Gröbner Bases and Integer Programming*

Günter M. Ziegler
Fachbereich Mathematik, MA 6-1
Technische Universität Berlin
Straße des 17. Juni 136
D-10623 Berlin, Germany
ziegler@math.tu-berlin.de

May 15, 1996

1 Introduction

"Integer programming" is a basic mathematical problem, of central importance in Optimization and Operations Research. While a systematic body of theory has been developed here in the last thirty years (see Schrijver [11]), it has been realized only very recently, first by Conti & Traverso [5], that the Buchberger algorithm provides a solution strategy for the general integer programming problem, in particular in the case of families of programs with "varying right-hand side."

In my **first lecture** (see Section 2 in the following notes), I intend to give a short introduction to "what integer programming is about". Then I present a simple combinatorial-geometric version of the Buchberger algorithm applied to integer programming (Section 3), mostly following Thomas [14, 15]. While the theory of Gröbner bases [1, 2, 3, 6] yields most of the basic ideas and tools, the presentation here will stay in an "elementary geometry" setting — nevertheless I assume you will enjoy it more with the intuition from Mora's talks.

My **second lecture** will sketch some connections between the geometry of integer programs, and the combinatorics of Gröbner bases (Section 4), following Sturmfels & Thomas [12]. Finally, I will also present a simple adaption of Buchberger's algorithm that seems to be even more useful for integer programming (Section 5). The ideas for this are quite fresh, see [18]. Perhaps I can report about some computational tests (encouraging, but without a breakthrough, yet).

Some of the **problems** to this talk deal with the relation between lattice vectors and binomials in more depth, and thus with the "Gröbner basis of a lattice": this is a situation closely linked to integer programming (the "local situation", the "group problem"), but also, for example, to the ideals of toric varieties.

2 "What is Integer Programming"

A *polyhedron* P is any intersection $P := \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$ of closed halfspaces, in some $\mathbb{R}^n$. Here $A \in \mathbb{R}^{m \times n}$ is a matrix, while $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$ are column vectors. A bounded polyhe-

*Lecture Notes for the EIDMA minicourse "Computer Algebra with emphasis on discrete algebra and geometry," Eindhoven, September 1994 (slightly revised version, February 1995).

dron is a *polytope*. (We refer to [19] for a recent exposition of the geometry and combinatorics of polytopes.)

Given any linear function $\mathbf{x} \longmapsto \mathbf{c}^t\mathbf{x}$ on $\mathbb{R}^n$, the *linear programming problem* is to determine a point $\mathbf{x}_0 \in P \subseteq \mathbb{R}^n$ in the polyhedron P for which the linear function $\mathbf{c}^t\mathbf{x}$ is minimal. The points in P are called *feasible*. If P is a non-empty polytope, then the existence of an *optimal* point $\mathbf{x}_0$ is guaranteed. Furthermore, if the linear function $\mathbf{c}^t$ is *generic* (with respect to the inequalities that define P), then the optimal point $\mathbf{x}_0$ is unique.

It is both useful and customary to deal only with a restricted class of rational polyhedra in some "standard form." That is, for example, one considers only polyhedra of the form

$$P \quad = \quad \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

with the additional assumptions that A and $\mathbf{b}$ have only integral coordinates. This is not much of a loss of generality: non-rational data do not occur "in nature," and general polyhedra can be represented by polyhedra in this *equality standard form* by suitable coordinate transformation, introduction of slack variables, etc.

Linear programming is a well-established theory (see Chvátal [4], Schrijver [11]). In some sense the linear programming problem is "solved": after essential progress in the eighties (including the construction of polynomial algorithms such as the "ellipsoid method", the rise of "interior point methods" that are both theoretically polynomial and practically competitive, and a considerable refinements of the classical "simplex method"), we have now codes available (such as CPLEX, by Bob Bixby) which will solve virtually every linear program that you can cook up and store in a computer.

The situation is vastly different with *integer programming*: the task to compute an *integral vector* $\mathbf{x}_I \in P \cap \mathbb{Z}^n$ that minimizes $\mathbf{c}^t$. In this situation, the points in $P \cap \mathbb{Z}^n$ are called *feasible*. Here a basic result of polyhedral theory states that the convex hull

$$P_I \quad = \quad \text{conv}\{\mathbf{x} \in \mathbb{Z}^n : A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

is also a polyhedron with finitely many facets. However, the facet-defining inequalities for the polyhedron P_I are not usually known in general (otherwise we would be done by linear programming), they are hard to determine, and their number may be huge. Thus the integer programming problem — of essential importance in many practical applications — is still a great challenge. It is still not difficult to produce integer programs of reasonable size ($m = 20$ and $n = 30$, say) that none of the currently available codes can solve. Here *solution*, as we will see, really comprises two separate tasks, both of them non-trivial: to *find* an optimal solution, and to *prove* that it is optimal.

While several "good", or at least "interesting", solution strategies exist, integer programming is still a difficult problem. The object of these lectures is an introduction to one such strategy: the construction of *test sets* for integer programming via the Buchberger algorithm. In a basic version, we will present this in the following section.

3 A Buchberger Algorithm for Integer Programming

In this section, we present a "Buchberger algorithm for integer programming." As mentioned in the introduction, the basic idea for this is due to Conti & Traverso [5]. Our presentation follows Thomas [14] [15, Chap. 2]. A successful application of the Buchberger approach to a class of integer programming problems was reported in Natraj, Tayur & Thomas [10] [15, Sect. 4.3].

For the following exposition, we consider the integer programs for which $A \in \mathbb{N}^{m \times n}$ is a fixed, *non-negative* integer matrix. The right hand side vector, $\mathbf{b} \in \mathbb{Z}^m$, is considered as variable. Thus we consider the integer polyhedra

$$P_I(\mathbf{b}) \quad := \quad \text{conv}\{\mathbf{x} \in \mathbb{N}^n : A\mathbf{x} = \mathbf{b}\}.$$

Now let $\mathbf{c} \neq \mathbf{0}$ be a (fixed) linear objective function. To make life easier, we will also assume that the linear program

$$LP(\mathbf{b}) \qquad \min \mathbf{c}^t\mathbf{x} : \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$$

is bounded for every $\mathbf{b}$. This is not much of a restriction: for example, it is satisfied if $\mathbf{c} \geq \mathbf{0}$, or if A has a $\mathbf{1}$ row. In particular, this implies that the integer programs

$$IP(\mathbf{b}) \qquad \min \mathbf{c}^t\mathbf{x} : \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathbb{N}^n$$

are bounded.

In the following, we also need that the objective function is generic. To simulate this, we choose a term order $\succ$ (for example, lexicographic) that can be used as a tie breaker, by defining

$$\mathbf{x} \prec_{\mathbf{c}} \mathbf{y} \qquad \Longleftrightarrow \qquad \begin{cases} \mathbf{c}^t\mathbf{x} < \mathbf{c}^t\mathbf{y}, & \text{or} \\ \mathbf{c}^t\mathbf{x} = \mathbf{c}^t\mathbf{y} & \text{and } \mathbf{x} \prec \mathbf{y}. \end{cases}$$

One might note that $\prec_{\mathbf{c}}$ is a term order if and only if $\mathbf{c} \geq \mathbf{0}$. With this, we get that the following integer programs

$$IP(\mathbf{b}) \qquad \min_{\prec_{\mathbf{c}}} : \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathbb{N}^n$$

either have no feasible solution, or they have a unique optimal solution. We will use $IP_{A,\mathbf{c}}$ to denote the whole family of these integer programs, with fixed A and $\mathbf{c}$, but varying right-hand side $\mathbf{b}$. The key idea is to consider this whole class of programs simultaneously.

Lemma 3.1 *There is a* unique *minimal (finite!) set of vectors* $\mathbf{a}_1, \ldots, \mathbf{a}_t \in \mathbb{N}^n$ *such that*

$$\{\mathbf{x} \in \mathbb{N}^n : \mathbf{x} \textit{ non-optimal}\} \quad = \quad \{\mathbf{x} \in \mathbb{N}^n : \mathbf{x} \geq \mathbf{a}_i \textit{ for some } i\}.$$

Proof. Use the Gordan-Dickson lemma! □

Definition 3.2 *A subset* $\mathcal{G} \subseteq \mathbb{Z}^d$ *is a* test set *for the family* $IP_{A,\mathbf{c}}$ *of integer programs if and only if*

- $A\mathbf{g} = \mathbf{0}$ *and* $\mathbf{g} \succ_{\mathbf{c}} \mathbf{0}$ *for all* $\mathbf{g} \in \mathcal{G}$, *and*
- *for every non-optimal point* $\mathbf{x} \in \mathbb{N}^n$, *there is some* $\mathbf{g} \in \mathcal{G}$ *with* $\mathbf{x} - \mathbf{g} \geq \mathbf{0}$.

The definition of a "test set" immediately provides the following algorithm for integer programming — provided we have a feasible point to start with, and we know how to compute a test set! Note the similarity to "improvement heuristics", such as the ones used to find good solutions to traveling salesman problems.

Algorithm 3.3 *to solve programs in a family* $IP_{A,\mathbf{c}}$*:*

INPUT $\mathcal{G}$, *some* $\mathbf{x} \in \mathbb{N}^n$, *(feasible for* $IP(A\mathbf{x})$*)*
REPEAT *find* $\mathbf{g} \in \mathcal{G}$, *such that* $\mathbf{x} - \mathbf{g} \geq \mathbf{0}$,
$\mathbf{x} \longrightarrow \mathbf{x} - \mathbf{g}$
UNTIL *optimal.*

Theorem 3.4 (Thomas [14, Cor. 2.1.10])
The unique minimal test set for the family $IP_{A,\mathbf{c}}$ *of integer programs is given by*

$$\mathcal{G}_{\mathbf{c}} \;:=\; \Big\{\mathbf{a}_i - \mathbf{b}_i : \mathbf{a}_i \in \mathrm{MIN}_{\leq}\{\mathbf{x} \in \mathbb{N}^n \text{ non-optimal}\},\ A\mathbf{a}_i = A\mathbf{b}_i, \mathbf{b}_i \text{ optimal}\Big\}.$$

This minimal test set corresponds to the (unique!) reduced Gröbner basis of the "toric ideal"

$$I_A \;:=\; \langle \mathbf{x}^{\mathbf{a}^+} - \mathbf{x}^{\mathbf{a}^-} : A\mathbf{a} = \mathbf{0},\ \mathbf{a} \in \mathbb{Z}^n \rangle$$

with respect to the term order $\prec_{\mathbf{c}}$ *(for* $\mathbf{c} \geq \mathbf{0}$*).*

(Here $\mathbf{a}^+$ is our shorthand for the vector we obtain by replacing all negative coordinates of $\mathbf{a}$ by zero. Similarly, we will use $\mathbf{a}^- := (-\mathbf{a})^+$, so that we have $\mathbf{a} = \mathbf{a}^+ - \mathbf{a}^-$, with $\mathbf{a}^+, \mathbf{a}^- \geq \mathbf{0}$.)

The information that $\mathcal{G}_{\mathbf{c}}$ is a Gröbner basis of I_A is not terribly helpful, since in general we do not know a generating set for the ideal — so we can't compute a Gröbner basis, either. Thus we use a small dirty trick: we create a larger integer program, which has an obvious integer feasible point, and for which the ideal has an nice generating set to start from.

For this, we consider the "extended integer programs"

$$EIP(\mathbf{b}) \qquad \min_{\prec_{(M\mathbf{1},\mathbf{c})}} :\quad E\mathbf{y} + A\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathbb{N}^n, \mathbf{y} \in \mathbb{N}^m$$

where $M \in \mathbb{N}$ is a large constant, E is the identity matrix, and $\mathbf{1}$ denotes the vector of all ones. We will use $EIP_{A,\mathbf{c}}$ to denote the whole family of these integer programs, with fixed A and $\mathbf{c}$, but varying right-hand side $\mathbf{b}$. What have we gained? On the one hand, all of the programs $EIP(\mathbf{b})$ are feasible: they have the obvious solution $\mathbf{x} = 0$, $\mathbf{y} = \mathbf{b}$. However, an optimal solution will satisfy $\mathbf{y} = \mathbf{0}$, $\mathbf{x} = \mathbf{x}_0$ if the program $IP(\mathbf{b})$ is feasible, because M is sufficiently large. If $IP(\mathbf{b})$ is infeasible, then the extended program $EIP(\mathbf{b})$ has an optimal solution with $\mathbf{y} > \mathbf{0}$. Putting both cases together, we see that it is sufficient to solve the extended programs.

Proposition 3.5 (Conti & Traverso [5])
The ideal

$$I_{(I,A)} \;:=\; \langle \mathbf{y}^{\mathbf{a}_1^+}\mathbf{x}^{\mathbf{a}_2^+} - \mathbf{y}^{\mathbf{a}_1^-}\mathbf{x}^{\mathbf{a}_2^-} : (I,A)\begin{pmatrix}\mathbf{a}_1\\ \mathbf{a}_2\end{pmatrix} = \mathbf{0},\ \begin{pmatrix}\mathbf{a}_1\\ \mathbf{a}_2\end{pmatrix} \in \mathbb{Z}^{m+n} \rangle$$

is generated by the binomials

$$\mathbf{y}^{A\mathbf{e}_j} - x_j \qquad \text{for } 1 \leq j \leq n.$$

The reduced Gröbner basis of $I_{(I,A)}$ *with respect to the term order* $\prec_{(M\mathbf{1},\mathbf{c})}$ *yields the minimal test set* $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$ *for that family* $EIP_{A,\mathbf{c}}$*, via the canonical bijection*

$$\begin{pmatrix}\mathbf{a}_1\\ \mathbf{a}_2\end{pmatrix} \;\longleftrightarrow\; \mathbf{y}^{\mathbf{a}_1^+}\mathbf{x}^{\mathbf{a}_2^+} \;-\; \mathbf{y}^{\mathbf{a}_1^-}\mathbf{x}^{\mathbf{a}_2^-}.$$

Proof. The binomials $\mathbf{y}^{A\mathbf{e}_j} - x_i$ in fact form a Gröbner basis for $I_{(I,A)}$, for a lexicographic term order with $x_i \succ_{lex} y_j$. Thus they certainly generate the ideal. □

Putting things together, we get a simple algorithm for integer programming: we "only" need to compute a reduced Gröbner basis with respect to the term order $\prec_{(M\mathbf{1},\mathbf{c})}$, and then use this with the above algorithm to solve the extended programs $EIP_{A,\mathbf{c}}$.

Algorithm 3.6 "Integer programming via Buchberger's algorithm"
The following procedure solves the integer program

$$IP(\mathbf{b}) \qquad \min_{\prec_{\mathbf{c}}} : \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathbb{N}^n$$

for $A \in \mathbb{N}^{m \times n}$, $\mathbf{b} \in \mathbb{N}^m$, $\mathbf{c} \in \mathbb{N}^n$.

First Phase: Compute a test set

INPUT A, $\mathbf{c}$
COMPUTE *reduced Gröbner basis* $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$ *for* $I := \langle \mathbf{y}^{A\mathbf{e}_j} - x_j \rangle$,
OUTPUT *test set* $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$.

Second Phase: Reduction

INPUT $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$, $\mathbf{b}$
REDUCE $\mathbf{y}^{\mathbf{b}}$ *with respect to* $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$, *get* $\mathbf{y}^{\mathbf{a}_1}\mathbf{x}^{\mathbf{a}_2}$.
OUTPUT *if* $\mathbf{a}_1 \neq \mathbf{0}$, *output "infeasible".*
if $\mathbf{a}_1 = \mathbf{0}$, *output "*$\mathbf{x}_0 = \mathbf{a}_2$ *is optimal".*

However, there is still quite a detour involved: one can formulate the Buchberger algorithm in such a way that it directly operates on lattice points (no ideals, binomials, etc. involved!) This geometric formulation (given in the next lecture) yields an extremely simple algorithm for integer programming: also one that is very easy to implement! The basic version is not terribly efficient: but we will discuss a few basic issues in "how to speed it up".

Note here that while the first phase is hard work, the second one is quite trivial (if we manage to search efficiently the Gröbner basis, which may be huge). If we don't have a complete Gröbner basis, then we can still use any partial basis to reduce the monomial $\mathbf{y}^{\mathbf{b}}$, which may yield a feasible, or even the optimal, point.

4 The Geometry of Buchberger's Algorithm

The Buchberger algorithm for integer programming is just a special case of the general Buchberger algorithm. However, there is a lot of special features in the special situation of "toric ideals" we consider here. In particular, one only has to deal with "binomials with disjoint supports": thus we can get an entirely geometric formulation of the algorithm, dealing with lattice vectors in $\mathbb{Z}^n$ — no polynomials whatsoever appear. This simplifies the data structures considerably!

The first observation is that

$$I := \langle \mathbf{y}^{A\mathbf{e}_j} - x_j \rangle$$

is a binomial ideal. (See [7] for more on binomial ideals.) Now the S-pair of two binomials is a binomial (see Problem 6.3 for a sharper version of this fact). Also the reduction of a binomial by binomials leads to binomials. Thus the reduced Gröbner basis of I will consist of binomials.

As a second step, notice that in this situation, if $\mathbf{x^a - x^b} \in I$ (with $\mathbf{a, b} \geq \mathbf{0}$) is a binomial such that the two monomials have common factors, then we can remove them: $\mathbf{x^{(a-b)^+} - x^{(b-a)^+}}$ is also contained in I. To see this, apply Exercise 6.4 to the lattice

$$\mathcal{L} \;:=\; \mathbb{Z}^{m+n} \cap \{\begin{pmatrix}\mathbf{y}\\ \mathbf{x}\end{pmatrix} : I\mathbf{y} + A\mathbf{x} = \mathbf{0}\}.$$

Thus every reduced Gröbner basis of I can only contain binomials of the form $\mathbf{x^{a^+} - x^{a^-}}$ with $\mathbf{a} \in \mathcal{L}$.

Thus we are really dealing with a geometric algorithm operating on lattice vectors. The following two algorithms compute the *reduced Gröbner basis of a lattice*, that is, the finite subset $\mathcal{G} \subseteq \mathcal{L}^{\succ \mathbf{0}}$ corresponding to the reduced Gröbner basis of $I_\mathcal{L}$. The assumption for this is that we know a "good" generating set for the lattice, i.e., a subset of the lattice corresponding to a set of binomials that generates $I_\mathcal{L}$.

Algorithm 4.1 "Reduction"
The following algorithm computes the reduction of a vector $\mathbf{f} \in \mathbb{Z}^{m+n}$ by a set $\mathcal{G}$ of integer vectors.

INPUT $\mathcal{G} \subseteq \mathcal{L}^{\succ 0}$, $\mathbf{f} \succ \mathbf{0}$.

REPEAT *If there is some* $\mathbf{g} \in \mathcal{G}$ *with* $\mathbf{g}^+ \leq \mathbf{f}^+$, *then replace* $\mathbf{f}$ *by* $\pm(\mathbf{f} - \mathbf{g}) \succ \mathbf{0}$.
If there is some $\mathbf{g} \in \mathcal{G}$ *with* $\mathbf{g}^+ \leq \mathbf{f}^-$, *then replace* $\mathbf{f}$ *by* $\mathbf{f} + \mathbf{g}$.

OUTPUT $\overline{\mathbf{f}} := \mathbf{f}$.

Algorithm 4.2 "Buchberger algorithm on lattice vectors"
The following algorithm computes the reduced Gröbner basis of the lattice $\mathcal{L}$, for a fixed term order $\succ$.

First Step: Construct a "Gröbner basis"

INPUT *A basis* $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\} \subseteq \mathcal{L}$ *of the lattice* $\mathcal{L}$ *such that the binomials* $\mathbf{x^{a^+} - x^{a^-}}$ *generate* $I_\mathcal{L}$.

SET $\mathcal{G}_{old} := \emptyset$, $\mathcal{G} := \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$

REPEAT *While* $\mathcal{G}_{old} \neq \mathcal{G}$, *repeat the following steps*
$\mathcal{G}_{old} := \mathcal{G}$
(S-pairs) construct the pairs $\mathbf{g} := \mathbf{a} - \mathbf{a}' \succ \mathbf{0}$ *with* $\mathbf{a}, \mathbf{a}' \in \mathcal{G}$.
(reduction) reduce the vectors $\mathbf{g}$ *by the vectors in* $\mathcal{G}_{old}$. *If* $\overline{\mathbf{g}} \neq \mathbf{0}$, *set* $\mathcal{G} := \mathcal{G} \cup \overline{\{\mathbf{g}\}}$.

Second Step: Construct a minimal "Gröbner basis"

REPEAT *If for some* $\mathbf{g} \in \mathcal{G}$ *the point* $\mathbf{g}^+$ *can be reduced by some* $\mathbf{g}' \in \mathcal{G} \backslash \mathbf{g}$, *then delete* $\mathbf{g}$ *from* $\mathcal{G}$.

Third Step: Construct the reduced "Gröbner basis"

REPEAT *If for some* $\mathbf{g} \in \mathcal{G}$ *the point* $\mathbf{g}^-$ *can be reduced by some* $\mathbf{g}' \in \mathcal{G} \backslash \mathbf{g}$, *then replace* $\mathbf{g}$ *by the corresponding reduced vector:* $\mathcal{G} := \mathcal{G} \backslash \mathbf{g} \cup \overline{\mathbf{g}}$.

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

All these operations are easy to visualize (at least in the 2-dimensional situation), see the lecture. They are also easily implemented — just do it. There is also a lot of flexibility: in fact, for a successful implementation it is important to reduce earlier, otherwise the Gröbner bases in constructed in the "First Step" will be too large. See the ideas in [5, 9, 18] for further ideas about how to make this efficient.

Let us just remark that the elements that can occur in a reduced Gröbner basis can be characterized geometrically in a different way. The following theorem is due to Sturmfels & Thomas [12], see also Thomas & Weismantel [16].

Theorem 4.3 *The universal Gröbner basis (that is, the union of all the reduced Gröbner bases $\mathcal{G}_{\mathbf{c}}$ of $IP_{A,\mathbf{c}}$, for different objective functions $\mathbf{c}$) consists of all the primitive lattice vectors $\mathbf{a} \in \mathbb{Z}^n$ (with $A\mathbf{a} = \mathbf{0}$, and $\frac{1}{\lambda}\mathbf{a} \notin \mathbb{Z}^n$ for $\lambda > 1$) such that $[\mathbf{a}^+, \mathbf{a}^-]$ is an edge of the polyhedron*

$$P_I(\mathbf{a}^+) \quad = \quad \text{conv}\{\mathbf{x} \in \mathbb{N}^n : A\mathbf{x} = A\mathbf{a}^+\}.$$

This theorem be applied as well to the extended integer programs EIP_A, where we know how the minimal test sets (Gröbner bases) can be computed via Buchberger's algorithm.

We just mention that Thomas & Sturmfels [12] also have a technique to compute universal Gröbner bases via a single application of a Buchberger algorithm (to a larger problem).

5 A Variant of Buchberger's Algorithm

In the following I sketch the basic version of a simple adaption of Buchberger's algorithm that seems to be even more useful for integer programming. The ideas for this are quite fresh, from Urbaniak, Weismantel & Ziegler [18].

Given a matrix $A \in \mathbb{N}^{m \times n}$, an objective function $\mathbf{c} \in \mathbb{N}^n$, and a right hand side vector $\mathbf{b} \in \mathbb{N}^m$, we denote by $IP_{A,b,c}$ the optimization problem

$$\max_{\prec_{\mathbf{c}}} \{\mathbf{x} \in \mathbb{N}^n : A\mathbf{x} \leq \mathbf{b},\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\}.$$

(This is again a quite special type of integer program, which we call a problem in *inequality standard form*. See also Exercise 6.2. Again, in order to avoid dealing with degenerate cases we have refined the objective function $\mathbf{c}^t\mathbf{x}$ to get a term order $\prec_{\mathbf{c}}$.)

We now present the basic form of an algorithm that computes a test set for the above integer programming problem. It is not too hard to see that the algorithm terminates after finitely many steps and determines a test set for $IP_{A,b,c}$.

Roughly speaking, a test set, $\mathcal{G}$ say, can be computed as follows. Start with the n unit vectors, i.e., set $\mathcal{G} := \{e_i : 1 \leq i \leq n\}$. Iteratively, compute the difference vectors between all pairs of vectors that are in $\mathcal{G}$ and direct each such difference vector such that it is greater than 0 with respect to the order. All such difference vectors which are currently not in $\mathcal{G}$ and which are the difference of feasible vectors for $IP_{A,b,c}$, are now added to $\mathcal{G}$. The algorithm terminates if no more vectors are added to $\mathcal{G}$.

More precisely, the basic algorithm can be formulated as follows:

Algorithm 5.1

(1) *Set $\mathcal{G}_{old} := \emptyset$, $\mathcal{G} := \{e_i : 1 \leq i \leq n\}$.*

(2) *While $\mathcal{G}_{old} \neq \mathcal{G}$ perform the following steps:*

(2.1) *Set* $\mathcal{G}_{old} := \mathcal{G}$.

(2.2) *For all pairs of vectors* $\mathbf{v}, \mathbf{w} \in \mathcal{G}$ *such that* $\mathbf{v} \prec \mathbf{w}$, $-\mathbf{b} \leq A(\mathbf{w} - \mathbf{v}) \leq \mathbf{b}$ *and* $-\mathbf{u} \leq \mathbf{w} - \mathbf{v} \leq \mathbf{u}$, *set* $\mathcal{G} := \mathcal{G} \cup \{\mathbf{w} - \mathbf{v}\}$.

Whenever Step 2 of the above algorithm is executed (except for the last time), a new vector was added to the set $\mathcal{G}_{old}$. Since the number of integral vectors $\mathbf{x}$ satisfying $-\mathbf{u} \leq \mathbf{x} \leq \mathbf{u}$ is bounded by $\prod_{i=1}^{n}(2u_i + 1)$, the above algorithm terminates after finitely many steps.

Let us now show that the set $\mathcal{G}$ generated by the above algorithm is a test set for $IP_{A,b,c}$. Suppose that $\mathbf{x}$ is a feasible point, i.e., $A\mathbf{x} \leq \mathbf{b}$, $\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$, that is not optimal and that cannot be improved via some element in $\mathcal{G}$. Let $\mathbf{x}'$ be some feasible vector with $\mathbf{x} \prec \mathbf{x}'$. Then, $\mathbf{x}' - \mathbf{x}$ is not an element of $\mathcal{G}$. However, as $\mathbf{x}' - \mathbf{x}$ can be written as a certain linear combination of unit vectors and since unit vectors are elements of $\mathcal{G}$, we can decrease from $\mathbf{x}$ to reach $\mathbf{0}$, then increase to reach $\mathbf{x}'$. Hence, there exists a sequence $P = (\mathbf{x}^0, \ldots, \mathbf{x}^p)$ of vectors $\mathbf{x}^i$ and a number $1 \leq \tau < p$ with the properties:

(i) $\mathbf{x}^0 = \mathbf{x}$, $\mathbf{x}^p = \mathbf{x}'$,

(ii) for all $i = 1, \ldots, \tau$, $-(\mathbf{x}^i - \mathbf{x}^{i-1}) \in \mathcal{G}$,

(iii) for all $i = \tau + 1, \ldots, p$, $(\mathbf{x}^i - \mathbf{x}^{i-1}) \in \mathcal{G}$,

(iv) every vector in P is feasible.

Let p_{min} be the smallest number such that there exists some sequence

$$P_{min} \quad = \quad (\mathbf{x}^0, \ldots, \mathbf{x}^{p_{min}})$$

of vectors $\mathbf{x}^i$ and a number $1 \leq \tau < p_{min}$ satisfying (i), (ii), (iii) and (iv).

Since $-(\mathbf{x}^\tau - \mathbf{x}^{\tau-1}) \in \mathcal{G}$ and $(\mathbf{x}^{\tau+1} - \mathbf{x}^\tau) \in \mathcal{G}$, the difference vector

$$\mathbf{v} := (\mathbf{x}^{\tau+1} - \mathbf{x}^\tau) \; - \; (-(\mathbf{x}^\tau - \mathbf{x}^{\tau-1})) \quad = \quad \mathbf{x}^{\tau+1} \; - \; \mathbf{x}^{\tau-1}$$

has been computed in Step 2.2 of Algorithm 5.1. Moreover, both vectors $\mathbf{x}^{\tau+1}$ and $\mathbf{x}^{\tau-1}$ are feasible. It follows that $-\mathbf{b} \leq A\mathbf{v} \leq \mathbf{b}$ and $-u_i \leq v_i \leq u_i$ for all i.

In case that $0 \prec v$, the vector v was added to $\mathcal{G}$. Consequently,

$$P' \quad := \quad (\mathbf{x}^0, \ldots, \mathbf{x}^{\tau-1}, \mathbf{x}^{\tau+1}, \ldots, \mathbf{x}^{p_{min}})$$

and $\tau - 1$ again satisfy properties (i) – (iv), yet involving $p_{min} - 1$ vectors, a contradiction. Therefore $\mathbf{v} \prec \mathbf{0}$. In this case the vector $-\mathbf{v}$ was added to $\mathcal{G}$ in Step 2.2 of Algorithm 5.1. Then,

$$P' \quad := \quad (\mathbf{x}^0, \ldots, \mathbf{x}^{\tau-1}, \mathbf{x}^{\tau+1}, \ldots, \mathbf{x}^{p_{min}})$$

and τ satisfy properties (i) – (iv). Since again only $p_{min} - 1$ vectors belong to P', we obtain a contradiction. □

From our discussions we derive the following theorem.

Theorem 5.2 *Algorithm 5.1 terminates after a finite number of steps. The output is a test set for the integer programming problem* $IP_{A,b,c}$.

This extremely simple algorithm has some essential advantages as compared to Algorithm 3.6: in particular, we do not have to increase dimension to obtain the extended problem: here the computation takes place in original space. However, in Algorithm 5.1 we still have the problem that we compute way to many elements: the partial Gröbner basis computed here is way too large. Thus one has to reduce elements, and discard superfluous elements during the computation. Also, this algorithm sometimes makes way too many comparisons, while generating only relatively few new basis elements. Such observations, made on a practical implementation, led to further variations of the algorithm that are currently still under investigation [18].

References

[1] W. W. Adams & P. Loustaunau: *An Introduction to Gröbner Bases,* American Mathematical Society, Graduate Studies in Math., Vol. III, 1994.

[2] T. Becker & V. Weispfennig: *Gröbner bases: a computational approach to commutative algebra,* Graduate Texts in Mathematics **141**, Springer-Verlag 1993.

[3] B. Buchberger: *Gröbner bases: an algorithmic method in polynomial ideal theory,* in: N.K. Bose (ed.), "Multidimensional Systems Theory", D. Reidel 1985, 184-232.

[4] V. Chvátal: *Linear Programming,* Freeman, New York 1983.

[5] P. Conti & C. Traverso: *Buchberger Algorithm and Integer Programming,* Proceedings AAECC-9 (New Orleans), Springer LNCS **539**, 1991, pp. 130-139.

[6] D. A. Cox, J. B. Little & D. O'Shea: *Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra, Undergraduate Texts in Mathematics*, Springer-Verlag, New York 1992.

[7] D. Eisenbud & B. Sturmfels: *Binomial ideals,* preprint 1994, 44 pages; *Duke Math. Journal*, to appear.

[8] R. E. Gomory: *Some polyhedra related to combinatorial problems, Linear Algebra and its Applications* **2** (1969), 451-455.

[9] C. Moulinet & L. Pottier: *Gröbner bases of toric ideals: properties, algorithms, and applications,* preprint, INRIA Sophia Antipolis, 10 pages.

[10] N. R. Natraj, S. R. Tayur & R. R. Thomas: *An algebraic geometry algorithm for scheduling in presence of setups and correlated demands, Mathematical Programming* **69** (1995), 369-402.

[11] A. Schrijver: *Theory of Linear and Integer Programming,* Wiley-Interscience, Chichester 1986.

[12] B. Sturmfels & R. R. Thomas: *Variation of cost functions in integer programming,* preprint, Cornell University 1994, 31 pages.

[13] B. Sturmfels, R. Weismantel & G. M. Ziegler: *Gröbner bases of lattices, corner polyhedra, and integer programming, Beiträge zur Algebra und Geometrie/Contributions to Algebra and Geometry* **36** (1995), 281-298.

[14] R. R. THOMAS: *A geometric Buchberger algorithm for integer programming,* preprint, Cornell 1993, 25 pages; *Mathematics of Operations Research*, to appear.

[15] R. R. THOMAS: *Gröbner basis methods for integer programming,* Ph. D. Thesis, Cornell University 1994, 157 pages.

[16] R. R. Thomas & R. Weismantel: Truncated Gröbner bases for integer programming, Preprint SC-95-09, ZIB Berlin, May 1995, 12 pages.

[17] R. R. THOMAS & R. WEISMANTEL: *Test sets and inequalities for integer programs: extended abstract,* in: Proc. IPCO-96, Vancouver, June 1996, to appear.

[18] R. URBANIAK, R. WEISMANTEL & G. M. ZIEGLER: *A variant of Buchberger's algorithm for integer programming,* Preprint SC 94-29, ZIB-Berlin, January 1995, 19 pages; *SIAM J. Discrete Math.*, to appear.

[19] G. M. ZIEGLER: *Lectures on Polytopes, Graduate Texts in Mathematics* **152**, Springer-Verlag New York Berlin Heidelberg 1995, 370 pages; *Updates, corrections, and more* available at `http://www.math.tu-berlin.de/~ziegler`

6 Problems

Problem 6.1 "Upper bounds"
For a problem in the form

$$\max\{\mathbf{c}^t\mathbf{x} : A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

with $A \in \mathbb{N}^{n\times n}$, $\mathbf{b} \in \mathbb{N}^m$, and $\mathbf{c} \in \mathbb{Z}^n$, how can we compute upper bounds u_i for the variables x_i? What happens in the special case where A has a zero-column?

Problem 6.2 "Standard forms of integer programs"
Show that for the "equality standard form"

$$\min\{\mathbf{c}^t\mathbf{x} : A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

and the "inequality standard form"

$$\max\{\mathbf{c}^t\mathbf{x} : A\mathbf{x} \leq \mathbf{b},\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\}.$$

of linear programs are equivalent: for any problem in one form we can construct a problem in the other form that solves it.
(Assume $A \in \mathbb{N}^n$, $\mathbf{b} \in \mathbb{N}^m$, and $\mathbf{c} \in \mathbb{Z}^n$.)

The following problem sharpens our observation that S-pair formation corresponds to difference of vectors: we explicitly identify the "superfluous" monomial factors occur in the formation of S-pairs.

Problem 6.3 "S-pairs and difference vectors"
For $\mathbf{a}, \mathbf{b} \in \mathcal{L}^{\prec \mathbf{0}}$, the S-polynomial of $\mathbf{x}^{\mathbf{a}^+} - \mathbf{x}^{\mathbf{a}^-} \in I_{\mathcal{L}}$ and $\mathbf{x}^{\mathbf{b}^+} - \mathbf{x}^{\mathbf{b}^-} \in I_{\mathcal{L}}$ is

$$\mathbf{x}^{\min(\mathbf{a}^-,\mathbf{b}^-)}\,(\mathbf{x}^{(\mathbf{a}-\mathbf{b})^+} - \mathbf{x}^{(\mathbf{a}-\mathbf{b})^-}),$$

a monomial times the binomial corresponding to $\mathbf{a}-\mathbf{b}$.

Problem 6.4 "Binomial criterion"
Let I be an ideal generated by the binomials corresponding to some sublattice $\mathcal{L} \subseteq \mathbb{Z}^n$:

$$I_{\mathcal{L}} \ := \ \langle \mathbf{x}^{\mathbf{a}^+} - \mathbf{x}^{\mathbf{a}^-}; \mathbf{a} \in \mathcal{L}\rangle.$$

Then a binomial $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}}$, with $\mathbf{a}, \mathbf{b} \geq \mathbf{0}$, is contained in $I_{\mathcal{L}}$ if and only if $\mathbf{a}-\mathbf{b} \in \mathcal{L}$.
(Hint: $\mathbf{x}^{\mathbf{a}}$ and $\mathbf{x}^{\mathbf{b}}$ reduce to the same standard monomial.)

The following problems deal with the relation between lattice vectors and binomials in more depth, and thus with the "Gröbner basis of a lattice": this is a situation closely linked to integer programming (the "local situation", Gomory's [8] "group problem"), but also, for example, to the ideals of toric varieties.

Let $\mathcal{L} \subseteq \mathbb{Z}^n$ be an n-dimensional integral lattice, and associate with it the *lattice ideal*

$$I_{\mathcal{L}} \ := \ \langle \mathbf{x}^{\mathbf{a}^+} - \mathbf{x}^{\mathbf{a}^-} : \mathbf{a} \in \mathcal{L}\rangle.$$

Here we assume that the lattice $\mathcal{L}$ is generated by the columns of a nonnegative matrix $A \in \mathbb{N}^{n\times n}$.

Problem 6.5 "Ideal of a lattice: generators"
Show that if the columns $(\mathbf{a}_1, \ldots, \mathbf{a}_n)$ *of* A *generate* $\mathcal{L}$*, and if they are all non-negative, then the ideal* $I_{\mathcal{L}}$ *is generated by the binomials*

$$\mathbf{x}^{\mathbf{a}_i^+} \; - \; \mathbf{x}^{\mathbf{a}_i^-} \quad = \quad \mathbf{x}^{\mathbf{a}_i} - 1.$$

Show that this can fail if we do not assume A *to be non-negative.*

Problem 6.6 "Gröbner bases of a lattice: example"
Let $\mathcal{L}_A \subseteq \mathbb{Z}^2$ *be the 2-dimensional lattice generated by the columns of* $A = \begin{pmatrix} 1 & 4 \\ 4 & 3 \end{pmatrix}$.
Compute all the (four) different reduced Gröbner bases for the corresponding ideal. Describe the structure of the various Gröbner basis elements.
How many standard monomials are there in each case?

Problem 6.7 "Gröbner bases of a lattice: geometry"
Show that if $\mathcal{L} \subseteq \mathbb{Z}^2$ *is a 2-dimensional integral lattice, then the universal Gröbner basis (that is, the union of all the reduced Gröbner bases) consists of the following lattice vectors:*

- *the vertices* $\mathbf{a} \in \mathbb{Z}^2$ *of the polyhedron* $\operatorname{conv}(\mathcal{L} \cap \mathbb{N}^2 \backslash \mathbf{0})$,
- *the vectors* $\mathbf{a} \in \mathcal{L}$ *that have one positive and one negative component, and for which* $\mathbf{0}$ *and* $\mathbf{a}$ *are two adjacent vertices of the polyhedron* $\operatorname{conv}(\mathcal{L} \cap \{\mathbf{x} \in \mathbb{Z}^2 : \mathbf{x} \geq -\mathbf{a}^-\})$.

(Remark: a similar structure theorem is true in higher dimensions as well, but harder to prove. See [13].)

Problem 6.8 "The variant of Buchberger's algorithm: example"
Apply Algorithm 5.1 to compute a test set $\mathcal{G}$ *for the 0/1 knapsack problem*

$$\max\{x_1 + 2x_2 + 3x_3 : x_1 + 2x_2 + 3x_3 \leq 3, \; x_i \in \{0, 1\}, \; i = 1, 2, 3\}.$$

Identify a minimal test set $\mathcal{G}_{min} \subseteq \mathcal{G}$.