MATHEMATICS

Shortest Non-Zero Vector in Integer Lattices with Given Points

Asked 8 years, 6 months ago Modified 3 years, 1 month ago Viewed 2k times

10

5

There are two questions related to the shortest non zero vector problem that have left me scratching my head. Please bear with me as I describe the problem. **Disclaimer: this is homework.**

For the first question there is a lattice that is generated by the vectors (1, 2) and (3, 1). Vectors (5, 5) and (14, 13) generate the same lattice.

The question asks us to calculate the distance from origin for all these points, which I do using $\sqrt{x^2+y^2}$ (Euclidean norm).

Then the question asks us to state the shortest non zero vector for this lattice. From the calculations above, the shortest non zero vector for this lattice is (1, 2).

For the second question, there is a lattice generated by vectors (7, 7) and (14, 13). It asks us to prove that the length (magnitude) of the shortest non zero vector in this lattice is equal to 1.

For this to be true, the norm calculated using the formula $\sqrt{x^2+y^2}$ will need to equal 1. The only way that is possible is if either x or y is 0 and the other 1.

Using trial and error and using simultaneous equations (linear or matrix equations both give the same results), I can find out that the vector (0, 1) lies on the lattice generated by (7, 7) and (14, 13). $\sqrt{0^2+1^2}=1$ which proves the required statement.

Then the question goes on to ask what all of these exercises tell us about the difficulty of finding the shortest non zero vector.

What it tells me is that without specific information such as multiple points that generate the lattice, or the magnitude of the shortest non zero vector on the lattice, it is *pretty darn* difficult to find the shortest non zero vector.

Now all of this is well and good but my problem is that I am basing all of this on my *intuition*, which means that I could easily be dead wrong. What I would like to know is if there is a more *scientific* way of going about this. And if there is, can someone either explain the concepts in simple terms or direct me somewhere that explains it in simple terms?

I have no formal education in lattices. The questions were dumped on us as a last minute (torture) exercise by the assignment writer. There is nothing in the textbooks and the subject guides on this

subject. I am studying via distance learning so I don't really have easy access to professors. I would like to understand this in layman's terms because most of the papers are a little too technical for me.

algorithms integer-lattices

Share Cite Follow

edited Jan 3, 2014 at 20:24

asked Jan 3, 2014 at 20:06

absurdrefusal
203 1 7

1 Answer

Sorted by:

Highest score (default)

12

It is, in fact, a difficult problem. But not in dimension 2.

From a given basis (a set of independant vectors) of an integer lattice, you are trying to find a shortest non-zero) vector of that lattice. This is the shortest vector problem (SVP), and for larger dimensions it is in fact hard (known results include NP-hardness for randomized reductions and infinity norm).

The problem is that you could potentially get a short vector from a weird integer combination of the basis vectors, as the way they interact with each other is really hard to predict (especially in larger dimensions).

In dimensions larger than two, there are in fact no better known ways to generate a short vector than to enumerate all possible vectors in a given sphere. There are ways to avoid generating too many vectors, but the complexity is still exponential. The LLL algorithm is a polynomial approach that exploits the basic idea of trying to create shorter vectors by working on only two vectors at a time (a generalisation of the 2-dimensional case), but it only gives you a relativly short vector (best proof gives you a factor of $2^{O(n)}$ on the size :().

In dimension two however, the LLL algorithm (Called Lagrange's algorithm, sometimes erroneously attributed to Gauss, a generalisation of the euclidean algorithm.) gives you a basis of the lattice made of the shortest and the second shortest (independant) vector of the lattice in polynomial time.

Short story in dimension 2 : from the largest vector of your basis, you add/remove the shortest one as to make it shorter. Rinse and repeat. (works the same way as the euclidean algorithm to find the pgcd of two integers)

Exemple : from $\begin{pmatrix}5\\5\end{pmatrix}$ and $\begin{pmatrix}14\\13\end{pmatrix}$.

you remove two times $\begin{pmatrix}5\\5\end{pmatrix}$ from $\begin{pmatrix}14\\13\end{pmatrix}$, forming $\begin{pmatrix}4\\3\end{pmatrix}$. as an integer combination, this new vector is inside your lattice.

your new vectors are $\begin{pmatrix}5\\5\end{pmatrix}$ and $\begin{pmatrix}4\\3\end{pmatrix}$. You remove the latter from the former, and you get $\begin{pmatrix}1\\2\end{pmatrix}$.

now you have $\begin{pmatrix}1\\2\end{pmatrix}$ and $\begin{pmatrix}4\\3\end{pmatrix}$. You remove the former twice from the latter, and get $\begin{pmatrix}2\\-1\end{pmatrix}$.

now you have $\begin{pmatrix}2\\-1\end{pmatrix}$ and $\begin{pmatrix}1\\2\end{pmatrix}$. You can't make any of them shorter by adding/removing the other. The LLL algorithm in dimension 2 (called the Lagrange or Gauss algorithm) tells you that those are the shortest vectors in the lattice (non dependant).

This is all usually done by using unitary transformation (multiplication by integer lattices of determinant ± 1) on the lattice basis (a $n \times n$ matrix). Here we did :

$$\begin{pmatrix}5&14\\5&13\end{pmatrix}\begin{pmatrix}1&-2\\0&1\end{pmatrix}\begin{pmatrix}1&0\\-1&1\end{pmatrix}\begin{pmatrix}1&-2\\0&1\end{pmatrix}=\begin{pmatrix}1&2\\2&-1\end{pmatrix}$$

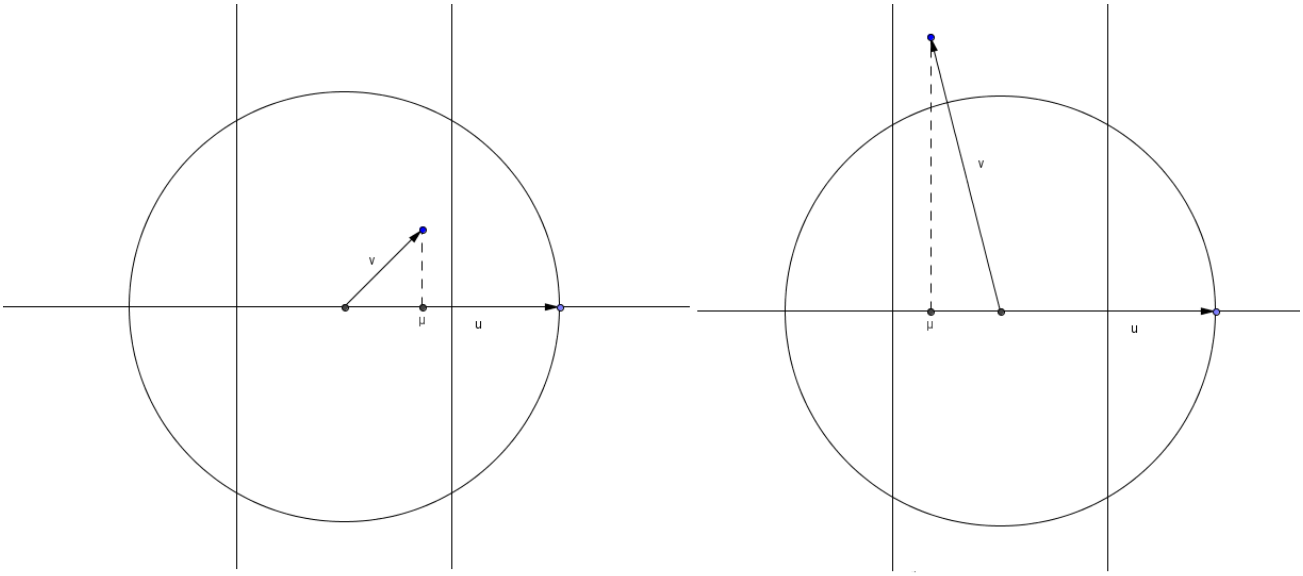
the original basis is on the left, and each step is one of the multiplying matrices (from left to right). The end result is on the right. There usually are $\begin{pmatrix}0&1\\1&0\end{pmatrix}$ matrices in between for swapping the two vectors (that way the shortest one is on the left), but I removed those here for clarity.

For larger dimensions, it gets ugly. Couldn't find anything nice for the 2 dimensional case, so let's go :

Inner working :

You have two vectors u and v, sorted by euclidean length. We define the coefficient $\mu = \dfrac{u \cdot v}{||u||^2}$.

that coefficient μ represents the relative length of the projection of v over u (see picture below). Then making the vector v as short as possible is done by removing $\lceil \mu \rfloor \times u$ from v. We get a new vector v such that one of the following happens :



the new μ is in $\left[-\frac{1}{2};\frac{1}{2}\right]$, meaning that the v vector is inside the two vertical bars. This is called size-reduced.

On the left side the vector v is inside the circle of radius $\|u\|$, meaning that v is now shorter than u. Therefore swapping position of u and v will yield $|\mu| > \frac{1}{2}$ (you can see it by projecting u on v)

On the right side however, v is outside the circle (the Lovasz condition is said to be verified) and therefore swapping the places of u and v wouldn't yield anything. The basis is said LLL-reduced, and u is the shortest non-zero vector of the lattice.

Proof :

let's say it isn't. There are x and y integers ($x \neq 0$) such that $xu + yv$ is shorter than u.

$$\begin{aligned}\|xu+yv\|^2 &= x^2\|u\|^2 + 2xy(u\cdot v) + y^2\|v\|^2 \\ &= x^2\|u\|^2 + 2\mu xy\|u\|^2 + y^2\|v\|^2 \\ &\geq (x^2 + 2\mu xy + y^2)\|u\|^2 \\ &\geq (x^2 - |xy| + y^2)\|u\|^2\end{aligned}$$

Either x^2 or y^2 is larger than $|xy|$. Therefore $(x^2 - |xy| + y^2)$ is larger than $\min(x^2; y^2)$. If that minimum is 0, $xu + yv$ is a non-zero integer combination of u **or** v, and therefore larger than u (contradiction). If it isn't, then $(x^2 - |xy| + y^2) > 0$ and we get a contradiction.

Therefore u is the shortest vector.

For LLL in larger dimension, you do the same two vectors at a time, selecting the two vectors using a process that's similar to a bubble sort : once a pair of vectors is done you get to the next vector, and you move down each time you swap vectors. For the μ coefficient you use the Gram-Schmidt Orthogonalisation.

Share Cite Follow

edited Jun 9, 2019 at 7:04

answered Jan 3, 2014 at 20:30

imj
1,372 7 10

1 In 2 dimensions, it's essentially the Euclidean algorithm, no? Calling it LLL makes it seem scarier than it really is. – Gerry Myerson Jan 3, 2014 at 20:39

1 @GerryMyerson : I think the 2-dimensional case is actually Gauss's algorithm, if I remember correctly. I'll edit – imj Jan 3, 2014 at 20:46

1 @nickecarlo : I added a few lines in the beggining to explain it a little better. I'll try to find something that's accessible in my archives for both dimension 2 and larger – imj Jan 3, 2014 at 21:00

1 @nickecarlo : actually the correct name is Lagrange's algorithm. The attribution to Gauss is common but incorrect. – imj Jan 3, 2014 at 21:10

1 @nickecarlo : added a buch of stuff, including the workings of the algorithm and its proof. For more technical stuff in larger dimensions you can check out the original LLL paper "Factoring polynomials with rational coefficients" (google). The one by E. Kaltofen "On the complexity of finding short vectors in integer lattices." if I remember correctly is easier to read and as a better complexity bound. If you read French, I have a few PDh theses on the subject where the introduction could help. – imj Jan 3, 2014 at 22:29