# Polynomial Algorithms for Linear Programming over the Algebraic Numbers

Ilan Adler and Peter A. Beling
Department of Industrial Engineering and Operations Research
University of California, Berkeley
Berkeley, CA 94720

## Abstract

*We derive an algorithm based on the ellipsoid method that solves linear programs whose coefficients are real algebraic numbers. By defining the encoding size of an algebraic number to be the bit size of the coefficients of its minimal polynomial, we prove the algorithm runs in time polynomial in the dimension of the problem, the encoding size of the input coefficients, and the degree of any algebraic extension which contains the input coefficients. This bound holds even if all input and arithmetic is performed symbolically, using rational numbers only.*

## 1 Introduction

Linear programming with rational numbers is usually modeled in terms abstracted from the Turing machine model of computation. Problem input is assumed to consist only of rational numbers, and an algorithm is permitted to perform only the elementary operations of addition, subtraction, multiplication, division, and comparison. The *dimension* of a problem instance is defined to be the number of entries in the matrices and vectors that define the instance, and the *size* of an instance is defined to be the total number of bits needed to encode these entries in binary form. A linear programming algorithm is said to run in *polynomial time* if the number of elementary operations it performs is bounded by a polynomial in the problem dimension and encoding size. Typically, it is further required that the binary encoding size of any number generated during the course of a polynomial-time algorithm be polynomial in the size of the instance.

The polynomial-time solvability of rational-number linear programs (LPs) was demonstrated in a landmark paper by Khachiyan in 1979. In fact, both Khachiyan's ellipsoid method [7] and Karmarkar's interior point method [6] solve LPs with rational coefficients in time that is polynomial in the number of input coefficients and the total number of bits in a binary encoding of the problem data. Unfortunately, these results do not extend in any obvious way to LPs whose coefficients are real numbers.

In the case of real numbers (i.e., numbers that are not necessarily rational), linear programming is usually modeled in terms of a machine that can perform any elementary arithmetic operation in constant time, regardless of the nature of the operands. (See [2] for a treatment of the theory of general computation over the real numbers.) Problem dimension is defined as in the rational case, but because a general real number cannot be represented by a finite string of digits, there is no corresponding notion of the size of a real-number LP. An algorithm is said to run in *polynomial time* if the number of elementary operations it performs is bounded by a polynomial in the problem dimension. By this definition, no polynomial-time algorithm is known for general real-number LPs.

Complexity results for both the ellipsoid and interior point methods depend in a fundamental way


24th ANNUAL ACM STOC - 5/92/VICTORIA, B.C., CANADA
© 1992 ACM 0-89791-512-7/92/0004/0483...$1.50

on upper and lower bounds on the magnitude of certain numbers related to basic solutions of the LP. If the problem data is rational, the bounds are a function of the bit size of the data and can be computed in polynomial time. If the data is not rational, it is still possible to compute the necessary upper bounds in polynomial time, but no polynomial method for computing the lower bounds is known [12]. In fact, in this case it is possible to construct examples in which the running time of the ellipsoid method is arbitrarily bad compared with the problem dimension [20].

Using an approach that is quite different from that of the existing polynomial algorithms for rational-number LPs, Megiddo has shown that several special classes of real-number LPs can be solved in polynomial time. In [10] a polynomial algorithm is given for feasibility problems in which at most two variables appear in each inequality, and in [11] one is given for LPs in which the number of variables is fixed. In fact, these algorithms are *strongly polynomial:* they are polynomial in both the rational and the real senses.

Adler and Beling [1] have shown that a variant of the interior point method can solve LPs whose coefficients belong to a particular subring of the algebraic integers. The combination of this algorithm with a variant of the Tardos scheme [19] leads to a strongly polynomial algorithm for LPs whose coefficient matrices are circulant. (LPs of this kind are related to discrete convolution and arise frequently in image processing.)

In this paper, we show that the ellipsoid method can be used to solve LPs whose coefficients are real algebraic numbers, in spite of the fact that these numbers are generally not rational. The key to our result is a notion of problem size that is analogous in form and function to the binary size of a rational-number problem. Specifically, we measure the size of an algebraic number in terms of bit size of the rational numbers that define its minimal polynomial. We also view the problem coefficients as members of a finite algebraic extension of the rational numbers. The degree of this extension is an upper bound on the degree of any algebraic number that can occur during the course of the algorithm, and in this sense can be viewed as a supplementary measure of problem dimension.

An essential feature of our construction is a tool for obtaining upper and lower bounds on linear and polynomial forms involving algebraic integers (see proposition 3.1). In addition to being interesting in its own right, this tool permits us to obtain 'reasonable' upper and lower bounds on certain quantities involving the basic solutions of the LP. These bounds are a function of the degree of the extension in which we work and the size of the data. We use these bounds to show that our algorithm's running time is polynomial in the dimension of the LP, the degree of the extension defined by the input coefficients, and the size of the data.

As an analytic convenience, we work initially under a model of computation that allows real-number input and arithmetic. In fact, this assumption is stronger than necessary. Later we show that, by handling algebraic numbers in a symbolic fashion, we can achieve materially the same results under a rational-number model of computation.

The paper is organized in the following manner: In section 2, we provide a brief review of terminology and concepts from algebra and number theory that we use in the remainder of the paper. In section 3, we derive basic complexity bounds for LPs whose coefficients are real algebraic integers (we lose no generality by the working with the algebraic integers rather than the full set of algebraic numbers). In particular, we establish a chain of polynomial problem equivalencies that leads from the linear programming problem to a problem that can be solved by the ellipsoid method in polynomial time. In section 4, we show how we can use our earlier results to obtain complexity bounds under several different assumptions about the form of the input data. In section 5, we discuss how one can perform all input and computations in the linear programming algorithm using rational numbers only. Finally, in section 6, we conclude with some remarks.

## 2 Algebraic Preliminaries

In this section we give a brief review of the terminology and concepts from algebra and number theory that we shall use throughout the remainder of the paper. Proofs and detailed discussion of the

results stated here can be found in most texts on algebraic number theory (see, e.g., [5], [16], or [17]).

We begin by stating the basic terminology we shall use to describe polynomials.

**Definition.** Let $F(t) = q_d t^d + \cdots + q_1 t + q_0$ be a polynomial in indeterminate $t$ with coefficients $q_0, \ldots, q_d \in \boldsymbol{Q}$ where $q_d \neq 0$. (We use the standard notation $\boldsymbol{Z}$ for the integers, $\boldsymbol{Q}$ for the rationals, $\boldsymbol{R}$ for the reals, and $\boldsymbol{C}$ for the complex numbers.) We define the *degree* of $F$ to be $d$, the index of the leading coefficient of $F$, and we write this quantity as $\deg(F)$. We say $F$ is a *monic polynomial* if $q_d = 1$. $F$ is *reducible over the rationals* if there exist polynomials $F_1$ and $F_2$ with rational coefficients and strictly positive degrees such that $F(t) = F_1(t)F_2(t)$; otherwise $F$ is *irreducible over the rationals.*

Polynomials with rational coefficients are intimately related to the algebraic numbers, a subset of the complex numbers that is central to our work.

**Definition.** A complex number $\alpha$ is an *algebraic number* if there exists a polynomial $F$ with rational coefficients such that $F(\alpha) = 0$.

Clearly, each algebraic number is the root of many polynomials. Among these, we distinguish one polynomial as being of particular importance.

**Proposition 2.1** *Let $\alpha$ be an algebraic number. Then $\alpha$ is the root of a unique monic, irreducible polynomial with rational coefficients.*

The polynomial whose existence is asserted in proposition 2.1 is known as the *minimal polynomial* of $\alpha$. We define two key attributes of an algebraic number in terms of its minimal polynomial.

**Definition.** Let $\alpha$ be an algebraic number with minimal polynomial $G$ of degree $d$. We define the *degree of the algebraic number* $\alpha$ to be $d$, the degree of its minimal polynomial, and we write this quantity as $\deg(\alpha)$. By the fundamental theorem of algebra, $G$ has $d$ (possibly complex) roots, say $\alpha_1, \ldots, \alpha_d$. We call $\alpha_1, \ldots, \alpha_d$ the *conjugates* of $\alpha$. (Note that the conjugates of $\alpha$ include $\alpha$ itself.)

One can show that the conjugates of an algebraic number are distinct, and that they share the same minimal polynomial.

Certain classes of algebraic numbers enjoy the property that the class is closed under arithmetic operations among its members; other classes are defined on the basis of this property. Before introducing several such classes, we review standard terminology for sets that are closed under arithmetic operations.

**Definition.** A subset $\boldsymbol{V}$ of the complex numbers is called a *subring of the complex numbers* if $1 \in \boldsymbol{V}$ and if, for any $\alpha, \beta \in \boldsymbol{V}$, we have $-\alpha, \alpha + \beta, \alpha\beta \in \boldsymbol{V}$. A subset $\boldsymbol{W}$ of the complex numbers is called a *subfield of the complex numbers* if $1 \in \boldsymbol{W}$ and if, for any $\alpha, \beta \in \boldsymbol{W}$ with $\alpha \neq 0$, we have $-\alpha, 1/\alpha, \alpha + \beta, \alpha\beta \in \boldsymbol{W}$.

**Proposition 2.2** *The algebraic numbers form a subfield of the complex numbers.*

Given an algebraic number $\alpha$, we shall often be interested in the set of all numbers that can be 'built up' by a sequence of arithmetic operations using rational numbers and $\alpha$. We define this set in terms of a subfield.

**Definition.** Let $\alpha$ be an algebraic number. We define $\boldsymbol{Q}(\alpha)$ to be the smallest subfield of the complex numbers that contains both $\alpha$ and the rationals $\boldsymbol{Q}$. We call $\boldsymbol{Q}(\alpha)$ a *single algebraic extension* of the rational numbers by $\alpha$. We define the *degree of the extension* $\boldsymbol{Q}(\alpha)$ to be $\deg(\alpha)$.

The following property of single algebraic extensions is particularly important for our purposes.

**Proposition 2.3** *Let $\alpha$ be an algebraic number of degree $d$. Then every $\beta \in \boldsymbol{Q}(\alpha)$ is an algebraic number of degree at most $d$. Moreover, every $\beta \in \boldsymbol{Q}(\alpha)$ has the representation $\beta = q_0 + q_1\alpha + \cdots + q_{d-1}\alpha^{d-1}$ for a unique set of rational coefficients $q_0, q_1, \ldots, q_{d-1}$.*

The next proposition characterizes the conjugates of every member of a single algebraic extension in terms of the conjugates of the algebraic number that defines the extension.

**Proposition 2.4** *Let $\alpha$ be an algebraic number of degree $d$ with conjugates $\alpha_j$, where $j = 1, \ldots, d$. Let $F$ be a polynomial with rational coefficients. Then the conjugates of $\beta = F(\alpha)$ are the distinct members of the collection $\{F(\alpha_j); j = 1, \ldots, d\}$.*

As a natural generalization of the notion of a single algebraic extension, we have the following definition:

**Definition.** Let $\alpha_1, \ldots, \alpha_n$ be algebraic numbers. Then we define the *multiple algebraic extension* $\boldsymbol{Q}(\alpha_1, \ldots, \alpha_n)$ to be the smallest field that contains $\alpha_1, \ldots, \alpha_n$ and $\boldsymbol{Q}$.

Rather surprisingly, every multiple extension is also a single extension.

**Proposition 2.5** *Let $\alpha_1, \ldots, \alpha_n$ be algebraic numbers with degrees $d_1, \ldots, d_n$, respectively. Then there exists an algebraic number $\theta$ of degree at most $\prod_{j=1}^{n} d_j$ such that $\boldsymbol{Q}(\alpha_1, \ldots, \alpha_n) = \boldsymbol{Q}(\theta)$.*

The algebraic number $\theta$ whose existence is asserted in proposition 2.5 is not unique. Indeed, for any algebraic number $\alpha$, it is evident from the definition of a single algebraic extension that $\boldsymbol{Q}(\alpha) = \boldsymbol{Q}(\alpha + 1)$. The degree of a single algebraic extension, on the other hand, is uniquely defined.

**Proposition 2.6** *Let $\alpha$ and $\beta$ be algebraic numbers such that $\boldsymbol{Q}(\alpha) = \boldsymbol{Q}(\beta)$. Then* $\deg(\alpha) = \deg(\beta)$.

In light of proposition 2.6, we are justified in defining the *degree of a multiple algebraic extension* to be the degree of any equivalent single extension.

Combining propositions 2.3 and 2.5, we see that every member of a multiple extension can be expressed as a polynomial function of a single algebraic number.

**Corollary 2.1** *Let $\boldsymbol{Q}(\alpha_1, \ldots, \alpha_n)$ be a multiple algebraic extension of degree $d$. Then every $\beta \in \boldsymbol{Q}(\alpha_1, \ldots, \alpha_n)$ is an algebraic number of degree at most $d$. Moreover, there exists an algebraic number $\theta$ of degree $d$ such that every $\beta \in \boldsymbol{Q}(\alpha_1, \ldots, \alpha_n)$ has the representation $\beta = q_0 + q_1\theta + \cdots + q_{d-1}\theta^{d-1}$ for a unique set of rational coefficients $q_0, \ldots, q_{d-1}$.*

At this point we introduce a particular subset of the algebraic numbers that will, when we turn to linear programming, prove to be somewhat easier to work with than the full set of algebraic numbers.

**Definition.** A complex number $\alpha$ is an *algebraic integer* if there exists a monic polynomial $F$ with integer coefficients such that $F(\alpha) = 0$.

It follows that every algebraic integer is also an algebraic number. The converse is not true, as can be seen from the next result.

**Proposition 2.7** *Let $\alpha$ be an algebraic number. Then $\alpha$ is an algebraic integer if and only if the minimal polynomial of $\alpha$ over the rationals has integer coefficients.*

In light of the following proposition, one can view an algebraic number as the quotient of an algebraic integer and an integer.

**Proposition 2.8** *Let $H(t) = t^d + q_{d-1}t^{d-1} + \cdots + q_1 t + q_0$ be a polynomial with coefficients $q_0, \ldots, q_{d-1} \in \boldsymbol{Q}$, and let $z$ be a common denominator for $q_0, \ldots, q_{d-1}$. Then if $\alpha$ is a root of $H$, the product $z\alpha$ is a root of $F(t) = t^d + zq_{d-1}t^{d-1} + \cdots + z^{d-1}q_1 t + z^d q_0$. Moreover, the coefficients of $F$ are all integers.*

Although always an algebraic number, the quotient of two algebraic integers is not, in general, an algebraic integer. The algebraic integers are closed, however, under addition, multiplication, and negation.

**Proposition 2.9** *The algebraic integers form a subring of the complex numbers.*

It will prove convenient to have the following shorthand notation for the algebraic integers.

**Definition.** We define $\boldsymbol{\mathcal{A}}$ to be the set of all algebraic integers, and we define $\boldsymbol{\mathcal{A}}_R$ to be $\boldsymbol{\mathcal{A}} \cap \boldsymbol{R}$, the set of all algebraic integers that are also real numbers.

As a final preliminary, we define some notation concerning matrices and vectors. Given a set $\boldsymbol{K}$, we use $\boldsymbol{K}^{r \times s}$ and $\boldsymbol{K}^r$ to denote the set of all $r \times s$ matrices and the set of all column $r$-vectors whose components belong to $\boldsymbol{K}$. Given a matrix or vector $M$, we use $M^T$ to denote the transpose of $M$.

# 3 Linear Programming over the Algebraic Integers

We consider the following linear program:

$$(P) \quad \max \quad c^T x$$
$$\text{s.t.} \quad Ax \leq b,$$

where $A \in \mathcal{A}_R^{m\times n}$ with full column rank, $b \in \mathcal{A}_R^m$, $c \in \mathcal{A}_R^n$, and $x \in \boldsymbol{R}^n$. Our goal in this section is to bound the complexity of problem $(P)$.

It is important to note that we lose no generality by working with the algebraic integers instead of the full set of algebraic numbers. Recall from proposition 2.8 that we can view an algebraic number as the quotient of an algebraic integer and an integer. It follows that, if we are given a LP of the form $(P)$ in which the coefficients are algebraic numbers — but not necessarily algebraic integers — then, by multiplying the constraints and the objective by an appropriate integer, we can transform the problem into an equivalent problem that involves only algebraic integers. It is straightforward to show that this transformation is polynomial in the input measures that will be defined later.

We now develop the tools that we shall use in analyzing the complexity of problem $(P)$. Our immediate goal is to establish upper and lower bounds on certain functions of $\alpha \in \mathcal{A}$ (there is no need to specialize to real numbers until we discuss systems of inequalities and linear programs, entities which are not generally defined in terms of complex numbers). As a first step toward this goal, we define a measure of the magnitude of the roots of the minimal polynomial of $\alpha$.

**Definition.** Let $\alpha$ be an algebraic integer of degree $d$, and let $\alpha_1, \ldots, \alpha_d$ be the conjugates of $\alpha$. We define the *conjugate norm* $S(\alpha)$ of $\alpha$ to be:

$$S(\alpha) = \max\{|\alpha_1|, \ldots, |\alpha_d|\},$$

where we use the standard notation $|\beta|$ for the magnitude of the complex number $\beta$ (i.e., $|\beta| = \sqrt{\beta\bar{\beta}}$, where $\bar{\beta}$ is the complex conjugate of $\beta$).

The following proposition gives the main algebraic and metric properties of the conjugate norm.

**Proposition 3.1** *Let $\alpha, \beta \in \mathcal{A}$. Then,*

*(i)* $S(a\alpha + b\beta) \le |a|S(\alpha) + |b|S(\beta)$, *for any integers $a$ and $b$,*

*(ii)* $S(\alpha\beta) \le S(\alpha)S(\beta)$,

*(iii)* $|\alpha| \le S(\alpha)$,

*(iv) If* $\alpha \neq 0$, *then* $|\alpha| \ge (S(\alpha))^{1-d}$, *where* $d = \deg(\alpha)$.

**Proof.** *(i)* By corollary 2.1, there exists an algebraic number $\theta$ such that $\alpha$ and $\beta$ have the unique representations $\alpha = F_\alpha(\theta)$ and $\beta = F_\beta(\theta)$ for some polynomials $F_\alpha$ and $F_\beta$ with rational coefficients. By proposition 2.4, we know that the conjugates of $\alpha$ and $\beta$ are the distinct members of $\{F_\alpha(\theta_j)\}$ and $\{F_\beta(\theta_j)\}$, respectively, where the $\theta_j$, $j = 1, \ldots, \deg(\theta)$, are the conjugates of $\theta$. Hence, $S(\alpha) = \max_j\{|F_\alpha(\theta_j)|\}$ and $S(\beta) = \max_j\{|F_\beta(\theta_j)|\}$. Since $a\alpha + b\beta = aF_\alpha(\theta) + bF_\beta(\theta)$ is also a rational polynomial in $\theta$, we also know that the conjugates of $a\alpha + b\beta$ are the distinct members of $\{aF_\alpha(\theta_j) + bF_\beta(\theta_j)\}$. Thus, we have $S(a\alpha + b\beta) = \max_j\{|aF_\alpha(\theta_j) + bF_\beta(\theta_j)|\}$
$\le |a|S(\alpha) + |b|S(\beta)$.
*(ii)* The proof of this statement follows by an argument similar to that used in *(i)*.
*(iii)* This statement is obvious from the definition of $S(\alpha)$.
*(iv)* Let $G(t) = (t - \alpha_1)(t - \alpha_2)\cdots(t - \alpha_d)$ be the minimal polynomial of $\alpha$, where we may assume $\alpha = \alpha_1$. Since $\alpha$ is an algebraic integer, we can also write $G(t) = t^d + z_{d-1}t^{d-1} + \cdots + z_1 t + z_0$ for some integer coefficients $z_0, \ldots, z_{d-1}$. It is clear that the constant term, $z_0$, must not be zero, since if it were zero we could divide $G(t)$ by $t$ to obtain a (monic) polynomial of strictly smaller degree, contradicting the irreducibility of $G(t)$. Hence, we have $|z_0| \ge 1$. But, by matching coefficients between the two expansions of $G(t)$, we see that $z_0 = \prod_{j=1}^d \alpha_j$. It follows that $\prod_{j=1}^d |\alpha_j| \ge 1$. Using the inequality $|\alpha_j| \le S(\alpha)$ implied by the definition of conjugate norm, we then have $|\alpha| = |\alpha_1| \ge (S(\alpha))^{1-d}$, which completes the proof of the proposition. □

We shall use proposition 3.1 to derive several useful results concerning matrices and systems of inequalities whose coefficients are algebraic integers. These results are most easily stated in terms of the notation introduced below.

**Definition.** Let $M \in \mathcal{A}^{r\times s}$ and let $M_{jk}$ denote the $jk^{\text{th}}$ entry of $M$. We define the *conjugate norm of the matrix $M$* to be

$$T(M) = \max_{j,k}\{S(M_{jk})\}.$$

Let $\ell$ denote the rank of $M$. Then we define the

*conjugate size of the matrix* $M$ to be

$$L(M) = \ell \log(\ell T(M)),$$

where by log we mean the base-2 logarithm. Additionally, we define the *degree of the matrix* $M$ to be the degree of the multiple algebraic extension $\boldsymbol{Q}(\{M_{jk}\})$, and we write this quantity as $\deg(M)$.

We use similar notation when discussing linear programs. Let $(Q)$ denote the problem $\{\max f^T v : Mv \leq g\}$, where $M \in \mathcal{A}_R^{r \times s}$, $g \in \mathcal{A}_R^r$, and $f \in \mathcal{A}_R^s$. Let $\Psi$ denote the set of all entries in $M$, $g$, and $f$. We define the *conjugate norm of the linear program* $(Q)$ to be

$$T(M, g, f) = \max_{\alpha \in \Psi} \{S(\alpha)\}.$$

Let $\ell$ denote the rank of $M$. We define the *conjugate size of the linear program* $(Q)$ to be

$$L(M, g, f) = \ell \log(\ell T(M, g, f)).$$

Additionally, we define the *degree of the linear program* $(Q)$ to be the degree of the multiple algebraic extension $\boldsymbol{Q}(\Psi)$, and we write this quantity as $\deg(M, g, f)$. We use analogous notation with respect to systems of linear inequalities.

Having fixed notation, we now derive some characteristics of a matrix determinant that are fundamental to our later work.

**Proposition 3.2** *Let* $B \in \mathcal{A}^{r \times r}$, *and let* $L = L(B)$ *and* $d = \deg(B)$. *If* $B$ *is nonsingular then*

*(i)* $\det(B) \in \mathcal{A}$,

*(ii)* $\deg(\det(B)) \leq d$,

*(iii)* $S(\det(B)) \leq 2^L$,

*(iv)* $2^{(1-d)L} \leq |\det(B)| \leq 2^L$.

**Proof.** The statement follows easily from the properties of conjugate norm given in 3.1. □

As a consequence of the last proposition, we obtain apriori bounds on the magnitude and conjugate norm of the vertices of polyhedra whose defining coefficients belong to $\mathcal{A}_R$.

**Proposition 3.3** *Let* $M \in \mathcal{A}_R^{r \times s}$, $g \in \mathcal{A}_R^r$, *and let* $L = L(M, g)$ *and* $d = \deg(M, g)$. *Suppose* $\bar{v}$ *is a vertex of* $\{v \in \boldsymbol{R}^s | Mv \leq g\}$. *Then every component* $\bar{v}_j$ *of* $\bar{v}$ *can be written in the form* $\bar{v}_j = \alpha_j / \beta$, *where*

*(i)* $\alpha_j, \beta \in \mathcal{A}_R$,

*(ii)* $\deg(\alpha_j) \leq d$, $\deg(\beta) \leq d$,

*(iii)* $S(\alpha_j) \leq 2^L$, $S(\beta) \leq 2^L$.

*Moreover, if* $\bar{v} \neq 0$ *then* $2^{-dL} \leq \|\bar{v}\|_\infty \leq 2^{dL}$.

**Proof.** The statement follows trivially from proposition 3.2 and Cramer's rule. □

Propositions 3.1 through 3.3 constitute our basic analytical tools. Using them, we can show how to modify almost any variant of the ellipsoid method [7] or the interior point method [6] to solve problem $(P)$ in time polynomial in the problem dimension, degree, and conjugate size. In the remaining part of this section, we outline a procedure that is centered on the ellipsoid method. We shall loosely follow the analysis given for problems with rational data by Papadimitriou and Steiglitz [15].

For the purposes of the complexity analysis, we assume that we have a machine that performs addition, subtraction, multiplication, division, and comparison of real numbers in constant time per operation. We refer to this model as the *real-number model of computation* and to algorithms derived under it as *real-number algorithms.* (See [2] for a treatment of the theory of general computation over the real numbers.)

We assume that the input of an instance of the linear program $(P)$ consists of the following items:

*(i)* $A \in \mathcal{A}_R^{m \times n}$, $b \in \mathcal{A}_R^m$, $c \in \mathcal{A}_R^n$,

*(ii)* $\deg(A, b, c)$,

*(iii)* $L(A, b, c)$.

Our strategy for establishing the polynomial-time solvability of $(P)$ is to establish a chain of polynomial problem transformations leading from $(P)$ to a problem that can be solved by the ellipsoid method in polynomial time.

We begin by noting that, given the duality theorem of linear programming, solving the linear program $(P)$ is no harder than solving a set of linear closed inequalities. As a matter of language, we say an algorithm *solves a system of inequalities* if it gives us a report that the instance is infeasible or a feasible solution to the instance, as appropriate.

**Proposition 3.4** *Let $M \in \mathcal{A}_R^{r\times s}$ with full column rank, $g \in \mathcal{A}_R^r$, and let $\delta, \lambda \in \boldsymbol{R}$ be such that $\deg(M,g) \leq \delta$ and $L(M,g) \leq \lambda$. Suppose there exists a real-number algorithm that, given $M$, $g$, $\delta$, and $\lambda$, solves $Mv \leq g$ in time polynomial in $r$, $\delta$, and $\lambda$. Then there exists a real-number algorithm that solves the linear program $(P)$ in time polynomial in $m$, $\deg(A,b,c)$, and $L(A,b,c)$.*

**Proof.** By writing the conditions of primal feasibility, dual feasibility, and equality of objectives values as a system of closed inequalities, we can reduce linear programming to linear closed inequalities. Proof of the proposition follows by noting that the row dimension, degree, and conjugate size of the system constructed from $(P)$ in this way are individually bounded from above by a polynomial in the corresponding quantities associated with $(P)$. □

Next we note that we can solve a system of linear closed inequalities by solving a closely related system of linear open inequalities.

**Proposition 3.5** *Let $M \in \mathcal{A}_R^{r\times s}$ with full column rank, $g \in \mathcal{A}_R^r$, and let $L = L(M,g)$ and $d = \deg(M,g)$. Let $e_r$ denote the $r$-vector of all ones. Then the open system $2^{2dL}Mv < 2^{2dL}g + e_r$ is feasible if and only if the closed system $Mv \leq g$ is feasible. Moreover, there exists a real-number algorithm that, given a solution to one system, finds a solution to the other system in time polynomial in $r$.*

**Proof.** The proof is an adaptation of that given by Papadimitriou and Steiglitz [15, lemma 8.7, pp. 173–174] for an analogous result concerning systems with rational coefficients. The main difference lies in the use of the properties of conjugate norm (proposition 3.1) to derive lower bounds on linear and polynomial forms. □

We require one more preliminary result before we can state the main result of the section.

**Proposition 3.6** *Let $M \in \mathcal{A}_R^{r\times s}, g \in \mathcal{A}_R^r$, and let $L = L(M,g)$ and $d = \deg(M,g)$. Let $\boldsymbol{K} = \{v \in \boldsymbol{R}^s | Mv < g\}$. Then if $\boldsymbol{K}$ is bounded and nonempty the $s$-dimensional volume of $\boldsymbol{K}$ satisfies $\mathrm{vol}_s(\boldsymbol{K}) \geq 2^{-4sdL}$.*

**Proof.** The proof of proposition 3.6 is an adaptation of a standard proof given for the analogous result concerning systems with rational coefficients (cf. [4] or [15]). The main difference lies in the use of the properties of conjugate norm (proposition 3.1) to derive lower bounds on linear and polynomial forms. □

We now state the main result of the section.

**Theorem 3.1** *There exists a real-number algorithm that solves the linear program $(P)$ in time polynomial in $m$, $\deg(A,b,c)$, and $L(A,b,c)$.*

**Proof.** (Outline) In light of proposition 3.4, it suffices to demonstrate the existence of a polynomial algorithm for solving systems of linear closed inequalities. By proposition 3.5, this problem is, in turn, polynomially reducible to that of solving a system of linear open inequalities. It follows from propositions 3.3 and 3.6, that we can always structure the reduction so that the resulting system is sufficiently well-bounded that it can be solved by the ellipsoid method (cf. [15] or [4]) in polynomial time. □

# 4 Extensions to Other Input Models

In the previous section, we derived a solution procedure and associated complexity bounds for problem $(P)$ under the assumption that, in addition to the coefficients that define the instance, input of an instance of $(P)$ includes bounds on the degree and conjugate size of the instance. In many situations these bounds may not be known apriori and so must be deduced, if possible, from information about the individual problem coefficients.

In this section, we discuss ways of obtaining problem degree and conjugate size bounds under several different assumptions about the form of the problem coefficients. Because we cannot anticipate every possible input form, the discussion is more

illustrative than exhaustive. We begin by working through an example in which the problem degree and conjugate size, although not apparent from the given data, are easy to bound. The nature of this analysis leads us to consider some results from number theory that serve as useful tools for obtaining bounds for other input forms. We then consider several examples that illustrate the use of these results. We conclude the section by showing how our results generalize from the algebraic integers to the algebraic numbers.

Throughout this section, we shall use $d$ to denote the degree and $L$ to denote the conjugate size of $(P)$; that is

$$d = \deg(A, b, c), \quad L = L(A, b, c).$$

Additionally, we shall use $\Psi$ to denote the set of all entries in $A$, $b$, and $c$.

**Example 4.1.** For every $\alpha \in \Psi$, suppose that, in addition to the actual numerical value of $\alpha$, we know a set of four integers $z_0$, $z_1$, $z_2$, $z_3$ such that $\alpha = z_0 + z_1\sqrt{2} + z_2\sqrt{3} + z_3\sqrt{6}$.

Our first goal is to bound the problem conjugate size, $L$, by a function of the quantities available to us. As it will turn out, a useful input measure for this purpose is the total number of bits in a binary representation of the integers that define the members of $\Psi$ in terms of the square roots. We use $E$ to denote this number.

Recall that $L$ is a nondecreasing function of $S(\alpha)$, $\alpha \in \Psi$. Hence, we may first bound the conjugate norms of the individual input coefficients and then calculate a bound on $L$ from these quantities. Proceeding along these lines, we note that if $\alpha = z_0 + z_1\sqrt{2} + z_2\sqrt{3} + z_3\sqrt{6}$, then by the properties of the conjugate norm (proposition 3.1) we have

$$\begin{aligned} S(\alpha) &= S(z_0 + z_1\sqrt{2} + z_2\sqrt{3} + z_3\sqrt{6}) \\ &\leq |z_0| + |z_1|S(\sqrt{2}) + |z_2|S(\sqrt{3}) + |z_3|S(\sqrt{6}). \end{aligned}$$

The problem of bounding $L$ is thus reduced to that of finding (or bounding) the conjugate norms of the square-root terms.

Let $p$ be a positive integer such that $p$ is not the square of another integer. It is easy to show that $F(t) = t^2 - p$ is the minimal polynomial of $\sqrt{p}$. It follows that the conjugates of $\sqrt{p}$ are $\sqrt{p}$ and $-\sqrt{p}$. Hence, we have $S(\sqrt{p}) = \max\{|\sqrt{p}|, |-\sqrt{p}|\} = \sqrt{p}$. Substitution in our earlier inequality then gives

$$S(\alpha) \leq |z_0| + |z_1|\sqrt{2} + |z_2|\sqrt{3} + |z_3|\sqrt{6},$$

which holds for any $\alpha \in \Psi$. Using this bound in the formula that defines $L$, it is straightforward to show that $L \leq mE$.

It remains for us to bound $d$, the degree of $(P)$. First note that, by the definition of the degree of a linear program, $d$ equals the degree of the multiple algebraic extension $\boldsymbol{Q}(\Psi)$. It is obvious from the form of the problem coefficients that every $\alpha \in \Psi$ also belongs to $\boldsymbol{Q}(\sqrt{2}, \sqrt{3}, \sqrt{6})$. This implies $\boldsymbol{Q}(\Psi) \subset \boldsymbol{Q}(\sqrt{2}, \sqrt{3}, \sqrt{6})$. It follows from corollary 2.1 that $d$ is no larger than the degree of $\boldsymbol{Q}(\sqrt{2}, \sqrt{3}, \sqrt{6})$. Letting $\bar{d}$ denote this last quantity and using proposition 2.5, we then have

$$\begin{aligned} d \leq \bar{d} &\leq (\deg(\sqrt{2}))(\deg(\sqrt{3}))(\deg(\sqrt{6})) \\ &= 8, \end{aligned}$$

where we have used the fact that $\deg(\sqrt{2}) = \deg(\sqrt{3}) = \deg(\sqrt{6}) = 2$.

Actually, we can tighten the bound on $d$ somewhat by noting that, since $(\sqrt{2})(\sqrt{3}) = \sqrt{6}$, the extensions $\boldsymbol{Q}(\sqrt{2}, \sqrt{3}, \sqrt{6})$ and $\boldsymbol{Q}(\sqrt{2}, \sqrt{3})$ are, in fact, the same, and so must have the same degree. Hence we have $d \leq 4$. Although insignificant in this simple example, the savings from observations of this kind can sometimes be quite substantial (cf. example 4.2).

As a final observation we note that, based on our bounds for $d$ and $L$, theorem 3.1 implies that $(P)$ can be solved in time polynomial in $m$ and $E$.

The ad hoc analysis given in example 4.1 illustrates an effective strategy for many input forms. To bound $L$, we first bound $S(\alpha)$ for each $\alpha \in \Psi$. If we know a representation for $\alpha$ in terms of other algebraic integers, we use the properties of the conjugate norm to reduce the problem to that of bounding the conjugate norm of those algebraic integers. To bound $d$, we identify a set of algebraic numbers — hopefully smaller than $\Psi$ itself — such that the multiple algebraic extension generated by these numbers includes $\boldsymbol{Q}(\Psi)$. We then can claim that $d$ is at most the product of the degrees of these algebraic integers.

The input form considered in example 4.1 is a special case of the form $\alpha = \sum z_j \theta_j$, where $z_j$ is an integer and $\theta_j$ is an algebraic integer. It is clear that in order to use the general approach outlined above to find bounds for problems with this form we must know something about the degree and conjugate norm of $\theta_j$, for each $j$. This reflects a general caveat; in any attempt to bound the degree and conjugate size of a linear program, we ultimately reach a point where we must bound the degree and conjugate norm of an individual algebraic integer. Therefore, it is worth considering how to obtain these bounds from the auxiliary information commonly associated with an algebraic integer.

We begin along these lines by introducing some additional terminology concerning polynomials and algebraic integers.

**Definition.** Let $F(t) = z_d t^d + \cdots + z_1 t + z_0$ be a polynomial with coefficients $z_0, \ldots, z_d \in Z$. We define the *height of the polynomial* $F$ to be $\max_j\{|z_j|\}$. We define the *height of an algebraic integer* to be the height of its minimal polynomial.

Next we state a well-known result from the theory of transcendental numbers that relates the conjugate norm of an algebraic integer to its height.

**Proposition 4.1** *Let $\alpha$ be an algebraic integer of height $h$. Then $S(\alpha) \leq 2h$.*

See, e.g., [18] for proof of proposition 4.1.

For the purpose of bounding the degree and height of an algebraic integer, it suffices to know the degree and height of any monic, integral polynomial of which the algebraic integer is a root, as the following proposition shows.

**Proposition 4.2** *Let $F$ be a monic polynomial with integer coefficients, degree $\ell$, and height $h$. Then every root of $F$ is an algebraic integer of degree at most $\ell$ and height at most $4^\ell h$.*

See, e.g., [13] for proof of a generalization of proposition 4.2.

As an illustration of the use of the preceding results, we next obtain bounds for an input form that arises in the theory of LPs whose coefficient matrices are circulant.

**Example 4.2.** Let $\omega$ be the first primitive $p^{th}$ root of unity; that is,

$$\omega = e^{2\pi i/p}, \text{ where } i = \sqrt{-1} \text{ and } p \text{ is integer.}$$

In addition to $\Psi$, suppose that we know the following:

(*i*) a positive integer $p$ such that every $\alpha \in \Psi$ can be written as $\alpha = \sum_{j=0}^{p-1} z_j \omega^j$ for integers $z_0, \ldots, z_{p-1}$.

(*ii*) for every $\alpha \in \Psi$, a set of integers $z_0, \ldots, z_{p-1}$ such that $\alpha = \sum_{j=0}^{p-1} z_j \omega^j$.

Let $E$ denote the total number of bits in a binary representation of the integers $z_j$ in (*ii*) above.

We begin by finding a bound on the conjugate norm of each member of $\Psi$. If $\alpha = \sum_{j=0}^{p-1} z_j \omega^j$, then we have

$$S(\alpha) \leq \sum_{j=0}^{p-1} |z_j| S(\omega^j).$$

We now concentrate on bounding $S(\omega^j)$, for all $j \in \{0, \ldots, p-1\}$. First note that since $\omega^{jp} = 1$ for all $j = 0, \ldots, p-1$, it is clear that $\omega^j$ is a root of the polynomial $F(t) = t^p - 1$. Although $F$ is not the minimal polynomial of $\omega^j$, it does give us enough information to bound $S(\omega^j)$. In particular, since $F$ has degree $p$ and height 1, proposition 4.2 implies that $\omega^j$ has height at most $4^p$. Proposition 4.1 then gives $S(\omega^j) \leq 4^{p+1}$, which in turn gives

$$S(\alpha) \leq 4^{p+1} \sum_{j=0}^{p-1} |z_j|.$$

Using this last bound it is easy to show that $L \leq 3mE$.

To bound $d$, note that $\boldsymbol{Q}(\Psi) \subset \boldsymbol{Q}(\omega)$, and so $d \leq \deg(\omega)$. But $\omega$ is a root of $F(t) = t^p - 1$ and so by proposition 4.2 has degree at most $p$. Hence we have $d \leq p$. (In fact, it is possible to sharpen our bounds on $L$ and $d$ somewhat by using additional results from number theory concerning the roots of unity.)

Combining our bounds on $d$ and $L$ with theorem 3.1, we see that $(P)$ can be solved in time polynomial in $m$, $p$, and $E$.

LPs of the form considered in example 4.2 are analyzed in a different manner but with similar results in [1].

We next consider LPs in which we have direct knowledge of the minimal polynomial of each algebraic integer in the problem.

**Example 4.3.** For every $\alpha \in \Psi$, suppose that we know both the numerical value of $\alpha$ and the minimal polynomial of $\alpha$.

Let $\bar{d} = \prod_{\alpha \in \Psi} \deg(\alpha)$, the product of the degrees of the minimal polynomials associated with the members of $\Psi$. Also, let $E$ denote the total number of bits in a binary representation of the coefficients of these polynomials.

Using proposition 4.1, it is straightforward to show that $L$ is bounded from above by the problem encoding size. To bound $d$, we again make use of the fact that $d$ is the degree of the multiple algebraic extension $\boldsymbol{Q}(\Psi)$. Recall that proposition 2.5 states that the degree of a multiple algebraic extension is at most the product of the degrees of the algebraic numbers that define the extension. Hence, we have $d \leq \bar{d}$.

Using theorem 3.1 and our bounds on $d$ and $L$, we see that $(P)$ can be solved in time polynomial in $m$, $\bar{d}$, and $E$.

Using proposition 4.2, it is easy to generalize the results in example 4.3 to the case in which the polynomials associated with the problem coefficients are monic and integral but not necessarily irreducible. The results are similar to the irreducible case.

# 5 Rational-number Model of Computation

The complexity bounds given in sections 3 and 4 are derived under a model of computation that allows real-number input and arithmetic. In fact, this assumption is stronger than necessary. One can achieve materially the same results under a rational-number model of computation by using a well-known scheme for the symbolic manipulation of algebraic numbers in conjunction with the results of section 3. We provide an outline of such a procedure below, but defer a full exposition of the (somewhat messy) details to a subsequent paper.

Recall from proposition 2.4 that the conjugates of an algebraic number are distinct. Hence, if $\alpha$ is a real algebraic number with minimal polynomial $G$, then there exists an interval with rational endpoints, say $[q_1, q_2]$, that contains $\alpha$ but does not contain any other root of $G$. Since the triplet $(G; q_1, q_2)$ unambiguously defines $\alpha$, we can represent $\alpha$ in a rational-number machine by storing $q_1$, $q_2$, and the coefficients of $G$.

It can be shown that the roots of $G$ are sufficiently small and well-separated that there exists an isolating interval whose binary encoding size is polynomial in the binary encoding size of the coefficients of $G$. In fact we can construct such an interval; there are several well-known algorithms that isolate the roots of a polynomial in time polynomial in the binary size of its coefficients (see [3] for a survey of such algorithms).

To be useful as part of a linear programming algorithm, the triplet scheme must allow us to manipulate algebraic numbers as well as represent them. It follows from some results of Lovasz [9], that, given triplets for two algebraic numbers, we can find a triplet that represents their sum, difference, product, or quotient in time polynomial in the encoding lengths of the given triplets. We can also compare two triplets in polynomial time.

Equipped with a means of representing and manipulating algebraic numbers, we can establish complexity bounds for linear programming in much the same manner, and with similar results, as is done in section 3. The overall bound is polynomial in the problem dimension, problem degree, and the bit size of the triplets that represent the input coefficients.

An unavoidable extra complication of working with a rational-number model of computation is that we must be careful to control the bit size of the rational numbers that occur during the course of the linear programming algorithm. As in the analysis of ordinary rational-number LPs, the two subalgorithms of particular concern are the ellipsoid method and Gaussian elimination. Also as in the standard analysis, we can ensure that these procedures stay under control by relating the size of each intermediate number to that of the input data.

In the case of the ellipsoid method, this can be

done with the aid of the standard finite precision variant developed for rational-number LPs (see, e.g. [15] or [4]). Briefly, we use limited-precision rational numbers to calculate and represent each ellipsoid. Exact (triplet) arithmetic is used only for the linear separation subroutine. A feasible triplet can then be recovered using Gaussian elimination (cf. proposition 3.5).

In the case of Gaussian elimination, we can arrange the computation so that each intermediate number is a subdeterminant of the input matrix (see, e.g., [4]). But by using the results in section 3, it is easy to show that the encoding size of any subdeterminant is bounded by a polynomial in the encoding size of the input matrix and the degree of the multiple algebraic extension defined by its entries.

# 6 Remarks

1) Using the basic analytical tools presented in section 3, one can modify the algorithm for combinatorial LPs given by Tardos [19] so that it works with LPs whose coefficients are algebraic integers. As in the rational case, the running time of the resulting algorithm is independent of the data in the objective and right-hand side. The details of this extension are quite similar to those given in [1], where Tardos' scheme is extended from the rationals to the cyclotomic integers.

We also note that it may be possible to make a similar extension to the algorithm of Norton, Plotkin, and Tardos [14], which solves LPs in time independent of the data in a fixed number of rows or columns of the coefficient matrix.

2) In [1] it is shown that standard form LPs with circulant coefficient matrices can be solved in strongly polynomial time. The basic idea is to transform the given problem into an equivalent problem in which the entries of the coefficient matrix belong to the subring of the algebraic integers discussed in example 4.3 and are small in conjugate norm. This can be done by multiplying the equality constraints by the pseudoinverse of the coefficient matrix. The transformed problem can then be solved in strongly polynomial time by using a variant of the Tardos scheme (see the previous remark) in conjunction with a polynomial-time algorithm for LPs whose coefficients belong to the subring.

In light of the results in this paper, which extend the linear programming results in [1] to general algebraic numbers, it is worth investigating whether other classes of problems can be shown to be strongly polynomial by arguments similar to those used for circulant LPs. Equivalently, we can ask whether there are other simultaneously diagonalizable families of matrices whose diagonalizing matrix is composed of small algebraic numbers (not necessarily from the subring in [1]). Indeed, it appears that such families do exist and that considerable progress in identifying them can be made by using results from the theory of group representations. We plan to report on this topic in a subsequent paper.


## Acknowledgments

We thank Leonid Khachiyan, Uriel Rothblum, and Ron Shamir for valuable discussions on extending our earlier work with cyclotomic integers to the present case. We are also indebted to Michael Todd for perceptive comments about the generalization from algebraic integers to algebraic numbers.

This research was funded by the National Science Foundation under grant DMS88-10192. We greatly appreciate this financial support.


# References


[1] I. Adler and P. A. Beling, Polynomial Algorithms for LP over a Subring of the Algebraic Integers with Applications to LP with Circulant Matrices, *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science*, pp. 480–487, 1991.

[2] L. Blum, M. Shub and S. Smale, On a Theory of Computation and Complexity over the Real Numbers; NP-completeness, Recursive Functions and Universal Machines, *Bulletin of the AMS* **21**, No. 1, pp. 1–46, 1989.

[3] G. E. Collins and R. Loos, Real Zeros of Polynomials, in: B. Buchberger, G. E. Collins, and

R. Loos (eds.), *Computer Algebra*, Springer-Verlag, Wien, 1983, pp. 83-94.

[4] M. Grotschel, L. Lovasz and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[5] K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory*, Springer-Verlag, New York, 1972.

[6] N. Karmarkar, A New Polynomial Time Algorithm for Linear Programming, *Combinatorica* **4**, pp. 373–395, 1984.

[7] L. Khachiyan, A Polynomial Algorithm in Linear Programming, *Soviet Mathematics Doklady* **20**, pp. 191–194, 1979.

[8] R. Loos, Computing in Algebraic Extensions, in: B. Buchberger, G. E. Collins, and R. Loos (eds.), *Computer Algebra*, Springer-Verlag, Wien, 1983, pp. 173-187.

[9] L. Lovasz, *An Algorithmic Theory of Numbers, Graphs and Convexity*, Society for Industrial and Applied Mathematics, Pennsylvania, 1986.

[10] N. Megiddo, Towards a Genuinely Polynomial Algorithm for Linear Programming, *SIAM Journal on Computing* **12**, No. 2, pp. 347–353, 1983.

[11] N. Megiddo, Linear Programming in Linear Time When the Dimension is Fixed, *Journal of the Association for Computing Machinery* **31**, pp. 114–127, 1984.

[12] N. Megiddo, On Solving the Linear Programming Problem Approximately, in: J.C. Lagarias and M.J. Todd (eds.), *Contemporary Mathematics* **114**: *Mathematical Developments Arising from Linear Programming*, American Mathematical Society, 1990, pp. 35–50.

[13] M. Mignotte, Some Useful Bounds, in: B. Buchberger, G. E. Collins, and R. Loos (eds.), *Computer Algebra*, Springer-Verlag, Wien, 1983, pp. 259–263.

[14] C. Norton, S. Plotkin and E. Tardos, Using Separation Algorithms in Fixed Dimension, *Proceedings of the 1st ACM/SIAM Symposium on Discrete Algorithms*, pp. 377–387, 1990.

[15] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, Inc., New Jersey, 1982.

[16] H. Pollard and H. G. Diamond, *The Theory of Algebraic Numbers*, 2nd Edition, The Mathematical Association of America, 1975.

[17] I. N. Stewart and D. O. Tall, *Algebraic Number Theory*, Chapman and Hall, New York, 1987.

[18] A. B. Shidlovskii, *Transcendental Numbers*, Walter de Gruyter & Co., Berlin, 1989.

[19] E. Tardos, A Strongly Polynomial Algorithm to Solve Combinatorial Linear Programs, *Operations Research* **34**, pp. 250–256, 1986.

[20] J. F. Traub and H. Wozniakowski, Complexity of Linear Programming, *Operations Research Letters* **1**, pp. 59–62, 1982.