In this lecture we present the fastest known algorithm for solving the shortest vector problem, due to Ajtai, Kumar, and Sivakumar [1]. This is a randomized algorithm that solves the shortest vector problem exactly. Its running time is $2^{O(n)}$ times some polynomial in the length of the input where n is the dimension of the lattice. The previous best algorithm runs in time $2^{O(n \log n)} = n^{O(n)}$ (times some polynomial in the input length) and is due to Kannan [3]. In fact, his algorithm is still the fastest known *deterministic* algorithm.

1 The algorithm

We start by noting that it is enough to solve SVP for instances where the length of the shortest vector is in some known range.

LEMMA 1
Given an algorithm A that finds a shortest nonzero vector in lattices for which $2 \leq \lambda_1 < 3$, we can find a shortest nonzero vector in any lattice in time that is greater by a factor of at most $O(n)$.

PROOF: Given any input basis B we can apply the LLL algorithm and obtain an estimate $\tilde{\lambda}_1$ such that $\lambda_1(\mathcal{L}(B)) \leq \tilde{\lambda_1} \leq 2^n \lambda_1(\mathcal{L}(B))$. We now apply algorithm A on B_0, B_1 , ..., B_{2n} where $B_k := \frac{(1.5)^k}{\tilde{\lambda}_1} B$. Let $v_0, \ldots, v_{2n}$ be the vectors returned and let v'_k denote the vector $\frac{\tilde{\lambda}_1}{(1.5)^k} v_k$. We output the shortest vector among the vectors $v'_0, \ldots, v'_{2n}$ that are contained in $\mathcal{L}(B) \setminus \{0\}$.

To prove the correctness of this algorithm, notice that there exists a $k \in \{0, \ldots, 2n\}$ for which $2 \leq \lambda_1(\mathcal{L}(B_k)) < 3$. By our assumption on A, for that k, v_k is the shortest vector in B_k. Therefore, v'_k is a shortest nonzero vector in B and hence the algorithm outputs a shortest nonzero vector in B. □

The next lemma lies at the core of the algorithm. It says that in any set of points inside a ball of radius R, one can find a subset of at most 5^n 'centers' such that any point has a center within distance at most $R/2$.

LEMMA 2 (SIEVE)
Let $R > 0$ be some real. For any set of points $x_1, \ldots, x_N$ in $\mathbf{B}(0, R)$ we can find a subset $J \subseteq [N]$ of size at most 5^n and a mapping $\eta : [N] \to J$ such that for any $i \in [N]$, $\|x_i - x_{\eta(i)}\| \leq \frac{R}{2}$. The running time of this procedure is polynomial in the input size.

PROOF: Consider the following procedure. We initialize J to be the empty set and then, for each $i = 1, \ldots, N$, do the following. If there exists a $j \in J$ such that $\|x_i - x_j\| \leq \frac{R}{2}$ then define $\eta(i) = j$ and continue to the next i. Otherwise, add i to J and define $\eta(i) = i$.

We now show that $|J| \leq 5^n$. First, notice that the distance between any two points in J is greater than $\frac{R}{2}$. Hence, if we take balls of radius $\frac{R}{4}$ around each point in J then these balls are disjoint. On the other hand, their union is contained in $\mathbf{B}(0, \frac{5}{4}R)$. Therefore, the number of balls (and hence also $|J|$) is at most

$$\frac{\mathrm{vol}(\mathbf{B}(0, \frac{5}{4}R))}{\mathrm{vol}(\mathbf{B}(0, \frac{1}{4}R))} = 5^n.$$

□

We now describe the SVP algorithm.

Algorithm:

INPUT: A lattice basis $B = (b_1, \ldots, b_n)$, with $\lambda_1(\mathcal{L}(B)) \in [2, 3)$
OUTPUT: a shortest nonzero vector in B

1. $R_0 \leftarrow n \max_i \|b_i\|$
   Choose $N = 2^{8n} \log R_0$ points $x_1, \ldots, x_N$ uniformly in $\mathbf{B}(0, 2)$
   Compute $y_i = x_i \mod \mathcal{P}(B)$ for $i = 1, \ldots, N$
   Let $Z = \{(x_1, y_1), \ldots, (x_N, y_N)\}$
   $R \leftarrow R_0$

2. **while** $R > 6$ **do**
   { Invariant 1: For all $(x_i, y_i) \in Z$, $y_i - x_i \in \mathcal{L}(B)$ }
   { Invariant 2: For all $(x_i, y_i) \in Z$, $\|y_i\| \leq R$ }
   Apply the sieving procedure to the y vectors in Z
   The result is a set J of at most 5^n 'centers' and a mapping η s.t. $\forall i. \|y_i - y_{\eta(i)}\| \leq \frac{R}{2}$
   Remove from Z all pairs (x_i, y_i) corresponding to $i \in J$
   Replace each remaining pair (x_i, y_i) in Z with $(x_i, y_i - (y_{\eta(i)} - x_{\eta(i)}))$
   $R \leftarrow \frac{R}{2} + 2$

3. For any two pairs $(x_i, y_i), (x_j, y_j)$ in Z consider the difference $(y_i - x_i) - (y_j - x_j)$
   and output the shortest nonzero vector among these differences

We now analyze this algorithm. First, notice that the number of iterations of Step 2 is at most, say, $2 \log R_0$. Second, the running time of the sieving procedure is polynomial in the size of Z. Hence, the total running time is $2^{O(n)} \text{poly} \log R_0$. Since the input size is at least $\log R_0$, this running time is $2^{O(n)}$ times some polynomial in the input size, as claimed.

Next, we show that the two invariants are maintained. The first invariant is satisfied when we begin Step 2 by the choice of y_i in Step 1, and is maintained in Step 2 since we only subtract from y_i vectors of the form $y_j - x_j$ which are themselves lattice vectors. Next, let us consider the second invariant. It holds in the first iteration because $y_i \in \mathcal{P}(B)$ and hence $\|y_i\| \leq \sum \|b_i\| \leq R_0$. Moreover, it is maintained in Step 2 because

$$\|y_i - (y_{\eta(i)} - x_{\eta(i)})\| \leq \|y_i - y_{\eta(i)}\| + \|x_{\eta(i)}\| \leq \frac{R}{2} + 2.$$

The total number of pairs removed from Z during Step 2 is at most $5^n \cdot 2 \log R_0$. Hence, when we get to Step 3, Z contains at least $(2^{8n} - 2 \cdot 5^n) \log R_0$ pairs (x_i, y_i), which, by the second invariant, satisfy

$$\|y_i - x_i\| \leq \|y_i\| + \|x_i\| \leq 6 + 2 = 8.$$

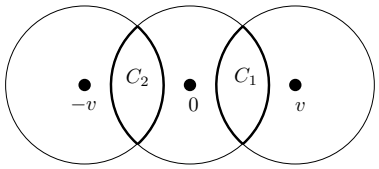
Moreover, by the first invariant, $y_i - x_i \in \mathcal{L}(B)$ and hence, when we begin Step 3 we have a list of many short lattice vectors. At first, it might seem that this already shows that we have a very good approximation to the SVP (recall that $\lambda_1 \in [2, 3)$). However, notice that so far we have not excluded the possibility that all these vectors are the zero vector!

We deal with this crucial issue next. For this we use the randomization in the algorithm. We argue that due to the randomization, the algorithm must produce some nonzero vectors. Intuitively, instead of directly analyzing the loop in Step 2, we regard it as a black box, and argue that since it does not 'know' exactly the randomization performed in Step 1, it must sometimes output nonzero vectors. In fact, the analysis shows that in Step 3 we obtain a shortest nonzero vector and not just an approximation of it.

Let v be a shortest nonzero vector in $\mathcal{L}(B)$, so $\|v\| \in [2, 3)$. Denote $C_1 = \mathbf{B}(0, 2) \cap \mathbf{B}(v, 2)$, $C_2 = \mathbf{B}(0, 2) \cap \mathbf{B}(-v, 2)$, as illustrated in the following figure.

CLAIM 3

$$\frac{\text{vol}(C_1)}{\text{vol}(\mathbf{B}(0, 2))} = \frac{\text{vol}(C_2)}{\text{vol}(\mathbf{B}(0, 2))} \geq 2^{-2n}$$



PROOF: The equality is obvious. To prove the inequality, notice that C_1 contains a cylinder of height $\frac{1}{2}$ and radius $\sqrt{15/16}$ centered around $v/2$. So,

$$\mathrm{vol}(C_1) \geq \mathrm{vol}(\mathbf{B}_{n-1}(0, \sqrt{15/16}))/2$$

where $\mathbf{B}_{n-1}$ indicates an $n-1$-dimensional ball. On the other hand, $\mathbf{B}(0,2)$ is contained in a cylinder of height 4 and radius 2, so

$$\mathrm{vol}(\mathbf{B}(0,2)) \leq 4\mathrm{vol}(\mathbf{B}_{n-1}(0,2)).$$

Hence,

$$\frac{\mathrm{vol}(C_1)}{\mathrm{vol}(\mathbf{B}(0,2))} \geq \frac{\mathrm{vol}(\mathbf{B}_{n-1}(0,\sqrt{15/16}))}{8\mathrm{vol}(\mathbf{B}_{n-1}(0,2)} = \frac{1}{8}\left(\frac{15}{64}\right)^{(n-1)/2} \geq 2^{-2n}$$

for large enough n. □

Define a bijection τ on $\mathbf{B}(0,2)$ that maps C_1 to C_2, C_2 to C_1, and $\mathbf{B}(0,2) \setminus (C_1 \cup C_2)$ to itself:

$$\tau(x) = \begin{cases} x+v & x \in C_2 \\ x-v & x \in C_1 \\ x & \text{otherwise} \end{cases}$$

Since $v \in \mathcal{L}(B)$, we have that for any $x \in \mathbf{B}(0,2)$, $x \mod \mathcal{P}(B) = \tau(x) \mod \mathcal{P}(B)$. Moreover, if x is chosen uniformly from $\mathbf{B}(0,2)$ then $\tau(x)$ is also uniform on $\mathbf{B}(0,2)$.

Consider now the following modification to Step 1 of the algorithm. After choosing each x_i, we toss a fair coin and if it comes up heads, we replace x_i with $\tau(x_i)$. We refer to this operation as 'tossing x_i'. Since this modified algorithm is only used for the analysis, we do not need to worry about its running time (so there is no problem with the fact that computing τ involves v, a shortest nonzero vector). Since $\tau(x_i)$ and x_i are identically distributed, and $x \bmod \mathcal{P}(B) = \tau(x) \bmod \mathcal{P}(B)$, the modified algorithm behaves exactly as the original algorithm. So, in the sequel, we analyze it instead.

Our next crucial observation is that we can postpone the tossing of x_i to the first time in which it has an effect on the algorithm. More precisely, we modify the algorithm once again: we no longer toss any x_i in Step 1; instead, in Step 2, right after we call the sieving procedure, we toss all x_i for $i \in J$. Moreover, in the beginning of Step 3, we toss all remaining x_i. This does not affect the algorithm since the operation $y_i = x_i \bmod \mathcal{P}(B)$ in Step 1 is independent of the tossing of x_i, and elsewhere, x_i is not used before it is tossed.

Let us call a point x_i 'green' if $x_i \in C_1 \cup C_2$.

CLAIM 4 *With high probability, there are at least $2^{6n-1}\log R_0$ green points in the initial set $x_1, \ldots, x_N$.*

PROOF: By Claim 3, each x_i is green with probability at least $p := 2^{-2n}$. Hence, the expected number of green points is pN and the variance of this number is at most pN (since the variance of a Bernoulli trial is $p(1-p) < p$). By Chebyshev's inequality, the probability that there are less than $pN/2$ points is at most $4/pN$ which is exponentially small. □

Consider now the situation at the beginning of Step 3 (before tossing the x_i). By the above claim, the number of green points in Z is at least $(2^{6n-1} - 2\cdot 5^n)\log R_0 > 2^{5n}$. For each such point, $y_i - x_i$ is a lattice point inside $\mathbf{B}(0,8)$. We now show that this implies that there are many repetitions among $y_i - x_i$.

CLAIM 5 $|\mathbf{B}(0,8) \cap \mathcal{L}(B)| < 2^{4n}$

PROOF: The distance between any two lattice points is at least $\lambda_1(\mathcal{L}(B)) \geq 2$. Hence if we put a ball of radius 1 around each point in $\mathbf{B}(0,8) \cap \mathcal{L}(B)$, the balls are disjoint. On the other hand, these balls are contained in $\mathbf{B}(0,9)$. Hence,

$$|\mathbf{B}(0,8) \cap \mathcal{L}(B)| \leq \frac{\mathrm{vol}(\mathbf{B}(0,9))}{\mathrm{vol}(\mathbf{B}(0,1))} = 9^n < 2^{4n}.$$

□

Hence, there exists a lattice vector w and at least $2^{5n}/2^{4n} = 2^n$ indices i for which $y_i - x_i = w$ and x_i is green. After we toss all vectors in Step 3, for each such i, $y_i - x_i$ remains w with probability $1/2$ or otherwise becomes one of $w + v, w - v$. Therefore, with very high probability, in Step 3 we find a pair of indices i, j for which $(y_i - x_i) - (y_j - x_j) = v$, as required.

2 Final notes

In an earlier paper, Kumar and Sivakumar [4] presented a randomized algorithm that *approximates* the SVP to within n^c for some constant c. Its running time is similar to the above algorithm, namely, $2^{O(n)}$ times some polynomial in the input length. Although this result is a clearly weaker than the one above, their proof is intriguing as it builds heavily on Ajtai's worst-case to average-case reduction. It also has the advantage of being somewhat simpler.

In a more recent paper, Ajtai, Kumar, and Sivakumar presented an algorithm for solving the CVP with a running time similar to the above SVP algorithm [2]. Their algorithm is randomized and is based on the SVP algorithm.

Finally, we mention two interesting open questions:

- Can the dependence of the running time on n be improved?
- Is it possible to derandomize the SVP algorithm?

References

[1] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 601–610, 2001.

[2] M. Ajtai, R. Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *Annual IEEE Conference on Computational Complexity*, volume 17, 2002.

[3] R. Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.

[4] R. Kumar and D. Sivakumar. On polynomial approximation to the shortest lattice vector length. In *Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 126–127, 2001.