Bjarke Hammersholt Roune

Department of Computer Science
University of Aarhus
IT-parken, Aabogade 34
DK-8200 Aarhus N, Denmark

Solving Thousand Digit Frobenius Problems Using Gröbner Bases

Published in Journal of Symbolic Computation
volume 43 issue 1, January 2008
doi:10.1016/j.jsc.2007.06.002

Abstract

A Gröbner basis-based algorithm for solving the Frobenius Instance Problem is presented, and this leads to an algorithm for solving the Frobenius Problem that can handle numbers with thousands of digits. Connections to irreducible decompositions and Hilbert functions are also presented.

1. Introduction

Let $p_1, \ldots, p_n$ be relatively prime positive integers and let $p = (p_1, \ldots, p_n)$. An integer is *p-representable* if it can be written as $v \cdot p$ for some $v \in \mathbb{N}^n$. Determining p-representability is known as the *Frobenius Instance Problem*, and we present a new Gröbner basis-based algorithm that solves it.[1] Our algorithm differs from the classical way to solve integer programs using Gröbner bases due to Conti and Traverso (1991) (see also Pottier (1994)) by not adding any auxiliary variables to the problem.

The *Frobenius Number* f_p^* is the largest integer that is not p-representable, and such an integer exists by Proposition 2 below. E.g. if $p = (6, 10, 15)$ then $f_p^* = 29$.

The *Frobenius Problem* is to compute the Frobenius Number f_p^*. A recent algorithm due to Einstein et al. (2007) can solve Frobenius problems even if the p_i have thousands of decimal digits. Here we describe a simpler variant of that algorithm that performs better.

URL: `http://www.broune.com/` (Bjarke Hammersholt Roune).

[1] Malkin (2006) has discovered this independently.

The algorithm first computes a Gröbner basis and then determines the Frobenius number based on that. Einstein et al. (2007) use a novel algorithm based on the Fundamental Domain that in effect computes this Gröbner basis, and they report that this is much faster than using Buchberger's algorithm in their implementation.

However, the benchmarks in section 7 show that the program 4ti2 [4ti2 team (2006)], which implements Buchberger's algorithm in a special case, outperforms the Fundamental Domain-based implementation from Einstein et al. (2007). The record for random p_i with 11 decimal digits was $n = 11$, but we can now reach $n = 13$. Performance is also improved on smaller examples.

We show that the second step of the algorithm of Einstein et al. (2007) can be rephrased as the computation of the irreducible decomposition of a certain monomial ideal. The final step of the algorithm is to maximize a linear function over the decomposition, and we show that this essentially computes the index of regularity of the ideal. We define these terms when they are needed.

See Jensen et al. (2007) for a geometric version of some of the ideas in this paper in terms of maximal lattice free bodies. That paper is joint work with Niels Lauritzen and Anders Nedergaard Jensen.

We refer to the book by Ramírez Alfonsín (2005) for more background on the Frobenius Problem. We wish to thank Anders Nedergaard Jensen, Niels Lauritzen, Daniel Lichtblau and Stan Wagon for helpful discussions about Frobenius numbers.

2. Preliminaries

Let $e_i \in \mathbb{N}^n$, $i = 1, \ldots, n$, be the vector whose entries are all zero except that there is a 1 at position i. The *p-degree* of a vector $v \in \mathbb{Z}^n$ is $v \cdot p$. If $v \in \mathbb{Z}^n$ then define $v^+ \in \mathbb{N}^n$ as below and define $v^- := (-v)^+$.

$$v_i^+ := \begin{cases} v_i, & \text{for } v_i \geq 0 \\ 0, & \text{for } v_i < 0 \end{cases}$$

We will need to consider the lattice ideal $I_\mathcal{L}$ defined by

$$I_\mathcal{L} := \left\langle x^{v^+} - x^{v^-} \,\middle|\, v \in \mathbb{Z}^n \text{ and } v \cdot p = 0 \right\rangle.$$

We will compute a Gröbner basis G of $I_\mathcal{L}$. The term order $\leq$ we will use first considers the p-degree of the exponent vector of a term and then the reverse lexicographic order where $x_1 < x_2 < \cdots < x_n$.

Example 1. If $p = (2, 3)$ then $1 = x^{(0,0)} \leq x^{(1,0)} \leq x^{(3,0)} \leq x^{(0,2)} \leq x^{(4,5)} = x_1^4 x_2^5$.

We refer to Cox et al. (1997) for more details.

Proposition 2. *Only finitely many integers $t \in \mathbb{N}$ are not p-representable.*

Proof. As $p_1, \ldots, p_n$ are relatively prime, iterated use of the Extended Euclidean Algorithm provides us with a vector $v \in \mathbb{Z}^n$ such that $v \cdot p = 1$. Let $m := \min_{i=1}^n v_i$ and define $u := v + p_1|m|(1, \ldots, 1)$. Then $u + iv \in \mathbb{N}^n$ for $i = 0, \ldots, p_1 - 1$ and therefore $(u + iv) \cdot p = u \cdot p + i$ are p_1 consecutive p-representable numbers. □

3. An Algorithm That Solves The Frobenius Instance Problem

We claim that the following algorithm determines if $t \in \mathbb{N}$ is p-representable.

Step 1: Compute an $a \in \mathbb{Z}^n$ such that $a \cdot p = t$, $a_1 \leq 0$ and $a_i \geq 0$ for $i = 2, \ldots, n$.

Step 2: Divide $x^{a^+} - x^{a^-}$ by G giving remainder $x^w(x^{c^+} - x^{c^-})$ for some $w \in \mathbb{N}^n$ where $x^{c^-} \leq x^{c^+}$.

Step 3: Then t is p-representable if and only if $c \in \mathbb{N}^n$.

Step 1 We first find an $a \in \mathbb{Z}^n$ such that $a \cdot p = t$. One way to do this is to use the Extended Euclidean Algorithm iteratively to find a $b \in \mathbb{Z}^n$ such that $b \cdot p = 1$ and then let $a := tb$. As $(-\sum_{i=2}^n p_i, p_1, \ldots, p_1)$ has p-degree zero and the sign pattern $(-, +, \cdots, +)$, we can assume that a also has this sign pattern by adding a sufficiently large multiple of this vector to a.

Step 2 This step requires knowing the Gröbner basis G, and computing G is the most time consuming part of the algorithm. Once G has been computed the polynomial division itself is comparatively fast.

Step 3 Observe that the division algorithm ensures that c will have no more negative entries than a does, so c is negative at most in the first coordinate. As x^{c^+} is not reducible by G, the following lemma tells us that if t is p-representable then $c_1 \geq 0$ so that c is a p-representation of t since $c \cdot p = a \cdot p = t$.

Lemma 3. *Let $a \in \mathbb{Z}^n$ such that $a_i \geq 0$ for $i = 2, \ldots, n$ and $a_1 < 0$. If $a \cdot p$ is p-representable then there exists a $g \in G$ such that $in_{\leq}(g) | x^{a^+}$.*

Proof. As $a \cdot p$ is p-representable there exists a $b \in \mathbb{N}^n$ such that $a \cdot p = b \cdot p$. Letting $d := a - b$ this implies that $d \cdot p = 0$ whereby $h := x^{d^+} - x^{d^-} \in I_{\mathcal{L}}$. Thus there exists a $g \in G$ such that $\mathrm{in}_{\leq}(g) | \mathrm{in}_{\leq}(h)$. We will prove that $\mathrm{in}_{\leq}(g) | \mathrm{in}_{\leq}(h) = x^{d^+} | x^{a^+}$.

$\mathbf{in}_{\leq}(\mathbf{h}) = \mathbf{x}^{\mathbf{d}^+}$**:** d^+ and d^- have the same p-degree and $d_1 = a_1 - b_1 < 0$.

$\mathbf{x}^{\mathbf{d}^+} | \mathbf{x}^{\mathbf{a}^+}$**:** This follows from $b \in \mathbb{N}^n$. □

Note that $\leq$ can be replaced with any term order that first considers the p-degree of the exponent vector and then the reverse lexicographic order on the first variable.

4. An Algorithm That Solves The Frobenius Problem

The general idea of the algorithm is that we can represent f_p^* by a certain vector that has p-degree f_p^*, and that this vector has certain properties (see Proposition 4). It turns out that only finitely many vectors have these properties, so we can look through all of them, and then the one with maximal p-degree will be the vector that represents f_p^*, i.e. it will have p-degree equal to f_p^* (see Proposition 5).

Proposition 4 spells out the properties mentioned above.

Proposition 4. *Let $c \in \mathbb{Z}^n$ be the vector resulting from running the algorithm from Section 3 on $t = f_p^*$. Then the following holds.*

(M1) $c_1 = -1$ *and* $c_i \geq 0$ *for* $i = 2, \ldots, n$.

(M2) x^{c^+} *cannot be reduced by any element of* G.

(M3) $x^{(c+e_i)^+}$ can *be reduced by some* $g_i \in G$ *for* $i = 2, \ldots, n$.

Proof. (M1): It holds by construction that $c_1 < 0$ and that $c_i \geq 0$ for $i = 2, \ldots, n$. Let $c' := c + e_1$. Then $x^{(c')^+} = x^{c^+}$ is not reducible by G and $c' \cdot p$ is p-representable as $c' \cdot p > f_p^*$. Then Lemma 3 implies that $c'_1 \geq 0$ whereby $c_1 = -1$.

(M2): This holds by construction.

(M3): We see that $(c + e_i) \cdot p$ is p-representable as it is strictly larger than f_p^*. Thus Lemma 3 provides a $g_i \in G$ such that $\text{in}_\leq(g_i)|x^{(c+e_i)^+}$. □

Let M_p be the set of vectors $c \in \mathbb{Z}^n$ that have the properties (M1), (M2) and (M3) from Proposition 4. The idea is to compute M_p and then use Proposition 5 below to find f_p^*.

Proposition 5. *The following holds.*

(i) M_p *is finite.*

(ii) If $a \in M_p$ *then* $a \cdot p$ *is not* p*-representable.*

(iii) $f_p^* = \max\{a \cdot p | a \in M_p\}$

Proof. (i): Let $a \in M_p$ and $i \in \{2, \ldots, n\}$. Let $g \in G$ such that $\text{in}_\leq(g)|x^{(a+e_i)^+}$. As $\text{in}_\leq(g)$ does not divide x^{a^+} we can infer that $a_i + 1$ is the exponent of x_i in $\text{in}_\leq(g)$. Thus there are at most $|G|$ possibilities for what a_i can be.

(ii): Lemma 3 shows that this follows from (M1) and (M2).

(iii): Proposition 4 shows that there is an $a \in M_p$ such that $a \cdot p = f_p^*$. This and part (ii) above shows what we need. □

We can compute M_p as follows. We saw in the proof of Proposition 5 that if $i \in \{2, \ldots, n\}$ and $a \in M_p$ then $a_i + 1$ is the exponent of x_i in $\text{in}_\leq(g_i)$ for some $g_i \in G$. Thus we can run through all possible values of a_i for $i = 2, \ldots, n$ and only keep those a that have properties (M1), (M2) and (M3).

This algorithm is easy to understand and implement, but it requires us to look through up to $|G|^{n-1}$ possibilities. The External Corner Algorithm from Einstein et al. (2007) is a more efficient algorithm for computing M_p which usually dramatically reduces the number of possibilities that need to be examined.

In Section 5 we will define the irreducible decomposition of a monomial ideal, and we will prove that M_p corresponds to the irreducible decomposition of the initial ideal of $I_{\mathcal{L}}$. Roune (2007) shows how an algorithm that is very similar to the External Corner Algorithm can compute irreducible decompositions of monomial ideals in general in much less time than the best competing programs.

5. A Connection To Monomial Irreducible Decompositions

We need a few more definitions. Let I be a monomial ideal and define the function ϕ by $\phi(v) = \langle x_i^{v_i} | v_i > 0 \rangle$ for $v \in \mathbb{N}^n$ except that $\phi(1) = \langle 1 \rangle$. An ideal of the form $\phi(v)$ is called *irreducible* and the *irredundant irreducible decomposition* of I is the unique minimal subset $D \subseteq \mathbb{N}^n$ such that $I = \bigcap_{v \in D} \phi(v)$. Thus the irredundant irreducible decomposition of $I := \langle x_1^2, x_1 x_2 \rangle$ is $\{(1, 0), (2, 1)\}$ as $I = \langle x_1 \rangle \cap \langle x_1^2, x_2 \rangle$.

An ideal is *artinian* if there exists a $t \in \mathbb{N}$ such that $x_i^t \in I$ for $i = 1, \ldots, n$. Note that $\mathrm{in}_\leq(I_\mathcal{L})$ is artinian if we first project out the variable x_1, since $x_i^{p_1} - x_1^{p_i} \in I_\mathcal{L}$ and therefore $\mathrm{in}_\leq(x_i^{p_1} - x_1^{p_i}) = x_i^{p_1} \in \mathrm{in}_\leq(I_\mathcal{L})$ for $i = 2, \ldots, n$.

We claimed in Section 4 that M_p corresponds to the irreducible decomposition of the initial ideal of $I_\mathcal{L}$, and Proposition 6 proves this claim.

Proposition 6. *The set* $M' := \{(a_1+1, \ldots, a_n+1) | a \in M_p\}$ *is the irreducible irredundant decomposition* D *of* $in_\leq(I_\mathcal{L})$*.*

Proof. Let $a \in \mathbb{Z}^n$ and let $a' := a - \sum_{i=1}^n e_i$. Consider the following statements.
(1) $a \in D$
(2) $a_1 = \ \ 0$, $a_i \geq 1$, $\mathrm{in}_\leq(I_\mathcal{L}) \subseteq \phi(a)$ and $\mathrm{in}_\leq(I_\mathcal{L}) \not\subseteq \phi(a + e_i)$ for $i = 2, \ldots, n$.
(3) $a'_1 = -1$, $a'_i \geq 0$, $x^{(a')^+} \notin \mathrm{in}_\leq(I_\mathcal{L})$ and $x^{(a'+e_i)^+} \in \mathrm{in}_\leq(I_\mathcal{L})$ for $i = 2, \ldots, n$.
(4) $a' \in M_p$
We will prove that these statements are equivalent.

(1) ⇔ (2): The initial ideal of $I_\mathcal{L}$ contains no monomial that is divisible by x_1, and this implies that $a_1 = 0$ for $a \in D$. Now that we have handled the first entry, we can project out x_1 and thereby get an artinian ideal.

When working with artinian ideals the elements of the decomposition consists of vectors without zero entries, and if a has no zero entries, then $\phi(a + e_i) \subsetneq \phi(a)$.

To get the irredundant irreducible decomposition of an ideal we write the ideal as an intersection of irreducible ideals that are as small as possible, and this is exactly what (2) expresses since the projected ideal is artinian.

(2) ⇔ (3): By Lemma 7 below.

(3) ⇔ (4): By definition. □

Lemma 7. *Let* $m \in \mathbb{N}^n$ *and let* I *be a non-zero monomial ideal. Then* $x^m \in I$ *if and only if* $I \not\subseteq \phi(m + \sum_{i=1}^n e_i)$*.*

Proof. Let $a \in I$ be a monomial. Then $a | x^m$ if and only if $a \notin \phi(m + \sum_{i=1}^n e_i)$. □

6. A Connection To The Hilbert Function

The *p-weighted Hilbert function* $\mathrm{HF}_I \colon \mathbb{N} \to \mathbb{N}$ of a monomial ideal I is defined such that $\mathrm{HF}_I(t)$ is the number of monomials $x^m \notin I$ where m has p-degree t.

Proposition 8. $HF_{in_\leq(I_\mathcal{L})}(t)$ *is equal to 1 if t is p-representable and 0 otherwise.*

Proof. $\mathbf{HF_{in_\leq(I_\mathcal{L})}(t) \leq 1}$**:** Let m_1, m_2 be two vectors of p-degree t. Then $x^{m_1} - x^{m_2} \in I_\mathcal{L}$ whereby the initial term is in $\mathrm{in}_\leq(I_\mathcal{L})$. Thus at most one of x^{m_1} and x^{m_2} is not in $\mathrm{in}_\leq(I_\mathcal{L})$.

$\mathbf{HF_{in_\leq(I_\mathcal{L})}(t) = 1}$ **⇒ representability:** If $\mathrm{HF}_{\mathrm{in}_\leq(I_\mathcal{L})}(t) = 1$ then there is a monomial $x^m \notin \mathrm{in}_\leq(I_\mathcal{L})$ where m has p-degree t. Thus t is p-representable.

representability ⇒ $\mathbf{HF_{in_\leq(I_\mathcal{L})}(t) = 1}$**:** Run the algorithm from Section 3 on t. The resulting vector $c \in \mathbb{N}^n$ is such that $x^c \notin \mathrm{in}_\leq(I_\mathcal{L})$. □

We can infer that $\mathrm{HF}_{\mathrm{in}_{\leq}(I_{\mathcal{L}})}(f_p^*) = 0$ and that $\mathrm{HF}_{\mathrm{in}_{\leq}(I_{\mathcal{L}})}(t) = 1$ for all integers $t > f_p^*$. The integer at which the Hilbert function becomes equal to a polynomial is known as the *index of regularity*, so in this case the index of regularity is $f_p^* + 1$.

The elements of M_p correspond to the maximal monomials outside of $\mathrm{in}_{\leq}(I_{\mathcal{L}})$ according to divisibility (disregarding the first variable), and what the algorithm from Section 4 does is to maximize the dot product with p over M_p. This amounts to maximizing the dot product over the vectors m such that $x^{m+e_1} \notin \mathrm{in}_{\leq}(I_{\mathcal{L}})$ and $m_1 = -1$.

We can multiply any monomial not in $\mathrm{in}_{\leq}(I_{\mathcal{L}})$ with x_1 and get something still not in $\mathrm{in}_{\leq}(I_{\mathcal{L}})$, so anything that goes on in the first variable does not prevent the Hilbert function from becoming a polynomial. Thus the algorithm can be interpreted as finding the maximal point that prevents the Hilbert function from becoming a polynomial, which is to say that it computes the index of regularity.

We conclude that the algorithm of Einstein et al. (2007) can be interpreted as computing an index of regularity.

7. Benchmarks

Roune (2006) has written an implementation called Frobby of the algorithm described in this paper, and here we compare Frobby to the implementation of Einstein et al. (2007). Figure 1 displays the collected data. "Intractable" means intractable according to the authors of that software package. Note that the most time consuming part of the algorithm usually is to compute the Gröbner basis. Frobby and Mathematica are the only programs that can handle input numbers p_i as large as those in figure 1.

Frobby uses the program fplll due to Stehlé (2006) to obtain an LLL-reduced lattice basis and then computes the Gröbner basis of the lattice ideal $I_{\mathcal{L}}$ from that using the program 4ti2 [4ti2 team (2006)]. Frobby then computes the Frobenius number by computing an irreducible decomposition of $\mathrm{in}_{\leq}(I_{\mathcal{L}})$ using the algorithm due to Roune (2007), which is similar to the External Corner Algorithm. Frobby is written in C++ and the source code is available under the GNU General Public License (GPL).

The implementation of Einstein et al. (2007) is available as a part of Mathematica and is written mostly in C. The major difference from Frobby is that the Gröbner basis is computed using a different algorithm than Buchberger's.

All the inputs were randomly generated using genuinely random radioactive decay via the service provided by Walker (2006) except the $n = 11$ input which was provided by Stan Wagon who pseudo-randomly generated it using Mathematica. We wish to thank Daniel Lichtblau for carrying out the Mathematica benchmarks.

All the benchmarks were run on machines with a 3.0 GHz Pentium 4 CPU with 1 GB RAM except the Mathematica benchmark on the $n = 11$ input which was run on a 3.2 GHz Pentium 4.

References

4ti2 team, 2006. 4ti2 version 1.3 – a software package for algebraic, geometric and combinatorial problems on linear spaces. Available at `http://www.4ti2.de`.

Conti, P., Traverso, C., 1991. Buchberger algorithm and integer programming. In: AAECC-9: Proceedings of the 9th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. Springer-Verlag, London, UK, pp. 130–139.

n	$\lfloor \log_{10}(\min p_i) \rfloor + 1$	$\|G\|$	Mathematica	Frobby
4	10000	7	620s	3.5s
4	800	10	4.0s	0.3s
5	150	34	3.8s	0.3s
6	70	131	5.5s	0.3s
6	80	148	5.6s	0.3s
6	90	112	6.8s	0.3s
6	100	140	9.5s	0.3s
8	30	2099	80.2s	11.8s
11	11	27037	43.3h	0.5h
12	11	56693	intractable	2.5h
13	11	170835	intractable	49.7h

Fig. 1. The benchmark data.

Cox, D., Little, J., O'Shea, D., 1997. Ideals, Varieties, and Algorithms. Springer.

Einstein, D., Lichtblau, D., Strzebonski, A., Wagon, S., 2007. Frobenius numbers by lattice point enumeration. Integers 7, available at `http://www.integers-ejcnt.org/`.

Jensen, A. N., Lauritzen, N., Roune, B. H., 2007. Maximal lattice free bodies, Frobenius numbers and test sets. Eprint arXiv:0705.4439.

Malkin, P. N., 2006. Truncated Markov bases and Gröbner bases for integer programming. Eprint arXiv:math/0612615v1.

Pottier, L., 1994. Gröbner bases of toric ideals. INRIA Rapport de recherche 2224, available at `http://hal.inria.fr/inria-00074446/en/`.

Ramírez Alfonsín, J. L., 2005. The Diophantine Frobenius Problem. Oxford University Press.

Roune, B. H., 2006. Frobby – a software package for computing Frobenius numbers and irreducible decompositions of monomial ideals. Available at `http://www.broune.com/frobby/`.

Roune, B. H., 2007. The label algorithm for irreducible decomposition of monomial ideals. Eprint arXiv:0705.4483.

Stehlé, D., 2006. fplll version 1.3 – an lll-reduction program. Available at `http://perso.ens-lyon.fr/damien.stehle/english.html`.

Walker, J., 2006. Hotbits – genuine random numbers, generated by radioactive decay. Available at `http://www.fourmilab.ch/hotbits`.