ be the variety they define.

If we let k be the field of complex numbers, then $\tilde{V}$ is the feasible set of the 0-1 IP with matrix A and right hand side b. (The data is assumed to be in $\mathcal{R}$.) This is based on the simple observation that either $\tilde{V}$ is empty, or if there is an element, it is actually in $\mathcal{R}$ (it is in fact integral).

We consider the ideal $\tilde{I} := I(\tilde{V})$ and the ideal $J :=< f_1, \ldots, f_m, g_1, \ldots, g_n >$. The algorithm we propose enumerates all feasible 0-1 solutions, or detects that no feasible solution exists.

Algorithm A.

Input: Matrix A and vector b.

Output: All feasible solutions $(a_1, \ldots, a_n)$ to Problem (1).

1. Find a Groebner basis G of J using lex order $x_1 > x_2 > \ldots > x_n$. If $G = \{1\}$, then the 0-1 IP has no feasible solutions. `Exit`.

2. If $G \neq \{1\}$: Consider for $1 \leq l \leq n-1$, the sets $G_l = G \cap k[x_{l+1}, \ldots, x_n]$. Starting from $l = n-1$, and working sequentially:

- Find a_n in $V(G_{n-1})$.
- Extend a_n to (a_{n-1}, a_n) such that $(a_{n-1}, a_n) \in V(G_{n-2})$.
- $\vdots$
- Find a_2 such that $(a_2, \ldots, a_n) \in V(G_1)$.
- Find a_1 such that $(a_1, \ldots, a_n) \in V(G)$.

We next show that the algorithm correctly solves 0-1 feasibility IPs (Problem (1)).

Theorem 1 *Algorithm A either provides all feasible solutions for Problem (1), or provides a certificate of infeasibility whenever the Groebner basis $G = \{1\}$.*

Proof: Lemma A implies that

$$J \subseteq \tilde{I} = I(\tilde{V}).$$

Consequently, $\tilde{V} \subseteq V(J)$.

Consider the elimination ideals $J_1, \ldots, J_{n-1}$, where $J_k = J \cap k[x_{k+1}, \ldots x_n]$. If we find the Groebner basis G of J using `lex order` (via Buchberger's algorithm), then Theorem B implies that $G_k := G \cap k[x_{k+1}, \ldots x_n]$ is a Groebner basis of J_k, further implying that $V(G_k) = V(J_k)$. Observe that one of the following two cases holds:
(a) The Groebner basis $G = \{1\}$. Then the ideal J coincides with $k[x_1, \ldots, x_n]$ indicating that $V(J)$ is empty, since we are working on an algebraic closed field. Therefore, $\tilde{V}$ is an empty set implying that we have an infeasible integer program.
(b) The Groebner basis $G \neq \{1\}$. In this case, we notice that G_{n-1} has elements $x_n - 1$ or x_n or $x_n^2 - x_n$ belonging to it. This follows from the observation that J_{n-1} is a polynomial ideal in one variable, namely a subset in $k[x_n]$ and therefore, needs only one generator. We interpret this to mean that points 1 or 0 or both 0 and 1 are *partial* solutions respectively. That is, we get a a_n in $V(J_{n-1})$. Subsequently we obtain a_{n-1} by extending the partial solution a_n to $(a_{n-1}, a_n) \in V(J_{n-2})$. That is, we take a partial solution in $V(J_k)$ and extend it to a partial solution in $V(J_{k-1})$ and so on. By Theorem C, this is always possible as

the appropriate leading co-efficients (starting from f_i and g_j) in our setting are constants. Continuing in this way and applying Theorem C, we can obtain a solution $(a_1, \ldots, a_n)$ in $V(J)$.

We are interested, however, in a point of $\tilde{V}$. It is easy to see that Lemma E implies that the projection π_k of $\tilde{V}$ to the last $n-k$ coordinates, satisfies $\pi_k(\tilde{V}) \subset V(I_k)$. We notice that a repeated application of Corollary D indicates that

$$\pi_k(\tilde{V}) = V(I_k),$$

and therefore, $(a_2, \ldots, a_n) \in V(J_1)$ obtained by the Algorithm A is in $\pi_1(\tilde{V})$. We can find a point in $\tilde{V}$ using any one of the f_i. ■

Remark:
It is important to observe that using different a_n from $V(J_{n-1})$, we can find **all** solutions to the given 0-1 integer program.

We next illustrate the use of Algorithm A. All computations in this paper have been done using an implementation of Buchberger's algorithm in Mathematica on a personal computer.

Example 1. Consider the IP

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 15x_6 = 15, \quad x_j \in \{0, 1\} \; \forall j.$$

The ideal we consider is

$$J = < x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 15x_6 - 15, x_1^2 - x_1, x_2^2 - x_2, \ldots, x_6^2 - x_6 > .$$

A Groebner basis of J (with lex order) is

$$G = \{x_6^2 - x_6, x_5 + x_6 - 1, x_4 + x_6 - 1, x_3 + x_6 - 1, x_2 + x_6 - 1, x_1 + x_6 - 1\}.$$

Therefore, we have $G_5 = \{x_6^2 - x_6\}$, indicating that $a_6 = 1$ and $a_6 = 0$ are both partial solutions. Starting from $a_6 = 1$, we get $a_1 = a_2 = \ldots = a_5 = 0$. Starting from $a_6 = 0$, we get $a_1 = \ldots = a_5 = 1$. Therefore, we have two feasible solutions: (0,0,0,0,0,1) and (1,1,1,1,1,0).

An interesting feature of the Groebner basis, for example, is the interpretation of the term $x_5 + x_6 - 1$. This implies that in all solutions exactly one of x_5 and x_6 is equal to 1. This example illustrates the structural information that the Groebner basis contains regarding a 0-1 IP.

The next example amplifies the observation that Algorithm A captures logical interactions between variables.

Example 2. Consider the IP

$$x_1 + 3x_2 + 2x_3 + 2x_4 + 4x_5 + 4x_6 = 4, \quad x_j \in \{0,1\}\ \forall j.$$

A Groebner basis is

$$G = \{x_6^2 - x_6, x_5x_6, x_5^2 - x_5, x_4x_5, x_4^2 - x_4, x_3 - x_4, x_2 + x_4 + x_5 + x_6 - 1, x_1 + x_4 + x_5 + x_6 - 1\}.$$

The element $x_3 - x_4$ implies that both variables should always have the same value in any feasible solution. Other elements are interpreted accordingly.

3.1 On the structure of the reduced Groebner basis obtained from Algorithm A

We have illustrated that algorithm A finds a Groebner basis and therefore, all the feasible solutions of Problem (1). Nevertheless, for a given ideal the Groebner basis is not typically unique. However, we may use the notion of `the reduced Groebner basis` (see Cox et. al. (1997)) which is unique for a given ideal. Some computer algebra packages find a reduced Groebner basis, while others do not. The reduced Groebner basis has some advantages. From a theoretical perspective, we can show the following properties.

Properties:

When Algorithm A uses the a reduced Groebner basis, then the following hold:

1. *If the solution of Problem (1) is unique, then all the reduced Groebner basis elements*

are of the form $x_i - a_i$ with $a_i \in \{0,1\}$.

Although this is an intuitive property, we also provide a formal proof in Appendix A (Theorem 3).

2. *The reduced Groebner basis applied with the lexicographic term order to Problem (1), has the following diagonalized structure:*
   - *(i) The first element is either x_n, $x_n - 1$ or $x_n^2 - x_n$.*
   - *(ii) There are several terms of up to second degree polynomials only involving variables x_n, x_{n-1}.*
   - *(iii) For $i = 2, \ldots, n-1$, there are several terms involving variables $x_n, x_{n-1}, \ldots, x_{n-i}$.*

These observations follow from the fact that we are dealing with elimination ideals as well as the fact that we are restricted to 0-1 IPs.

The following example illustrates Property 1.

Example 3.

Consider the following 0-1 feasibility integer programming problem.

$$x_1 + x_2 + x_3 = 2,$$

$$x_1 + x_2 = 2,$$

$$x_2 + x_3 = 1,$$

$$x_i \in \{0,1\}, \quad i = 1, \ldots, 3.$$

This problem has the unique feasible solution $x_1 = x_2 = 1$, $x_3 = 0$. Furthermore, Algorithm A yields this solution through the reduced Groebner basis $G = \{x_3, -1 + x_2, -1 + x_1\}$.

To illustrate property 2 as well as the importance of using the reduced Groebner basis we provide the following example.

Example 4.

Consider the 0-1 feasibility IP

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 6x_5 = 6, \quad x_j \in \{0,1\}\ \forall j.$$

A Groebner basis of J (but not the reduced one) in this case is

$$G' = \{x_5^2 - x_5, x_4x_5, x_4^2 - x_4, x_3 + x_4 + x_5 - 1, x_2 - 144x_3x_4x_5 + 24x_3x_4$$

$$-26x_3 - 24x_4x_5 - 26x_4 - 25x_5 + 25, x_1 + 2x_2 + 3x_3 + 4x_4 + 6x_5 - 6\}.$$

We notice that triplets such as $x_3x_4x_5$ can be part of a polynomial in a Groebner basis. Moreover, we obtain polynomials with very large coefficients. However, we notice that the reduced Groebner basis for lex order is

$$G = \{x_5^2 - x_5, x_4x_5, x_4^2 - x_4, x_3 + x_4 + x_5 - 1, x_2 + x_5 - 1, x_1 + x_4 + x_5 - 1\}.$$

Algorithm A enumerates all 0-1 solutions:

$$(0,0,0,0,1), \quad (0,1,0,1,0), \quad (1,1,1,0,0).$$

Although this example indicates that the degree of no term exceeds two, this is just a coincidence. It is possible to construct examples where the reduced Groebner basis has terms with degree larger than two. The following example illustrates this fact.

Example 5.

Consider the following 0-1 feasibility IP

$$x_1 + x_2 + x_3 + x_5 + 2x_6 = 2$$

$$x_2 + x_4 + x_7 = 1$$

$$x_3 + x_4 + x_8 = 1$$

$$x_i \in \{0,1\}, \quad i = 1, ..., 8.$$

The reduced Groebner basis G for the term order $x_8 > x_7 > x_6 > x_5 > x_1 > x_2 > x_3 > x_4$ is

$$G = \{x_4^2 - x_4, x_3x_4, x_3^2 - x_3, x_2x_4, x_2^2 - x_2, x_1x_2x_3, -x_1 - x_2 + 2x_1x_2 - x_3 + 2x_1x_3 + 2x_2x_3 + x_5,$$

$$x_1^2 - x_1, -1 + x_1 + x_2 - x_1x_2 + x_3 - x_1x_3 - x_2x_3 + x_6, -1 + x_2 +$$

$$x_4 + x_7, -1 + x_3 + x_4 + x_8\}.$$

Notice that there is a third degree polynomial in the reduced Groebner basis of this example, that is, $x_1x_2x_3$.

3.2 On the relation of Gaussian elimination and Algorithm A

Gaussian elimination for the $n \times n$ linear system $Ax = b$ provides a diagonalized system of equations in which variable x_n appears only in one equation, variables x_n and x_{n-1} appear in another, and so on. In this sense, the properties we described in Section 3.1 are the generalization of Gaussian elimination for 0-1 IPs.

Example 6. Consider the IP

$$2x_1 + 2x_2 + 3x_3 + 2x_4 = 5$$

$$x_1 + x_2 + 2x_3 + x_4 = 3$$

$$x_j^2 = x_j, \quad j = 1, \dots, 4.$$

The reduced Groebner basis is

$$G = \{x_4^2 - x_4, x_3 - 1, x_2x_4, x_2^2 - x_2, x_1 + x_2 + x_4 - 1\}.$$

The diagonalized structure of the properties in Section 3.1 is $(x_4^2 - x_4)$, $(x_3 - 1)$, $(x_2x_4, x_2^2 - x_2)$, and $(x_1 + x_2 + x_4 - 1)$.

3.3 An interpretation of Algorithm A as Farkas lemma for 0-1 IPs

Farkas lemma, which is the central idea of duality in linear programming, provides a certificate of infeasibility for a linear programming problem. *What is a certificate of infeasibility of a 0-1 IP ?* An obvious (and very inefficient) certificate is the enumeration of all possible 2^n vectors. Nevertheless, Algorithm A provides a potentially certificate in a more efficient way. If the 0-1 IP is infeasible, then $G = \{1\}$. In other words, the certificate is the computation of the Groebner basis.

Example 7. Consider the following example:

$$2x_1 + 2x_2 + 4x_3 + 6x_4 = 11, \quad x_j \in \{0, 1\}.$$

Clearly, this is an infeasible IP, as the lhs is an even integer, while the rhs is an odd one. In this case $G = \{1\}$.

3.4 Constructing an IP from a given set of points in $\{0,1\}^n$

To this point, this paper has addressed the problem of finding the feasible points of a 0-1 feasibility integer programming problem (Problem (1)). In this subsection, we will address the reverse problem. That is, we will establish how to construct a 0-1 feasibility integer programming problem for a given set of integer points in $\{0,1\}^n$.

Theorem 2 *Suppose we are given a set of points S in $\{0,1\}^n$ representing the feasible space of some 0-1 IP. We can provide an underlying IP and construct a Groebner basis of an ideal I such that $V(I) = S$.*

Proof:

Given a subset S of points in $\{0,1\}^n$ we can enumerate the points $P^i = (p_1^i, \dots, p_n^i)$ in $\{0,1\}^n$, for $i = 1, \dots m$, that do not lie in the set S. The following inequality describes a set of points in R^n that excludes only point P_i,

$$\sum_{j=1}^{n}(x_j - p_j^i)^2 \geq 1.$$

Therefore, we can describe the set S through the following set of quadratic inequalities,

$$\sum_{j=1}^{n}(x_j - p_j^i)^2 \geq 1, \quad i = 1, \dots, m,$$
$$x_j^2 - x_j = 0, \quad j = 1, \dots, n.$$

Nevertheless, the observation that the variables x_j are 0 or 1, allows us to rewrite this as a set of linear inequalities combined with separable, quadratic equalities for each variable as follows

$$\sum_{j=1}^{n}[x_j(1 - 2p_j^i) + p_j^i] \geq 1, \quad i = 1, \dots, m,$$
$$x_j^2 - x_j = 0, \quad j = 1, \dots, n.$$

Introducing a binary expression for the excess variables corresponding to each inequality yields the following set of linear equalities with integer variables,

$$\sum_{j=1}^{n}[x_j(1 - 2p_j^i) + p_j^i] - \sum_{k=0}^{\lceil \log n \rceil - 1} 2^k x_{n+1+(i-1)\lceil \log n \rceil + k} = 1, \quad i = 1, \dots, m,$$

$$x_j^2 - x_j = 0, \quad \forall j.$$

The previous representation formulates the set of points S in the form of Problem (1). For this new representation we can apply Buchberger's algorithm to find a Groebner basis. ∎

Theorem 3 in Appendix A also provides an alternative construction.

4 Optimization of IPs

In this section, we will generalize our results of Section 3 for solving general integer programming problems. In the next subsection, we will illustrate how to solve 0-1 IPs.

4.1 Optimization of 0-1 IPs

We consider the optimization problem

$$\begin{array}{ll} \text{minimize} & c'x \\ \text{subject to} & Ax = b \\ & x_j^2 = x_j, \quad \forall j. \end{array}$$

The largest objective function value is $\bar{Z} = \sum_{j:c_j \geq 0} c_j$. Let Z_{LP} be the value of the LP relaxation. Clearly, $\lceil Z_{LP} \rceil \leq Z_{IP} \leq \bar{Z}$. This observation allows us to apply binary search on Z_{IP} and solve the optimization problem as follows. We add the polynomial $h := \sum_{j=1}^{n} c_j x_j - C$ to the generators of J, for specific values of C in the range $[\lceil Z_{LP} \rceil, \bar{Z}]$. We thus need to apply Algorithm A at most $\log(\bar{Z} - \lceil Z_{LP} \rceil)$ times.

A more direct method is as follows. We work in $k[x_1, \ldots, x_n, y]$. Let

$$h := y - \sum_{j=1}^{n} c_j x_j$$

and we look at $\hat{V} := V(f_1, \ldots, f_m, g_1, \ldots, g_n, h)$. Following the same approach as the one for feasibility, with `lex` order $x_1 > x_2 > \ldots x_n > y$, we notice that the Groebner basis $\hat{G}$ of $\hat{J} :=< f_1, \ldots, f_m, g_1, \ldots, g_n, h >$ is either $\{1\}$ (indicating infeasibility) or we will have $\hat{G}$ intersected with $k[y]$ which leads to a polynomial in y. We interpret this polynomial in y

as follows: Every root of the polynomial is a feasible cost of the IP. Therefore, we can find the minimum root, and work upwards to get the associated x_j values.

Example 8.

Consider the IP:

$$\begin{array}{rl} \text{minimize} & x_1 + 2x_2 + 3x_3 \\ \text{subject to} & x_1 + 2x_2 + 2x_3 = 3 \\ & x_j^2 = x_j, \quad j = 1, \dots, 3. \end{array}$$

The reduced Groebner basis of

$$J =< y - x_1 - 2x_2 - 3x_3, 2x_1 + 4x_2 + 4x_3 - 6, x_1^2 - x_1, x_2^2 - x_2, x_3^2 - x_3 >$$

is

$$G = \{12 - 7y + y^2, 3 + x_3 - y, -4 + x_2 + y, 1 - x_1\}.$$

The two roots of the polynomial $12 - 7y + y^2$ are $y = 3$ and $y = 4$. Thus the minimum value is $y = 3$, and the corresponding solution is $(1, 1, 0)$.

We next illustrate that we can simplify the calculation of Algorithm A if we have partial information on the optimal cost.

Example 9.

Consider the IP:

$$\begin{array}{rl} \text{minimize} & x_1 + 2x_2 + 3x_3 + 3x_4 \\ \text{subject to} & x_1 + x_2 + 2x_3 + x_4 = 3 \\ & x_j^2 = x_j, \quad j = 1, \dots, 3. \end{array}$$

A Groebner basis of $J =< x_1 + 2x_2 + 3x_3 + 3x_4 - y, 2x_1 + 2x_2 + 4x_3 + 2x_4 - 6, x_1^2 - x_1, x_2^2 - x_2, x_3^2 - x_3, x_4^2 - x_4 >$ is

$$G = \{120 - 74y + 15y^2 - y^3,\ -20 + 2x_4 + 9y - y^2,\ -6 + 6x_3 + y - x_3 y,$$

$$x_3 - x_3^2,\ -23 - x_2 - x_3 - x_3 + 10y - y^2,\ 32 - 2x_1 - 2x_3 - 11y + y^2\},$$

which suggests that all feasible y's are $y = 4, 5, 6$, i.e., the roots of the first equation of the Groebner basis. If however, we have additional information that $4 \leq y \leq 5$, we can add the polynomial $(y-4)(y-5)$ and rerun algorithm A, to obtain the solution:

$$G = \{20 - 9y + y^2,\ -1 + x_3,\ -4 - x_2 + y,\ 5 - x_1 - y\}.$$

A natural question is to compare the performance of Algorithm A to branch and bound. Our next example addresses this issue.

Example 10.

We consider the class of integer programming problem

$$\begin{aligned} \text{minimize} \quad & x_{n+1} \\ \text{subject to} \quad & 2x_1 + 2x_2 + \cdots + 2x_n + x_{n+1} = n \\ & x_i \in \{0, 1\}. \end{aligned}$$

It is easy to show that any branch and bound algorithm that uses linear programming relaxations to compute lower bounds, and branches by setting a fractional variable to either zero or one, will require the enumeration of an exponential number of subproblems when n is odd (see Bertsimas and Tsitsiklis (1997)). It is thus interesting to observe the performance of Algorithm A. Applying Algorithm A to this problem we obtain the reduced Groebner basis very quickly:

$$G = \left\{ x_{n+1} - 1,\ x_j^2 - x_j, j = 1, \ldots, n,\ \sum_{j=1}^{n} x_j - \frac{n-1}{2} \right\}.$$

In this particular class of examples, Algorithm A is stronger than branch and bound.

4.2 Optimization of general IPs

An arbitrary IP, in which the variables are only restricted to be nonnegative integers, can be reduced in a standard way to the 0-1 case as follows: If $x_j \in \{0, 1, \ldots, U_j\}$ with U_j known, then for each j, we write $x_j = \sum_{p=0}^{\lceil \log U_j \rceil - 1} 2^p x_j^p$, with the auxiliary variables x_j^p taking values either 0 or 1. We then substitute the above expression for x_j in the objective function and the constraints. Alternatively, we can include the polynomial

$$h = x_j(x_j - 1) \cdots (x_j - U_j)$$

to the ideal J and apply Algorithm A. The next example illustrates an application of this idea. Instead of considering $x_j \in \{0, 1, 2\}$, we consider without loss of generality the case $x_j \in \{-1, 0, 1\}$, i.e., $x_j^3 = x_j$.

Example 11. We consider the IP

$$2x_1 - 2x_2 + x_3 = 1$$

$$3x_1 + x_2 + 2x_3 = 1$$

$$x_j \in \{-1, 0, 1\}.$$

Then, we apply algorithm A on the ideal $J =< 2x_1 - 2x_2 + x_3 - 1,\ 3x_1 + x_2 + 2x_3 - 1,\ x_1^3 - x_1,\ x_2^3 - x_2,\ x_3^3 - x_3 >$. The reduced Groebner basis is

$$G = \{1 - x_3, -1 - x_2, -1 - x_1\}.$$

The general case of x_j nonnegative integer can be reduced to the bounded case as follows. Papadimitriou and Steiglitz (1982), prove that if the IP $Ax = b$, $x_j \in Z^+$ has a solution, then it has a solution with $x \in \{0, 1, \ldots, M\}^n$, where $M = n(ma_{max})^{2m+3}(1 + b_{max})$, with $a_{max} = \max |a_{ij}|$, and $b_{max} = \max |b_i|$.

4.3 IPs with inequality constraints

In this subsection, we will illustrate how to solve IPs with inequality constraints. The key idea in showing this, is to convert the given IP into an IP with linear equality constraints and integer variables.

Consider the general IP problem

$$
\begin{aligned}
\text{minimize} \quad & c'x \\
\text{subject to} \quad & a_i'x \le b_i, \qquad i = 1, \dots, m \\
& x_j \text{ integer}, \qquad j = 1, \dots, n.
\end{aligned}
$$

For each inequality $i = 1, \dots, m$, we will introduce a nonnegative integer slack variable s_i, which is less than b_i. Nevertheless, we can rewrite a binary expression of these variables as before. This observation allows us to convert our problem into an IP with linear equalities and integer variables.

The following example illustrates this.

Example 12. We consider the IP

$$x_1 + x_2 + x_3 \le 2,$$

$$x_2 + x_3 + x_4 \le 4,$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, 4.$$

We introduce slack variables s_1 and s_2 for the first and second constraint respectively. Furthermore, we rewrite these variables as $s_1 = x_5 + 2x_6$ and $s_2 = x_7 + 2x_8 + 4x_9$, with $x_5, \dots, x_9 \in \{0, 1\}$.

Therefore, we rewrite the IP as

$$x_1 + x_2 + x_3 + x_5 + 2x_6 = 2,$$

$$x_2 + x_3 + x_4 + x_7 + 2x_8 + 4x_9 = 4,$$

$$x_i^2 - x_i = 0, \quad i = 1, \dots, 9.$$

We are now able to solve this problem using Algorithm A, as we have shown in the previous subsections.

5 Sensitivity analysis of IPs

Our results in this paper also allow us to perform sensitivity analysis for integer programming problems. That is, they allow us to address the problem of finding the optimal objective function value as a function of one of the right hand side coefficients b_i. To achieve this we work in $k[x_1, \ldots, x_n, y, b_i]$ with lex order $x_1 > \ldots > x_n > y > b_i$, and find the Groebner basis G. We find that either (i) $G = \{1\}$ indicating that there is no value of b_i for which the problem is feasible, or (ii) $G \cap k[b_i]$ is a polynomial in b_i, and each of the roots of this polynomial represents a value for which the problem has a feasible solution. In this case, $G \cap k[y, b_i]$ are polynomials in y and b_i, and so we have an explicit representation of the `value function`.

Example 13.

Consider the IP

$$\begin{array}{rl} \text{minimize} & x_1 + 2x_2 + 3x_3 \\ \text{subject to} & 2x_1 + 2x_2 + 4x_3 = b \\ & x_j^2 = x_j, \quad j = 1, \ldots, 3. \end{array}$$

Suppose we are interested to find the optimal solution value as a function of b, when $4 \leq b \leq 6$.

A Groebner basis of

$$J = < x_1 + 2x_2 + 3x_3 - y, 2x_1 + 2x_2 + 4x_3 - b, (b-4)(b-5)(b-6), x_1^2 - x_1, x_2^2 - x_2, x_3^2 - x_3 >$$

is

$$G = \{24 - 10b + b^2,\ 18 - 3b - 6y + by,\ -14 - b + 9y - y^2,\ -4 + b + 4x_3 - bx_3,$$

$$3 - 3x_3 - y + x_3 y,\ -x_3 + x_3^2,\ -b - 2x_2 - 2x_3 + 2y, b - x_1 - x_3 - y\},$$

which implies that for $b = 4$, $y = 3$, and the two solutions are: $(0,0,1)$ and $(1,1,0)$. For $b = 5$, there is no solution, and for $b = 6$, the feasible y are roots of the equation $-20 + 9y - y^2 = 0$, i.e., $y = 4$ and $y = 5$.

6 Conclusions

In this paper we used ideas of algebraic geometry to present a method for solving integer programming problems. We started by presenting a method for solving the 0-1 feasibility integer programming problem, which we subsequently extended to solving general integer optimization problems. For the feasibility problem, our method provides a systematic enumeration of all feasible solution. Our results may be viewed as a natural generalization of Farkas Lemma to integer programming and allow us to check easily whether a given problem is infeasible. Our results also lead to a way of performing sensitivity analysis. Finally, we also addressed the reverse problem, that is, how to provide an IP formulation for a given set of integer points in $\{0,1\}^n$.

We have experimented with several integer programming problems of up to 25 variables. We have used a general purpose implementation of Buchberger's algorithm in Mathematica that does not exploit the particular structure of the IP. We are currently developing an implementation of Buchberger's algorithm that is tailored for Algorithm A. We used a personal computer with limited memory to perform the computation.

In preliminary computational work, we have observed that Algorithm A is computationally faster, when the problem is either infeasible or it has very few solutions. In such situations, we have been able to solve problems with 25 variables using Mathematica on a personal computer. The following is such an example.

Example 14. Consider the following 0-1 feasibility IP.

$$x_i + x_{i+1} = 1, \quad i = 1, \ldots, 24$$

$$x_j^2 = x_j, \quad j = 1, \ldots, 25.$$

Algorithm A, gives the Groebner basis

$$G = \{x_1, x_2 - 1, x_3, \ldots, x_{24} - 1, x_{25}\},$$

which indicates that the unique solution is $x_{2r-1} = 0$, $r = 1, \ldots, 13$, and $x_{2k} = 1$, $k = 1, \ldots, 12$.

Our goal in our future research is to explore the computational performance of Algorithm A.

References

[1] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*, Athena Scientific, Belmont, Massachusetts, 1997.

[2] P. Conti and C. Traverso. Buchberger Algorithm and Integer Programming. *Proceedings AAECC-9, New Orleans*, LNCS, Vol. 539 pp. 130-139.

[3] D. Cox, J. Little and D. O'Shea. *Ideals, Varieties and Algorithms*, Springer (Berlin), 1997 (second edition).

[4] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*, Wiley, New York, 1988.

[5] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization; Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, 1982.

[6] S. Tayur, R. Thomas and N. R. Natraj. An Algebraic Geometry Algorithm for Scheduling in Presence of Setups and Correlated Demands. *Mathematical Programming*, Vol. 69, pp. 369-401, 1995.

[7] R. Thomas. A Geometric Buchberger Algorithm for Integer Programming. *Math. Oper. Res.*, Vol. 20, pp. 864-884, 1995.

APPENDIX A

We need the following additional definitions and results from Cox, Little, O'Shea (1997).

Definition 7 *An ideal I is* `radical` *if $f^m \in I$ for any integer $m \geq 1$ implies that $f \in I$.*

Definition 8 *An ideal $I \subset k[x_1, \ldots, x_n]$ is* `prime` *if whenever $f, g \in k[x_1, \ldots, x_n]$ and $fg \in I$, then either $f \in I$ or $g \in I$.*

Definition 9 *An ideal $I \subset k[x_1, \ldots, x_n]$ is* `maximal` *if $I \neq k[x_1, \ldots, x_n]$ and any ideal J containing I is such that either $J = I$ or $J = k[x_1, \ldots, x_n]$.*

The following results are well known (see Cox et. al. (1997)).

If k is an algebraically closed field, then every maximal ideal of $k[x_1, \ldots, x_n]$ is of the form $< x_1 - a_1, \ldots, x_n - a_n >$ for some $a_1, \ldots, a_n \in k$.

Let $I =< f_1, \ldots, f_r >$ and $J =< g_1, \ldots, g_s >$. Then the product $I \cdot J$, also an ideal, is generated by the set of generators of I and J:

$$I \cdot J =< f_i g_j : 1 \leq i \leq r, 1 \leq j \leq s > .$$

Furthermore, $V(I \cdot J) = V(I) \cup V(J)$.

Definition 10 *The* `Zariski closure` *of a subset of affine space is the smallest affine algebraic variety containing the set. If $S \subset k^n$, the Zariski closure of S is equal to $V(I(S))$.*

Definition 11 *An affine variety $V \subset k^n$ is* `irreducible` *if whenever V is written in the form $V = V_1 \cup V_2$, where V_1 and V_2 are affine varieties, then either $V_1 = V$ or $V_2 = V$. It is easy to show that V is irreducible if and only if $I(V)$ is a prime ideal.*

The `Ideal-Variety` correspondence states that, for an arbitrary field k and variety S, we have

$$V(I(S)) = S.$$

We are now ready to state and prove Theorem 3 that is used to establish Property 1 in Section 3.1. The proof of Theorem 3 is an alternative construction to the reverse problem addressed in Section 3.4.

Theorem 2

Let $S \in k^n$ denote the set of feasible solutions to some 0-1 IP. (Recall that S is a variety and that we are working over a algebraically closed field k.) Then, we obtain

(a) *The set $I(S) := \{f : f(a) = 0 \; \forall a \in S\} \subset k[x_1, \ldots, x_n]$ is a radical ideal.*

(b) *The Zariski closure satisfies $V(I(S)) = S$.*

(c) *Let $S = \cup_{i=1}^{P} S_i$ be a decomposition of S into irreducible varieties. Then*

$$I(S) = I(\cup_{i=1}^{P} S_i) = \prod_{i=1}^{P} I(S_i)$$

where $I(S_i) =< x_{1i} - a_{1i}, \ldots, x_{ni} - a_{ni} >$ for $i = 1, \ldots, P$ and

$$S_i = \{(a_{1i}, \ldots, a_{ni})\} = V(x_{1i} - a_{1i}, \ldots, x_{ni} - a_{ni}).$$

In fact, this decomposition is unique upto the order of the S_i.

(d) *There is finite basis of $I(S)$ where the co-efficients of the polynomials are in $\{0, -1, 1\}$. In fact, this can be constructed sequentially if the elements of S are known.*

Proof: **(a)** We can easily verify that indeed $I(S)$ is an ideal. To prove that $I(S)$ is also radical, that is $I = \sqrt{I}$, pick an $f \in \sqrt{I}$. (Note that $I \subset \sqrt{I}$ always.) Then, by definition, $f^m = 0$ for some m, for all $a \in S$. Nevertheless, this implies that $f(a) = 0$, for all $a \in S$, indicating that $f \in I$.

(b) It follows from the Ideal-Variety correspondence since S is a variety.

(c) S_i is irreducible and $S_i = V(x_{1i} - a_{1i}, \ldots, x_{ni} - a_{ni})$. This implies that $I(S_i)$ is prime. In fact, it is maximal, and therefore,

$$I(S_i) =< x_{1i} - a_{1i}, \ldots, x_{ni} - a_{ni} > .$$

Observe that

$$S_1 \cup S_2 = V(f_l g_j : 1 \leq l \leq n;\ 1 \leq j \leq n)$$

where $f_l := x_{l1} - a_{l1}$ and $g_j := x_{j2} - a_{j2}$. Therefore,

$$I(S_1) \cdot I(S_2) =< f_l g_j : 1 \leq l \leq n;\ 1 \leq j \leq n > .$$

$$V(I(S_1) \cdot I(S_2)) = V(I(S_1)) \cup V(I(S_2)),$$

and part **(b)** above implies that

$$V(I(S_1)) \cup V(I(S_2)) = S_1 \cup S_2.$$

Applying the above logic sequentially to $S_3, \ldots, S_P$ implies the result. The uniqueness follows from the fact that a minimal decomposition of a variety S into irreducibles $S_1, \ldots, S_P$ is unique up to the order of the S_i.

(d) It follows from part **(c)** and the fact that $x_i^2 = x_i$ and $(1 - x_i)^2 = (1 - x_i)$. ■

We illustrate the construction with the following example.

Example 15. Consider the set of integer points in $\{0, 1\}^3$,

$$S = \{(0, 0, 1), (1, 0, 0)\}.$$

Let $S_1 = \{(0, 0, 1)\}$, and $S_2 = \{(1, 0, 0)\}$. Observe that $I(S_1) =< x_1, x_2, x_3 - 1 >$ and $I(S_2) =< x_1 - 1, x_2, x_3 >$. Therefore, $I(S)$ is

$$< x_1^2 - x_1, x_1x_2, x_1x_3, x_1x_2 - x_2, x_2^2, x_2x_3, x_1x_3 - x_3 - x_1 + 1, x_2x_3 - x_2, x_3^2 - x_3 >$$

which yields the following Groebner basis

$$I(S) =< x_1 + x_3 - 1, x_3^2 - x_3, x_2, x_1^2 - x_1 > .$$

Although this is a Groebner basis, it, however, is not the reduced Groebner basis. Nevertheless, if we assume that $x_3 > x_1$ in term order, then we may remove $x_3^2 - x_3$ to obtain

$$I(S) =< x_3 + x_1 - 1, x_2, x_1^2 - x_1 > .$$

This is in fact the unique (for this term order) reduced Groebner basis.