> I felt a bit like a dentist. I had dealt with this problem before, so why were they asking the same question again?
>
> H. W. Lenstra

1 Integer Linear Programming: An Introduction

In this class and following ones, we will be coming up with an FPT algorithm for Integer Linear Programming, with the parameter being the number of dimensions.

As a quick reminder, the ILP problem denotes checking the feasibility of $K = \left\{\mathbf{x} : \begin{smallmatrix}\mathbf{Ax} \leq \mathbf{b} \\ \mathbf{x} \in \mathbb{Z}^d\end{smallmatrix}\right\}$. In other words, this is a decision problem asking if $K \neq \varnothing$, and can be extended to finding the objective via binary search.

For completeness, $\mathbf{A} \in \mathbb{Z}^{m \times d}$ and $\mathbf{b} \in \mathbb{Z}^m$ where m is the number of rows and d is the number of dimensions. Finally, for simplicity we assume that K is bounded within a region of length M.

For FPT purposes, we are interested in an algorithm with runtime $f(d) \cdot \text{poly}(m, \log ||\mathbf{A}||_\infty, \log ||\mathbf{b}||_\infty)$. We note that the latter two elements are the largest entries in $\mathbf{A}$ and $\mathbf{b}$.

1.1 Early History

In 1980, van Emde Boas and Marchetti-Spacamela asked H. W. Lenstra about the feasibility of a triangular K in 2 dimensions. Lenstra replied that this was done nearly 200 years back by Lagrange and Gauss, known as Lagrange-Gauss Basis Reduction. [Sme10]

1.2 Algorithm for triangular K

At a high level, the full algorithm is as follows. We use $|\cdot|$ to denote the Euclidean norm.

1. Apply a linear transformation $\mathbf{M}$ to the integer lattice so that K is transformed into a unit-length equilateral triangle. Let (b_1, b_2) denote the transformed standard basis vectors $(\vec{e}_1, \vec{e}_2)$.
2. Apply Lagrange-Gauss Basis Reduction to form new basis vectors (b'_1, b'_2).
3. Case on $|b'_2|$.
   - If $|b'_2| < \frac{1}{10}$, then we are guaranteed an integer point.
   - Else, at most a constant number of lines (in particular, 10) parallel to b'_1 can intersect with the triangle. Check each.

Notice that an integer point (x, y) is mapped to $x \cdot b'_1 + y \cdot b'_2$. We examine steps 2 and 3 in more detail.

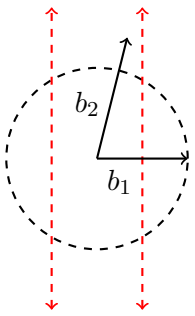
1.2.1 Lagrange-Gauss Basis Reduction

We assume that $|b_1| \leq |b_2|$.

Algorithm 1 Basis Reduction

```
procedure REDUCE(b_1, b_2)
    μ ← ⟨b_1, b_2⟩ / |b_1|^2
    if |μ| > 1/2 then
        b_2 ← b_2 − [μ]b_1
        if |b_2| < |b_1| then return REDUCE(b_1, b_2)
        end if
    end if
    return (b_1, b_2)
end procedure
```

The algorithm uses $[\mu]$, where $[\cdot]$ denotes the nearest integer function. We claim that updating b_2 forces $|\mu| \leq \frac{1}{2}$. To show this, note that for any ℓ, $\frac{\langle b_1, b_2 - \ell b_1 \rangle}{|b_1|^2} = \mu - \ell$. So, choosing $\ell = [\mu]$ suffices as the nearest integer is at most $\frac{1}{2}$ away from μ. Pictorially, our returned basis looks as follows:



The angle between b_1 and b_2 is at least $60°$, where we represent that boundary with the dashed red lines.

1.2.2 Finishing it off

If $|b_2'| < \frac{1}{10}$ (the first case), recall that the inradius[1] of a unit equilateral triangle is $\frac{\sqrt{3}}{6} > \frac{2}{10}$. Consider the incenter of the triangle: $p = \ell_1 b_1 + \ell_2 b_2$, and let $p^* = [\ell_1]b_1 + [\ell_2]b_2$. Because $\frac{1}{10} > |b_2| \geq |b_1|$, $|p^* - p| \leq \sqrt{(\frac{1}{2} \cdot \frac{1}{10})^2 + (\frac{1}{2} \cdot \frac{1}{10})^2} < \frac{2}{10}$, so p^* is an integer point in the unit equilateral triangle, as desired.

Otherwise, suppose that $|b_2| \geq \frac{1}{10}$. Then consider the lines parallel to b_1 and offset by b_2 (i.e., $t \cdot b_1 + k \cdot b_2$, parametrized by t). The distance between any two such lines is at least $\sin\theta \cdot |b_2| \geq \frac{\sqrt{3}}{20}$, where $\theta \geq 60°$ is the angle between b_1 and b_2. Hence we only need to check a constant number of lines for having points within the equilateral triangle.

[1] Recall that the inradius and incenter of a triangle are defined as follows: take the largest circle completely inside the given triangle (which is tangent to all three sides of the triangle). The inradius of the triangle is the radius of this circle, and the incenter is the center of this circle.

1.3 General algorithm in 2-D

The general algorithm is very similar to the one for triangular K.

1. Convert K to some "nice" K'.
2. Use L-G Basis Reduction to get b_1, b_2 with the same properties as above.
3. We case on $|b_2|$.
   - If $|b_2|$ is sufficiently small, there must exist an integer point within K.
   - Else, there are only a small number of 1-D problems to solve.

For the general, d-dimensional case, basis reduction is done differently and uses the LLL [LLL82] algorithm. Doing similar recursive steps (substitute 1-D with $(d-1)$-D) of depth d gives an FPT algorithm.[2]

2 Lattices

Remark 11.1. We specifically consider point lattices.

We begin with some useful definitions about lattices.

Definition 11.2. Given linearly independent vectors $b_1, b_2, \ldots, b_d \in \mathbb{R}^d$, let

$$B = \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_d \\ | & | & & | \end{bmatrix}$$

We define $\Lambda(B) = \left\{ \sum_{i=1}^{d} \lambda_i b_i : \lambda_i \in \mathbb{Z} \right\}$ as the *lattice* of B.

In general, there are an infinite number of bases for a lattice. The following operations are permitted.

- Negate b_i.
- Replace $b_i \leftarrow b_i + \ell b_j$ where $j \neq i$ and $\ell \in \mathbb{Z}$.
- Permute the order of the b_i.

In line with having a set of operations on a set of vectors, we define a linear transformation with some desirable properties.

Definition 11.3. U is a *unimodular* matrix if for all entries $U_{i,j} \in \mathbb{Z}$ and $|\det(U)| = 1$.

As it turns out, unimodular matrices have exactly the same properties as the operations we permit.

Lemma 11.4. *B_1 and B_2 are bases of the same lattice Λ if and only if there exists a unimodular matrix U such that $B_1 = B_2 U$.*

[2] XP does not seem to be any easier!

Finally, we define a notion of *volume* for general lattices.

Definition 11.5. Define $P(B) = \left\{ \sum_{i=1}^{d} x_i b_i : 0 \le x_i \le 1 \right\}$ as the *fundamental domain/parallelipiped* of B.

Furthermore, we define for a lattice over B the volume via

$$\text{vol } \Lambda \triangleq \det(\Lambda) \triangleq \text{vol } P(B) = |\det(B)|$$

Finally, the *density* of a lattice is $\text{density}(\Lambda) = \frac{1}{\text{vol } \Lambda}$.

Remark 11.6. Note that volume is invariant under choice of basis for the lattice, as if $B_1 = B_2 U$ we have

$$|\det B_1| = |\det B_2 U| = |\det B_2||\det U| = |\det B_2|$$

It seems that these definitions are a bit arbitrary, but as it turns out there is a natural hard problem often used in cryptography relating to them.

Definition 11.7. The *Shortest Vector Problem* (SVP) asks for the shortest nonzero vector in a lattice.

Remark 11.8. LLL gives a pretty bad approximation of $2^{\frac{d}{2}}$. The best known is $(1+\varepsilon)^d$.

Given all of these definitions, we can finally state a result which will be important in our overall discussion of FPT algorithms for ILP.

Minkowski. *Given a d-dimensional lattice Λ, the following are true.*

(a) Suppose K is a bounded, symmetric[3]*, and convex body with $\text{vol}(K) > 2^d$. Then $K \cap (\mathbb{Z}^d \backslash \mathbf{0}) \neq \varnothing$. In other words, there is a nonzero vector in K.*

(b) There exists a nonzero vector in the lattice of length $\sqrt{d} \cdot \text{vol}(\Lambda)^{\frac{1}{d}}$

We present some helpful notation before beginning on the proof.

Notation 11.9. Let $cK = \{cx \mid x \in K\}$ and $||x||_K = \inf\{c \in \mathbb{R} \mid x \in cK\}$. Observe that the latter is a valid norm (in particular, it satisfies the triangle inequality because K is convex).

Proof. We prove (a) first. Consider the following diagram, which shows K and some translated versions of $\frac{1}{2}K$ by integer vectors $v \in \mathbb{Z}^d$.

[3] $x \in K \iff -x \in K$.

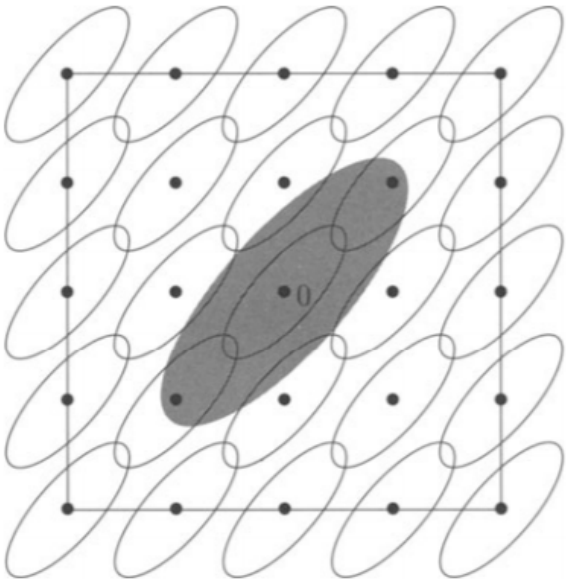


Figure 11.1: Image from [Mat02]

Let $C = \frac{1}{2}K$. We claim that $C + x$ and $C + y$ intersect for some integer vectors x and y. Indeed, suppose not and consider a hypercube of radius $S + R(C)$ centered at some integer point, where $R(C) = \frac{1}{2}D(C)$ is the radius of C (and $D(C)$ is the diameter). This hypercube has volume $(2S + D(C))^d$, and contains at least $(2S + 1)^d$ translated copies of C. So, we have

$$(2S + D(C))^d \geq (2S + 1)^d \cdot \text{vol}(C) \implies \text{vol}(C)^{\frac{1}{d}} \leq \frac{2S + D(C)}{2S + 1}$$

Noting that the right hand side can be made arbitrarily small, this is a contradiction as $2^d \text{vol}(C) = \text{vol}(K) > 2^d$.

So, $C + x$ and $C + y$ intersect at some point w. Note that by the triangle inequality, $||x - y||_K \leq ||x - w||_K + ||w - y||_K \leq 1$ because each of the summands are at most $\frac{1}{2}$. So, $x - y$ must be an integer vector in K.

We now prove (b), using (a). To do so, we prove that for any (bounded, symmetric, convex) body K with $\text{vol}(K) > 2^d \cdot \text{vol}(\Lambda)$, it must be true that $K \cap (\Lambda \backslash \mathbf{0}) \neq \varnothing$. Indeed, we transform $\Lambda \mapsto \mathbb{Z}^d$ and $K \mapsto K'$, and note that K's volume is scaled by $\text{vol}(\Lambda)$, as $\text{vol}(\mathbb{Z}^d) = 1$. So, as $\text{vol}(K') > 2^d$ we apply part (a) to obtain a nonzero integer vector in K' and hence K.

Choose K to be the hypercube of radius $\text{vol}(\Lambda)^{\frac{1}{d}} + \varepsilon$ for $\varepsilon > 0$, which must contain an integer vector of length at most $\sqrt{d} \cdot (\text{vol}(\Lambda)^{\frac{1}{d}} + \varepsilon)$. Choosing ε small enough guarantees a vector of length at most $\sqrt{d} \cdot \text{vol}(\Lambda)^{\frac{1}{d}}$, as desired.[4] □

2.1 An application of Minkowski's Theorem

Suppose we are given d irrational numbers $\alpha_1, \alpha_2, \ldots, \alpha_d$ and we wish to approximate them with fractions $\frac{p_1}{q}, \frac{p_2}{q}, \ldots, \frac{p_d}{q}$ for some bounded q. How well can we do this? More next time.

[4] Recall that the length of an integer vector must be of the form $k^{\frac{1}{d}}$, so choose ε small enough that $d^{\frac{d}{2}}(\text{vol}(\Lambda)^{\frac{1}{d}} + \varepsilon)^d < d^{\frac{d}{2}} \text{vol}(\Lambda) + 1$.

References

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, (261):515–534, 1982. 1.3

[Mat02] Jiří Matoušek. *Lectures on Discrete Geometry*. Springer, 2002. 11.1

[Sme10] Ionica Smeets. The History of the LLL-Algorithm. In Phong Q. Nguyen and Brigitte Vallée, editors, *The LLL Algorithm: Survey and Applications*, chapter 1, pages 1–17. Springer, 2010. 1.1