Annals of Operations Research 108, 87–109, 2001

Minimised Geometric Buchberger Algorithm for Integer Programming

QIANG LI
Research Center, PFU Limited, Japan

YI-KE GUO and JOHN DARLINGTON
Department of Computing, Imperial College, UK

TETSUO IDA
Institute of Information Sciences and Electronics, University of Tsukuba, Japan

Abstract. Recently, various algebraic integer programming (IP) solvers have been proposed based on the theory of Gröbner bases. The main difficulty of these solvers is the size of the Gröbner bases generated. In algorithms proposed so far, large Gröbner bases are generated by either introducing additional variables or by considering the generic IP problem $IP_{A,C}$. Some improvements have been proposed such as Hosten and Sturmfels' method (GRIN) designed to avoid additional variables and Thomas' truncated Gröbner basis method which computes the reduced Gröbner basis for a specific IP problem $IP_{A,C}(b)$ (rather than its generalisation $IP_{A,C}$). In this paper we propose a new algebraic algorithm for solving IP problems. The new algorithm, called *Minimised Geometric Buchberger Algorithm*, combines Hosten and Sturmfels' GRIN and Thomas' truncated Gröbner basis method to compute the fundamental segments of an IP problem $IP_{A,C}$ directly in its original space and also the truncated Gröbner basis for a specific IP problem $IP_{A,C}(b)$. We have carried out experiments to compare this algorithm with others such as the geometric Buchberger algorithm, the truncated geometric Buchberger algorithm and the algorithm in GRIN. These experiments show that the new algorithm offers significant performance improvement.

1. Introduction

In this paper, we consider the following IP problem:

$$IP_{A,C}(b) = \min\{Cx\colon\ Ax = b,\ x \in \mathbb{N}^n\},$$

where C is an objective vector in $\mathbb{R}^n$, A is an $m \times n$ matrix of integers, and b is a vector in $\mathbb{Z}^m$. We use $IP_{A,C}$ to denote a generic IP problem where b is not taken into account.

Recently, the tools of commutative algebra and algebraic geometry have brought new insights to IP via the theory of Gröbner bases [2]. The key idea is to encode an IP problem into a special ideal associated with the constraint matrix A and the cost (objective) function Cx. An important property of such an encoding is that its Gröbner bases correspond directly to the test sets of the IP problem. Thus, by employing an algebraic package such as MACAULAY [6] or MAPLE [7], the test sets of the IP problem can be directly computed. Using a proper test set (such as the minimal test set which corresponds directly to the reduced Gröbner basis of the encoded ideal), the optimal value of the cost function can be computed by constructing a monotonic path from the initial non-optimal solution of the problem to the optimal solution. Thus, IP problems can be solved in a similar fashion to the simplex method for linear programming without using intensive heuristic searching algorithms.

There are two strategies for encoding an IP problem into a special ideal.

- Indirect encoding: encoding by adding extra variables.
- Direct encoding: encoding without adding extra variables.

The first strategy was originally given by Conti and Traverso [3]. The scheme involves two encoding mechanisms: encoding the cost function of $IP_{A,C}$ into a linear order and encoding the coefficient matrix into a polynomial ideal. With this translation, IP problems are transformed into solving the subalgebra membership problems (see section 2.1 in detail).

In [15], Thomas proposed a geometric interpretation of Conti–Traverso method. The key idea of Thomas' *Geometric Buchberger Algorithm* (GBA) is to relate the Gröbner bases of the encoded polynomial ideal of an IP problem $IP_{A,C}$ to the notion of test sets for $IP_{A,C}$. Each binomial is now directly interpreted as directed line segments, i.e., vectors, in a lattice of all feasible solutions of $IP_{A,C}$. The Buchberger algorithm is then directly applied to a directed graph, where nodes of the graph are lattice points corresponding to feasible solutions of $IP_{A,C}$ and the edges at the beginning correspond to the input basis of the binomial ideal I. Finding the reduced Gröbner basis amounts to rebuilding the graph such that the edges correspond to the members of the reduced Gröbner basis of I, which can be geometrically understood as a test set of the IP problem. Thus, by this graph, an optimal solution of $IP_{A,C}$ can be found along the directed path in the graph from a feasible solution. Thomas' work provides not only a succinct understanding of an algebraic IP solver but also a practical computational procedure for its implementation. In particular, this "generate and test" approach provides great inherent parallelism. In [9], we presented a parallel implementation of GBA on a Fujitsu AP1000+. The experiment showed that the algebraic approach towards IP provides a very promising mechanism. It also showed that the new method can be improved in various ways.

In the above strategy, the first problem is that the strategy is applied to an extended IP (EIP) with additional variables (y), of the form:

$$\min\{My + Cx\}$$

subject to $Iy + Ax = b$ and $(y, x) \in \mathbb{Z}^{m+n}$. I is the $m \times m$ identity matrix and $M \in \mathbb{R}^m$ is a vector whose components have large magnitude (it is assumed, without loss of generality, that all entries in A, C and b are nonnegative integers). In practice the additional variables will lead to a considerable increase in the space and time requirements of the algorithms considered.

The second problem is that the test set generated by both algorithms are generic in the sense that it is only determined by A and C for an IP problem $IP_{A,C}$. Thus, the search space for computing the reduced Gröbner basis for such a generalised problem is quite large. In [16], Thomas proposed the "Truncated Göbner basis" method by fixing b to reduce the cardinality of the reduced Gröbner basis, but the size of Gröbner basis computed by the algorithm is still not optimal since the basis is for the EIP with respeet to $IP_{A,C}(b)$, not for $IP_{A,C}(b)$ itself. So, many vectors in the reduced Gröbner basis are needed to move from an initial solution of EIP to an initial solution of IP.

The second strategy was given by Hosten and Sturmfels. In [8], Hosten and Sturmfels proposed an algorithm in which a set of fundamental segments of $IP_{A,C}$ can be computed without going through EIP. This algorithm starts with a basis for the lattice $\ker(A)$ and then proceeds to refine this to a set of fundamental segments for $IP_{A,C}$. But the basis constructed by the algorithm is not a truncated basis since the vector b is not taken into account. Thus, the efficiency is still a problem when the method is applied to large scale IP problems due to the complexity of the Buchberger algorithm.

In this paper we propose a new algebraic algorithm for solving integer programming. The new algorithm, called the *Minimised Geometric Buchberger Algorithm* (MGBA), combines Hosten and Sturmfels' method GRIN and Thomas' truncated Gröbner basis method to compute the fundamental segments of an IP problem $IP_{A,C}$ directly in its original space and also the truncated Gröbner basis for the fixed b.

This paper is organized as follows. In section 2 and section 3, we give a brief sketch of the approaches of strategy 1 and strategy 2, respectively. Section 4 introduces the idea of the truncated Gröbner basis method. In section 5, we present the new Buchberger algorithm to compute a test set for IP. We also show some computational results to compare MGBA with other algorithms such as the algorithm in GRIN, the geometric Buchberger algorithm and the truncated geometric Buchberger algorithm in section 6. Finally, we draw a conclusion in section 7.

2. Strategy 1: indirect encoding

In this strategy, we first translate IP into the extended IP (EIP) by introducing additional variables y:

$$\mathrm{EIP}(b) = \min\bigl\{My + Cx:\ Iy + Ax = b,\ (y, x) \in \mathbb{N}^{m+n}\bigr\},$$

where I is the identity matrix, $M \in \mathbb{N}^m$ is a vector whose components have large magnitude, A is an $m \times n$ matrix of non-negative integers, and b is a vector of non-negative integers. We use $\mathrm{EIP}_{A,C}$ to denote the whole family of these integer programs, with fixed A and C, but varying right-hand side b.

From EIP(b), we can see all of the programs in it are feasible: they have the obvious solution $x = 0$, $y = b$. An optimal solution will satisfy $y = 0$, $x = x_0$ if the $IP_{A,C}(b)$ is feasible, because the components of M are sufficiently large comparing to C. If $IP_{A,C}(b)$ is infeasible, then EIP(b) has an optimal solution with $y > 0$. The value of M will not affect the computation of x and y if M is greater than C. So, in actual computation, we can select any integers for M which are much greater than the maximum in C, for instance we assign $M = 100$ which is much greater than the maximum in C 3 in example 2.1 and example 2.2.

There are two approaches as follows to solve the EIP(b).

2.1. Algebraic approach

The algebraic approach was proposed by Conti and Traverso [3]. The scheme involves two encoding mechanisms:

(1) Encoding the cost function $(M\ C)$ into a linear order on $\mathbb{Z}^{m+n}$. This can be done by choosing an arbitrary term order, such as lexicographic order $\prec_o$ and use it as a "tie breaker" on the points that have the same value under function $(M\ C)$; that is, we define $\prec_{M,C}$ to encode $(M\ C)$ as a linear order on $\mathbb{Z}^{m+n}$:

$$x_1 \prec_{M,C} x_2 \quad \Longleftrightarrow \quad \begin{cases} (M\ C)x_1 < (M\ C)x_2, \\ (M\ C)x_1 = (M\ C)x_2, \quad x_1 \prec_o x_2. \end{cases}$$

(2) Encoding $(I\ A)$ into a polynomial ideal:

$$I = \langle y^{Ae_j} - x_j\colon\ j = 1, \ldots, n \rangle,$$

where e_j is the jth unit vector on $\mathbb{Z}^n$.
Actually, Ae_j is a projection of jth column of A. So,

$$I = \langle y^{a^1} - x_1, \ldots, y^{a^n} - x_n \rangle,$$

where $a^1, \ldots, a^n$ are the columns of A.

Let $\phi : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_n]$ be the image defined by

$$x_j \longmapsto y_1^{a_{1j}} \cdots y_m^{a_{mj}}.$$

So, with this translation, IP problems are transformed into solving the subalgebra membership problem for determining whether y^b is in the image of ϕ.

Let $\mathcal{G}$ be the reduced Gröbner basis of I with respect to $\prec_{M,C}$. According to Shannon and Sweedler subalgebra membership theorem [1], $g \in k[y_1, \ldots, y_m]$ is in the image of ϕ iff the remainder $r_{\mathcal{G}}(g)$ of g on division by $\mathcal{G}$ is in $k[x_1, \ldots, x_n]$. Thus, first we compute the reduced Gröbner basis of I. Then we divide y^b by $\mathcal{G}$. If $r_{\mathcal{G}}(y^b) \in k[x_1, \ldots, x_n]$ then the remainder is a monomial whose exponent vector is the optimal solution to the problem $IP_{A,C}(b)$, otherwise $IP_{A,C}(b)$ has no feasible solution.

Example 2.1.

$$\text{IP:} \quad \min\{x_1 + x_2\colon\ 3x_1 + 2x_2 = 6,\ (x_1, x_2 \in \mathbb{N})\}.$$

Firstly, we translate it into extended IP:

$$\text{EIP:} \quad \min\{100y + x_1 + x_2\colon\ y + 3x_1 + 2x_2 = 6,\ (y, x_1, x_2 \in \mathbb{N})\}.$$

Encoding $(I\ A)$ into the polynomial ideal:

$$I = \left\langle y^{(3\ 2)(1\ 0)^{\mathrm{T}}} - x_1,\, y^{(3\ 2)(0\ 1)^{\mathrm{T}}} - x_2 \right\rangle = \left\langle y^3 - x_1,\, y^2 - x_2 \right\rangle.$$

Computing the reduced Gröbner basis:

$$\mathcal{G} = \left\{ y^2 - x_2,\, yx_2 - x_1,\, yx_1 - x_2^2,\, x_2^3 - x_1^2 \right\}.$$

Computing the optimal solution of $IP_{A,C}(b)$:

$$r_{\mathcal{G}}\left(y^6\right) = x_1^2.$$

So, the optimal solution is: $x_1 = 2$, $x_2 = 0$.

2.2. Geometric approach

The Geometric Buchberger Algorithm (GBA), proposed by Thomas [15], is based on a geometric interpretation of above algebraic approach. The key idea is to relate the Gröbner basis of encoded polynomial ideal to the notion of test set of IP. A test set for an IP problem $IP_{A,C}$ is a set of vectors in $\mathbb{Z}^n$ such that for each non-optimal solution u to a problem in this family, there is at least one element g in this set such that $u - g$ has an improved cost value as compared to u.

Definition 2.1. A test set of $IP_{A,C}$, $\mathcal{G} \subseteq \ker(A) \cap \mathbb{N}^n$ such that for every feasible solution u of $IP_{A,C}(b)$ either u is the optimal solution of $IP_{A,C}(b)$ or there exists $g \in \mathcal{G}$ such that $u - g$ is also the solution and $Cu > C(u - g)$.

Thus, a test set for $IP_{A,C}$ provides an obvious algorithm to find the optimal solution of a feasible problem $IP_{A,C}(b)$. That is, starting an initial feasible solution u, a better solution can be found by moving from u a unit step along $-g$ in the test set until the optimal is reached.

By the theorem in [15], the unique minimal test set of EIP is the reduced Gröbner basis $\mathcal{G}_{M,C}$ of the polynomial ideal $I = \langle y^{Ae_j} - x_j\colon\ j = 1, \ldots, n\rangle$. Thus, the optimal solution of $IP_{A,C}(b)$ can be therefore computed by using $\mathcal{G}_{M,C}$ to improve the obvious feasible solution $(b, 0)$ to optimum $(0, v)$ of EIP(b). Here, we are really dealing with this algorithm operating on lattice vectors.

Segment vector: geometric polynomial. Each binomial in the ideal can be interpreted as a directed line segment, i.e., a vector by reading off its exponents. For example, we translate $x_1^2x_3x_5^3 - x_2^3x_4x_6^2$ directly into the vector $[(2, 0, 1, 0, 3, 0), (0, 3, 0, 1, 0, 2)]$. For a vector $d = [\alpha, \beta]$, $\alpha, \beta \in \mathbb{N}^{m+n}$, the tail $d^t = \alpha$ of the vector is more expensive than the head $d^h = \beta$ according to the order $\prec_{M,C}$ defined as follow:

$$\beta \prec_{M,C} \alpha \quad \Longleftrightarrow \quad \begin{cases} \beta(M\ C)^{\mathrm{T}} < \alpha(M\ C)^{\mathrm{T}}, \\ \beta(M\ C)^{\mathrm{T}} = \alpha(M\ C)^{\mathrm{T}}, \quad \beta \prec_o \alpha. \end{cases}$$

The generating set of the ideal I, which is called *fundamental segment*, can be easily constructed by translating each binomial $y_1^{a_{1j}}, \ldots, y_m^{a_{mj}} - x_j$ of I into a vector d_j.

S-vector: geometric S-polynomial. The geometric correspondence of S-polynomial is called S-vector of two segment vectors d_1, d_2, which is computed as the difference of the heads of two vectors yielded by translated d_1, d_2 into a fiber on which their tails meet. Here a fiber is defined as a set of feasible solutions to an IP problem. Specially, given a right-hand side vector b, the set of feasible solutions to $IP_{A,C}(b)$ is identified as b-fiber (refer to [15] for the detailed definition). The computation of S-vector is shown in figure 1. Here the fiber is 3-fiber.

Geometric reduction. A vector f is reduced by a set of vectors $\{f_i\}$. If the tail of f can be reduced by f_i ($f_i^t \leqslant f^t$), then translate f_i so that f_i^t meets f^t and replace f by $\bar{f}$ joining f^h and f_i^h of translated f_i directed from the expensive to cheap end; if the head of f can be reduced by f_i ($f_i^t \leqslant f^h$), then translate f_i so that f_i^t meets f^h and replace f by $\bar{f}$ joining f^t and f_i^h of translated f_i directed from the expensive to cheap end. Repeat this procedure until there exists no such f_i. Actually, the geometric interpretation of reducing a vector f by a set of vectors $\{f_i\}$ is constructing a path from the tail of f to the head of $\bar{f}$, as in figure 2.

Algorithm 2.1 (Geometric Buchberger algorithm). The algorithm computes the reduced Gröbner basis of the ideal I with the term order $\prec_{M,C}$.

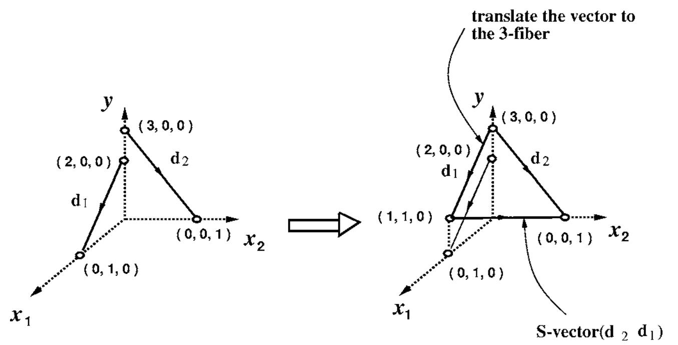

Figure 1. Computing S-vector.

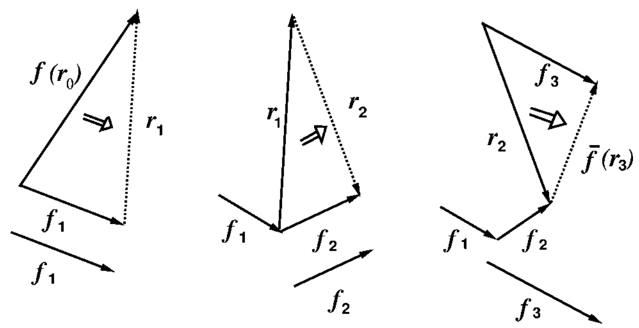

Figure 2. Geometric reduction.

First step: Construct a Gröbner basis

INPUT $F = \{d_1, \ldots, d_n\}$, the fundamental segments of EIP(b) directed according to $\prec_{M,C}$

SET $\mathcal{G}_{\text{old}} := \emptyset, \mathcal{G} := F$

REPEAT While $\mathcal{G}_{\text{old}} \neq \mathcal{G}$, repeat the following steps

$\mathcal{G}_{\text{old}} := \mathcal{G}$

(S-vector) construct the pair $g := d_i - d_j$

(reduction) reduce the vector g by the vectors in $\mathcal{G}_{\text{old}}$. If $\bar{g} \neq 0$, set $\mathcal{G} := \mathcal{G} \cup \{\bar{g}\}$.

Second step: Construct a minimal Gröbner basis

REPEAT If for some $g \in \mathcal{G}$ the tail g^t can be reduced by some $g' \in \mathcal{G}\backslash\{g\}$, then delete g from $\mathcal{G}$.

Third step: Construct the reduced Gröbner basis

REPEAT If for some $g \in \mathcal{G}$ the head g^h can be reduced by some $g' \in \mathcal{G}\backslash\{g\}$, then replace g by the corresponding reduced vector: $\mathcal{G} := \mathcal{G}\backslash\{g\} \cup \bar{g}$.

OUTPUT $\mathcal{G}_{\text{red}} := \mathcal{G}$.

Example 2.2 (The same as Example 2.1).

$$\text{EIP:} \quad \min\bigl\{100y + x_1 + x_2\colon\ y + 3x_1 + 2x_2 = 6,\ (y, x_1, x_2 \in \mathbb{N})\bigr\}.$$

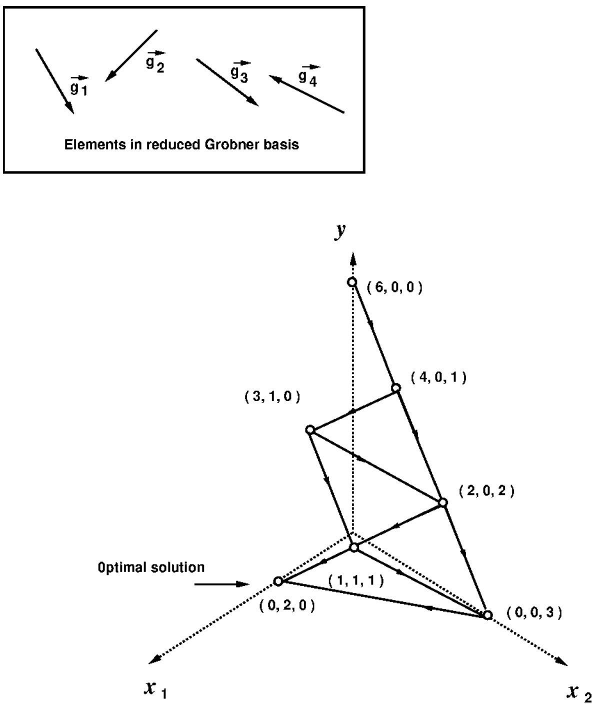

Figure 3. Example 2.2.

Computing fundamental segments by interpreting binomials of I geometrically.

$$I = \langle y^3 - x_1, y^2 - x_2 \rangle = \langle y^3 x_1^0 x_2^0 - y^0 x_1^1 x_2^0, y^2 x_1^0 x_2^0 - y^0 x_1^0 x_2^1 \rangle.$$

So, the fundamental segments are:

$$d_1 = [(3, 0, 0), (0, 1, 0)], \qquad d_2 = [(2, 0, 0), (0, 0, 1)].$$

Computing reduced Gröbner basis $\mathcal{G}_{M,C}$:

$$g_1 = [(2, 0, 0), (0, 0, 1)], \qquad g_2 = [(1, 0, 1), (0, 1, 0)],$$

$$g_3 = [(1, 1, 0), (0, 0, 2)], \qquad g_4 = [(0, 0, 3), (0, 2, 0)].$$

Deriving the optimal solution of $IP_{A,C}(b)$ from the feasible solution (6,0,0) by using $\mathcal{G}_{M,C}$, we obtain the optimal solution: $x_1 = 2, x_2 = 0$, as in figure 3.

3. Strategy 2: direct encoding

A typical approach of this strategy is the GRIN method. GRIN (GRöbner basis for INteger programming) is an experimental software system developed by Serkan Hosten and Bernd Sturmfels for computing the Gröbner basis of a toric ideal, in particular, for solving an IP problem using Gröbner bases. The algorithm in GRIN introduces a new method for computing the reduced Gröbner basis of the toric ideal which operates entirely in $k[x_1, \ldots, x_n]$ rather than in the auxiliary polynomial ring $k[y_1, \ldots, y_m, x_1, \ldots, x_n]$. In GRIN, two algorithms are implemented. Here we discuss only one.

First we give a definition of the toric ideal.

Definition 3.1. The toric ideal I_A is a binomial ideal constructed from matrix A:

$$I_A = \langle x^u - x^v \colon u, v \in N^n,\ u - v \in \ker(A) \rangle.$$

The algorithm in GRIN works in three stages: stage 1 encodes an IP problem $IP_{A,C}$ into a subideal of the toric ideal I_A; stage 2 computes the toric ideal I_A from the subideal; stage 3 computes the reduced Gröbner basis of I_A with respect to cost function Cx.

Stages 1 and 3 are easy. We can encode A into a subideal of the toric ideal I_A by finding an arbitrary lattice basis for $\ker(A)$ with some methods such as Hermit normal form algorithm [11] or Smith normal form algorithm [12]. After we get the toric ideal I_A, we can use Buchberger algorithm to compute the reduced Gröbner basis for I_A. Here we focus on stage 2 computing the toric ideal I_A from a subideal.

Definition 3.2. If f is a polynomial in $k[x_1, \ldots, x_n]$ and $J \subset k[x_1, \ldots, x_n]$ is an ideal, then the following two subsets of $k[x_1, \ldots, x_n]$ are again ideals:

$$(J : f) = \{g \in k[x_1, \ldots, x_n] \colon\ fg \in J\},$$
$$(J : f^\infty) = \{g \in k[x_1, \ldots, x_n] \colon\ f^r g \in J \text{ for some } r \in \mathbb{N}\}.$$

A basic formula involving ideal quotients is $(I : fg) = ((I : f) : g)$. A general method for computing Gröbner bases of the ideals from generators of J can be found in [4]. If J is a homogeneous ideal and f is one of the variables, say, $f = x_n$, then the algorithm for computing the Gröbner basis of the ideal from J is provided by the following lemma in [13].

First we give a definition of the graded reverse lexicographic order.

Definition 3.3 (Graded reverse lexicographic order $\succ_{\text{grevlex}}$). Let $\alpha, \beta \in \mathbb{N}^n$, we say $\alpha \succ_{\text{grevlex}} \beta$ if

$$|\alpha| = \sum_{i=1}^{n} \alpha_i > |\beta| = \sum_{i=1}^{n} \beta_i, \quad \text{or} \quad |\alpha| = |\beta| \quad \text{and} \quad \alpha \succ_{\text{revlex}} \beta,$$

where $\succ_{\text{revlex}}$ is the reverse lexicographic order [4].

We call this order graded reverse lexicographic order.

Lemma 3.1. Fix the graded reverse lexicographic order induced by $x_1 > \cdots > x_n$, and let $\mathcal{G}$ be the reduced Gröbner basis of a homogeneous ideal $J \subset k[x_1, \ldots, x_n]$. Then the set

$$\mathcal{G}' = \big\{ f \in \mathcal{G}\colon\ x_n \text{ does not divide } f \} \cup \{ f/x_n\colon\ f \in \mathcal{G} \text{ and } x_n \text{ divides } f \big\}$$

is a Gröbner basis of $(J : x_n)$. A Gröbner basis of $(J : f^{\infty})$ is obtained by dividing each element $f \in \mathcal{G}$ by the highest power of x_n that divides f.

The term order used in lemma 3.1 makes sense whenever the ideal J is homogeneous with respect to some positive grading $\deg(x_i) = d_i > 0$. By iterating the Gröbner basis computation n times with respect to different graded reverse lexicographic orders, that is, by applying lemma 3.1 one variable at a time, one can compute the ideal quotient

$$\big(J : (x_1 x_2 \cdots x_n)^{\infty}\big) = \big(\big(\cdots \big(J : {x_1}^{\infty}\big) : {x_2}^{\infty}\big) \cdots\big) : {x_n}^{\infty}\big).$$

So, if we find the relationship between the toric ideal I_A and ideal quotient, we can compute I_A and the reduced Gröbner basis of I_A. Let $B \subset \ker(A)$, we associate a subideal of I_A:

$$J_B := \big\langle x^{v^+} - x^{v^-}\colon\ v \in B \big\rangle,$$

where $v = v^+ - v^-$ is the usual decomposition into positive and negative part.

We have the following lemma whose proof is in [13].

Lemma 3.2. A subset B spans the lattice $\ker(A)$ if and only if

$$\big(J_B : (x_1 \cdots x_n)^{\infty}\big) = I_A.$$

From lemmas 3.1 and 3.3, we can prove the following proposition.

Proposition 3.1. Let $J_0 = \langle x^{v^+} - x^{v^-} \colon v \in B\rangle$ and $J_i = (J_{i-1} : x_i{}^\infty)$ $(i = 1, \dots, n)$ with the graded reverse lexicographic order by making x_i the reverse lexicographically cheapest variable. Then J_n is the toric ideal I_A.

The lemmas and proposition stated above give the following algorithm which computes a Gröbner basis of a toric ideal.

Algorithm 3.1 (Algorithm 1 in GRIN).

1. Find any lattice basis B for ker(A).
2. (Optional) Replace B by a reduced lattice basis B_{red}.
3. Let $J_0 := \langle x^{u^+} - x^{u^-} \colon u \in B_{\text{red}}\rangle$.
4. For $i = 1, \dots, n$: Compute $J_i := (J_{i-1} : x_i{}^\infty)$ using lemma 3.1, that is, by making x_i the reverse lexicographically cheapest variable.
5. Compute the reduced Gröbner basis of $J_n = I_A$ for the desired term order. If the term order is obtained from an objective function Cx, then the computed reduced Gröbner basis is the minimal test set of $IP_{A,C}$.

Example 3.1. Let $d = 4$, $n = 8$ and consider the matrix A

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 0 & 1 & 4 & 5 \\ 2 & 3 & 4 & 1 & 1 & 4 & 5 & 0 \\ 3 & 4 & 1 & 2 & 4 & 5 & 0 & 1 \\ 4 & 1 & 2 & 3 & 5 & 0 & 1 & 4 \end{pmatrix}.$$

Steps 1 and 2: Compute basis for the lattice ker(A). In this case we get the reduced basis B_{red} as follows.

$$B_{\text{red}} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & -3 & 0 & 0 & 0 & 2 \\ 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 2 & 0 & -2 & 0 & -1 & 0 & 1 & 0 \end{pmatrix}.$$

Here the basis for ker(A) is expressed as a matrix whose every row is in ker(A). In the new algorithm *Minimised Geometric Buchberger Algorithm* in section 5, we will give another expression of a basis for ker(A) which is better than this one.

Step 3: By splitting the vectors of B_{red} into positive and negative parts, we get the binomial ideal J_0 associated with B_{red}:

$$J_0 = \langle x_2x_4 - x_6x_8,\ x_2x_8^2 - x_4^3,\ x_1x_3 - x_5x_7,\ x_1^2x_7 - x_3^2x_5\rangle.$$

Step 4: In this step, we need to make eight Gröbner basis computations with respect to certain graded reverse lexicographic orders, starting with J_0. After each Gröbner basis computation we need to divide out certain variables. What we get after these eight Gröbner bases computations is a generating set of I_A.

Entering the loop in this step, we first compute the reduced Gröbner basis for J_0 with respect to the graded reverse lexicographic order that makes x_1 the cheapest variable. Here is

$$x_1 < x_2 < x_3 < x_4 < x_5 < x_6 < x_7 < x_8.$$

The result is

$$G_0 = \{x_3^3x_1 - x_7^2x_1^2, x_4^3 - x_8^2x_2, x_5x_3^2 - x_7x_1^2, x_7x_5 - x_3x_1, x_8x_6 - x_4x_2\}.$$

Next we divide each binomial in G_0 by x_1 whenever possible. So for example $x_3^3x_1 - x_7^2x_1^2$ when divided by x_1 gives $x_3^3 - x_7^2x_1$ and none of the other can be divided by x_1(the cheapest variable in the above order).

Then we get a new set J_1 which consists of all the binomials in G_0 divided by x_1 whenever possible.

$$J_1 = \langle x_3^3 - x_7^2x_1, x_4^3 - x_8^2x_2, x_5x_3^2 - x_7x_1^2, x_7x_5 - x_3x_1, x_8x_6 - x_4x_2\rangle.$$

Now we compute the reduced Gröbner basis G_1 for J_1 by using the graded reverse lexicographic order that makes x_2 the cheapest variable. For example we use the order

$$x_1 > x_8 > x_7 > x_6 > x_5 > x_4 > x_3 > x_2.$$

The result is

$$G_1 = \{x_4^3 - x_8^2x_2, x_7x_5 - x_1x_3, x_8x_6 - x_4x_2, x_1x_7^2 - x_3^3, x_1^2x_7 - x_5x_3^2, x_1^3x_3 - x_5^2x_3^2\}.$$

Dividing each binomial in G_1 by x_2 whenever possible, we get J_2:

$$J_2 = \langle x_4^3 - x_8^2x_2, x_7x_5 - x_1x_3, x_8x_6 - x_4x_2, x_1x_7^2 - x_3^3, x_1^2x_7 - x_5x_3^2, x_1^3x_3 - x_5^2x_3^2\rangle.$$

Then we can repeat the process, each time computing the reduced Gröbner basis G_i for J_i by using the graded reverse lexicographic order that makes x_{i+1} the cheapest variable and then dividing each binomial in G_i by x_{i+1} to get J_{i+1}. Finally we get J_8, that is the generating set of the toric ideal I_A.

$$J_8 = \langle x_2^4 - x_6^3x_8, x_3^3 - x_7^2x_1, x_4x_2 - x_6x_8, x_4^3 - x_2x_8^2,$$
$$x_5x_3^2 - x_7x_1^2, x_5^2x_3 - x_1^3, x_6x_4^2 - x_2^2x_8, x_6^2x_4 - x_2^3, x_7x_5 - x_3x_1\rangle.$$

Step 5: Now, we can use J_8 as a generating set to compute the reduced Gröbner basis of I_A with a fixed term order. Here, the reduced Gröbner basis of I_A with respect to the lexicographic term order given by

$$x_1 > x_2 > x_3 > x_4 > x_5 > x_6 > x_7 > x_8$$

equals

$$\mathcal{G} = \{ x_1^3 - x_3x_5^2,\ x_1^2x_7 - x_3^2x_5,\ x_1x_3 - x_5x_7,\ x_1x_7^2 - x_3^3,\ x_2^3 - x_4x_6^2,\ x_2^2x_8 - x_4^2x_6,$$
$$x_2x_4 - x_6x_8,\ x_2x_8^2 - x_4^3,\ x_3^4 - x_5x_7^3,\ x_4^4 - x_6x_8^3\}.$$

4. Truncated Gröbner bases

The computation of the entire reduced Gröbner basis associated with the family of programs $IP_{A,C}$, is often expensive or infeasible. In practice, we are often interested in solving $IP_{A,C}(b)$ for a fixed right-hand side vector b, which typically requires only a subset of the entire Gröbner basis. In [16], Thomas proposed a truncated Buchberger algorithm called b-Buchberger algorithm for toric ideals that finds a sufficient test set for $IP_{A,C}(b)$. This set is a proper subset of the reduced Gröbner basis of I_A, with respect to C. So, by the algorithm, we can produce a minimal test set for $IP_{A,C}$ whose right-hand side vector is smaller than or equal to b in a specific sense, which greatly improves the computation.

Let $C_{\mathbb{N}}(A) = \{\sum_{i=1}^{n} m_i a_i\colon\ m_i \in \mathbb{N}\}$, where a_i is the ith column of the matrix A $(i = 1, \ldots, n)$. Then $C_{\mathbb{N}}(A)$ is a monoid and the $IP_{A,C}(b)$ is feasible if and only if b lies in $C_{\mathbb{N}}(A)$. We have the following lemma:

Lemma 4.1. The toric ideal $I_A = \bigoplus_{\beta \in C_{\mathbb{N}}(A)} I_A(\beta)$ where $I_A(\beta)$ is the vector space spanned by the binomials $\{x^u - x^v\colon\ Au = Av = \beta,\ u, v \in \mathbb{N}^n\}$.

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 π_A-degree of x^u is denoted by $\pi_A(x^u) = \pi_A(u) = Au \in C_{\mathbb{N}}(A)$. Let $M(f)$ denote the monomials in a polynomial $f \in k[x]$.

Definition 4.1. 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 $M(f)$.

With above lemma 4.1 and definition 4.1, we have the following lemma:

Lemma 4.2. The toric ideal I_A is homogeneous with respect to the grading induced by π_A.

Associated with the monoid $C_{\mathbb{N}}(A)$ there is a "natural" partial order $\succeq$ such that for $b_1, b_2 \in C_{\mathbb{N}}(A)$, $b_1 \succeq b_2$ if and only if $b_1 - b_2 \in C_{\mathbb{N}}(A)$. Notice that when $C_{\mathbb{N}}(A) = \mathbb{N}^m$, the partial order $\succeq$ coincides with the componentwise partial order $\geqslant$.

Based on the above lemmas, we give b-Buchberger algorithm as follows. Let $\mathrm{NF}_{\{G,>_c\}}(g)$ denote the normal form of a binomial g, modulo a set of binomials G with term order $>_c$ and $S\text{-}bin(g_1, g_2)$ denote the S-binomial of two binomials g_1 and g_2.

Algorithm 4.1 (b-Buchberger algorithm for toric ideals).

Input: A finite homogeneous binomial basis F of I_A and the term order $>_c$.

Output: A truncated (with respect to b) Gröbner basis of I_A.

$i = -1$
$G_0 = F$

Repeat

$i = i + 1$
$G_{i+1} = G_i \cup \big(\big\{\mathrm{NF}_{\{G_i,>_c\}}\big(S\text{-}bin(g_1, g_2)\big)\colon\ g_1, g_2 \in G_i, \pi_A\big(S\text{-}bin(g_1, g_2)\big) \preceq b\big\}\backslash\{0\}\big)$

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

The algorithm 4.1 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 if the system $\{x \in \mathbb{N}^n\colon\ Ax = b - Au\}$ which is as hard as solving the original IP problem $IP_{A,C}(b)$. Therefore, in order to implement the algorithm in practice, Thomas proposed two relaxations of the above check. Consider the S-binomial $g = x^u - x^v \in I_A$ for reduction if:

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

In our implementation, we use the second check by introducing the Hermite normal form, see the *Minimised Buchberger Geometric Algorithm* in section 5. When $C_{\mathbb{N}}(A) = \mathbb{N}^m$, we just use $b - Au \geqslant 0$, as the following example.

Example 4.1 (Continue with example 3.1). Consider example 3.1 in section 3. Suppose we are interested only in right-hand side vector $b = (n_1, n_2, n_3, n_4)$ which satisfies $n_1 \leqslant 8$ and $n_2 \leqslant 8$. According to the condition, the truncated Gröbner basis equals $\{x_1^3 - x_3x_5^2,\ x_1x_3 - x_5x_7,\ x_2x_4 - x_6x_8\}$. Because here $C_{\mathbb{N}}(A) = \mathbb{N}^4$, the degree Au of these three binomials are (3,6,9,12), (4,6,4,6) and (6,4,6,4), which satisfy $b - Au \geqslant 0$, while for the other seven binomials, $b - Au < 0$. Thus, they lie outside of $C_{\mathbb{N}}(A)$.

5. The Minimised Geometric Buchberger Algorithm

Combining the GRIN method, the truncated Gröbner bases and geometric Buchberger algorithm together, we propose a new Buchberger algorithm called *Minimised Geometric Buchberger Algorithm* (MGBA) for IP problems. The idea behind the new algorithm is that for a special IP problem $IP_{A,C}(b)$ with fixed b, its minimal test set corresponds to a truncated reduced Gröbner basis of the toric ideal I_A. We encode $IP_{A,C}(b)$ into a subideal of I_A first, and then compute I_A using GRIN method, finally compute the truncated reduced Gröbner basis of I_A with b-Buchberger algorithm. We formulate the algorithm in the original space of $IP_{A,C}$ without introducing any additional variables and interpret all steps of the algorithm geometrically. This truncated reduced Gröbner basis, i.e., the minimal test set for $IP_{A,C}(b)$ that we obtained is a subset of the test set for $IP_{A,C}$. So, with the new algorithm, we can achieve considerable improvements in the efficiency and applicability of the Gröbner basis technique for IP.

Before giving the description of the algorithm, we need to introduce the Hermite normal form [11] from which we compute the lattice basis of $\ker(A)$.

Definition 5.1. Let H be an $m \times m$ nonsingular integer matrix and $h_{ij} \in H$ for $i = 1, \ldots, m$ and $j = 1, \ldots, m$. H is said to be in *Hermite normal form* if:

(a) $h_{ij} = 0$ for $i < j$,

(b) $h_{ii} > 0$ for $i = 1, \ldots, m$, and

(c) $h_{ij} \leqslant 0$ and $|h_{ij}| < h_{ii}$ for $i > j$.

Definition 5.2. Let R be a nonsingular integer matrix. Then R is called unimodular if R has determinant ± 1.

Theorem 5.1. If A is an $m \times n$ integer matrix with $\operatorname{rank}(A) = m$, then there exists an $n \times n$ unimodular matrix R such that:

(a) $AR = (H, 0)$ and H is in *Hermite normal form*, and

(b) $H^{-1}A$ is an integer matrix.

$(H, 0)$ is called the *Hermite normal form* of A. In [11], is shown a polynomial-time algorithm, called Hermite normal form algorithm for finding R and H which serves as a constructive proof of theorem 5.1. It also can be shown that H is unique. We have the following theorem in [11].

Theorem 5.2. Let $S = \{x \in \mathbb{Z}^n\colon\ Ax = b\}$ and let H and $R = (R_1, R_2)$ be as in theorem 5.1, with R_1 an $n \times m$ matrix and R_2 an $n \times (n - m)$ matrix.

(a) $S \neq \emptyset$ if and only if $H^{-1}b \in \mathbb{Z}^m$.

(b) If $S \neq \emptyset$, every solution of S is of the form

$$x = R_1 H^{-1} b + R_2 z, \quad z \in \mathbb{Z}^{n-m}.$$

From theorem 5.2, we have a computation of a basis for $\ker(A)$ as stated in the following theorem:

Theorem 5.3. Let B be a basis for $\ker(A)$ and let H and $R = (R_1, R_2)$ be as in theorem 5.2. Then $B = \{r_i\colon\ r_i$ is the ith column of R_2 and $i = 1, \ldots, n - m\}$.

Proof. Suppose $x \in \ker(A)$. Then $Ax = 0$. From theorem 5.2, we have

$$x = R_1 H^{-1} 0 + R_2 w_2 = R_2 w_2,$$

where w_2 is an arbitrary $(n-m)$-vector of integers.

Thus,

$$x = \sum_{i=1}^{n-m} \nu_i r_i,$$

where r_i is the ith column of R_2 and $\nu_i \in \mathbb{Z}$, $i = 1, \ldots, n-m$.

Therefore, $B = \{r_i\colon\ r_i$ is the ith column of R_2 and $i = 1, \ldots, n-m\}$. □

So, by the Hermite normal form algorithm [11], we can compute H and $R = (R_1, R_2)$ for a matrix A, as well as a basis for $\ker(A)$.

Now we are ready to describe our new algorithm MGBA. In this algorithm, the segment vector is slightly different from one in the geometric Buchberger algorithm. For a vector $d = [\alpha, \beta]$ in our algorithm, $\alpha, \beta \in \mathbb{N}^n$ and α is more expensive than β according to the term order $\prec_c$ defined as follow:

$$\beta \prec_c \alpha \quad \Longleftrightarrow \quad \begin{cases} \beta C^{\mathrm{T}} < \alpha C^{\mathrm{T}} \\ \beta C^{\mathrm{T}} = \alpha C^{\mathrm{T}} \quad \text{and} \quad \beta \prec_o \alpha, \end{cases}$$

where $\prec_c$ is encoded from the objective function Cx of $IP_{A,C}$ and $\prec_o$ is the lexicographic order. The fundamental segments in MGBA are constructed by interpreting the binomials of toric ideal $I_A = \langle x^{u^+} - x^{u^-}\colon\ u \in \ker(A)\rangle$ geometrically.

Algorithm 5.1 (Minimised Buchberger geometric algorithm).

1. **Compute lattice basis B for ker(A)**

   (1.1) Use the Hermite normal form algorithm in [11] to compute H and $R = (R_1, R_2)$.

   (1.2) Compute the basis B:
   $B = \{r_i\colon\ r_i$ is the ith column of R_2 and $i = 1, \ldots, n-m\}$.

2. **Reduce B into reduced lattice basis B_{red}**
   Here we use *Reduced Basis Algorithm* in [11] to compute the reduced lattice basis B_{red}. The purpose of this step is to make smaller some big numbers in B so that we can speed up the following computation by using the simplified (reduced) basis.

3. **Compute toric ideal I_A**

   (3.1) Compute a subideal of I_A based on B_{red}

   $$J_0 := \left\langle x^{u^+} - x^{u^-}\colon\ u \in B_{\text{red}}\right\rangle.$$

   Then interpret each binomial in J_0 as a vector by reading off its exponents. Here we can directly translate an element of B_{red} into a vector of J_0. For example, let $u \in B_{\text{red}}$ and $u = (1, 2, -1, -2)$, then the translated vector $v = [(1, 2, 0, 0), (0, 0, 1, 2)]$.

(3.2) For $i = 1, 2, \ldots, n$: Compute $J_i := (J_{i-1} : x_i{}^{\infty})$ geometrically by making x_i the reverse lexicographically cheapest variable. Here each J_i is in geometric term, i.e., its elements are all vectors. So, we use *Geometric Buchberger Algorithm* (algorithm 2.1) to compute the reduced Gröbner basis for each J_i, $i = 1, \ldots, n$.

4. **Compute the truncated Gröbner basis $\mathcal{G}_{\prec_c}(b)$ of I_A with order $\prec_c$**
Input: generating set J_n of toric ideal I_A and term order $\prec_c$
Output: truncated reduced Gröber basis $\mathcal{G}_{\prec_c}(b)$

(4.1) Construct a Gröbner basis
In the first step of algorithm 2.1 (*Geometric Buchberger algorithm*) we add a related check of the truncated Gröbner basis into the computation of S-vector:

$$b - Au \in C(A) \cap ZA, \quad \text{where } ZA = \left\{Az\colon z \in \mathbb{Z}^n\right\}.$$

We check whether there exists a feasible solution for $S = \{x \in \mathbb{Z}^n\colon Ax = b - Au\}$ by (a) of theorem 5.2 stated as follow:

$$S \text{ is not empty if and only if } H^{-1}(b - Au) \in \mathbb{Z}^m.$$

(4.2) Construct a minimal Gröbner basis
This step is the same as the second step of algorithm 2.1.

(4.3) Construct the reduced Gröbner basis
This step is the same as the third step of algorithm 2.1.

We illustrate the whole procedure of the above algorithm by the following example.

Example 5.1. We consider the following IP problem $IP_{A,C}(b)$:

$$\begin{aligned}
&\text{minimise } x_1 + 8x_2 + 8x_3 + 16x_4 + 2x_5 + 2x_6 + 2x_7 + 2x_8\\
&\text{subject to} \quad x_1 + 2x_2 + 3x_3 + 4x_4 + x_6 + 4x_7 + 5x_8 = 7,\\
&\qquad\qquad 2x_1 + 3x_2 + 4x_3 + x_4 + x_5 + 4x_6 + 5x_7 = 7,\\
&\qquad\qquad 3x_1 + 4x_2 + x_3 + 2x_4 + 4x_5 + 5x_6 + x_8 = 13,\\
&\qquad\qquad 5x_1 + 2x_2 + 3x_3 + 4x_4 + 6x_5 + x_6 + 2x_7 + 5x_8 = 17.
\end{aligned}$$

We have the coefficient matrix A, vector C and b as follows.

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 0 & 1 & 4 & 5\\ 2 & 3 & 4 & 1 & 1 & 4 & 5 & 0\\ 3 & 4 & 1 & 2 & 4 & 5 & 0 & 1\\ 5 & 2 & 3 & 4 & 6 & 1 & 2 & 5 \end{pmatrix},$$

$$C = (1, 8, 8, 16, 2, 2, 2, 2), \qquad b = (7, 7, 13, 17).$$

Step 1.1: We use the Hermite normal form algorithm to compute H and R for A and obtain a basis B for $\ker(A)$ as follows.

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ -11 & -12 & -2 & 22 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$R = \begin{pmatrix} -3 & -3 & -2 & 7 & -3 & 0 & -4 & 0 \\ 2 & 1 & 1 & -3 & 0 & -3 & 0 & -4 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 2 & 0 & 3 & 0 \\ 0 & 1 & 0 & -1 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Then,

$$R_2 = \begin{pmatrix} -3 & 0 & -4 & 0 \\ 0 & -3 & 0 & -4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 3 & 0 \\ 0 & 2 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

From R_2 we can get the basis B for $\ker(A)$:

$$B = \{(-3, 0, 1, 0, 2, 0, 0, 0), (0, -3, 0, 1, 0, 2, 0, 0), (-4, 0, 0, 0, 3, 0, 1, 0),\\ (0, -4, 0, 0, 0, 3, 0, 1)\}.$$

Step 2: We compute the reduced basis B_{red} for $\ker(A)$ and get:

$$B_{\text{red}} = \{(-1, 0, -1, 0, 1, 0, 1, 0), (0, -1, 0, -1, 0, 1, 0, 1), (-2, 0, 2, 0, 1, 0, -1, 0),\\ (0, -2, 0, 2, 0, 1, 0, -1)\}.$$

Step 3.1: We interpret each element of B_{red} as a vector. For example, $(-1, 0, -1, 0, 1, 0, 1, 0)$ is translated to the vector

$$[(0, 0, 0, 0, 1, 0, 1, 0), (1, 0, 1, 0, 0, 0, 0, 0)].$$

So, we obtain a subideal of I_A as follow.

$$J_0 = \langle [(0, 0, 0, 0, 1, 0, 1, 0), (1, 0, 1, 0, 0, 0, 0, 0)],\\ [(0, 0, 0, 0, 0, 1, 0, 1), (0, 1, 0, 1, 0, 0, 0, 0)],$$

$$\begin{aligned}&[(0,0,2,0,1,0,0,0),(2,0,0,0,0,0,1,0)],\\&[(0,0,0,2,0,1,0,0),(0,2,0,0,0,0,0,1)]\rangle.\end{aligned}$$

Step 3.2: We compute the toric ideal I_A. First we use algorithm 2.1 to compute the reduced Gröbner basis for J_0 with respect to graded reverse lexicographic order that makes x_1 the cheapest variable. Here is $x_1 < x_2 < x_3 < x_4 < x_5 < x_6 < x_7 < x_8$. The result is

$$\begin{aligned}G_0 = \{&[(0,0,0,0,1,0,1,0),(1,0,1,0,0,0,0,0)],\\&[(0,0,0,0,0,1,0,1),(0,1,0,1,0,0,0,0)],\\&[(0,0,2,0,1,0,0,0),(2,0,0,0,0,0,1,0)],\\&[(0,0,0,2,0,1,0,0),(0,2,0,0,0,0,0,1)],\\&[(1,0,3,0,0,0,0,0),(2,0,0,0,0,0,2,0)],\\&[(0,1,0,3,0,0,0,0),(0,2,0,0,0,0,0,2)]\}.\end{aligned}$$

Next we divide each vector in G_0 by x_1 whenever possible, removing the common factor x_1 from two monomials. For example $[(1,0,3,0,0,0,0,0),(2,0,0,0,0,0,2,0)]$ when divided by x_1 gives $[(0,0,3,0,0,0,0,0),(1,0,0,0,0,0,2,0)]$ and none of the other can be divided by x_1 (the cheapest variable in the above order).

Then we get a new set J_1 which consists of all the vectors in G_0 divided by x_1 whenever possible.

$$\begin{aligned}J_1 = \langle&[(0,0,0,0,1,0,1,0),(1,0,1,0,0,0,0,0)],\\&[(0,0,0,0,0,1,0,1),(0,1,0,1,0,0,0,0)],\\&[(0,0,2,0,1,0,0,0),(2,0,0,0,0,0,1,0)],\\&[(0,0,0,2,0,1,0,0),(0,2,0,0,0,0,0,1)],\\&[(0,0,3,0,0,0,0,0),(1,0,0,0,0,0,2,0)],\\&[(0,1,0,3,0,0,0,0),(0,2,0,0,0,0,0,2)]\rangle.\end{aligned}$$

Now we compute reduced Gröbner basis G_1 for J_1 by using the graded reverse lexicographic order that makes x_2 the cheapest variable. The order is $x_1 > x_8 > x_7 > x_6 > x_5 > x_4 > x_3 > x_2$. The result is

$$\begin{aligned}G_1 = \{&[(0,0,0,0,1,0,1,0),(1,0,1,0,0,0,0,0)],\\&[(0,0,0,0,0,1,0,1),(0,1,0,1,0,0,0,0)],\\&[(2,0,0,0,0,0,1,0),(0,0,2,0,1,0,0,0)],\\&[(0,0,0,2,0,1,0,0),(0,2,0,0,0,0,0,1)],\\&[(1,0,0,0,0,0,2,0),(0,0,3,0,0,0,0,0)],\\&[(0,1,0,3,0,0,0,0),(0,2,0,0,0,0,0,2)],\\&[(3,0,1,0,0,0,0,0),(0,0,2,0,2,0,0,0)]\}.\end{aligned}$$

Dividing each binomial in G_1 by x_2 whenever possible, we get J_2:

$$\begin{aligned}
J_2 = \langle &[(0,0,0,0,1,0,1,0),(1,0,1,0,0,0,0,0)],\\
&[(0,0,0,0,0,1,0,1),(0,1,0,1,0,0,0,0)],\\
&[(2,0,0,0,0,0,1,0),(0,0,2,0,1,0,0,0)],\\
&[(0,0,0,2,0,1,0,0),(0,2,0,0,0,0,0,1)],\\
&[(1,0,0,0,0,0,2,0),(0,0,3,0,0,0,0,0)],\\
&[(0,0,0,3,0,0,0,0),(0,1,0,0,0,0,0,2)],\\
&[(3,0,1,0,0,0,0,0),(0,0,2,0,2,0,0,0)]\rangle.
\end{aligned}$$

Then we can repeat the process, each time computing reduced Gröbner basis G_i for J_i by using graded reverse lexicographic order that makes x_{i+1} the cheapest variable and then dividing each vector in G_i by x_{i+1} to get J_{i+1}. Finally we get J_8, that is the the toric ideal I_A.

$$\begin{aligned}
J_8 = \langle &[(0,0,0,0,1,0,1,0),(1,0,1,0,0,0,0,0)],\\
&[(0,1,0,1,0,0,0,0),(0,0,0,0,0,1,0,1)],\\
&[(0,0,2,0,1,0,0,0),(2,0,0,0,0,0,1,0)],\\
&[(0,0,0,2,0,1,0,0),(0,2,0,0,0,0,0,1)],\\
&[(0,0,3,0,0,0,0,0),(1,0,0,0,0,0,2,0)],\\
&[(0,0,0,3,0,0,0,0),(0,1,0,0,0,0,0,2)],\\
&[(0,0,1,0,2,0,0,0),(3,0,0,0,0,0,0,0)],\\
&[(0,0,0,1,0,2,0,0),(0,3,0,0,0,0,0,0)],\\
&[(0,4,0,0,0,0,0,0),(0,0,0,0,0,3,0,1)]\rangle.
\end{aligned}$$

Step 4: In this step, we can use J_8 as the fundamental segments to compute truncated reduced Gröbner basis of I_A with fixed right-hand side b and cost function Cx. The test set for $IP_{A,C}(b)$, i.e., the truncated reduced Gröbner basis is:

$$\begin{aligned}
\mathcal{G}_{>_c}(b) = \{&[(1,0,1,0,0,0,0,0),(0,0,0,0,1,0,1,0)],\\
&[(0,1,0,1,0,0,0,0),(0,0,0,0,0,1,0,1)],\\
&[(0,0,2,0,1,0,0,0),(2,0,0,0,0,0,1,0)],\\
&[(0,0,0,2,0,1,0,0),(0,2,0,0,0,0,0,1)],\\
&[(0,0,3,0,0,0,0,0),(1,0,0,0,0,0,2,0)],\\
&[(0,0,0,3,0,0,0,0),(0,1,0,0,0,0,0,2)],\\
&[(0,0,1,0,2,0,0,0),(3,0,0,0,0,0,0,0)],\\
&[(0,3,0,0,0,0,0,0),(0,0,0,1,0,2,0,0)]\}.
\end{aligned}$$

For any feasible solution of the problem $IP_{A,C}(b)$, we can derive an optimal solution by using the above test set to reduce this feasible solution. For example, we have a feasible solution:

$$x_1 = 1, \quad x_2 = 1, \quad x_3 = 0, \quad x_4 = 1, \quad x_5 = 1, \quad x_6 = 0, \quad x_7 = 0, \quad x_8 = 0.$$

Then we can get an optimal solution:

$$x_1 = 1, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 0, \quad x_5 = 1, \quad x_6 = 1, \quad x_7 = 0, \quad x_8 = 1.$$

Theorem 5.4. The algorithm MGBA terminates after a finite number of steps and its output is the unique minimal test set for $IP_{A,C}(b)$.

Proof. Finiteness of the algorithm is clear since the Hermite normal form algorithm, the Buchberger algorithm and b-Buchberger algorithm all terminate in finitely many steps.

By proposition 3.1, we obtain the toric ideal I_A in step 3. Because the generating set of the toric ideal I_A is a set of fundamental segments for $IP_{A,C}$, the geometric Buchberger algorithm and b-Buchberger algorithm guarantee that we can obtain the truncated reduced Gröbner basis $\mathcal{G}_{\prec_c}(b)$ for $IP_{A,C}(b)$ with the term order $\prec_c$ in step 4. Now, we study $\mathcal{G}_{\prec_c}(b)$ from a completely geometric point of view. With $\mathcal{G}_{\prec_c}(b)$, we can build a connected, directed graph for only one fiber (b-fiber) of $IP_{A,C}(b)$. The nodes of the graph are all the lattice points in the fiber and the edges are the translations of elements in $\mathcal{G}_{\prec_c}(b)$ by nonnegative integral vectors. By theorem 2.1.8 in [15], the graph has a unique sink at the unique optimum in this fiber. In this graph, there exists a directed path from every nonoptimal point to the unique optimum. So, the reduced Gröbner basis $\mathcal{G}_{\prec_c}(b)$ is a test set for $IP_{A,C}(b)$. By corollary 2.1.10 in [15], we can prove $\mathcal{G}_{\prec_c}(b)$ is the unique minimal test set for $IP_{A,C}(b)$, depending on A, $\prec_c$ and b. □

6. Implementation, experiment and comparison

We have implemented the Minimised Geometric Buchberger Algorithm in the language C and developed a solver, called MGBS (Minimised Geometric Buchberger Solver) for IP on a Sun Ultra Enterprise 3000. MGBS works in two stages: the first stage is to compute a test set (reduced Gröbner basis) $\mathcal{G}_{\prec_c}(b)$ for $IP_{A,C}(b)$ based on MGBA, the second is to find a feasible solution and derive the optimal solution for $IP_{A,C}(b)$ by using $\mathcal{G}_{\prec_c}(b)$ to reduce the feasible solution. MGBS is connected via MathLink and CGI to the modelling IP system TIP [10]. When MGBS is called in Web page with an IP model (an objective function and a set of constraints), MGBS will solve the IP model and send the result (an optimal solution or a message "no feasible solution") to the Web page via CGI.

The main difficulty with MGBS is the computation of Gröbner bases. MGBS uses conventional Gröbner basis techniques to speed up this computation whenever this is suitable. For example, we make effective use of criteria to cut down the number of

S-pairs, which is a bottleneck during the computations. However, the criteria which proved to be inefficient for the binomial case, such as Gebauer's B-criterion [5], are "switched off". It is a common strategy to keep the set of binomials throughout the entire Gröbner basis computation as reduced as possible. We implemented this idea by doing global reductions periodically (whenever new elements of the size of a fixed percentage of the current basis are created) as opposed to doing it every time a new binomial is added. Another important strategy is the extraction of common monomial factors in every newly created S-pair. This extraction is justified by the fact that the toric ideal I_A is a prime ideal not containing any common monomials. The above idea proved to be very effective, leading to reductions as much as 40–50% in execution time.

We have implemented the Geometric Buchberger Algorithm (GBA), the Truncated Geometric Buchberger Algorithm (TGBA) and the algorithm in GRIN, simply called GRIN by language C respectively on the Sun Ultra Enterprise 3000. We have carried out experiments by running MGBS on randomly generated matrices A of various sizes (ranging from 3×7 to 8×16) with nonnegative entries in a range between 0 and 20. We generate random right-hand sides b to compute truncated Gröbner basis $\mathcal{G}_{\prec_c}(b)$. For each test instance (A, C, b) three comparisons are made with GBA, TGBA and GRIN.

The result of the comparisons is summarised in table 1. The first column of the table indicates IP problems with their coefficient matrix A. For example, A3 $\times$ 7.1 represents No.1 IP problem with 3×7 integer matrix A. The range of the entries used in the problems are given in the second column. The third and fourth columns give the size of the truncated reduced Gröbner basis and the execution time for computing it for each problem with MGBA. The fifth and sixth columns give two kinds of data

Table 1
Experimental result.

Problems	Entries	MGBA		GRIN		TGBA		GBA	
		Size	Time	Size	Time	Size	Time	Size	Time
A3×7.1	0–20	17	0.34	31	0.55	212	68.84	245	420.90
A3×7.2	0–20	20	2.29	69	4.64	236	78.68	345	1129.26
A3×7.3	0–20	26	3.88	72	5.42	237	79.88	723	4746.13
A4×8.1	0–20	21	4.30	95	40.29	823	8476.86	3390	35118.24
A4×8.2	0–20	25	17.72	103	116.40	847	9913.19	3831	39426.20
A4×8.3	0–20	33	92.37	124	217.30	949	12118.89	4814	42042.16
A5×10.1	0–4	55	65.72	85	70.57	1143	826.27	1766	11887.24
A5×10.2	0–4	57	65.89	92	73.18	1171	896.76	1890	11995.97
A5×10.3	0–4	65	66.42	102	73.38	1284	930.23	2014	12207.70
A6×12.1	0–3	100	112.29	181	256.54	1300	1569.63	2353	18600.83
A6×12.2	0–3	153	189.87	418	1348.34	3304	3671.52	5026	24189.12
A6×12.3	0–3	267	358.39	709	3119.38	7872	8593.21	9590	35870.38
A8×16.1	0–1	11	6.76	20	7.46	48	13.89	63	15.67
A8×16.2	0–1	18	210.78	26	215.72	136	267.36	702	6385.31
A8×16.3	0–1	19	212.60	30	215.77	167	313.45	928	8250.13

(size and execution time) for GRIN. The seventh and eighth columns are for TGBA. The last two columns are for GBA. The timings are in CPU seconds on the Sun Ultra Enterprise 3000.

From table 1, we can see the reduced Gröbner basis in GBA is the biggest one among the all algorithms because of the introduction of additional variables. Also the execution time is longest. For TGBA, because the algorithm computes the reduced Gröbner basis by fixing b, we can see the size of the reduced Gröbner basis in TGBA is less than that in GBA. But it is greater than that in GRIN and MGBA, because TGBA still introduces the additional variables to compute the reduced Gröbner basis. The size of the reduced Gröbner basis generated by MGBA is much less than those in the other three algorithms and also the performance in MGBA is the best.

7. Conclusions

We have proposed a new algorithm called Minimised Geometric Buchberger Algorithm for integer programming. It combines the GRIN method and the truncated Gröbner bases method to compute a generating set of the Gröbner bases in the original space and then refine it into a minimal test set, i.e., a truncated reduced Gröbner basis of $IP_{A,C}(b)$ with fixed right-hand side. Our preliminary experiments indicate that the algorithm is much faster than others such as the geometric Buchberger algorithm, the truncated geometric Buchberger algorithm and the algorithm in GRIN.

At present, the application of our algorithm MGBA is confined to small and middle size of IP problems. The prototype MGBS is not competitive in computing speed to popular commercial software such as CPLEX. However, we clearly see a new direction of research in solving IP problem using the theory of Gröbner bases. We would like to emphasize the importance of the symbolic methods of applying Gröbner basis technique for solving IP problems. Especially MGBA becomes more attractive when we tackle stochastic IP problems in [14]. The property of Gröbner basis corresponding directly to the test set of the IP problem does seem particularly useful for solving the classes of stochastic problems where some or all variables are integer valued which the general numerical methods cannot handle. In this context, computing the Gröbner basis is still the main difficulty. So, with our algorithm for the test set of IP, we can provide an efficient method for solving the stochastic IP problems in [14]. Future research will be parallelisation of our algorithm and the application of the algorithm to stochastic IP problems.

References

[1] W.W. Adams and P. Loustaunau, *An Introduction to Gröbner Bases*, Vol. 3 (American Mathematical Society, 1994).

[2] B. Buchberger, Gröbner bases: an algorithm method in polynomial ideal theory, in: *Multidimensional Systems Theory*, ed. N.K. Bose (Reidel, Dordrecht, 1985) pp. 184–232.

[3] P. Conti and C. Traverso, Buchberger algorithm and integer programming, in: *Proceedings AAECC-9*, New Orleans, Lecture Notes in Comput. Sci. 539 (Springer), pp. 130–139.

[4] D. Cox, J. Little and D. O'Shea, *Ideals, Varieties and Algorithms* (Springer, New York, 1992).

[5] R. Gebauer and H.M. Möller, On an installation of Buchberger's algorithm, Journal of Symbolic Computation 6 (1988) 275–286.

[6] D.R. Grayson and M.E. Stillman, Macaulay 2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2.

[7] A. Heck, *Introduction to Maple, a Computer Algebra System* (Springer, 1993).

[8] S. Hosten and B. Sturmfels, Grin: An implementation of Gröbner bases for integer programming, in: *Integer Programming and Combinatorial Optimization*, eds. E. Balas, J. Clausen, Lecture Notes in Comput. Sci. 920 (Springer), pp. 207–276.

[9] Q. Li, Y.K. Guo and T. Ida, A parallel algebraic approach towards integer programming, in: *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems*, Washington, DC (1997) pp. 59–64.

[10] Q. Li, Y.K. Guo and T. Ida, Modelling integer programming with logic: Language and implementation, IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, to appear.

[11] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization (Wiley, New York, 1988).

[12] A. Schrijver, *Theory of Linear and Integer Programming* (Wiley, 1986).

[13] B. Sturmfels, *Gröbner Bases and Convex Polytopes*, Vol. 8 (American Mathematical Society, 1996).

[14] S.R. Tayur, R.R. Thomas and N.R. Natrj, An algebraic geometry algorithm for scheduling in presence of setups and correlated demands, Mathematical Programming 69 (1995) 369–401.

[15] R.R. Thomas, A geometric Buchberger algorithm for integer programming, Mathematics of Operations Research 20 (1995) 864–884.

[16] R.R. Thomas and R. Weismantel, Truncated Gröbner bases for integer programming, Applicable Algebra in Engineering, Communication and Computing 8 (1997) 241–257.