Frobenius Numbers by Toric Grö

Daniel Lichtblau

Wolfram Research, Inc.
100 Trade Centre Dr.
Champaign IL USA, 61820

danl@wolfram.com

ACA 2005, Nara, Japan
August 2005

Abstract

Given a set $A = \{a_1, \ldots, a_n\}$ of positive integers with gc
"sufficiently large" integers can be represented as a non
elements of A. The Frobenius number of the set is defin
representable. The Frobenius instance problem (also cal
stamp problem) is to determine, given a positive integer
$X = \{x_1, \ldots, x_n\}$ such that $X.A = n$, or else show no such
how this can be addressed via toric Gröbner bases.

It is known that the Frobenius number problem is NP−h
is trivial (Sylvester solved in two decades before Frober
dimension 3 a very efficient method was found indepen
For higher dimensions some quite effective methods are
element of A is not too large (say, less than 10^7).

Recent work has given rise to methods that are effective
hold, although the dimension must be bounded by 10 or
to recast this work using toric Gröbner bases, wherein th
set A is given by the staircase of the basis with respect t
reasonably efficient in dimensions 4 to 7, when the elem
or so. We will illustrate this.

Introduction: Background and brief history

We are given a set $A = \{a_1, \ldots, a_n\}$ of positive integers
later purposes that the set is in ascending order.

Problem 1 (Frobenius instance problem): Given a nonne
nonnegative integers $X = \{x_1, \ldots, x_n\}$ such that $X.A = n$

Problem 2 (Frobenius number problem): Find the larges
nonnegative integer combination of A.

In the 80's and 90's Greenberg and Davison independe
problem 2 when $n = 3$. Beyond this size no specialized
methods are known, and we must resort to general tacti

Reasonably effective methods based mostly on graph th
30 years or so. Some very nice new ones are presented,
very recent work by Beihoffer, Hendry, Nijenhuis, and
third author helped originate the graph theory approach.
size of a_1, but not by n.

Introduction: Background and brief history

This restriction apparently rankled the fourth author, wh
using different tactics. Forthcoming joint work by Davi
Wagon, and myself will show how one can attack this p
methods and integer programming. While we can do aw
we do get into some algorithmic complexity due to dim
beyond $n = 11$ or so. Some of the technology we use is
where we can manage higher dimension (25 or larger).

It so happens that much of this can be recast in a setting
1, which boils to integer linear programming, has long b
an approach (as per work by Conti and Traverso), this is
finding Frobenius numbers. We can exploit it to handle
knowledge, could not be done by methods known as of
soon see is that the needed code is quite short (three pag

We will define a "fundamental domain" which is a gene
both lattice diagrams in earlier literature and a graph de
Frobenius number will be the furthest corner from the o
norm. As we will see, an important domain feature is wl
to find those via a Gröbner basis "staircase" constitutes

Solving a Frobenius instance via toric Gröb

Say we are given the set and value

```
A = {200, 230, 528, 863, 905, 1355, 17
b = 7777;
```

We wish to know whether or how we can write b as a n
elements of A. We may do this as follows. Create a vari
powers in this set. Create a variable for each set elemen
Gröbner basis in these variables, using an order that ma
monomials that do not contain it. For this we use a weig
way as to be efficient for the task at hand. Basically it is
with degree−reverse−lexicographic on the remaining va
(homogenious) total degree we weight by the values in

Solving a Frobenius instance via toric Gröb

```
len = Length[A];
vars = Array[x, len];
polys = vars - t^A;
wtmat = RotateRight[Reverse[-Identi
wtmat[[2]] *= -1;
wtmat[[1]] = Prepend[A, 0];
wtmat[[{1, 2}]] = wtmat[[{2, 1}]];
wtmat
```

```
{{1, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 200, 230, 528, 863, 905, 1355, 1
 {0, 0, 0, 0, 0, 0, 0, 0, 0, -1},
 {0, 0, 0, 0, 0, 0, 0, 0, -1, 0},
 {0, 0, 0, 0, 0, 0, 0, -1, 0, 0},
 {0, 0, 0, 0, 0, 0, -1, 0, 0, 0},
 {0, 0, 0, 0, 0, -1, 0, 0, 0, 0},
 {0, 0, 0, 0, -1, 0, 0, 0, 0, 0},
 {0, 0, 0, -1, 0, 0, 0, 0, 0, 0},
 {0, 0, -1, 0, 0, 0, 0, 0, 0, 0}}
```

Solving a Frobenius instance via toric Gröb

```
Timing[
 gb = GroebnerBasis[polys, Prepend[v
    MonomialOrder → wtmat];]
```

```
{9.8775 Second, Null}
```

```
Length[gb]
```

```
264
```

To check whether we can represent b as a nonnegative i
reduce (using the same term ordering) t^b by this Gröbne

```
red = PolynomialReduce[t^7777, gb, P
   MonomialOrder → wtmat][[2]]
```

$x[2]^4 \, x[3] \, x[5] \, x[9]^3$

Solving a Frobenius instance via toric Gröb

Now replace variables by their values and get correspon

```
fax = Drop[FactorList[red], 1]
exponvec = fax /. x[j_] :→ A[[j]]
```

```
{{x[2], 4}, {x[3], 1}, {x[5], 1}, {x[
```

```
{{230, 4}, {528, 1}, {905, 1}, {1808,
```

We check this. We want to see that $4*230+1*528+1$

```
Total[Apply[Times, exponvec, 2]]
```

```
7777
```

Remark: The above illustrates more or less the original
toric Gröbner bases. Subsequent improvements have ap
done much more efficiently now. This concludes our br
a Frobenius instance problem using the method of Cont
main task at hand, which is computation of Frobenius n

The Fundamental Domain

We define a lattice by the set of integer combinations of
a_1. This is a full dimensional lattice in $\mathbb{Z}^{n-1}$. The set of
lattice gives rise to what we call the fundamental domai
space of dimension one less than the size of our set. It is
distinct residue classes, so we know the cardinality of th
"weight" of a vector $v \in \mathbb{Z}^{n-1}$ as $v.\{a_2, \ldots, a_n\}$. It can be
at least one element with all nonnegative entries. Among
weight. In case of tie, choose the one that is lexicograph
the set of residues that we take to comprise the fundame

The Fundamental Domain

This domain can be shown to have several interesting pr

- It is a staircase. If it contains a lattice element then it c with any coordinate strictly smaller.
- It tiles $\mathbb{Z}^{n-1}$.
- It is a cyclic group $\mathbb{Z} / a_1 \mathbb{Z}$.
- It can be given a circulant graph structure. It is this stru shortest−path graph methods. Old and new methods for recent work by Beihoffer et al.

For our purposes the property of most interest is the firs
by computing a toric Gröbner basis.

Definitions related to the Fundamental Dom

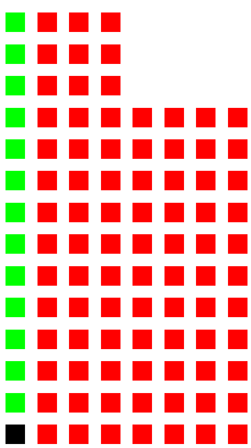
From the staircase property the fundamental domain ha
elbows. It has extremal points called corners. Specifical
domain, such that $c + e_j$ is not in the domain, where e_j i
elbow is a point x that is not in the domain, but is such t
$x - e_j$ is in the domain.

There are two other definitions that play a role in the alg
too carefully but, roughly, there are as follows.
(i) Protoelbows. These have both positive and negative
certain "minimal" equivalences (that is, reducing relatio
terms, these are given as exponent vectors of binomial p
(ii) Preelbows. These are the "positive parts" of the prot
elements in the partially ordered (ascending by inclusion

Fundamental Domain, illustrated

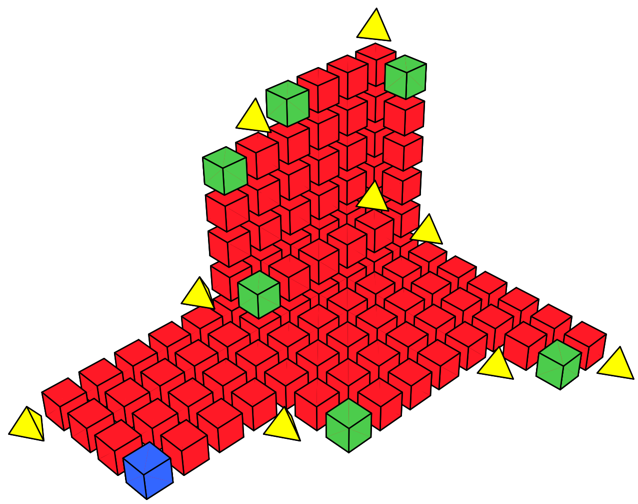
Since this region is one dimension smaller than the inpu
a_1), it can be illustrated for the cases $n = 3$ and $n = 4$. T
Wagon.

With respect to these domains, the Frobenius number co
from the origin, with distance an l_1 metric weighted by
diagram the "elbows" are the lattice points on the axes t
lattice point in the interior just outside the "ell". The cor
reached by intersecting vertical and horizontal lines thro
the entire story as regards the $n = 3$ case, because it can
interior elbow and two such corners, and finding them i



Fundamental Domain, illustrated

In the three dimensional diagram the elbows are again tl
as well as bounding points where the staircase goes up i
interior. They are demarcated by yellow tetrahedra. The
blue box is the maximal corner.



The Algorithm, in brief

The gist of our efficient algorithm is to use integer linea
containing elbows, then use a method David and Stan d
corners. We finish when we have the furthest corner. W
elbows) arise even at $n = 4$, the average case performan
nice. Some of the ILP ideas appear in work by Aardal, H
subsequent refinement by Aardal and Lenstra. I also had
AHL which I used to find large examples of what are kn

Getting back to Frobenius numbers, it turns out that one
find what are called "protoelbows". These are lattice po
one subset equal positive combinations of another. From
elbows, and we use a domination algorithm of Bentley,
"kernel" set which comprise the actual elbows. The hard
coming up with the protoelbows. This can be done with

As with instance solving, the idea is again to set up relat
polynomials for a toric ideal) of the form $x_j - t^{a_j}$, and el
at all...

Finding "elbow" relations

...For better efficiency we now use an improvement, du
computation of the elimination ideal. Instead of $x_j - t^{a_j}$
${x_j}^{{e_j}^+} - {x_j}^{{e_j}^-}$ where ${e_j}^+$ and ${e_j}^-$ are the positive and nega
generator for the null space of A. We can use any basis
that is lattice reduced so as to keep down the exponent s
$u - \prod x_j$ that in effect allows us to invert negative expo
that eliminates the new variable u.

From this basis we next want to find a new one that wil
provision of the definition. As our lattice is now represe
ideal, this amounts to an inverse lexicographical term o
well−founded ordering for monomials because it has co
products (it is an ordering appropriate for a local ring).
homogenizing, making the homogenizer variable larges
term order. As we want an inverse lexicographic order
reverse−lexicographic. So we now compute a new Gröb
and then dehomogenize. Note that this second basis con
the first, hence not problematic in regard to efficiency.

Finding "elbow" relations

Since it is the exponent vectors we are after, we extract
lattice elements by subtracting second term powers fron
but it is just the usual translation from toric ideal to latti
are only interested in equalities modulo a_1, we strip off
To satisfy a technical consideration for protoelbows (se
negate the lattice vector. Taking positive parts of the res
preelbows.

It is important to note that we are using the Gröbner bas
the instance solving usage, we do not work geometricall

Code to find elbows

```
preElbows[vals_] :=
 Module[{n = Length[vals], vars, x, t, nspace, pos, neg, p
  nspace = integerNullSpace[vals];
  vars = Array[x, n];
  pos = (nspace + Abs[nspace]) / 2;
  neg = -nspace + pos;
  polys = Map[Inner[Power, vars, #, Times] &, pos] - Map[I
  polys = Join[polys, {Apply[Times, vars] * y} - 1];
  polys = GroebnerBasis`ToricGroebnerBasis[polys, vars,
  polys = homogenize[polys, vars, h];
  polys = GroebnerBasis[polys, Reverse[Append[vars, h]],
     MonomialOrder → DegreeReverseLexicographic] /. h → 1
  exponvecs = First[GroebnerBasis`DistributedTermsList[
  exponvecs = Map[First[-#[[1]] + #[[2]]] &, exponvecs];
  exponvecs = Map[orient[#, vals] &, exponvecs];
  exponvecs = Map[Rest, exponvecs]; (*protoelbows*)Union
```

```
elbows[vals_] := Sort[dominationKernel[preElbows[vals]]
```

More code: utility functions

```
integerNullSpace[vec : {_Integer ..}] := Module[{mat, hnf}, mat = Transpose[Join[{vec}, I
  hnf = Last[Developer`HermiteNormalForm[mat]];
  LatticeReduce[Map[Drop[#, 1] &, Drop[hnf, 1]]]]
```

```
homogenize[poly_, vars_, new_] /; (Head[poly] =!= Plus && Head[poly] =!= List) := poly
homogenize[polys_List, vars_, v_] := Map[homogenize[#, vars, v] &, polys]
homogenize[poly_, vars_, new_] := Module[{degfunc, totdeg, j}, degfunc = Apply[Plus, Map[
  degfunc = Distribute[degfunc, Function, Plus];
  totdeg = Max[Map[degfunc, Apply[List, poly]]];
  Apply[Plus, Table[poly[[j]] * new^ (totdeg - degfunc[poly[[j]]]), {j, Length[poly]}]]]
```

```
orient[vec_, basevec_] := Module[{val = Rest[vec].Rest[basevec], j = 1},
  Which[val > 0, vec, val < 0, -vec, True, While[vec[[j]] === 0, j++]; -Sign[vec[[j]]] * ve
```

```
dominationKernel[X_] :=
 Module[{Y = X[[Ordering[Map[Tr, X]]]], Z, i = 2, len = 1, k}, Z = Table[{}, {Length[Y]}];
  Z[[1]] = Y[[1]];
  While[i ≤ Length[Y] && Tr[Y[[i]]] == Tr[Y[[1]]], len++;
   Z[[len]] = Y[[i]]; i++];
  Do[k = 1;
   While[k ≤ len,
    If[And @@ Thread[Z[[k]] ≤ Y[[j]]], k = len + 2; Break[], k++]];
   If[k == len + 1, len++; Z[[len]] = Y[[j]]], {j, len + 1, Length[Y]}];
  Sort[Take[Z, len]]]
```

```
ClearNegsAndDeleteZeroVector[vecs_] := If[vecs == {}, {}, Union[DeleteCases[vecs /. _?Ne
```

Rest of code: corners

The method below for using elbows to find the "Froben
Einstein, with refinements and code provided by Stan W

```
farthest[corns_, A_] := Fold[If[#2.Rest[A] > #1[[2]], {#2, #2.Rest[A]}, #1] &, {Table[0,

Options[FarthestCorner] = {TopLevelTraceQ → False, TraceStep → 1};

FarthestCorner[A_, elbows_?MatrixQ, opts___Rule] := Module[{pts, p1, p2, B, cc, trQ, far
    {tlQ, ts} = {TopLevelTraceQ, TraceStep} /. {opts} /. Options[FarthestCorner];
    skipct = counter = 0;
    currfar = farthest[elbows, A].Rest[A];
    pts = Select[elbows, #[[1]] > 0 &];
    Scan[(counter++;
       {p1, p2} = {First[#], Rest[#]};
       B = Rest /@ Select[elbows, #[[1]] < p1 &];
       B = dominationKernel[ClearNegsAndDeleteZeroVector[# - p2 & /@ B]];
       maxes = Prepend[Max /@ Transpose[B], 0];
       If[tlQ && Mod[counter, ts] == 0, Print[{counter, Length[pts], "Length of sent set
       If[(maxes + #).A1 > currfar, cc = FarthestCornerSub[A, B, currfar, #, opts];
        farvertex = farthest[Prepend[# + p2, p1] & /@ cc, A];
        currfar = Max[currfar, farvertex.A1], skipct++;
        If[tlQ && Mod[counter, ts] == 0, Print[{counter, Length[pts], "got a cutoff, tota
    currfar - Total[A]] /; Length[elbows[[1]]] > 2;

FarthestCorner[A_, elbows_?MatrixQ, opts___Rule] := farthest[FarthestCornerSub[A, elbo

FarthestCornerSub[A_, elbows_, currfar_, backdata_, opts___] :=
  Module[{p1, p2, B, maxes},
    Flatten[
     Map[
      ({p1, p2} = {#[[1]], Rest[#]};
        B = Rest /@ Select[elbows, First[#] < p1 &];
        B = dominationKernel[ClearNegsAndDeleteZeroVector[# - p2 & /@ B]];
        If[B == {}, {}, maxes = Prepend[Max /@ Transpose[B], 0];
         If[(PadLeft[maxes + #, Length[A] - 1] + backdata).Rest[A] > currfar,
          (Prepend[# + p2, p1] & /@ FarthestCornerSub[A, B, currfar, PadLeft[#, Length[A]
      Select[elbows, First[#] > 0 &]], 1]] /; Length[elbows[[1]]] > 2;

FarthestCornerSub[_, elbows_, ___] := (Partition[Take[Flatten[Reverse /@ Reverse[elbows
```

Examples

We show several examples to get some idea of time nee

```
vals1 = {200, 230, 528, 863, 905, 135
vals2 = {13557, 20002, 52831, 86312,
vals3 = {18543816, 27129592, 4322664
vals4 = {11615, 27638, 32124, 48384,
vals5 = {10^10, 18543816066, 2712959
   78522678316};
vals6 = {10000000000, 35550333799, 4
   67932625953, 75136205898, 790225
vals7 = {10000000000, 35550333799, 4
   67932625953, 75136205898, 790225
```

Examples, continued

```
Timing[elbows1 = elbows[vals1]]
Timing[f1 = FarthestCorner[vals1, el
```

```
{4.89226 Second, {{0, 0, 0, 0, 0, 2}, {0, 0, 0, 0, 2, 1}, {0
  {0, 0, 0, 1, 1, 0}, {0, 0, 0, 2, 0, 0}, {0, 0, 1, 0, 0, 1},
  {0, 0, 5, 0, 0, 0}, {0, 1, 0, 0, 1, 1}, {0, 1, 4, 0, 0, 0},
  {0, 3, 2, 0, 0, 0}, {0, 4, 1, 0, 0, 0}, {0, 5, 0, 0, 0, 0},
  {2, 0, 4, 0, 0, 0}, {3, 0, 0, 0, 2, 0}, {3, 0, 3, 1, 0, 0},
  {5, 0, 0, 0, 1, 0}, {5, 0, 2, 1, 0, 0}, {8, 1, 0, 0, 0, 0},
  {11, 0, 2, 0, 0, 0}, {13, 0, 1, 0, 0, 0}, {14, 0, 0, 1, 0,
```

```
{0.053992 Second, 4192}
```

```
Timing[elbows2 = elbows[vals2];
 f2 = FarthestCorner[vals2, elbows2]
```

```
{0.314952 Second, 2185053}
```

```
Timing[elbows3 = elbows[vals3];
 f3 = FarthestCorner[vals3, elbows3]
```

```
{0.030995 Second, 33335274131}
```

```
Timing[elbows4 = elbows[vals4];
 f4 = FarthestCorner[vals4, elbows4]
```

```
{2.50562 Second, 861905}
```

Examples, continued

```
Timing[elbows5 = elbows[vals5]]
Timing[f5 = FarthestCorner[vals5, el
```

```
{0.100984 Second,
 {{0, 0, 0, 210}, {0, 0, 162, 153}, {0
  {0, 234, 111, 89}, {0, 244, 307, 0}
  {0, 518, 0, 0}, {165, 174, 80, 26},
  {264, 448, 0, 83}, {358, 0, 0, 147}
  {454, 0, 327, 0}, {459, 458, 0, 0},
```

```
{0.022997 Second, 38563214973583}
```

```
Timing[elbows6 = elbows[vals6];]
Timing[f6 = FarthestCorner[vals6, el
```

```
{26.7119 Second, Null}
```

```
{0.619906 Second, 2202463617 9389}
```

```
Timing[elbows7 = elbows[vals7];]
Timing[f7 = FarthestCorner[vals7, el
```

{801.31 Second, Null}

{10.59 Second, 10155222194133}

To the best of my knowledge, the Frobenius numbers in
done with methods available as of one year ago.

Summary

We have seen how toric Gröbner bases can be used to fi
Domain for the Frobenius number problem. Unlike mos
involving finite varieties, toric and otherwise, what is pa
staircase for the "normal set" with respect to a certain m
Gröbner bases can be used at all to find Frobenius numb
to gain staircase information, both seem to be of interest

It is also of interest that this approach can handle Frober
not amenable to ANY method know as recently as a yea

The more efficient methods in our work in preparation h
programming. Much of this has been incorporated into t
functions `Reduce`, `FindInstance`, and `Minimize`.

Open questions:
(1) Would dedicated toric basis code do better? Probabl
(2) Are there better ways to do the term ordering so that
information about the fundamental domain, but faster?

References

K. Aardal, C. A. J. Hurkens, and A. K. Lenstra. Solving a system c
and upper bounds on the variables. Mathematics of Operations Res

K. Aardal and A. K. Lenstra. Hard equality constrained knapsacks.
Integer Programming and Combinatorial Optimization (IPCO 2002
Lecture Notes in Computer Science 233, 350−366. Springer−Verla

D. Beihoffer, J. Hendry, A. Nijenhuis, and S. Wagon. Faster algorit
Electronic Journal of Combinatorics 12. 2005.

J. L. Bentley, K. L. Clarkson, and D. B. Levine. Fast linear .expecte
and convex hulls. Algorithmica 9(2): 168−183. 1993.

P. Conti and C. Traverso. Gröbner bases and integer programming.
Springer−Verlag LNCS 539 130−139. 1991.

J. L. Davison. On the linear Diophantine problem of Frobenius. J. N

D. Einstein, D. Lichtblau, A. Strzebonski, and S. Wagon. Frobenius
preparation. 2005.

H. Greenberg. Solution to a linear Diophantine equation for nonneg
1988.

M. Keith. (Web page discussing Keith numbers)
http://users.aol.com/s6sj7gt/mikekeit.htm

D. Lichtblau. Solving knapsack and related problems. Manuscript.

L. Pottier. Gröbner bases of toric ideals. INRIA Rapport de recherc