THE DIOPHANTINE PROBLEM OF FROBENIUS: A CLOSE BOUND

by

H. Krawczyk and A. Paz

Technical Report #483

December 1987
The Diophantine Problem of Frobenius: A Close Bound

Hugo Krawczyk
Azaria Paz

Computer Science Dept.
Technion
Haifa, Israel

ABSTRACT

The conductor of $n$ positive integer numbers $a_1, a_2, \ldots, a_n$, whose greatest common divisor is equal to 1, is defined as the minimal $K$, such that for every $m \geq K$, the equation $a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = m$, has a solution over the nonnegative integers. In this note we give a polynomial algorithm computing a close bound $B$ for the conductor $K$ of $n$ given positive integers, when $n$ is fixed. The bound $B$ satisfies $B/n \leq K \leq B$.
INTRODUCTION

Definition: The conductor of \( n \) positive integer numbers \( a_1, a_2, \ldots, a_n \), whose greatest common divisor is equal to 1, is the minimal \( K \), such that for every \( m \geq K \), the equation

\[
 a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = m
\]

has a solution over the nonnegative integers. We denote this minimum by \( K = K(a_1, a_2, \ldots, a_n) \).

The Diophantine Problem of Frobenius is to determine the conductor of \( n \) given positive integers \([\text{Fro}].\)

In this paper we present an algorithm for the computation of a close bound \( B \) for the conductor \( K \). This bound has the property that \( B/n \leq K \leq B \). For fixed \( n \), the time complexity of the algorithm is bounded by a polynomial in the length of the input integers \( a_1, a_2, \ldots, a_n \).

Up to the present no polynomial algorithm for the computation of the conductor for \( n > 3 \) has been found, not even for the case where the number \( n \) of integers is not part of the input. For the case of \( n = 2 \), a very simple solution due to Sylvester is known [Syl]. While no such solution is known for \( n = 3 \), it is easy to show that a polynomial algorithm can be derived from the work of Brauer and Shockley [BS] combined with Lenstra's polynomial algorithm for solving integer linear programs with a fixed number of variables [Len]. We are not aware of any polynomial algorithm existing or implied in the literature for the \( n \geq 4 \) case. A non-polynomial algorithm for the computation of the conductor can be found in [Nij].

Several authors have tried to find a good upper bound for the conductor (see [Sel, Sch] for an extensive bibliography). Many such bounds have been found but all those bounds are of the order of magnitude of the square of the minimal \( a_i \) or higher. The bound we give in this note is, as far as we know, the first bound which is, for fixed \( n \), of the same order of magnitude as the conductor and computable in polynomial time.

THE MAIN RESULT

Let us denote by \( a_i, 1 \leq i \leq n \), the minimal integer \( \alpha \) such that there exists a solution over the nonnegative integers to the equation

\[
 a_1 x_1 + \cdots + a_{i-1} x_{i-1} + a_{i+1} x_{i+1} + \cdots + a_n x_n = \alpha a_i - 1
\]

Denote

\[
 B = (\alpha_i - 1) a_1 + (\alpha_{i-1} - 1) a_2 + \cdots + (\alpha_1 - 1) a_n
\]

**Theorem 1:** Let \( K \) be the conductor of \( n \) positive integers \( a_1, a_2, \ldots, a_n \), whose greatest common divisor is equal to 1, and let \( B \) be defined as above then

\[
 \frac{B}{n} \leq K \leq B
\]

**Proof:** First we prove the lower bound. We must show that \( B \leq n K \). We do this by showing for every \( i, 1 \leq i \leq n \), that \((\alpha_i - 1) a_i \leq K\). If this is not true, then it follows from the definition of \( K \) that the equation \( a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = (\alpha_i - 1) a_i - 1 \), has a
solution over the nonnegative integers, implying that the equation
\[ a_1 x_1 + \cdots + a_{i-1} x_{i-1} + a_i x_i + a_{i+1} x_{i+1} + \cdots + a_n x_n = (\alpha_{i-1} - x_i) a_i - 1 \]
has such a solution, contradicting the minimality of \( \alpha_i \).

To prove the upper bound we shall show that for any \( m \geq B \), there exists a solution to the equation \( a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = m \). We prove this by showing that if for any \( m > B \) a solution exists, then a solution exists also for \( m - 1 \). Let \( \beta_1, \beta_2, \ldots, \beta_n \), be a solution for such an \( m \). As \( m > B \), there exists some index \( i, 1 \leq i \leq n \), such that \( \beta_i > \alpha_i - 1 \). On the other hand, by the definition of \( \alpha_j \), there exist nonnegative integers \( \alpha'_1, \ldots, \alpha'_{i-1}, \alpha'_{i+1}, \ldots, \alpha'_n \) such that
\[ \alpha'_1 a_1 + \cdots + \alpha'_{i-1} a_{i-1} - \alpha_i a_i + \alpha'_{i+1} a_{i+1} + \cdots + \alpha'_n a_n = -1 \]

Combining the two equations we get that
\[ \beta_1 + \alpha'_1, \ldots, \beta_{i-1} + \alpha'_{i-1}, \beta_i - \alpha_i, \beta_{i+1} + \alpha'_{i+1}, \ldots, \beta_n + \alpha'_n \]
is a (nonnegative) solution for \( m - 1 \).

**Theorem 2:** The bound \( B \) can be computed in polynomial time for every fixed value \( n \).

**Proof:** The equation (*) can be solved for any value of \( \alpha \) as a linear integer program, and therefore every \( \alpha_i \) can be found by binary-search of the minimal value of \( \alpha \) for which a solution to (*) exists. Using Lenstra's polynomial algorithm for the Integer Linear Programming [Len], we can compute the bound \( B \) in polynomial-time when \( n \) is fixed.

**Remark 1:** In the proof of Theorem 1 we have shown that \( (\alpha_i - 1) a_i \leq K \). Combining this with the well-known bound \( K \leq (a_{\min} - 1)(a_{\max} - 1) \) [Bra] (where \( a_{\min}, a_{\max} \) are the minimal and maximal elements, respectively, among \( a_1, a_2, \ldots, a_n \)), we get that \( \alpha_i < a_{\max} \), a bound that can be used for initialization in the above binary search for \( \alpha_i \).

**Remark 2:** The above bound also induces a bound for the number of nonnegative integers \( m \) for which no solution exists to the equation \( a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = m \). This number, denoted by \( N \) was also investigated in the literature [NW] and it is easy to prove that \( K/2 \leq N \leq K \). Thus, our bound provides also a close bound for \( N \). Namely, \( B/2n \leq N \leq B \).
REFERENCES


