Lattice Algorithms: Design, Analysis and Experiments

Phong Nguyễn

http://www.di.ens.fr/~pnguyen

March 2017

Warning

- Interaction: please ask questions during my talks; interruptions are welcome.
- Slides will be available online.
- If you really want to understand an algorithm, it is helpful to implement it, using sage or NTL.

The Ubiquity of Lattices

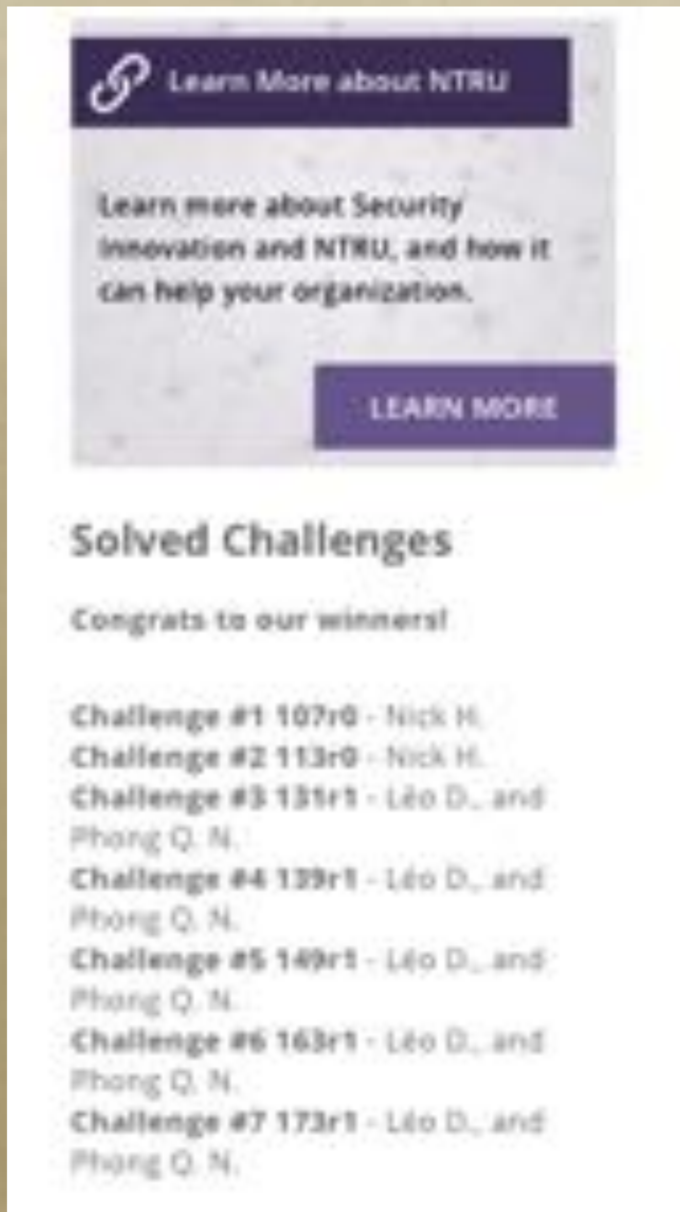
- In mathematics
  - Algebraic number theory, Algebraic geometry, Sphere packings, etc.
  - Fields medals: G. Margulis (1978), E. Lindenstrauss and S. Smirnov (2010), M. Bhargava (2014).
- Applications in computer science, statistical physics, etc.

Motivation

Motivation

- Many people want convincing security estimates for lattice-based cryptosystems (and other post-quantum cryposystems).
- Use numerical challenges as a sanity check of the state-of-the-art.

NTRU Challenges (2015-)



- Method used in largest records: Enumeration with BKZ.

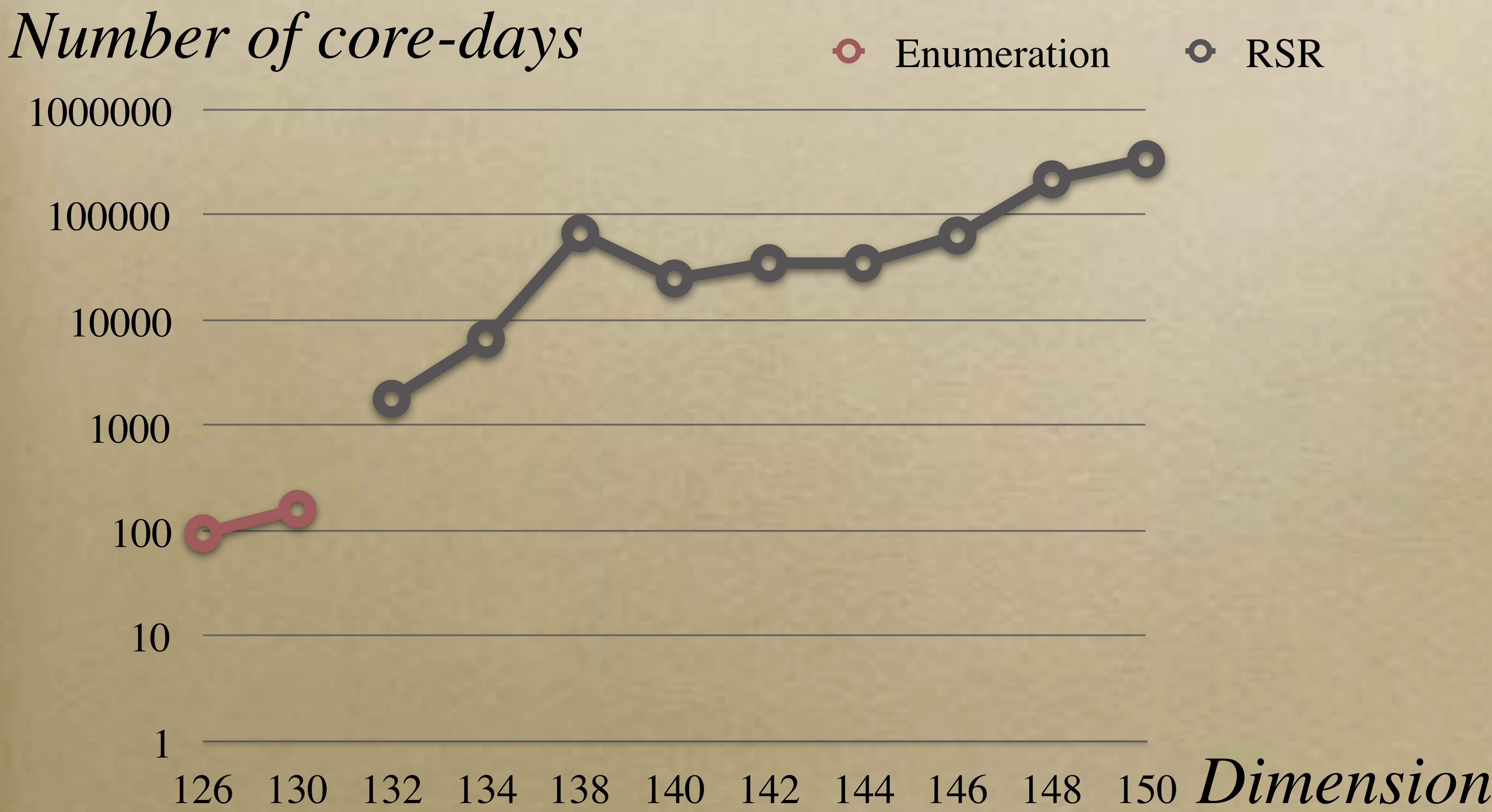
Darmstadt Lattice Challenge (2008-)

- Method used in largest records: Enumeration with BKZ.

Darmstadt SVP Challenge (2010-)

- Method used in largest records?

The SVP Challenges



Comparison with RSA Records

- The largest SVP-computation is for dim 150 (Jan. 2017): 340,000 core-days ≈ 2^{66} clock cycles.
- This is only half RSA-768 = 730,000 core-days ≈ 2^{67} clock cycles.

Goal

- Understand the main ideas and underlying the best lattice algorithms in practice.
- Understand their limitations.

Trends

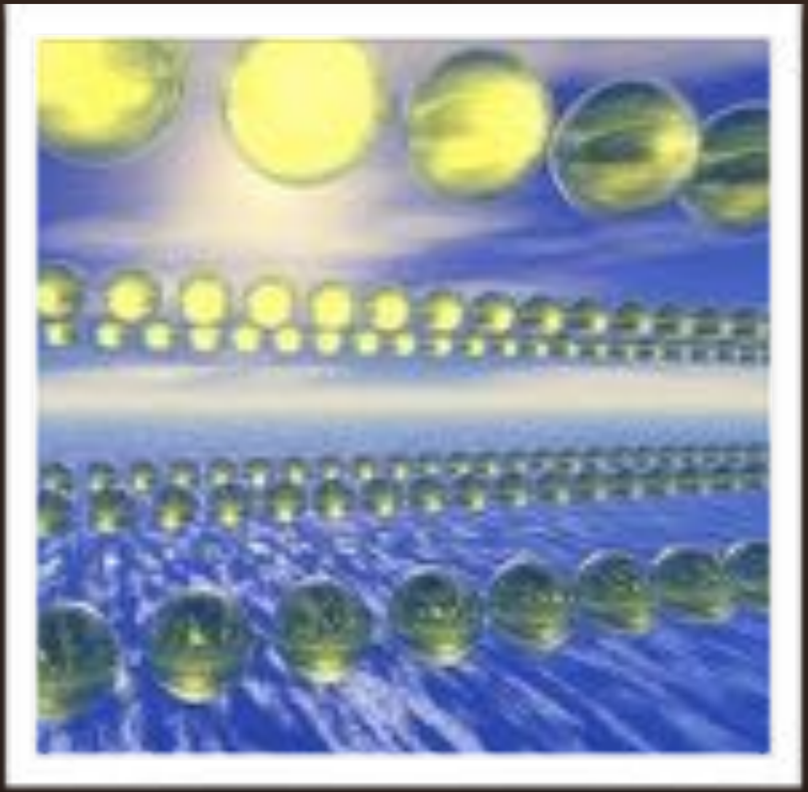
- Imbalance: much more publications on the design of lattice-based cryptographic schemes than lattice algorithms.
- The literature on lattice algorithms can be confusing:
  - Provable ≠ heuristic
  - Worst-case analysis ≠ typical behaviour
  - Sometimes, incorrect statements

Summary

- Mathematical background
- Enumeration
  - Cylinder pruning
  - Discrete pruning
- Algorithms from Hermite's constant
  - LLL and Hermite's inequality
  - Block-wise algorithms and Mordell's inequality
  - Mordell's proof of Minkowski's inequality
- Security Estimates

Overview

- The biggest distinction among lattice algorithms is space:
  - Poly-space algorithms
  - Exp-space algorithms

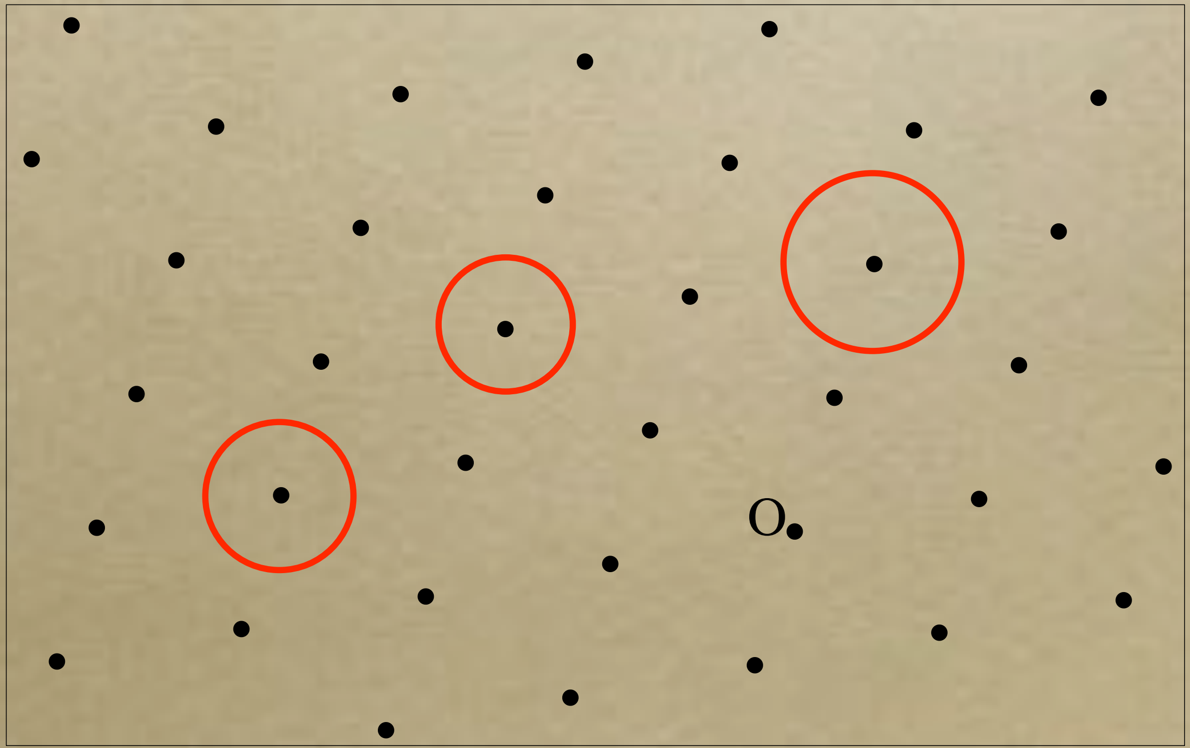


Mathematical Background

What is a Lattice?

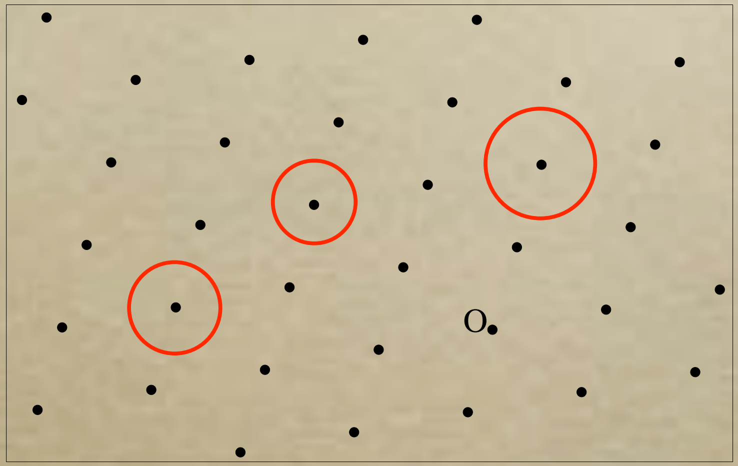
- A lattice is a discrete subgroup of $\mathbf{R}^n$, or the set $L(b_1,\ldots,b_d)$ of all linear combinations $\Sigma x_i b_i$ where $x_i \in \mathbf{Z}$, and the b_i's are linearly independent.

2	0	0	0	0
0	2	0	0	0
0	0	2	0	0
0	0	0	2	0
1	1	1	1	1



Integer Lattices

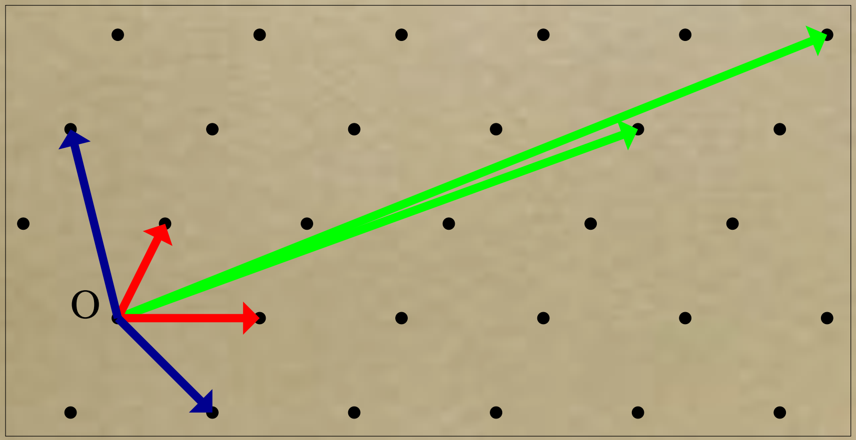
- A (full-rank) integer lattice is any subgroup L of $(\mathbf{Z}^n,+)$ s.t. $\mathbf{Z}^n/L$ is finite.



- A lattice is infinite, but lattice crypto implicitly uses the finite abelian group $\mathbf{Z}^n/L$: it works modulo the lattice L.

Lattice Invariants

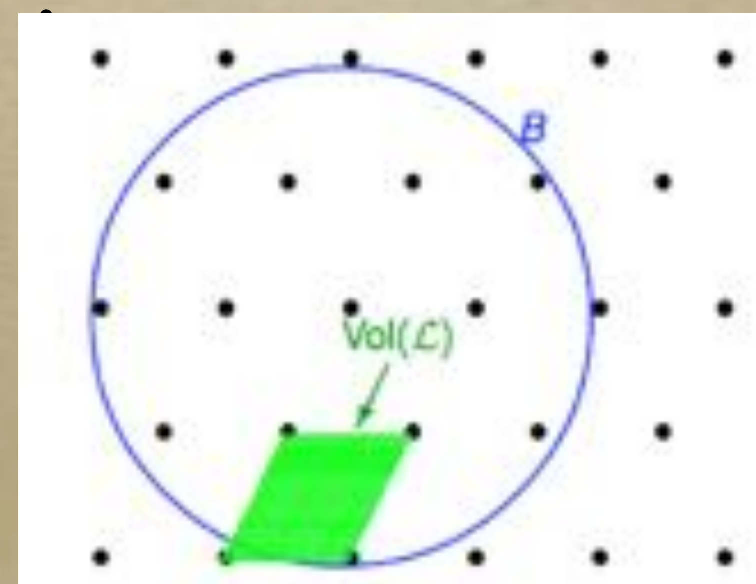
- The dim is the dim of span(L).
- The (co-)volume is the volume of any basis parallelepiped: can be computed in poly-time. Ex: vol($\mathbf{Z}^n$)=1.



The Gaussian Heuristic

- The volume measures the density of lattice points.
- For "nice" full-rank lattices L, and "nice" measurable sets C of $\mathbf{R}^n$:

$$\mathrm{Card}(L \cap C) \approx \frac{\mathrm{vol}(C)}{\mathrm{vol}(L)}$$



Volume of the Ball

The n-dimensional volume of a Euclidean ball of radius R in n-dimensional Euclidean space is:

$$V_n(R) = \frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}+1\right)} R^n,$$

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\, \mathrm{d}x$$

Short Lattice Vectors

- Th: Any d-dim lattice L has exponentially many vectors of norm ≤

$$O\left(\sqrt{d}\right) \mathrm{vol}(L)^{1/d}$$

- Th: In a random d-dim lattice L, all non-zero vectors have norm ≥

$$\Omega\left(\sqrt{d}\right) \mathrm{vol}(L)^{1/d}$$

Hermite's Constant (1850)

- This is the "worst-case" for short lattice vectors.
- Hermite showed the existence of this constant:

$$\sqrt{\gamma_d} = \max_L \frac{\lambda_1(L)}{\mathrm{vol}(L)^{1/d}}$$

- Here, $\lambda_1(L)$ is the minimal norm of a non-zero lattice vector.

Facts on Hermite's Constant

- Hermite's constant is asymptotically linear:

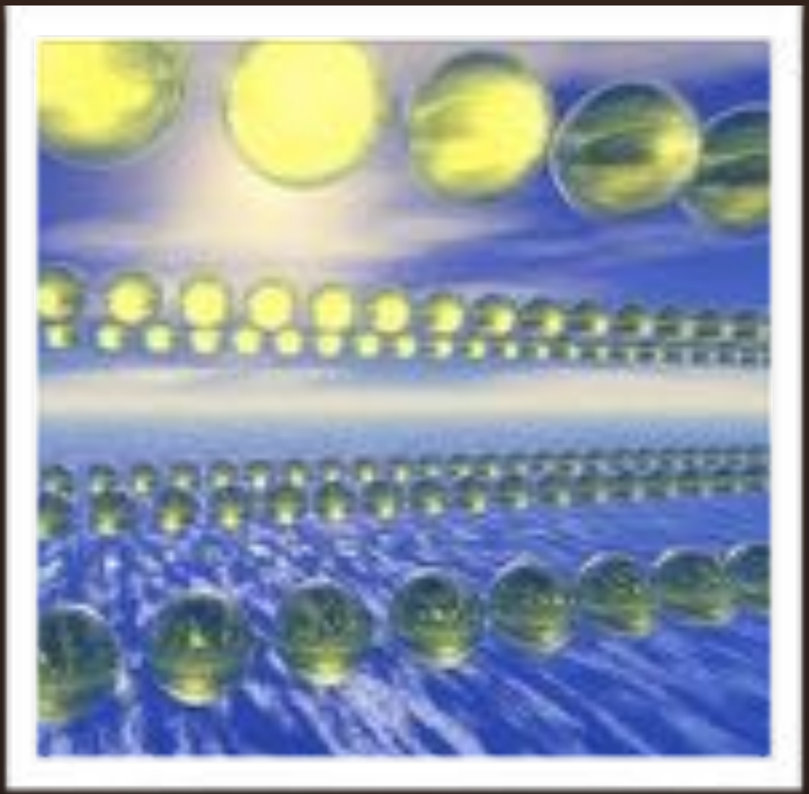
$$\Omega(n) \leq \gamma_n \leq O(n)$$

- The exact value of the constant is only known up to dim 8, and in dim 24 [2004].

dim n	2	3	4	5	6	7	8	24
γ_n	$2/\sqrt{3}$	$2^{1/3}$	$\sqrt{2}$	$8^{1/5}$	$(64/3)^{1/6}$	$64^{1/7}$	2	4
approx	1.16	1.26	1.41	1.52	1.67	1.81	2	4

Mathematical Goals

- Classical setting: the worst case.
  - Find the exact value of Hermite's constant.
- New trends: the average case.
  - Properties of random lattices, developing results from the 50s.
  - Properties of random lattice points



Overview of Lattice Algorithms

Lattice Algorithms

- Input = integer matrix, whose rows span the lattice. Parameters:
  - Size of basis coefficients
  - Lattice dimension
- Asymptotically:
  - dim increases
  - coeff-size polynomial in dim.

Hard Lattice Problems

- Since 1996, lattices are very trendy in classical and quantum complexity theory.
- Depending on the dimension d: approx. factor
  - NP-hardness — $O(1)$ — 1
  - non NP-hardness (NP∩co-NP) — $\sqrt{d}$
  - worst-case/average-case reduction — $d \log d$
  - cryptography — $d^{O(1)}$
  - **subexp-time** algorithms — $2^{\sqrt{d}}$
  - **poly-time** algorithms — $2^{\frac{d \log \log d}{\log d}}$ — ∞

Generic Lattice Problem

- Input: a lattice L and a ball C
- Output: decide if $L \cap C$ is non-trivial, and if it is, find a non-trivial point.
- Settings
    - Approx: $L \cap C$ has many points. Ex: SIS and ISIS.
    - Unique: essentially, L has one non-trivial point, even though C might be small.

The Shortest Vector Problem (SVP)

- Input: a basis of a d-dim lattice L
- Output: nonzero v∈L minimizing ||v|| i.e. $||v|| = \lambda_1(L)$

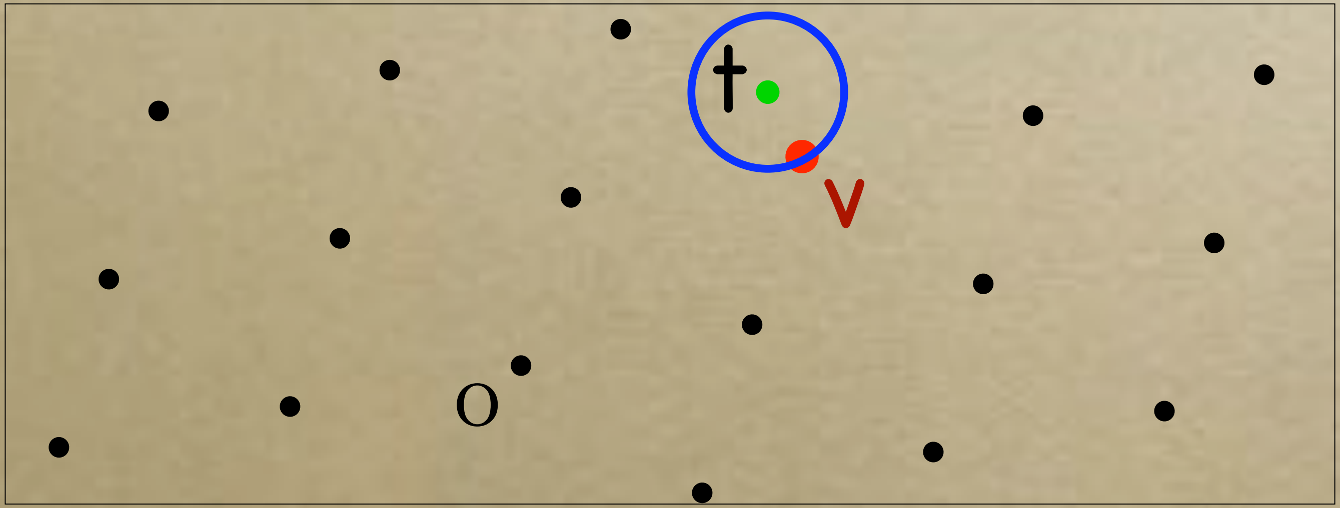
2	0	0	0	0
0	2	0	0	0
0	0	2	0	0
0	0	0	2	0
1	1	1	1	1

Relaxing SVP

- Input: a basis of a d-dim lattice L.
- Output: nonzero $v \in L$ such that
- Approximate-SVP: $||v|| \le f(d)\ \lambda_1(L)$ [relative]
- Hermite-SVP: $||v|| \le g(d)\ \mathrm{vol}(L)^{1/d}$ [absolute]

The Closest Vector Problem (CVP)

- Input: a basis of a lattice L of dim d, and a target vector t.
- Output: v∈L minimizing ||v-t||.



- BDD (bounded distance decoding): special case when t is very close to L.

Insight

- The most classical problem is to prove the existence of short lattice vectors.
- All known upper bounds on Hermite's constant have an algorithmic analogue:
  - Hermite's inequality: the LLL algorithm.
  - Mordell's inequality: Blockwise generalizations of LLL.
  - Mordell's proof of Minkowski's inequality: worst-case to average-case reductions for SIS and sieve algorithms [BJN14,ADRS15]

Hermite's Inequality and LLL

Hermite's Inequality

- Hermite proved in 1850:

$$\gamma_d \leq \gamma_2^{d-1} = \left(\frac{4}{3}\right)^{(d-1)/2}$$

- [LLL82] finds in polynomial time a non-zero lattice vector of norm ≤ $(4/3+\varepsilon)^{(d-1)/4}\mathrm{vol}(L)^{1/d}$.

  It is an algorithmic version of Hermite's inequality.

Proof of Hermite's Inequality

- Induction over d: obvious for d=1.
- Let b_1 be a shortest vector of L, and π the projection over $b_1^{\perp}$.
- Let $\pi(b_2)$ be a shortest vector of $\pi(L)$.
- We can make sure by lifting that:
  $\|b_2\|^2 \leq \|\pi(b_2)\|^2 + \|b_1\|^2/4$ (size-reduction)
- On the other hand, $\|b_1\| \leq \|b_2\|$ and $vol(\pi(L)) = vol(L)/\|b_1\|$.

Question

- Is the proof constructive?
- Does it build a non-zero lattice vector satisfying Hermite's inequality:

$$\|\vec{b}_1\| \leq \left(\frac{4}{3}\right)^{(d-1)/4} \mathrm{vol}(L)^{1/d}$$

An Algorithmic Proof

- Let b_1 be a primitive vector of L, and π the projection over $b_1^\perp$.
- Find recursively $\pi(b_2) \in \pi(L)$ satisfying Hermite's inequality.
- Size-reduce so that $\|b_2\|^2 \leq \|\pi(b_2)\|^2 + \|b_1\|^2/4$
- If $\|b_2\| < \|b_1\|$, swap(b_1, b_2) and restart, otherwise stop.

An Algorithmic Proof

- This algorithm will terminate and output a non-zero lattice vector satisfying Hermite's inequality:

$$\|\vec{b}_1\| \leq \left(\frac{4}{3}\right)^{(d-1)/4} \mathrm{vol}(L)^{1/d}$$

- But it may not be efficient: LLL does better by strengthening the test $\|b_2\| < \|b_1\|$.

Recursive LLL

- Input: $(b_1,b_2,\ldots,b_d)$ basis of L and $\varepsilon > 0$.
- LLL-reduce $(\pi(b_2),\ldots,\pi(b_d))$ where π is the projection over $b_1^{\perp}$.
- Size-reduce so that $\|b_i\|^2 \leq \|\pi(b_i)\|^2 + \|b_1\|^2/4$
- If $\|b_2\| \leq (1-\varepsilon)\|b_1\|$, swap$(b_1, b_2)$ and restart, otherwise stop.

Take Away

- Hermite's inequality and LLL are based on two key ideas:
  - Projection
  - Lifting projected vectors aka size-reduction.

LLL in Practice

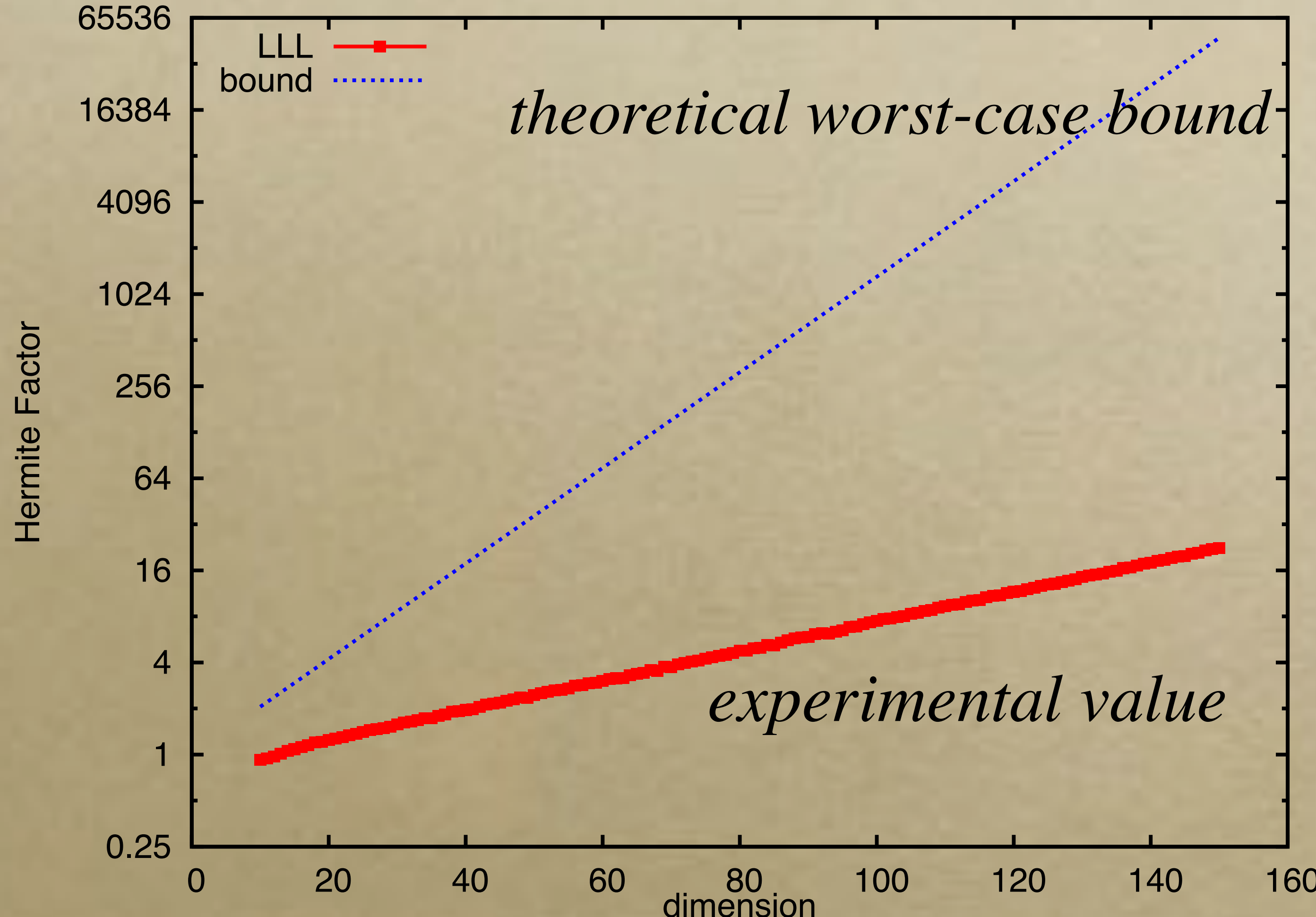
The Magic of LLL

- One of the main reasons behind the popularity of LLL is that it performs "much better" than what the worst-case bounds suggest, especially in low dimension.
- This is another example of worst-case vs. "average-case".

LLL: Theory vs Practice

- The approx factors $(4/3+\varepsilon)^{(d-1)/4}$ is tight in the worst case: but this is only for worst-case bases of certain lattices.
- Experimentally, $4/3+\varepsilon \approx 1.33$ can be replaced by a smaller constant ≈ 1.08, for any lattice, by randomizing the input basis.
- But there is no good explanation for this phenomenon, and no known formula for the experimental constant ≈ 1.08.

Illustration



Log(Hermite Factor)

Random Bases

- There is no natural probability space over the infinite set of bases.
- Folklore: generate a « random » unimodular matrix and multiply by a fixed basis. But distribution not so good.
- Better method:
  - Generate say n+20 random long lattice points
  - Extract a basis, e.g. using LLL.

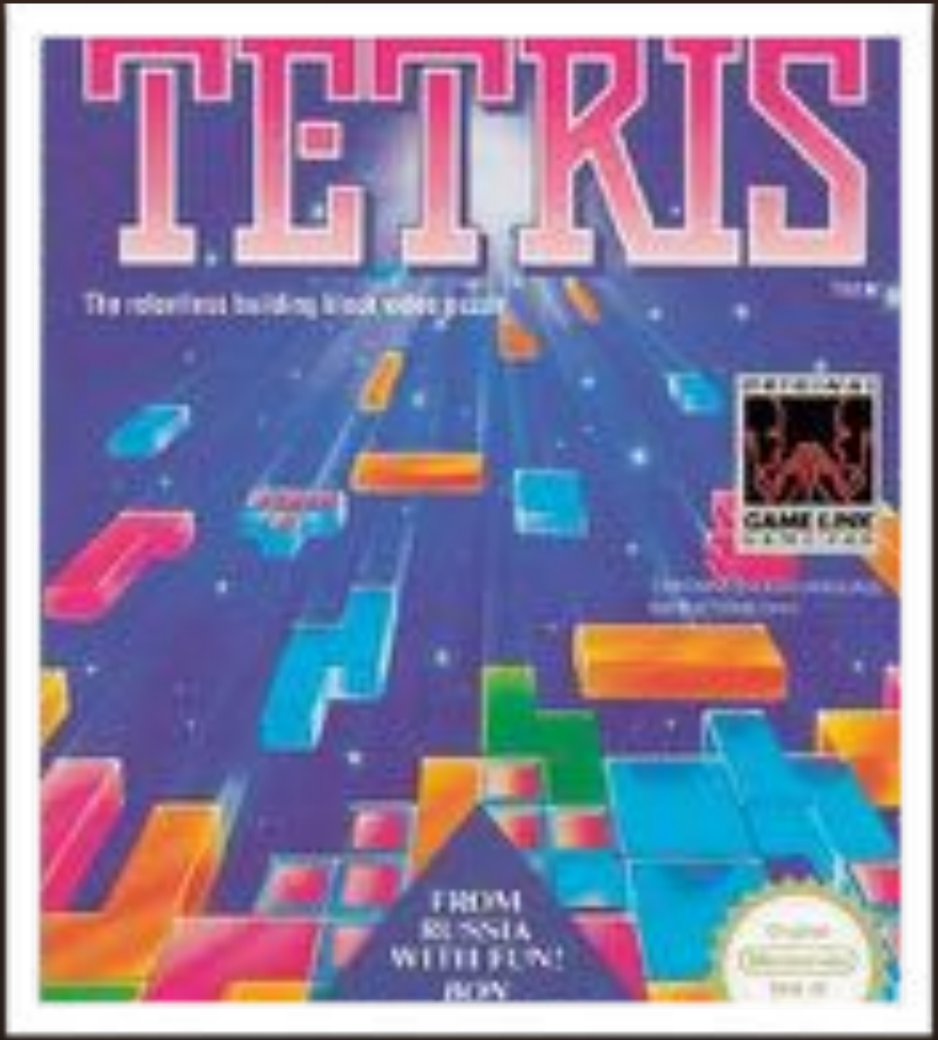
Random LLL

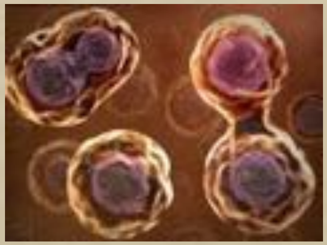
- Surprisingly, [KiVe16] showed that most LLL bases of a random lattice have a $||b_1||$ close to the worst case. Note: in fixed dimension, the number of LLL bases can be bounded, independently of the lattice.
- This means that LLL biases the output distribution.

Open problem

- Take a random integer lattice L.
- Let B be the Hermite normal form of L, or a « random » basis from the discrete Gaussian distribution.
- Is is true that with overwhelming probability, after LLL-reducing B, $||b_1|| \leq c^{d-1} vol(L)^{1/d}$ for some $c<(4/3)^{1/4}$?

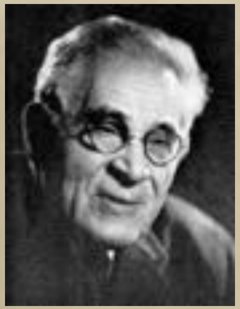
Mordell's Inequality and Blockwise Algorithms





Divide and Conquer

- LLL is based on a local reduction in dim 2.
- Blockwise algorithms find shorter vectors than LLL by using an « exact » SVP-subroutine in low dim k called blocksize.
  - Even if the subroutine takes exponential time in k, this is poly in d if k=log d.



Mordell's Inequality

- If we show the existence of very short lattice vectors in dim k, can we prove the existence of very short lattice vectors in dim d > k?
- [Mordell1944]'s inequality generalizes Hermite's inequality:

$$\sqrt{\gamma_d} \leq \sqrt{\gamma_k}^{(d-1)/(k-1)}$$

$$\lambda_1(L) \leq \sqrt{\gamma_k}^{(d-1)/(k-1)} \mathrm{vol}(L)^{1/d}$$

Approximation Algorithms for SVP

- Related to upper bounds on Hermite's constant, i.e. proving the existence of short lattice vectors.

- [LLL82] corresponds to [Hermite1850]'s inequality.

$$\|L\| \leq \left(\frac{4}{3}\right)^{(d-1)/4} \mathrm{vol}(L)^{1/d} = \sqrt{\gamma_2}^{d-1} \mathrm{vol}(L)^{1/d}$$

- Blockwise algorithms [Schnorr87, GHKN06, GamaN08,MiWa16] are related to

$$\|L\| \leq \sqrt{\gamma_k}^{(d-1)/(k-1)} \mathrm{vol}(L)^{1/d}$$

Achieving Mordell's Inequality

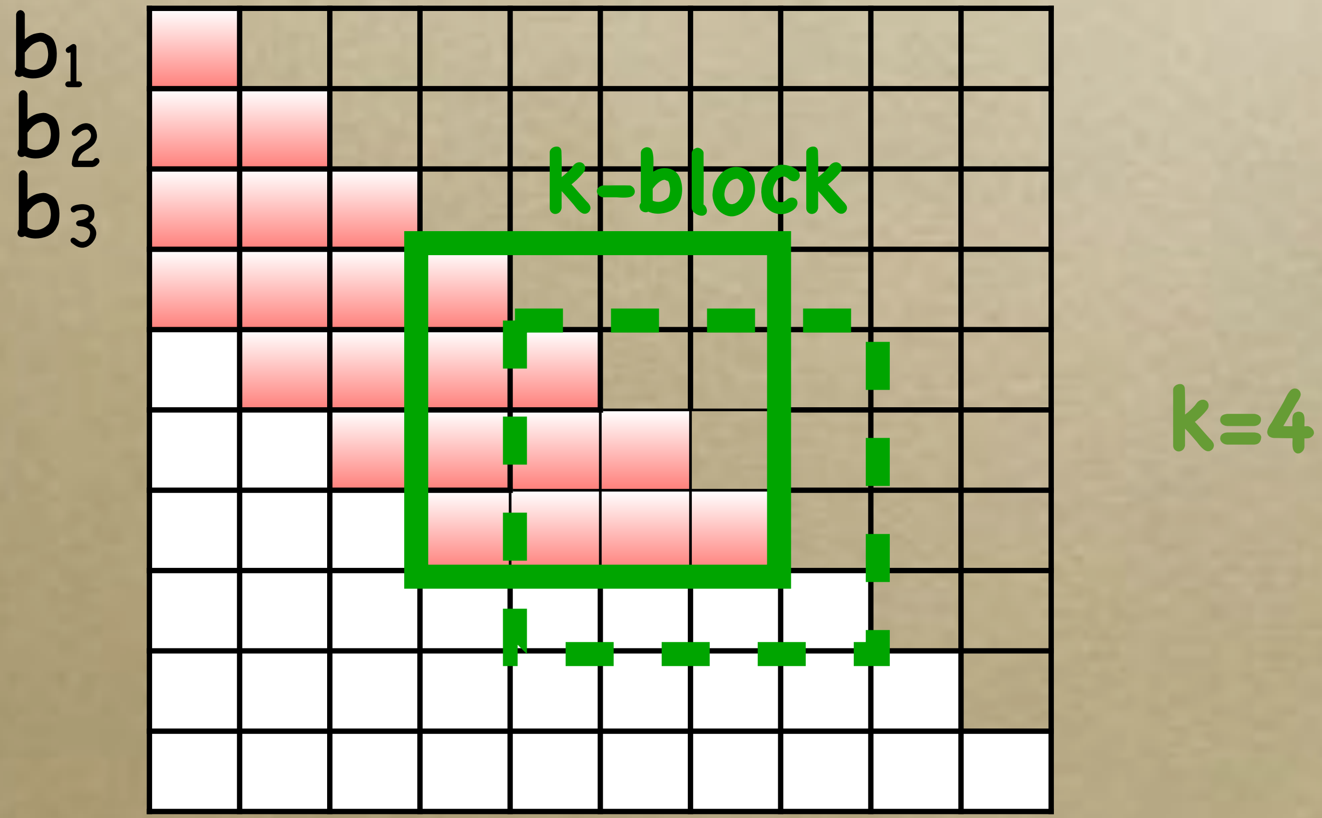
- All blockwise algorithms reaching Mordell's inequality use duality, which provides a different way of reducing the dimension.
  - Let v be a non-zero vector in the dual lattice $L^{\times}$.
  - Then $L \cap v^{\perp}$ is a lattice of dimension d-1.

What is BKZ?

- Among all blockwise algorithms, BKZ is the simplest, and seems to be the best in practice, though its bound is a bit worse than Mordell's inequality.
- Blockwise algorithms have different worst-case bounds, but in high blocksize, there may not be much differences in practice.

How BKZ Works

- BKZ repeatedly calls the k-dim SVP-subroutine to ensure that the first vector in each block is the first minimum.



Description of BKZ

- LLL-reduce the basis
- i = 1
- While some block is not reduced
  - Find the shortest vector in the k-block starting at index i.
  - If it is shorter than b_i^* : insert the new vector and run LLL to obtain a new basis.

Output of BKZ

- A basis output by BKZ is such that:
  - It is LLL-reduced
  - For each i, b_i^* is a (or near-) shortest vector in the k-block $(\pi_i(b_i), \pi_i(b_{i+1}), \ldots, \pi_i(b_{\min(d,i+k-1)}))$

Algorithms from Minkowski's Inequality

Short Lattice Vectors: Minkowski's Inequality

- [Minkowski]: Any d-dim lattice L has at least one non-zero vector of norm ≤

$$2\frac{\Gamma(1+d/2)^{1/d}}{\sqrt{\pi}}\mathrm{covol}(\mathrm{L})^{1/d} \leq \sqrt{d}\ \mathrm{covol}(\mathrm{L})^{1/d}$$

- This is Minkowski's inequality on Hermite's constant:

$$\sqrt{\gamma_d} \leq \frac{2}{v_d^{1/d}} = 2\frac{\Gamma(1+\frac{d}{2})^{1/d}}{\sqrt{\pi}} \leq \sqrt{d}$$

Four Proofs of Minkowski's Inequality

- Blichfeldt's proof: «continuous» pigeon-hole principle.

- Minkowski's original proof: sphere packings.
- Siegel's proof: Poisson summation.
- Mordell's proof: pigeon-hole principle.

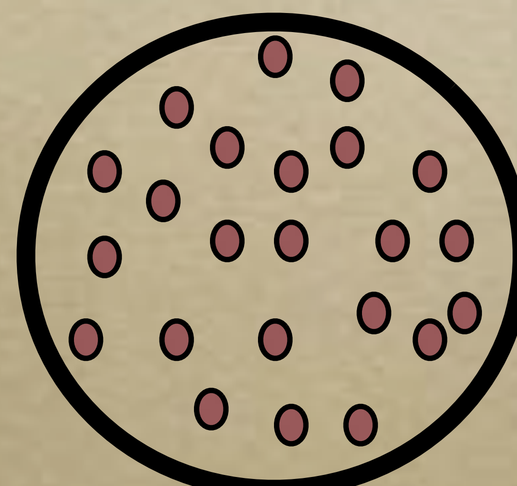
Mordell's Proof (1933)

Remember Blichfeldt's Proof

- The short lattice vector is some u-v where u,v∈F for a well-chosen convex (infinite) set F.
- Mordell's proof uses a finite F.

Mordell's Proof (1933)

- For $q \in \mathbf{N}$, let $\bar{L} = q^{-1}L$ then $[\bar{L}:L] = q^d$.
  Among $> q^d$ points $v_1, \ldots, v_m$ in $\bar{L}$, $\exists i \neq j$ s.t. $v_i - v_j \in L$.
- There are enough points in a large ball of radius r
  (r is close to Minkowski's bound in L, but large for $\bar{L}$)



- We obtain a short non-zero point in L: norm $\leq 2r$.

Key Point

- Mordell proved the existence of short lattice vectors by using the existence of short vectors in a special class of higher-dimensional integer lattices.
  - Let distinct $v_1,\dots,v_m \in \bar{L}=q^{-1}L$.
  - Consider the integer lattice L' formed by all $(x_1,\dots,x_m)\in\mathbf{Z}^m$ s.t. $\sum_i x_i v_i \in L$.
    - If $m>q^d$, $\lambda_1(L')\le\sqrt{2}$.

An Algorithm From Mordell's Proof

- Mordell's proof gives an (inefficient) algorithm:
  - Need to generate $>q^d$ lattice points in $\bar{L}$.
  - Among these exponentially many lattice points, find a difference in L, possibly by exhaustive search.
  - Both steps are expensive.
- [BGJ14] and [ADRS15] are more efficient randomized variants of Mordell's algorithm: sampling over $\bar{L}$ may allow to sample over L.

Sieve algorithms [AKS01,ADRS15]

- Initially, generate long random vectors.
- Using sieving, reduce iteratively the « average » norm of the distribution.
- After a while, the shortest vector can be extracted: the running time is $2^{O(d)}$.
- [ADRS15] uses the discrete Gaussian distribution and $\bar{L}=L/2$.
- [BGJ14] is somewhat a more efficient heuristic version of [ADRS15], by using a pool of vectors.

Wishful Thinking

- To apply the pigeon-hole principle, we need an exponential number m of lattice vectors in $\underset{\cdot}{\bar{L}}$.
- Can we get away with a small polynomial number m and make the algorithm efficient? (unlike [BGJ14] and [ADRS15])
  - Maybe if we could find short vectors in certain higher-dimensional random lattices.

Worst-case to Average-case Reductions from Mordell's Proof

The SIS Problem (1996): Small Integer Solutions

- Let $(G,+)$ be a finite Abelian group: $G=(\mathbf{Z}/q\mathbf{Z})^n$ in [Ajtai96]. View G as a $\mathbf{Z}$-module.
- Pick $g_1,\dots,g_m$ uniformly at random from G.
- Goal: Find short $(x_1,\dots,x_m)\in\mathbf{Z}^m$ s.t. $\sum_i x_i g_i = 0$, e.g. $\|x\| \leq m\,(\#G)^{1/m}$.
- This is essentially finding a short vector in a (uniform) random lattice of $L_m(G) = \{$ lattices $L\subseteq\mathbf{Z}^m$ s.t. $\mathbf{Z}^m/L \sim G$ $\}$.

Ex: Cyclic G

- Let $G = \mathbf{Z}/q\mathbf{Z}$
- Pick $g_1, \ldots, g_m$ uniformly at random mod q.
- Goal: Find short $x = (x_1, \ldots, x_m) \in \mathbf{Z}^m$ s.t. $\sum_i x_i g_i \equiv 0 \pmod{q}$.

Worst-case to Average-case Reduction

- [Ajtai96]: If one can efficiently solve SIS for $G=(\mathbf{Z}/q_n\mathbf{Z})^n$ on the average, then one can efficiently find short vectors in every n-dim lattice.
- [GINX16]: This can be generalized to any sequence (G_n) of finite abelian groups, provided that $\#G_n$ is sufficiently large $\geq n^{\Omega(\max(n,\mathrm{rank}(G)))}$ and m too. Ex: $(\mathbf{Z}/2\mathbf{Z})^n$ is not.

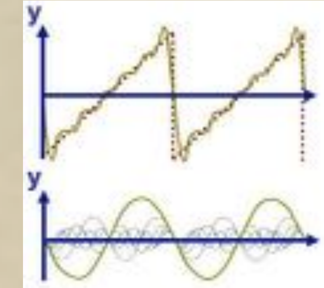
Overlattices and Groups

- If L is n-dim, $\bar{L}=q^{-1}L$ and $G=(\mathbf{Z}/q\mathbf{Z})^n$ then $\bar{L}/L \simeq G$.
- There is an exact sequence:

$$0 \to L \xrightarrow{1} \bar{L} \xrightarrow{\varphi} G \to 0$$

- $L=\mathrm{Ker}\,\phi$ where ϕ is efficiently computable.
- Let $v_1,\ldots,v_m \in \bar{L}$ and define $g_1,\ldots,g_m \in G$ by $g_i = \phi(v_i)$.
- If $\sum_i x_i g_i = 0$ for $(x_1,\ldots,x_m) \in \mathbf{Z}^m$ then $\sum_i x_i v_i \in L$.

Fourier Analysis



- Fourier analysis shows that if $v_1,\dots,v_m \in \bar{L}$ are chosen from a suitable (short) distribution, $g_i = \phi(v_i)$ has uniform distribution over G.
  - Any probability mass function f over $\bar{L}$ s.t. for any $x \in \bar{L}$, $\Sigma_{y \in L} f(x+y) \approx 1/\#G$. Ex: discrete Gaussian distribution.
- This is a key step: transforming a worst-case into an average-case.

Worst-to-average Reduction from Mordell's Proof

- Sample short $v_1,\dots,v_m \in \bar{L}$ from a suitable distribution, so that $g_i = \phi(v_i)$ has uniform distrib. over $G = (\mathbf{Z}/q\mathbf{Z})^n$
- Call the SIS-oracle on $(g_1,\dots,g_m)$ to find a short $x = (x_1,\dots,x_m) \in \mathbf{Z}^m$ s.t. $\sum_i x_i g_i = 0$ in G, i.e. $\sum_i x_i v_i \in L$.
- Return $\sum_i x_i v_i \in L$.

Generalized SIS Reduction

- The SIS reduction is based on this crucial fact: If B is a reduced basis of a lattice L, then $q^{-1}B$ is a reduced basis of the overlattice $\bar{L}=q^{-1}L$.
- But if G is an arbitrary finite Abelian group, we need to find a reduced basis of some overlattice $\bar{L}\supseteq L$ s.t. $\bar{L}/L \simeq G$, so that we can sample short vectors in $\bar{L}$.

Structural Lattice Reduction

- In classical lattice reduction, we try to find a good basis of a given lattice.
- In structural lattice reduction [GINX16], given a lattice L and a (sufficiently large) finite Abelian group G, we find a good basis of some overlattice $\bar{L}$ s.t. $\bar{L}/L \simeq G$.
  - Directly using backwards-LLL.
  - Or by reduction to the case $L=\mathbf{Z}^n$.

Easy Cases

- If $G=(\mathbf{Z}/q\mathbf{Z})^n$, any basis B of a full-rank lattice L in $\mathbf{Z}^n$ can be transformed into a basis $q^{-1}B$ of $\bar{L}=q^{-1}L$, which is $q=\#G^{1/n}$ times shorter.
- If $G=\mathbf{Z}^n/L$, the canonical basis of $\bar{L} = \mathbf{Z}^n$ is a short basis, typically $\#G^{1/n}$ times shorter than a short basis of L.

LWE:
A Dual Worst-case to Average-Case Reduction

Duality

- Remember the SIS lattice:
  - $g_1,\dots,g_m$ in some finite Abelian group $(G,+)$
  - $L=\{x=(x_1,\dots,x_m)\in\mathbf{Z}^m$ s.t. $\sum_i x_i\, g_i = 0\}$
- The dual lattice of L is related to the dual group $G^\vee$ of (additive) characters of G: morphisms from G to $\mathbf{T}=\mathbf{R}/\mathbf{Z}$
  - $L^\vee=\{(y_1,\dots,y_m)\in\mathbf{R}^m$ s.t. for some $s\in G^\vee$, for all i $y_i \equiv s(g_i) \pmod 1\}$

The LWE Problem: Learning (a Character) with Errors

- Let $(G,+)$ be any finite Abelian group e.g. $G=(\mathbf{Z}/q\mathbf{Z})^n$ in [Re05].
- Pick $g_1,\ldots,g_m$ uniformly at random from G.
- Pick a random character s in $G^{\vee}$.
- Goal: recover s given $g_1,\ldots,g_m$ and noisy approximations of $s(g_1),\ldots, s(g_m)$. Ex: Gaussian noise.

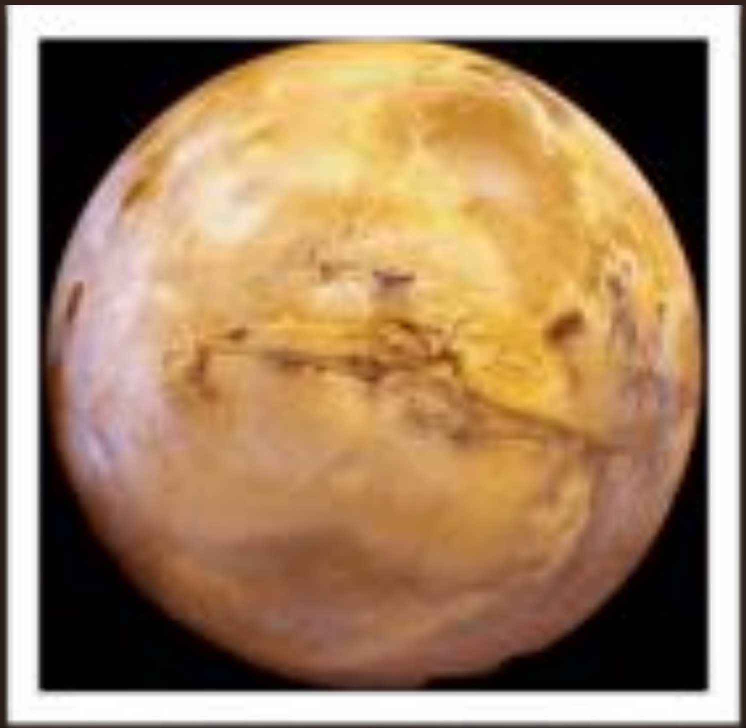
Ex: Cyclic G

- Let $G = \mathbf{Z}/q\mathbf{Z}$
- Pick $g_1,\ldots,g_m$ uniformly at random mod q.
- Goal: recover $s \in \mathbf{Z}$ given $g_1,\ldots,g_m$ and randomized approximations of sg_1 mod q,…, sg_m mod q.
- This is exactly a randomized variant of Boneh-Venkatesan's Hidden Number Problem from CRYPTO '96.

Hardness of LWE

- [Regev05]: If one can efficiently solve LWE for $G=(\mathbf{Z}/q_n\mathbf{Z})^n$ on the average, then one can quantum-efficiently find short vectors in every n-dim lattice.
- [GINX16]: This can be generalized to any sequence (G_n) of finite abelian groups, provided that $\#G_n$ is sufficiently large.

Conclusion



More Inequalities

- All known upper bounds on Hermite's constant have an algorithmic version.
- Is there a polynomial bound on Hermite's constant, possibly worse than Minkowski's inequality, but with a more efficient algorithmic version?

Thank you for your attention...

Any question(s)?

References

- [GINX16]: « Structural Lattice Reduction: Generalized Worst-Case to Average-Case Reductions and Homomorphic Cryptosystems », EUROCRYPT '16, full version on eprint.
- [N10]: « Hermite's constant and lattice algorithms » survey in the LLL+25 book.