AAECC 8, 241–256 (1997)

AAECC
Applicable Algebra in Engineering, Communication and Computing

Truncated Gröbner Bases for Integer Programming

R. R. Thomas[1], **R. Weismantel**[2]

[1] Department of Mathematics, Texas A&M University, College Station, TX 77843, USA (e-mail: rekha@math.tamu.edu)
[2] ZIB-Berlin, Takustrasse 7, D-14195 Berlin, Germany (e-mail: weismantel@zib-berlin.de)

Received: April 10, 1995; revised version: February 6, 1996

Abstract. The toric ideal I_A of a matrix $A = (a_1, \ldots, a_n) \in \mathbb{Z}^{d \times n}$ is the kernel of the monoid algebra map $\hat{\pi}_A : k[x_1, \ldots, x_n] \to k[t_1^{\pm 1}, \ldots, t_d^{\pm 1}]$, defined as $x_j \mapsto t^{a_j}$. It was shown in [4] that the reduced Gröbner basis of I_A, with respect to the weight vector c, can be used to solve *all* integer programs *minimize* $\{cx : Ax = b, x \in \mathbb{N}^n\}$, denoted $IP_{A,b,c,=}$, as b varies. In this paper we describe the construction of a truncated Gröbner basis of I_A with respect to c, that solves $IP_{A,b,c,=}$ for a fixed b. This is achieved by establishing the homogeneity of I_A with respect to a multivariate grading induced by A. Depending on b, the truncated Gröbner basis may be considerably smaller than the entire Gröbner basis of I_A with respect to c. For programs of the form *maximize*$\{cx : Ax \leqq b, x \leqq u, x \in \mathbb{N}^n\}$ in which all data are non-negative, this algebraic method gives rise to a combinatorial algorithm presented in [17].

Keywords: Integer programming, Toric ideal, Truncated Gröbner bases, Truncated Buchberger algorithm, Multivariate grading.

1 Introduction

We study integer programs of the form *minimize* $\{cx : Ax = b, x \in \mathbb{N}^n\}$, denoted $IP_{A,b,c,=}$, where the *coefficient matrix* $A = (a_1, \ldots, a_n) \in \mathbb{Z}^{d \times n}$ is a fixed matrix of full row rank, the *right hand side vector* $b \in \mathbb{Z}^d$, and the *cost vector* $c \in \mathbb{R}^n$. A vector $x \in \mathbb{N}^n$ such that $Ax = b$ is called a *solution* of $IP_{A,b,c,=}$. A solution y such that $cy \leqq cx$ for all $x \in \mathbb{N}^n$, $Ax = b$, is an *optimal solution* for $IP_{A,b,c,=}$ and cy, the cost value of y, is called the *optimal value* of $IP_{A,b,c,=}$. We say that $IP_{A,b,c,=}$ is *feasible*, or b is feasible if, $IP_{A,b,c,=}$ has at least one solution. The integer program $IP_{A,b,c,=}$ is assumed to be bounded (i.e, $IP_{A,b,c,=}$ has a bound optimal solution) and the cost vector c is assumed to induce a linear order on $\mathbb{N}^n$ via cx. (If a given vector c does not induce a linear order on $\mathbb{N}^n$, we use the lexicographic order on $\mathbb{N}^n$ to break ties among points with the same cost value. In what follows, c is always assumed to be a total order on $\mathbb{N}^n$. This refinement of c creates a unique optimal solution for $IP_{A,b,c,=}$, although the optimal value of the integer program is unchanged.) In [4], Conti and Traverso describe a Gröbner basis algorithm to solve all programs $IP_{A,b,c,=}$ as b varies. Their algorithm requires the computation of the *reduced Gröbner basis* with respect to the cost vector c, of the *toric ideal* I_A associated with A. Gröbner basis algorithms for finding non-negative integer solutions to systems of linear equations were also given by Pottier [9], [10] and Ollivier [11].

A set $T \subseteq \mathbb{Z}^n$ is a *test set* for the family of integer programs $\{IP_{A,b,c,=}, \forall b \in \mathbb{Z}^d\}$ if, for each non-optimal solution x to a program in this family, there exists $v \in T$ such that $x - v$ is a solution for the same program with $cx > c(x - v)$. See [7], [12] and [13] for finite test sets in integer programming. The algebraic algorithm in [4] allows a geometric interpretation which has been worked out in [16]. The geometry recognizes the reduced Gröbner basis produced by the Conti-Traverso algorithm as a minimal test set for the above family of integer programs. These test sets can be computed in practice by using a computer algebra package like MACAULAY [1], or the software GRIN [8] which is a specialized implementation of Gröbner bases for integer programming.

The computation of the entire reduced Gröbner basis associated with the family of programs $\{IP_{A,b,c,=}, \forall b \in \mathbb{Z}^d\}$, is often expensive or impossible. In practice, one is often interested in solving $IP_{A,b,c,=}$ for a fixed right hand side vector b, which typically requires only a subset of the entire Gröbner basis. In this paper, we provide a *truncated Buchberger algorithm* for toric ideals that finds a sufficient test set for $IP_{A,b,c,=}$. This set is often a proper subset of the reduced Gröbner basis of I_A, with respect to c. The algorithm follows from a *multivariate grading* induced by the matrix A, of the toric ideal I_A. This generalizes, in the case of toric ideals, the theory of truncated Gröbner bases for ideals that are homogeneous with respect to a grading given by a vector of non-negative integers (see Section 10.2 in [2]). We refer to [2] and [6] for the theory of Gröbner bases and to [14] for toric ideals, their Gröbner bases and connections to integer programming and convex polytopes.

This paper is organized as follows. In Section 2 we present the multivariate grading of I_A given by the matrix A and the truncated Buchberger algorithm to solve $IP_{A,b,c,=}$ for a fixed b. We introduce a partial order $\succeq$ on the monoid of all feasible right hand side vectors, and the truncated Buchberger algorithm, denoted *b-Buchberger*, produces a minimal test set for all programs $IP_{A,\beta,c,=}$ for which $\beta \preceq b$. The elements in the union of all test sets obtained by varying the cost function c are precisely the edge directions in the polytopes $P^I_\beta =$ *convex hull* $\{x \in \mathbb{N}^n : Ax = \beta\}$ for $\beta \preceq b$.

In Section 3 we apply the above algebraic method to the program *max*$\{cx : Ax \leqq b, x \leqq u, x \in \mathbb{N}^n\}$ for which all data are non-negative. Such programs allow for simplifications and in this case, a geometric interpretation of the truncated Buchberger algorithm gives a combinatorial algorithm presented in [17].

2 A Truncated Buchberger Algorithm for Toric Ideals

As earlier, let $IP_{A,b,c,=}$ denote the integer program minimize $\{cx : Ax = b, x \in \mathbb{N}^n\}$ satisfying all the stated assumptions. The program $IP_{A,b,c,=}$ is feasible if and only if b lies in the monoid $\mathscr{C}_{\mathbb{N}}(A) = \{\sum_{i=1}^n m_i a_i : m_i \in \mathbb{N}\} \subseteq \mathbb{Z}^d$. We assume that $\mathscr{C}(A) = \{\sum_{i=1}^n r_i a_i : r_i \in \mathbb{R}_+\}$ is a pointed cone in $\mathbb{R}^d$ and that $\{x \in \mathbb{R}^n_+ : Ax = 0\} = \{0\}$. The latter assumption ensures that $P^I_b =$ *convex hull* $\{x \in \mathbb{N}^n : Ax = b\}$ is a polytope for all $b \in \mathscr{C}_{\mathbb{N}}(A)$.

The matrix A induces a monoid homomorphism $\pi_A : \mathbb{N}^n \to \mathbb{Z}^d$ given by $\pi_A(u) = Au$. This lifts to the homomorphism of monoid algebras $\hat{\pi}_A : k[x_1, \ldots, x_n] \to k[t_1^{\pm 1}, \ldots, t_d^{\pm 1}]$ where $x_j \mapsto t^{a_j} = t_1^{a_{1j}} \ldots t_d^{a_{dj}}$. The toric ideal of A is the prime ideal $I_A = kernel(\hat{\pi}_A) \subset k[x_1, \ldots, x_n]$.

Lemma 2.1 *The toric ideal* $I_A = \bigoplus_{\beta \in \mathscr{C}_{\mathbb{N}}(A)} I_A(\beta)$ *where* $I_A(\beta)$ *is the k-vector space spanned by the binomials* $\{x^u - x^v : Au = Av = \beta, u, v \in \mathbb{N}^n\}$.

Proof. The ideal I_A is spanned as a k-vector space by the binomials $\{x^u - x^v : Au = Av,\ u, v \in \mathbb{N}^n\}$. The above decomposition is the obvious grading of I_A indexed by elements of $\mathscr{C}_{\mathbb{N}}(A)$, where the component $I_A(\beta)$ is the k-vectorspace spanned by $\{x^u - x^v : Au = Av = \beta \in \mathscr{C}_{\mathbb{N}}(A)\}$. ■

We call the polytope $P_\beta^I = convex\ hull\ \{x \in \mathbb{N}^n : Ax = \beta\}$, the *β-fiber* of π_A. Note that $x^u - x^v \in I_A(\beta)$ if and only if $u, v \in P_\beta^I \cap \mathbb{N}^n$. Hence, there is a bijection between the fibers of π_A and the components in the above direct sum via the elements in $\mathscr{C}_{\mathbb{N}}(A)$. Further, Lemma 2.1 implies that $I_A = \langle x^u - x^v : Au = Av, u, v \in \mathbb{N}^n \rangle$. Hence, I_A always has a finite generating set that consists of binomials of the above form and consequently, every reduced Gröbner basis of I_A again consists of such binomials.

The Conti-Traverso algorithm to solve all programs in the family $\{IP_{A,b,c,=}, \forall b \in \mathscr{C}_{\mathbb{N}}(A)\}$, involves the following two steps:

Step 1. Compute the reduced Gröbner basis $\mathscr{G}_c$ of the toric ideal I_A, with respect to the (refined) weight vector c.
Step 2. For a specified right hand side vector b, compute the *normal form* modulo $\mathscr{G}_c$ (remainder on division by elements in $\mathscr{G}_c$), of the monomial x^u, where u is any feasible solution to $IP_{A,b,c,=}$. The exponent vector of this normal form is the unique optimal solution of $IP_{A,b,c,=}$.

The Conti-Traverso algorithm achieves Step 1 by a Gröbner basis computation on a larger ideal in the polynomial ring $k[t_0, t_1, \ldots, t_d, x_1, \ldots, x_n]$. Methods for finding a generating set and Gröbner bases for I_A that operate entirely in the polynomial ring $k[x_1, \ldots, x_n]$ can be found in [5] and [8].

The reduced Gröbner basis $\mathscr{G}_c$ of I_A is a test set for all integer programs $IP_{A,b,c,=}$ such that $b \in \mathscr{C}_{\mathbb{N}}(A)$. For a fixed b, the set $\mathscr{G}_c$ often contains a number of elements that are not used in Step 2 of the algorithm. In the rest of this section we describe a modification of the Buchberger algorithm for I_A, inspired by Lemma 2.1, that provides a sufficient test set for $IP_{A,b,c,=}$ for fixed b. This set may be considerably smaller (depending on b) than the entire reduced Gröbner basis $\mathscr{G}_c$.

Example 2.2 Let $A = \begin{bmatrix} 3 & 1 & 11 & 2 & 3 & 5 & 3 \\ 4 & 5 & 0 & 1 & 7 & 4 & 6 \\ 5 & 6 & 1 & 9 & 2 & 3 & 3 \end{bmatrix}$ where rank(A) = 3. For the sake of clarity we associate the variables a, b, c, d, e, f, and g with the columns of A. For the cost vector $w = (23, 15, 6, 7, 1, 53, 4)$, the reduced Gröbner basis $\mathscr{G}_w$ of I_A consists of the 31 binomials: $fg^{10} - b^3ce^7$, $f^2g^5 - b^2ce^4$, $f^3 - bce$, $d^2ef^2g - a^2b^3c$, $\underline{d^2e^2f - b^4c}$, $bg^{10} - d^2e^9$, $bfg - a^2e$, $bef - ag^2$, $\underline{b^2f^2 - a^3g}$, $b^4ce^8 - ag^{12}$, $b^5c - ad^2eg^2$, $afg^7 - b^3ce^5$, $af^2g^2 - b^2ce^2$, $ae^2 - g^3$, $abg^7 - d^2e^7$, $ab^4c - d^2fg^3$, $a^2g^9 - b^4ce^6$, $a^2fg^4 - b^3ce^3$, $\underline{a^2f^2 - b^2cg}$, $a^2bg^4 - d^2e^5$, $\underline{a^2b^2f - d^2e^2g}$,

$a^3g^6 - b^4ce^4$, $a^3fg - b^3ce$, $a^3bg - d^2e^3$, $ag^{22} - b^3cd^2e^{17}$, $a^4g^3 - b^4ce^2$, $d^2f^2g^4 - a^3b^3ce$, $g^{25} - b^3cd^2e^{19}$, $a^4ef - b^3cg^2$, $a^4b - d^2eg^2$, $\underline{a^5 - b^4c}$, where the first term of each binomial is the leading monomial with respect to w.

For the right hand side vector $r = (31, 27, 38)$, the polytope P_r^I contains the four lattice points (0, 0, 1, 3, 2, 1, 1), (0, 4, 2, 1, 0, 0, 1), (2, 2, 1, 1, 0, 2, 0), and (5, 0, 1, 1, 0, 0, 1), of which (0, 0, 1, 3, 2, 1, 1) is the optimal solution to $IP_{A,r,w,=}$. Representing the solutions of $IP_{A,r,w,=}$ by the monomials cd^3e^2fg, b^4c^2dg, $a^2b^2cdf^2$, and a^5cdg, it can be seen that only the five underlined binomials in $\mathscr{G}_w$ are involved in the reduction of any non-optimal monomial to the optimum. Therefore, to solve $IP_{A,r,w,=}$ in practice, one would like to devise a method that uses r to shortcut the Buchberger algorithm to output just the five underlined binomials or some small superset of it.

Let M denote the set of all monomials in $k[x] = k[x_1, \ldots, x_n]$ where k is a field. The monoids M and $\mathbb{N}^n$ are isomorphic via the usual identification of a monomial x^u with its exponent vector. Under this identification, the monoid homomorphism π_A induces a multivariate grading of M and hence $k[x]$, where the π_A-*degree* of x^u denoted $\pi_A(x^u) = \pi_A(u) = Au \in \mathscr{C}_{\mathbb{N}}(A)$. Let $M(f)$ denote the monomials in a polynomial $f \in k[x]$.

Definition 2.3 *A polynomial* $0 \neq f \in k[x]$ *is said to be* π_A*-homogeneous if* $\pi_A(s) = \pi_A(t)$ *for all monomials* $s, t \in M(f)$. *The* π_A*-degree of a homogeneous polynomial* f, *denoted* $\pi_A(f)$, *equals the* π_A*-degree of any monomial in* $M(f)$.

A graded ideal I is homogeneous if and only if, all homogeneous components of every polynomial f in I, are also in I. Note that a polynomial $f \in I_A$ is π_A-homogeneous if and only if the exponent vectors of all monomials in $M(f)$ are lattice points in the same fiber of π_A.

Lemma 2.4 *The toric ideal* I_A *is homogeneous with respect to the grading induced by* π_A.

Proof. A binomial $x^u - x^v$ lies in I_A if and only if $u, v \in \mathbb{N}^n$ and $Au = Av$. Binomials of this form are clearly π_A-homogeneous. Let f_β denote the sum of all monomials of π_A-degree β, in a non-zero polynomial $f \in I_A$. By Lemma 2.1, f is a k-linear combination of homogeneous binomials with $f_\beta \in I_A(\beta) \subset I_A$ for all $\beta \in \mathscr{C}_{\mathbb{N}}(A)$. Hence I_A is homogeneous with respect to π_A. ■

From now on we use the word homogeneous to mean π_A-homogeneous. The above multivariate grading generalizes the usual grading of ideals by a vector of non-negative integers. An ideal that is homogeneous with respect to grading by a non-negative vector allows a natural truncation of the Buchberger algorithm that is compatible with the grading. We generalize this concept for the toric ideal I_A, which has been shown to be homogeneous with respect to the above multivariate grading. Our exposition follows Section 10.2 in [2].

Associated with the monoid $\mathscr{C}_{\mathbb{N}}(A)$ there is a "natural" partial order $\succeq$ such that for $b_1, b_2 \in \mathscr{C}_{\mathbb{N}}(A)$, $b_1 \succeq b_2$ if and only if $b_1 - b_2 \in \mathscr{C}_{\mathbb{N}}(A)$. Notice that when $\mathscr{C}_{\mathbb{N}}(A) = \mathbb{N}^d$, the partial order $\succeq$ coincides with the componentwise partial order $\geqq$, where $b_1 \geqq b_2$ if and only if $b_1 - b_2 \geqq 0$. Let $in_c(f)$ denote the initial (leading) monomial of $f \in k[x]$ with respect to the refined cost function c.

Lemma 2.5 *The following properties hold for the partial order* $\succeq$ *and the grading* π_A *of* I_A*:*

(i) *If* x^u *divides* x^v*, then* $\pi_A(x^u) \preceq \pi_A(x^v)$.

(ii) *Let* $f,\ g \in I_A$ *be homogeneous polynomials such that* $\pi_A(f) = \pi_A(g)$ *and* $f,\ g,\ f+g \neq 0$. *Then* $f+g$ *is again homogeneous with* $\pi_A(f+g) = \pi_A(f) = \pi_A(g)$.

(iii) *Let* $0 \neq f,\ g \in I_A$ *be homogeneous polynomials. Then* fg *is homogeneous with* $\pi_A(fg) = \pi_A(f) + \pi_A(g)$.

(iv) *Let* $0 \neq f, p \in I_A$ *be homogeneous polynomials and* $g \neq 0$ *be obtained by reducing* f *by* p *with respect to* c. *Then* $\pi_A(f) \succeq \pi_A(p)$ *and* g *is a homogeneous polynomial with* $\pi_A(g) = \pi_A(f)$.

Proof. We prove just (i) and (iv) since (ii) and (iii) follow from the definitions. (i) If x^u divides x^v then $v = u + w$ for some $w \in \mathbb{N}^n$. Therefore, $Av - Au = Aw \in \mathscr{C}_{\mathbb{N}}(A)$ which implies that $\pi_A(x^u) \preceq \pi_A(x^v)$. (iv) Since p reduces f, $in_c(p)$ divides some term of f. Using (i) and the homogeneity of f and p, $\pi_A(f) \succeq \pi_A(p)$. The polynomial g is again homogeneous with $\pi_A(g) = \pi_A(f)$ since reduction by a homogeneous polynomial preserves degree. ■

We are now ready to describe a truncated Buchberger algorithm for I_A called *b-Buchberger*, that produces a test set for the programs $IP_{A,\beta,c,=}$ for all $\beta \preceq b$. The normal form a binomial g, modulo a set of binomials G and cost function c, is denoted $normf_{\{G,c\}}(g)$ and the S-binomial of two binomials g_1 and g_2, with respect to c, is denoted S-$bin_c(g_1, g_2)$.

The b-Buchberger algorithm for toric ideals

Input: A finite homogeneous binomial basis F of I_A and the refined cost vector c.
Output: A truncated (with respect to b) Gröbner basis of I_A with monomial order given by c. $i = -1$, $G_0 = F$

Repeat

$i = i + 1$

$$G_{i+1} = G_i \cup (\{normf_{\{G_i,c\}}(S\text{-}bin_c(g_1, g_2)) : g_1, g_2 \in G_i,$$
$$\pi_A(S\text{-}bin_c(g_1, g_2) \preceq b)\} \setminus \{0\})$$

Until $G_{i+1} = G_i$.
Reduce G_{i+1} modulo the leading monomials of its elements.

The only difference between the usual Buchberger algorithm and the b-Buchberger algorithm described above is that the latter only considers those S-binomials with $\pi_A(S\text{-}bin_c(g_1, g_2)) \preceq b$. The algorithm terminates in finitely many steps since the Buchberger algorithm does so. Following the above notation, we may denote the usual Buchberger algorithm as ∞*-Buchberger*. Let F denote the starting binomial basis for I_A input to b-Buchberger and $\mathscr{G}_c(b)$ denote the output of the algorithm. From Lemma 2.5 we obtain the following proposition.

Proposition 2.6 (i) *Every* $g \in \mathscr{G}_c(b)$ *is a homogeneous binomial and* $\pi_A(g) \preceq b$ $\forall g \in \mathscr{G}_c(b) \backslash F$. (ii) *For all* $g_1, g_2 \in \mathscr{G}_c(b)$ *with* $\pi_A(S\text{-}bin_c(g_1, g_2) \preceq b$, $S\text{-}bin_c(g_1, g_2)$ *reduces to* 0 *modulo* $\mathscr{G}_c(b)$.

Notice that only those binomials in the starting basis F with $\pi_A(f) \preceq b$ play a role in the algorithm b-Buchberger. If there exists a binomial $f \in F$ such that $\pi_A(f)$ is not less than or equal to b with respect to the partial order $\preceq$, then it may be checked that the S-binomial formed by f and any other binomial will also inherit this property and hence will not be considered by b-Buchberger. However, $\mathscr{G}_c(b)$ generates I_A and the "passive" elements in F are carried along simply to preserve the generated ideal.

Example 2.2 continued. The twelve binomials listed in the following table form a minimal generating set for the toric ideal I_A from Example 2.2. For the right hand side vector $r = (31, 27, 38)$, an element of this minimal generating set participates in the algorithm r-Buchberger only if π_A (minimal generator) $\preceq r$, i.e., $r - \pi_A$ (minimal generator) $\in \mathscr{C}_{\mathbb{N}}(A)$. It can be seen from the table below that only the five underlined binomials are eligible to participate in r-Buchberger. The remaining seven are simply carried along by the algorithm to preserve the ideal.

binomial	π_A (binomial)	$\delta = r - \pi_A$ (binomial)	Is $\delta \in \mathscr{C}_{\mathbb{N}}(A)$?
$bce - f^3$	[15, 12, 9]	[16, 15, 29]	NO
$b^3f^2 - d^2e^3$	[13, 23, 24]	[18, 4, 14]	NO
$\underline{b^4c - d^2e^2f}$	[15, 20, 25]	[16, 7, 13]	YES
$ag^2 - bef$	[9, 16, 11]	[22, 11, 27]	NO
$ae^2 - g^3$	[9, 18, 9]	[22, 9, 29]	NO
$\underline{a^2f^2 - b^2cg}$	[16, 16, 16]	[15, 11, 22]	YES
$a^2e - bfg$	[9, 15, 12]	[22, 12, 26]	NO
$\underline{a^2b^2f - d^2e^2g}$	[13, 22, 25]	[18, 5, 13]	YES
$a^2b^3c - d^2ef^2g$	[20, 23, 29]	[11, 4, 9]	NO
$\underline{a^3g - b^2f^2}$	[12, 18, 18]	[19, 9, 20]	YES
$a^4b - d^2eg^2$	[13, 21, 26]	[18, 6, 12]	NO
$\underline{a^5 - d^2e^2f}$	[15, 20, 25]	[16, 7, 13]	YES

We call $T_c(b) = \{g \in \mathscr{G}_c(b) : \pi_A(g) \preceq b\}$ the *reduced b-Gröbner basis of I_A with respect to c*. Let $I_A[b] = \bigoplus_{\beta \preceq b} I_A(\beta)$ and $\mathscr{G}_c$ denote the usual reduced Gröbner basis of I_A with respect to c.

Proposition 2.7 *The reduced b-Gröbner basis $T_c(b) = \mathscr{G}_c \cap I_A[b]$.*

Proof. Since $\mathscr{G}_c(b)$ generates I_A, the algorithm ∞-Buchberger outputs $\mathscr{G}_c$ with starting basis $\mathscr{G}_c(b)$ and cost vector c. During the run of this algorithm, no binomial g is created such that $in_c(g)$ divides either the leading or trailing term of an element in $T_c(b)$. Suppose such a g is created and it is the first such. By Lemma 2.5, $\pi_A(g) = d \preceq b$ and hence, g is the normal form of an S-binomial $S\text{-}bin_c(g_1, g_2)$ of π_A-degree d. This implies that $\pi_A(g_1), \pi_A(g_2) \preceq d \preceq b$ and therefore, $g_1, g_2 \in T_c(b)$. But then by Proposition 2.6 (ii) and the definition of $\mathscr{G}_c(b)$, $S\text{-}bin_c(g_1, g_2)$ reduces to zero modulo $T_c(b)$. Hence no element of $T_c(b)$ is altered and no new binomial g with $\pi_A(g) \preceq b$ is created. Since $T_c(b) \subseteq \mathscr{G}_c(b)$, the result follows. ∎

The above proposition also proves that the set $T_c(b)$ is unique, although it may not be a generating set for I_A. We denote by $in_c(T_c(b))$ the set of all initial terms with respect to c, of the binomials in $T_c(b)$, and by $in_c(I_A)$ the initial ideal of I_A with respect to c.

Theorem 2.8 *The set* $T_c(b) = \mathcal{G}_c \cap I_A[b]$ *has the following properties*:

(i) *Every monomial* $s \in in_c(I_A)$ *such that* $\pi_A(s) \preceq b$ *is divisible by some* $t \in in_c(T_c(b))$.
(ii) *Every* $0 \neq f \in I_A[b]$ *reduces to zero modulo* $T_c(b)$.
(iii) *Every homogeneous* $f \in k[x]$ *with* $\pi_A(f) \preceq b$ *has a unique normal form modulo* $T_c(b)$.

Proof. (i) Suppose there exists some monomial $s \in in_c(I_A)$ with $\pi_A(s) \preceq b$ that is not divisible by any $t \in in_c(T_c(b))$. By Lemma 2.5 (i) and Proposition 2.7, it follows that there does not exist any $t \in in_c(\mathcal{G}_c)$ that divides s. This contradicts that $\mathcal{G}_c$ is a Gröbner basis of I_A with respect to c. Using (i) and Proposition 2.6, we get (ii) and (iii). ■

Corollary 2.9 *The set* $T_c(b)$ *is a minimal test set for all integer programs* $IP_{A,\beta,c,=}$ *with* $\beta \preceq b$.

Proof. Let u be a feasible non-optimal solution to $IP_{A,\beta,c,=}$ for some $\beta \preceq b$ for which the optimal solution is v. By Theorem 2.8, the binomial $x^u - x^v \in I_A[b]$ reduces to zero modulo $T_c(b)$ where $x^v \notin in_c(T_c(b))$. Hence, the unique normal form of x^u modulo $T_c(b)$ is x^v. This set is minimal by Proposition 2.7 since otherwise $\mathcal{G}_c$ would not be a minimal test set for the family of programs $\{IP_{A,b,c,=}, \forall b\}$. ■

Example 2.2 continued. For $r = (31, 27, 38)$ consider the five binomials $p_1 := b^4c - \underline{d^2e^2f}$, $p_2 := \underline{a^2f^2} - b^2cg$, $p_3 := \underline{a^2b^2f} - d^2e^2g$, $p_4 := a^3g - \underline{b^2f^2}$ and $p_5 := \underline{a^5} - d^2e^2f$ that are eligible to actively participate in r-Buchberger. For $w = (23, 15, 6, 7, 1, 53, 4)$, the leading terms of the binomials are the underlined monomials in the above list. The following table tabulates the S-binomials formed and the effect of truncation on them. When the degree of an S-binomial allows it to be considered for reduction by the algorithm, we record this division in the last column.

$q_{ij} := S-bin_w(p_i, p_j)$	$\pi_A(q_{ij})$	Is $r \succeq \pi_A(q_{ij})$?	effect of reduction
$q_{12} := a^2b^4cf - b^2cd^2e^2g$	[26, 32, 38]	NO	
$q_{13} := a^2b^6c - d^4e^4g$	[23, 38, 47]	NO	
$q_{14} := b^6cf - a^3d^2e^2g$	[22, 34, 40]	NO	
$q_{15} := a^5b^4c - d^4e^4f^2$	[30, 40, 50]	NO	
$q_{23} := d^2e^2fg - b^4cg$	[18, 26, 28]	YES	$q_{23} := g(-p_1)$
$q_{24} := a^5g - b^4cg$	[18, 26, 28]	YES	$q_{24} := g(p_5 - p_1)$
$q_{25} := d^2e^2f^3 - a^3b^2cg$	[25, 28, 31]	NO	
$q_{34} := a^5g - d^2e^2fg$	[18, 26, 28]	YES	$q_{34} := g(p_5)$
$q_{35} := b^2d^2e^2f^2 - a^3d^2e^2g$	[22, 34, 40]	NO	
$q_{45} := b^2d^2e^2f^3 - a^8g$	[27, 38, 43]	NO	

It may observed from above that the truncation allows only three of the S-binomials produced to be considered further and these reduce to zero modulo the existing binomials. Hence no new binomials are formed. After auto reducing the above five binomials modulo their leading terms, we obtain the reduced r-Gröbner basis $T_w(r) := \{\underline{d^2e^2f} - b^4c, \underline{b^2f^2} - a^3g, \underline{a^2f^2} - b^2cg, \underline{a^2b^2f} - d^2e^2g,$

$\underline{a^5} - b^4c\}$ which can be observed to be $\mathscr{G}_w \cap I_A[r]$ from earlier discussions of this example.

The algorithm b-Buchberger described above considers an S-binomial $g = x^u - x^v$ for reduction if and only if $\pi_A(g) = Au = Av \preceq b$. This amounts to checking feasibility of the system $\{x \in \mathbb{N}^n : Ax = b - Au\}$ which is as hard as solving the original integer program $IP_{A,b,c,=}$. Therefore, in order to implement b-Buchberger in practice, we propose two relaxations of the above check. Consider the S-binomial $g = x^u - x^v \in I_A$ for reduction if:

(i) $b - Au \in \mathscr{C}(A)$ where $\mathscr{C}(A) = \{Ax : x \in \mathbb{R}^n_+\}$. I.e., check feasibility of the linear programming relaxation of the original check.
(ii) $b - Au \in \mathscr{C}(A) \cap \mathbb{Z}A$ where $\mathbb{Z}A = \{Az : z \in \mathbb{Z}^n\}$. This is a relaxation of the original check since in general, $\mathscr{C}_{\mathbb{N}}(A)$ is strictly contained in $\mathscr{C}(A) \cap \mathbb{Z}A$.

Both the above relaxations consider all S-binomials that were considered by the original algorithm and possibly more. Hence, the output of the algorithm b-Buchberger with these relaxed checks will still provide a test set (not necessarily minimal) for all programs $IP_{A,\beta,c,=}$ with $\beta \preceq b$.

The truncated Buchberger algorithm can be sped up by applying Buchberger's first and second criteria to remove unnecessary S-binomials during the run of the algorithm. The first criterion allows $S\text{-}bin_c(g_1, g_2)$ to be discarded if, $in_c(g_1)$ and $in_c(g_2)$ are relatively prime. This condition is not affected by any truncation of the Buchberger algorithm. The second criterion states that the S-binomial $S\text{-}bin_c(g_1, g_2)$ can be discarded if there exists a binomial f in the current partial basis such that $S\text{-}bin_c(g_1, f)$ and $S\text{-}bin_c(f, g_2)$ have been already considered by the algorithm and $in_c(f)$ divides $lcm(in_c(g_1), in_c(g_2))$. We show that this criterion is also unaffected by truncation. If there exists f such that $in_c(f)$ divides $lcm(in_c(g_1), in_c(g_2))$ then $lcm(in_c(g_1), in_c(f))$ and $lcm(in_c(f), in_c(g_2))$ both divide $lcm(in_c(g_1), in_c(g_2))$. Therefore, if $\pi_A(S\text{-}bin_c(g_1, g_2)) \preceq b$, by Lemma 2.5 (i), $\pi_A(S\text{-}bin_c(g_1, f)) \preceq b$ and $\pi_A(S\text{-}bin_c(f, g_2)) \preceq b$. Hence, $S\text{-}bin_c(g_1, f)$ and $S\text{-}bin_c(f, g_2)$ are not victims of the truncation and Buchberger's second criterion can also be applied to remove unnecessary S-pairs during the run of the truncated Buchberger algorithm.

We remark that the theory of a truncated Buchberger algorithm and Gröbner basis, in the context of a multivariate grading induced by an integer matrix A, will hold for any ideal that is homogeneous with respect to this grading. The above results can be generalized to this situation.

We now examine the geometry of the elements in the set $T_c(b)$. The set $UGB_A = \cup_c \mathscr{G}_c$ is a well-defined unique finite set called the *universal Gröbner basis* of A (see [15]). This is a *universal test set* associated with A since it contains a test set for all programs $IP_{A,b,c,=}$ as b and c are varied. On similar lines we define the set $UGB_A(b) = \cup_c T_c(b)$ which we call the *universal b-Gröbner basis* of A. Clearly, $UGB_A(b)$ is a universal test set for all integer programs $IP_{A,\beta,c,=}$ with $\beta \preceq b$.

Lemma 2.10 *The set* $UGB_A(b) = UGB_A \cap I_A[b]$.

Proof. By proposition 2.7, $UGB_A(b) = \cup_c T_c(b) = \cup_c(\mathscr{G}_c \cap I_A[b]) = UGB_A \cap I_A[b]$. ∎

The above lemma implies that the set $UGB_A(b)$ is both unique and finite. The following theorem gives a geometric characterization of elements in UGB_A. A vector $v \in \mathbb{Z}^n$ is said to be *primitive* if the g.c.d. of its components is one.

Theorem 2.11 (Theorem 5.1 in [15]) *A binomial $x^{\alpha} - x^{\beta} \in UGB_A$ if and only if the vector $\alpha - \beta$ is primitive and the line segment $[\alpha, \beta]$ is an edge of the $A\alpha$-fiber of π_A.*

Corollary 2.12 *A binomial $x^{\alpha} - x^{\beta} \in UGB_A(b)$ if and only if the vector $\alpha - \beta$ is primitive and the line segment $[\alpha, \beta]$ is an edge of the $A\alpha$-fiber of π_A where $A\alpha \preceq b$.*

The Graver basis of A, introduced in [7] and denoted $\mathcal{GR}_A$, is a universal test set for all integer programs with coefficient matrix A. For $\sigma \in \{+, -\}^n$, let $K_\sigma = ker_{\mathbb{R}}(A) \cap \mathbb{R}^n_\sigma$, the pointed polyhedral cone in $\mathbb{R}^n$ obtained by intersecting the subspace $ker_{\mathbb{R}}(A)$ with the orthant in $\mathbb{R}^n$ of sign pattern σ. Let $\mathcal{H}_\sigma$ be the unique minimal *Hilbert basis* of K_σ. (A Hilbert basis [13] of a polyhedral cone $K \subseteq \mathbb{R}^n$ is a set of minimal generators over $\mathbb{N}$, for the monoid $K \cap \mathbb{Z}^n$. A pointed cone has a unique minimal Hilbert basis.) Then by definition, $\mathcal{GR}_A = \cup_\sigma \mathcal{H}_\sigma \backslash \{0\}$. We refer to [14] for details and results on the Graver basis of A. In particular, it can be shown that $UGB_A \subseteq \mathcal{GR}_A$. The binomial $g = x^u - x^v \in I_A$ (thought of as the line segment $[u, v]$), lies in the $Au = Av$-fiber of π_A. We call this fiber the *fiber of g*. By Theorem 2.11, the elements in the Graver basis of A that are in UGB_A are precisely those binomials that are edges in their fibers. Therefore, any algorithm to compute the Graver basis of A can be extended to an algorithm for computing UGB_A via a subroutine that checks whether $x^u - x^v \in \mathcal{GR}_A$ is an edge in its fiber.

We briefly describe the algorithm to compute $\mathcal{GR}_A$ and show how it may be modified to compute $UGB_A(b)$. Consider the $(d+n) \times 2n$-matrix $\Lambda(A) = \begin{pmatrix} A & \mathbf{0} \\ I_n & I_n \end{pmatrix}$, called the *Lawrence lifting* of A, where $\mathbf{0}$ is a $d \times n$ matrix of zeros and I_n is the identity matrix of size n. It may be checked that $ker_{\mathbb{Z}}(\Lambda(A)) = \{(u, -u) : u \in ker_{\mathbb{Z}}(A)\}$ and hence the toric ideal $I_{\Lambda(A)} = \langle x^p y^q - x^q y^p : p, q \in \mathbb{N}^n, Ap = Aq \rangle \subseteq k[x_1, \ldots, x_n, y_1, \ldots, y_n]$. The Lawrence lifting $\Lambda(A)$ has the property that *any* reduced Gröbner basis of $I_{\Lambda(A)}$ equals $\mathcal{GR}_{\Lambda(A)}$ as well as $UGB_{\Lambda(A)}$. See Theorem 4.1 in [15] for a proof. This along with the above discussion gives the following algorithm (Algorithm 4.3 in [15]) to compute $\mathcal{GR}_A$.

Algorithm to compute the Graver basis of A.

1. Compute the reduced Gröbner basis $\mathcal{G}_\succ$ of $I_{\Lambda(A)}$ with respect to any term order $\succ$.
2. The Graver basis of A consists of all binomials $x^p - x^q$ such that $x^p y^q - x^q y^p$ appears in $\mathcal{G}_\succ$.

Applying the decomposition in Lemma 2.1 to $I_{\Lambda(A)}$ we see that the component $I_{\Lambda(A)}(\beta, \beta')$ is the k-vector space spanned by all binomials of the form $\{x^p y^q - x^q y^p : Ap = Aq = \beta,\ p + q = \beta',\ p, q \in \mathbb{N}^n\}$. This implies that $x^p - x^q \in I_A(\beta)$ if and only if $x^p y^q - x^q y^p \in I_{\Lambda(A)}(\beta, \beta')$ for some $\beta' \in \mathbb{N}^n$. We define the *b-Graver basis* of A, denoted $\mathcal{GR}_A(b)$ to be the set $\mathcal{GR}_A \cap I_A[b]$. Let $(b, *)$-Buchberger be the truncated Buchberger algorithm on $I_{\Lambda(A)}$ that only considers those S-binomials $x^p y^q - x^q y^p$ such that $Ap \preceq b$. Note that $\pi_{\Lambda(A)}(x^p y^q - x^q y^p) = (Ap, p + q) \in \mathcal{C}_{\mathbb{N}}(A) \oplus \mathbb{N}^n$ and $(b, *)$-Buchberger only checks the first d components of $\pi_{\Lambda(A)}(x^p y^q - x^q y^p)$ in order to decide whether this S-binomial should be considered for reduction or not. An algorithm to compute $\mathcal{GR}_A(b)$ is then immediate.

Algorithm 2.13. How to compute the b-Graver basis of A.

1. Compute the test set $T_{\succ}(b,*)$ of $\Lambda(A)$ with respect to any term order $\succ$.
2. Then $\mathcal{GR}_A(b)$ consists of all binomials $x^p - x^q$ such that $x^p y^q - x^q y^p \in T_{\succ}(b,*)$.

Proof of correctness of Algorithm 2.13. By Proposition 2.7, $T_{\succ}(b,*) = \{x^p y^q - x^q y^p \in \mathcal{G}_{\succ} : Ap = Aq \preceq b\}$ where $\mathcal{G}_{\succ}$ is the reduced Gröbner basis of $I_{\Lambda(A)}$ with respect to $\succ$. By the above discussion, $\mathcal{G}_{\succ}$ is also the Graver basis of $\Lambda(A)$ and a binomial $x^p - x^q \in \mathcal{GR}_A(b)$ if and only if $x^p y^q - x^q y^p$ is in the Graver basis of $\Lambda(A)$ and $Ap = Aq \preceq b$. ■

Algorithm 2.13 and Corollary 2.12 give the following algorithm to compute $UGB_A(b)$.

Algorithm 2.14. How to compute the universal b-Gröbner basis $UGB_A(b)$.

1. Compute $\mathcal{GR}_A(b)$ using Algorithm 2.13. Then $UGB_A(b) \subseteq \mathcal{GR}_A(b)$.
2. A binomial $x^p - x^q \in \mathcal{GR}_A(b)$ is in $UGB_A(b)$ if and only if it is an edge in its fiber.

A very well-studied class of integer programs, those of the form *minimize*$(cx : Ax = b, x \in \{0,1\}^n\}$, are called 0, 1 programs. The above 0, 1 program can also be written as *minimize*$\{cx + 0s : \Lambda(A)(x,s)^t = (b,\mathbf{1})^t,\ x \in \mathbb{N}^n, s \in \mathbb{N}^n\}$, where $\mathbf{1}$ is an n-vector of ones. The reduced $(b,\mathbf{1})$-Gröbner basis of $I_{\Lambda(A)}$ would give a universal b-Gröbner basis for all 0, 1 integer programs with coefficient matrix A. Similarly, the reduced $(*,\mathbf{1})$-Gröbner basis of $I_{\Lambda(A)}$ would give a universal Gröbner basis for all 0, 1 integer programs of the above form as b and c are varied. We conclude this section with such an example.

Example 2.15 Consider the $1 \times n$ matrix $A_n = [1,2,3,\ldots,n]$ and the family of integer programs *minimize*$(cx : A_n x = b, x \in \{0,1\}^n\}$, obtained by varying the right hand side vector b in $\mathcal{C}_{\mathbb{N}}(A_n)$ and cost function c in $\mathbb{R}^n$. The reduced $(*,\mathbf{1})$-Gröbner basis of $I_{\Lambda(A_n)}$ is a universal Gröbner basis for the above family of programs. This set is precisely the set of square-free binomials in $\mathcal{GR}_{A_n}$. As an example, $\mathcal{GR}_{A_4} = \{bd - c^2,\ c^4 - d^3,\ bc^2 - d^2,\ b^2 - d,\ b^3 - c^2,\ \underline{ad - bc},\ ad^2 - c^3,\ \underline{ac - d}, ac - b^2, \underline{ab - c}, a^2 - b, a^2 d - c^2, a^2 b - d, a^3 - c, a^4 - d\}$ of which only the three underlined binomials are square free. The subset $\{ad - bc,\ ac - d,\ ab - c\}$ which may be identified with $\mathcal{GR}_{\Lambda(A_4)}(*,\mathbf{1})$, is a universal test set for the 0, 1 programs given by A_4. We tabulate below the cardinality of both $\mathcal{GR}_{A_n}$ and its subset of square-free binomials, for $n = 2,\ldots,22$. Up to $n = 13$, the computations can be done quite efficiently using MACAULAY. The computations for $n = 14,\ldots,22$ were obtained from Dimitrii V. Pasechnik for which he uses his special program for computing Hilbert bases of cones.

n	2	3	4	5	6	7	8	9	10	11	12	13	14
$\lvert\mathcal{GR}_{A_n}\rvert$	1	5	15	47	102	276	578	1261	2465	5362	9285	18900	33269
$\lvert\mathcal{GR}_{\Lambda(A_n)}(*,\mathbf{1})\rvert$	0	1	3	7	17	33	66	124	231	408	717	1229	2084

n	15	16	17	18	19	20	21	22
$\lvert\mathcal{GR}_{A_n}\rvert$	58171	99328	181514	287239	502116	775710	1239710	1956334
$\lvert\mathcal{GR}_{A(A_n)}(*, \mathbf{1})\rvert$	3480	5693	9157	14600	22911	35595	54744	83151

3 A Special Case

In this section we specialize the theory developed in the previous section to integer programs of the form $max\{cx : Ax \leqq b, x \leqq u, x \in \mathbb{N}^n\}$, denoted $IP_{A,b,c,u,\leqq}$, where all data are non-negative and integral. We show that a geometric interpretation of this specialization gives a combinatorial algorithm for $IP_{A,b,c,u,\leqq}$, presented in [17]. As before, we may assume that c has been refined to create a total order on $\mathbb{N}^n$. The non-negativity of the data ensures that $IP_{A,b,c,u,\leqq}$ is bounded with respect to every cost function. The vector of upper bounds $u \in \mathbb{N}^n$ and as before, A, b, c and u will be fixed throughout this section.

In order to apply the results of the previous section to $IP_{A,b,c,u,\leqq}$, we add slack vectors and transform the problem to $max\{cx + 0s + 0r : Ax + I_d s = b, I_n x + I_n r = u, x \in \mathbb{N}^n, s \in \mathbb{N}^d, r \in \mathbb{N}^n\}$ which we denote as $IP_{A',(b,u),c',=}$. Here I_p denotes the identity matrix of size p and s and r are slack vectors of the sizes specified. The matrix $A' = \begin{bmatrix} A & I_d & 0 \\ I_n & 0 & I_n \end{bmatrix}$ is in $\mathbb{N}^{(d+n)\times(2n+d)}$, right hand side vector $(b,u) \in \mathbb{N}^{d+n}$ and cost vector $c' = (c, 0, 0) \in \mathbb{N}^{2n+d}$. The monoid $\mathscr{C}_{\mathbb{N}}(A') = \mathbb{N}^{d+n}$ and the partial order $\succeq$ is just the componentwise partial order $\geqq$ on $\mathbb{N}^{d+n}$.

The associated monoid homomorphism $\pi_{A'} : \mathbb{N}^{2n+d} \to \mathbb{N}^{d+n}$ takes $(x,s,r) \mapsto A'(x,s,r)$. The toric ideal $I_{A'} = kernel(\hat{\pi}_{A'})$ where $\hat{\pi}_{A'} : k[x,s,r] \to k[t,z]$ such that $x_j \mapsto t^{a_j} z_j$, $s_i \mapsto t_i$ and $r_j \mapsto z_j$ for $j = 1, \ldots, n$ and $i = 1, \ldots, d$. Let J denote the polynomial ideal in $k[t,z,x,s,r]$ given by $J = \langle x_j - t^{a_j} z_j, s_i - t_i, r_j - z_j, j = 1, \ldots, n, i = 1, \ldots, d\rangle$. It follows from Theorem 2 in Section 3.3 of [6] that $I_{A'} = J \cap k[x,s,r]$.

Lemma 3.1 *The toric ideal* $I_{A'} = \langle x_j - s^{a_j} r_j, j = 1, \ldots, n\rangle \subseteq k[x,s,r]$.

Proof. The set $\mathscr{G}_> = \{\underline{x_j} - s^{a_j} r_j, \underline{t_i} - s_i, \underline{z_j} - r_j, j = 1, \ldots, n, i = 1, \ldots, d\}$ with the underlined terms as leading terms, is the reduced Gröbner basis of J with respect to any elimination order $>$ such that $t, z, x > s, r$. Hence $I_{A'} = J \cap k[x,s,r] = \langle \mathscr{G}_> \cap k[x,s,r]\rangle$. ∎

As in [15] and [16], we can think of the binomial $y^\alpha - y^\beta$ in a toric ideal $I_A \subseteq k[y] = k[y_1, \ldots, y_n]$, with no common factors in the two terms, as the vector $\alpha - \beta \in \mathbb{Z}^n$ or alternatively as the line segment $[\alpha, \beta] \subset \mathbb{R}^n$. In the case of the programs $IP_{A',(b,u),c',=}$ under consideration, we modify the usual interpretation so that a binomial $x^\alpha s^\beta r^\gamma - x^\delta s^\mu r^\lambda$ in $I_{A'}$ with no common factors in the two terms, is identified with the vector $\alpha - \delta \in \mathbb{Z}^n$, or the line segment $[\alpha, \delta] \subset \mathbb{R}^n$, by ignoring the slack components. Conversely, there exists a unique way in which a vector in $\mathbb{Z}^n$ can be interpreted as a binomial in $I_{A'}$. Given $v \in \mathbb{Z}^n$, we first write it uniquely as $v = v^+ - v^-$ where $v^+, v^- \in \mathbb{N}^n$. The binomial associated with

$[v^+, v^-]$ is then defined as $bin(v) = x^{v^+} s^{(Av)^-} r^{v^-} - x^{v^-} s^{(Av)^+} r^{v^+}$. It can be seen that the two terms in $bin(v)$ have no common factors and that the above choice of slack exponents is the smallest possible that will ensure $bin(v) \in I_{A'}$. Given two vectors $v, w \in \mathbb{Z}^n$ and the refined cost function c, S-$bin_c(bin(v), bin(w))$ equals $bin(v - w)$ up to sign and multiplication by a monomial.

The (b,u)-fiber of $\pi_{A'}$ is the polytope $P^I_{(b,u)} = conv\{(x,s,r) \in \mathbb{N}^{2n+d}: Ax + I_d s = b,\ I_n x + \mathrm{I}_n r = u\}$ and let $Q^I_{(b,u)} = conv\{x \in \mathbb{N}^n: Ax \leqq b, x \leqq u\}$. Under the above interpretation of binomials in $I_{A'}$, the generators of $I_{A'}$ in Lemma 3.1 are the n unit vectors in $\mathbb{R}^n$.

Observation 3.2 *There exists a connected undirected graph in every polytope $Q^I_{(b,u)}$ for $b \in \mathbb{N}^d$ and $u \in \mathbb{N}^n$, where the nodes are the lattice points in $Q^I_{(b,u)}$ and edges are translations of the unit vectors in $\mathbb{R}^n$.*

The above observation follows from the non-negativity of the data since one can construct a path from every lattice point $x \in Q^I_{(b,u)}$ to the origin by consecutively subtracting unit vectors and keeping all intermediate points in $Q^I_{(b,u)}$. The observation also follows from a general fact about generating sets for toric ideals: a set of binomials $\{y^{\alpha_i} - y^{\beta_i},\ A\alpha_i = A\beta_i,\ \alpha_i, \beta_i \in \mathbb{N}^n,\ i = 1, \ldots, p\}$ generates the toric ideal $I_A \subseteq k[y]$ if and only if, in every fiber of π_A, we can build a connected (undirected) graph in which nodes are the lattice points in the fiber and edges are translations of the segments $[\alpha_i, \beta_i]$. The argument is completed by noting the bijection between lattice points in $P^I_{(b,u)}$ and $Q^I_{(b,u)}$.

We now show that a number of algebraic operations required in Section 2 can be reduced to easy checks on vectors, for the programs $IP_{A,b,c,u,\leqq}$. As in the previous section, $\pi_{A'}$ defines a multivariate grading of $I_{A'}$ under which the degree of $bin(v)$ is $\pi_{A'}\ (bin(v)) = \begin{pmatrix} Av^- + (Av)^+ \\ v^+ + v^- \end{pmatrix}$. However, $Av^- + (Av)^+ = max\ \{Av^+, Av^-\}$ where max computes the componentwise maximum of vectors. The (b,u)-Buchberger algorithm considers the S-binomial $bin(v)$ for reduction if and only if $\pi_{A'}(bin(v)) \leqq (b,u)$. This yields the following lemma.

Lemma 3.3 *An S-binomial of the form bin(v) will be considered for reduction by the algorithm (b,u)-Buchberger if and only if $Av^+ \leqq b$, $Av^- \leqq b$ and $0 \leqq v^+, v^- \leqq u$.*

In this section we will assume that all S-binomials considered are of the form $bin(v)$, i.e., the common terms in the two monomials have been removed. This is not required for the truncated Buchberger algorithm described in the previous section. We do this here in order to be able to store a binomial without ambiguity, as a vector equal to the difference of its exponent vectors. A vector $v \in \mathbb{Z}^n$ satisfies $Av^+ \leqq b$, $Av^- \leqq b$ and $0 \leqq v^+, v^- \leqq u$ if and only if v is the difference of two feasible solutions of $IP_{A,b,c,u,\leqq}$. Therefore, the algorithm (b,u)-Buchberger considers an S-binomial $bin(v)$ only if, v is the difference of two feasible solutions of $IP_{A,b,c,u,\leqq}$. As remarked earlier, for a general integer matrix A and right hand side vector b, checking whether the π_A-degree of an S-binomial is less than or equal to b with respect to the partial order $\succeq$, amounts to checking feasibility of an integer program. In the case of the programs $IP_{A',(b,u),c',=}$ studied here, this check reduces to the above easy check on the vectors v^+ and v^-. This allows the algorithm (b,u)-Buchberger to be implemented without relaxations.

Since $IP_{A',(b,u),c',=}$ is a maximization problem, if $cv > 0$, the leading term of the binomial $bin(v)$ with respect to c is $x^{v^-} s^{(Av)^+} r^{v^+}$ which is the monomial corresponding to v^-. Therefore, the binomial $bin(v)$ reduces the leading term of the binomial $bin(w)$, where $cw > 0$, if $x^{v^-} s^{(Av)^+} r^{v^+}$ divides $x^{w^-} s^{(Aw)^+} r^{w^+}$. We may write this as an operation between the vectors v and w. For a vector $d \in \mathbb{Z}^n$, let $d^c = d$ if $cd > 0$ and $d^c = -d$ otherwise. If v is written without a superscript, we assume $cv > 0$.

Definition 3.4 [17] *A vector $w \neq 0$ can be reduced by v if $v^+ \leqq w^+$, $v^- \leqq w^-$ and $(Av)^+ \leqq (Aw)^+$. If the above conditions are satisfied, we obtain $(w - v)^c$ by reducing w by v.*

By the above definition, v reduces w if the leading term of $bin(v)$ divides the leading term of $bin(w)$. In the usual theory of Gröbner bases, the binomial $bin(v)$ reduces $bin(w)$ if the leading term of $bin(v)$ divides either term of $bin(w)$. By the above definition, if the leading term of $bin(v)$ divides the trailing term of $bin(w)$, we would have to say that v reduces $-w$. On the same lines, we may think of the reduction of a homogeneous binomial in $I_{A'}$ by a set of homogeneous binomials in $I_{A'}$ as an operation on vectors.

The specializations of the algebra to the case of $IP_{A',(b,u),c',=}$ described above, allow the algorithm (b,u)-Buchberger to be described combinatorially. This is precisely Algorithm 3.7 in [17].

A combinatorial (b,u)-Buchberger algorithm for $IP_{A,b,c,u,\leqq}$

(1) Set $B_{old} := \emptyset$, $B := \{e_i : i = 1, \ldots, n\}$
(2) While $B_{old} \neq B$ repeat the following:
  (2.1) Set $B_{old} := B$
  (2.2) For all pairs of vectors $v, v' \in B_{old}$ such that $cv < cv'$ perform the following steps:
    (2.2.1) If $Av^+ \leqq b$, $Av^- \leqq b$, $0 \leqq v^+, v^- \leqq u$, set $r = v' - v$.
    (2.2.2) As long as possible, find $v \in B$ such that $p \in \{r, -r\}$ can be reduced by v, and replace r by $p - v$.
    (2.2.3) Set $B := B \cup \{r^c\}$.

Theorem 3.5 *The output of the combinatorial* (b,u)-Buchberger *algorithm is a minimal test set for all programs $IP_{A,b',c,u',\leqq}$ for which $b' \leqq b$ and $u' \leqq u$.*

The set of generators of the toric ideal $I_{A'}$ that is used as input to the (b,u)-Buchberger algorithm is the set $\{x_j - s^{a_j} r_j, j = 1, \ldots, n\}$. The $\pi_{A'}$-degree of $x_j - s^{a_j} r_j$ is (a_j, e_j) for $j = 1, \ldots, n$. We may assume without loss of generality that $a_j \leqq b$ and $u_j \geqq 1$ for $j = 1, \ldots, n$ since otherwise, we could have removed column j from the matrix A. Therefore, all generators of $I_{A'}$ take part in the algorithm (b,u)-Buchberger.

We refer the reader to [17] for computational tests of the above combinatorial algorithm, on a wide variety of integer programming problems. Below we demonstrate on five test instances considered in [17], that the truncated Gröbner bases may be considerably smaller and computed much faster than the entire reduced Gröbner basis. We use the convention that x denotes the number of rows and y the number of columns of the instance *inst.x.y*. All instances are 0/1 integer programs, i.e., the vector u of upper bounds is equal to the vector of all ones. Columns 2 and 3 give the number of elements, denoted card$(0,1)$, in the truncated Gröbner bases (truncated by u) and the time needed for this computation, in minutes: seconds, denoted time(0, 1), on a SUN Sparc 5. Accordingly, the entries in Columns 4 and 5 represent the number of elements in the full reduced Gröbner bases (without the 0,1 condition on the variables) and the corresponding running times. The two entries marked with a $*$ show the number of elements in the partial Gröbner basis after 5 hours of CPU time. The entire Gröbner basis could not be computed within this time bound.

Problem	Time(0, 1)	Card(0, 1)	Time	Cardinality
knap.10.6	0:00	59	0:00	285
knap.10.10	0:03	749		70552*
knap.10.15	4:51	9395		88581*
knap.1.20	0:01	540	0:01	747
cov.13.9	0:01	549	0:03	992

In the remainder of this section we examine the geometry of the elements in the test sets produced by (b, u)-Buchberger. We denote by $UGB_A(b,u)$, the universal (b,u)-Gröbner basis of $I_{A'}$. As before, let $Q^I_{(b',u')} = conv\{x \in \mathbb{N}^n : Ax \leqq b',\ x \leqq u'\}$ and $P^I_{(b',u')} = conv\{(x,s,r) \in \mathbb{N}^{2n+d} : Ax + I_d s = b',\ I_n x + I_n r = u'\}$. If $[p,q]$ is an edge of a polytope P where p and q are adjacent vertices of P, we say that $p - q$ (up to sign) is an edge direction of P. By Corollary 2.12, we know that the elements in $UGB_A(b,u)$, thought of as vectors in $\mathbb{Z}^{2n+d}$, are the primitive edge directions in the polytopes $P^I_{(b',u')}$ for $b' \leqq b$ and $u' \leqq u$. Since all interpretations so far were done in n-space, we think of $UGB_A(b,u)$ as a subset of $\mathbb{Z}^n$ and give an elementary combinatorial proof of the following fact.

Proposition 3.6 *The set $UGB_A(b,u)$ consists of all primitive edge directions in the polytopes $Q^I_{(b',u')}$ with $b' \leqq b$ and $u' \leqq u$.*

Proof. We start by showing that every primitive edge direction among the polytopes $Q^I_{(b',u')}$, with $b' \leqq b, u' \leqq u$ must be contained in $UGB_A(b,u)$. Let $e = y - z$ be a primitive edge direction where z and y are adjacent vertices of $Q^I_{(b',u')}$ for some $b' \leqq b$ and $u' \leqq u$. Let c be a cost function such that $cy > cz > cp$, for all $p \in \{\mathbb{N}^n \cap Q^I_{(b',u')}\} \setminus \{y,z\}$. Such a c exists since $[z,y]$ is an edge of $Q^I_{(b',u')}$. Hence the only vector that can be added to z to get an improved solution is e. Therefore, $T_c(b,u)$ and hence $UGB_A(b,u)$ must contain $e = y - z$.

Next we consider the reverse inclusion. Let $v \in UGB_A(b,u)$ and c be a cost function such that $v \in T_c(b,u)$ with $cv^+ > cv^-$. Define $u' := v^+ + v^-$ and $b' = Av^- + (Av)^+$. Since $(b',u') = \pi_{A'}(bin(v)) \leqq (b,u)$ we have $u' \leqq u$ and $b' \leqq b$. The vector $v = v^+ - v^-$ is primitive since it belongs to $UGB_A(b,u)$. We will show that $[v^+, v^-]$ is an edge of $Q^I_{(b',u')}$. Notice that v^+ and $v^- \in Q^I_{(b',u')} \cap \mathbb{N}^n$ since, $0 \leqq v^+$, $v^- \leqq v^- + v^+ = u'$, $Av^- \leqq Av^- + (Av)^+ = b'$, and $Av^+ = Av^- + (Av^+ - Av^-) = Av^- + Av = Av^- + (Av)^+ - (Av)^- \leqq Av^- + (Av)^+ = b'$.

For any $z \in Q^I_{(b',u')} \cap \mathbb{N}^n$ distinct from v^+ and v^-, we have $(z - v^-) \geqq -v^-$ since $z \geqq 0$. Also, $z \leqq u' = v^+ + v^-$ implies $(z - v^-) \leqq v^+$. Therefore $(z - v^-)^+ \leqq v^+$ and $(z - v^-)^- \leqq v^-$. Moreover, $Az \leqq b'$ implies that $Az = Av^- + A(z - v^-) = Av^- + (A(z - v^-))^+ - (A(z - v^-))^- \leqq b' = Av^- + (Av)^+$. The last relation implies that $(A(z - v^-))^+ \leqq (Av)^+$ since $(A(z - v^-))^+$ and $(A(z - v^-))^-$ have disjoint supports. Putting these arguments together we see that every $z \in Q^I_{(b',u')}$ satisfies the conditions $(z - v^-)^+ \leqq v^+$, $(z - v^-)^- \leqq v^-$ and $(A(z - v^-))^+ \leqq (Av)^+$. Therefore, if $c(z - v^-) > 0$ then $z - v^-$ reduces v and v cannot be in $T_c(b,u)$ which is a contradiction. Therefore, $cz < cv^- < cv^+$ for every $z \in Q^I_{(b',u')}$ distinct from v^+ and v^-.

Now we show that v^+ and v^- are vertices of $Q^I_{(b',u')}$. Suppose that v^+ is not a vertex of $Q^I_{(b',u')}$. Then $v^+ = \sum_{w \in W} \lambda_w w + \lambda_0 v^-$ where W is a subset of vertices in $Q^I_{(b',u')}$ and $\lambda_0 + \sum_{w \in W} \lambda_w = 1$, $\lambda_w \geqq 0$ for all $w \in W$ and $\lambda_0 \geqq 0$. Clearly, $\lambda_0 = 0$ since v^- and v^+ have disjoint supports. Therefore, $v^+ = \sum_{w \in W} \lambda_w w$ which is impossible because $cw < cv^+$ for all $w \in W$. This implies that v^+ is a vertex of $Q^I_{(b',u')}$. Similarly v^- is a vertex of $Q^I_{(b',u')}$.

It remains to be shown that the vertices v^+ and v^- are adjacent. If not, then there exists a point z on the line connecting v^+ and v^- that can be written as a convex combination of vertices in $Q^I_{(b',u')}$ different from v^+ and v^-. I.e., $z = \mu v^+ + \sigma v^-$ with $\mu + \sigma = 1$, μ, $\sigma > 0$ has a representation as $z = \sum_{w \in W} \lambda_w w$, $\sum \lambda_w = 1$, $\lambda_w \geqq 0$, with W being a subset of vertices in $Q^I_{(b',u')}$ not containing v^+ and v^-. Again, we obtain a contradiction, since $cz > cv^- > cw$ for every $w \in W$. This completes the proof. ■

Acknowledgements. We thank Karin Gatermann and Bernd Sturmfels for helpful discussions and comments. This work was done while the first author was visiting ZIB

References

1. Bayer, D., Stillman, M.: MACAULAY: A computer algebra system for algebraic geometry. Available by anonymous ftp from zariski.harvard.edu.
2. Becker, T., Weispfenning, V.: Gröbner Bases: A Computational Approach to Commutative Algebra. Berlin Heidelberg New York: Springer 1993
3. Buchberger, B.: Gröbner bases: an algorithmic method in polynomial ideal theory. In: N. K. Bose (ed.), Multidimensional Systems Theory. D. Reidel, 1985, pp. 184–232
4. Conti, P., Traverso, C.: Buchberger algorithm and integer programming. Proceedings AAECC-9 (New Orleans). Berlin Heidelberg New York: Springer 1991; LNCS **539**, 130–139
5. Di Biase, F., Urbanke, R.: An algorithm to compute the kernel of certain polynomial ring homomorphisms. Experimental Mathematics (to appear)
6. Cox, D. A., Little, J. B., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra Undergraduate Texts in Mathematics. Berlin Heidelberg New York: Springer 1992
7. Graver, J. E.: On the foundations of linear and integer programming I. Mathematical Programming **8**, 207–226 (1975)
8. Hosten, S., Sturmfels, B.: GRIN: An implementation of Gröbner bases for integer programming. In: Balas, E., Clausen, J. (eds.) Integer Programming and Combinatorial Optimization. LNCS **920**, 267–276
9. Pottier, L.: Minimal solutions of linear diophantine systems: bounds and algorithms. Proceedings RTA (Como). Berlin Heidelberg New York: Springer 1991; LNCS **488**
10. Pottier, L.: Sub-groups of $\mathbb{Z}^n$, standard bases and linear diophantine systems. Research Report 1510, I.N.R.I.A. Sophia Antipolis, 1991
11. Ollivier, F.: Canonical bases: relations with standard bases, finiteness conditions and applications to tame automorphisms. Proceedings MEGA-90, Progress in Mathematics, pp. 379–400. Basel: Birkhäuser 1991
12. Scarf, H. E.: Neighborhood systems with production sets with indivisibilities. Econometrica **54**, 507–532 (1986)

13. Schrijver, A.: Theory of Linear and Integer Programming. Series in Discrete Mathematics and Optimization. New York Wiley 1986
14. Sturmfels, B.: Gröbner Bases and Convex Polytopes, American Mathematical Society University Lecture Series, Vol. 8, Providence, RI, 1995
15. Sturmfels, B., Thomas, R. R.: Variation of cost functions in integer programming. Technical Report, School of Operations Research, Cornell University, 1994, Mathematical Programming (to appear)
16. Thomas, R. R.: A geometric Buchberger algorithm for integer programming. Math. Operations Res. **20**(4) 864–884 (1995)
17. Urbaniak, R., Weismantel, R., Ziegler, G.: A variant of Buchberger's algorithm for integer programming, preprint SC 94-29, Konrad-Zuse-Zentrum Berlin, 1994. SIAM J. on Discrete Mathematics (to appear)