Computational complexity of lattice problems and cyclic lattices

Lenny Fukshansky
Claremont McKenna College

Undergraduate Summer Research Program
ICERM - Brown University
July 28, 2014

Euclidean lattices

A **lattice** in Euclidean space $\mathbb{R}^n$ is a nonzero discrete subgroup. If $\Lambda \subset \mathbb{R}^n$ is a lattice, then there exist $\mathbb{R}$-linearly independent vectors

$$\mathbf{a}_1, \ldots, \mathbf{a}_k \in \Lambda, \ 1 \leq k \leq n,$$

called a **basis** for Λ, such that

$$\Lambda = \left\{ \sum_{i=1}^{k} m_i \mathbf{a}_i : m_i \in \mathbb{Z} \right\} = A\mathbb{Z}^k,$$

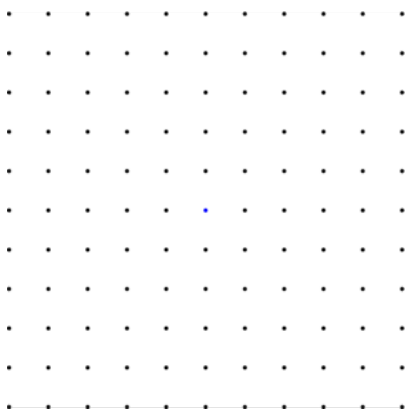
where

$$A = (\mathbf{a}_1 \ldots \mathbf{a}_k)$$

is the corresponding $n \times k$ **basis matrix**. Then k is called the **rank** of Λ, and $k = n$ if and only if the quotient group $\mathbb{R}^n/\Lambda$ is compact.

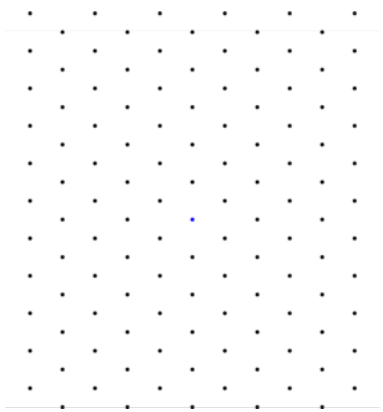
Examples of lattices in the plane

Square lattice



$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbb{Z}^2$$

Hexagonal lattice



$$\begin{pmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix} \mathbb{Z}^2$$

Determinant of a lattice

Determinant or **covolume** of a lattice $\Lambda = A\mathbb{Z}^k \subset \mathbb{R}^n$ is

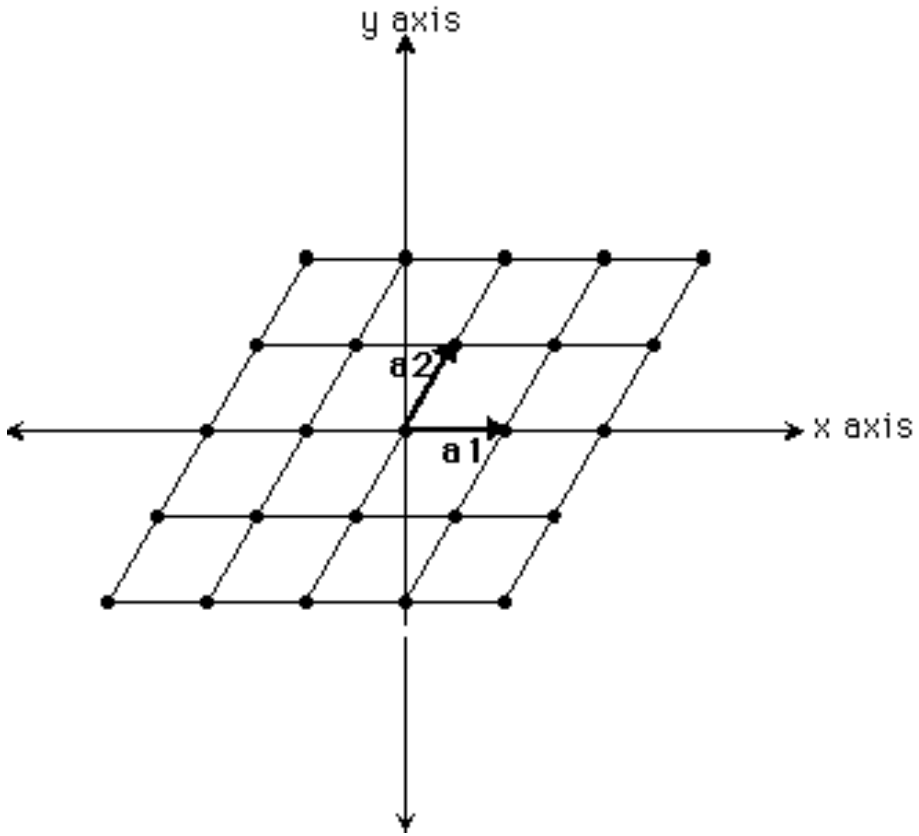
$$\sqrt{\det(A^t A)}.$$

This is equal to the volume of the compact quotient V/Λ, where

$$V = \operatorname{span}_{\mathbb{R}} \Lambda$$

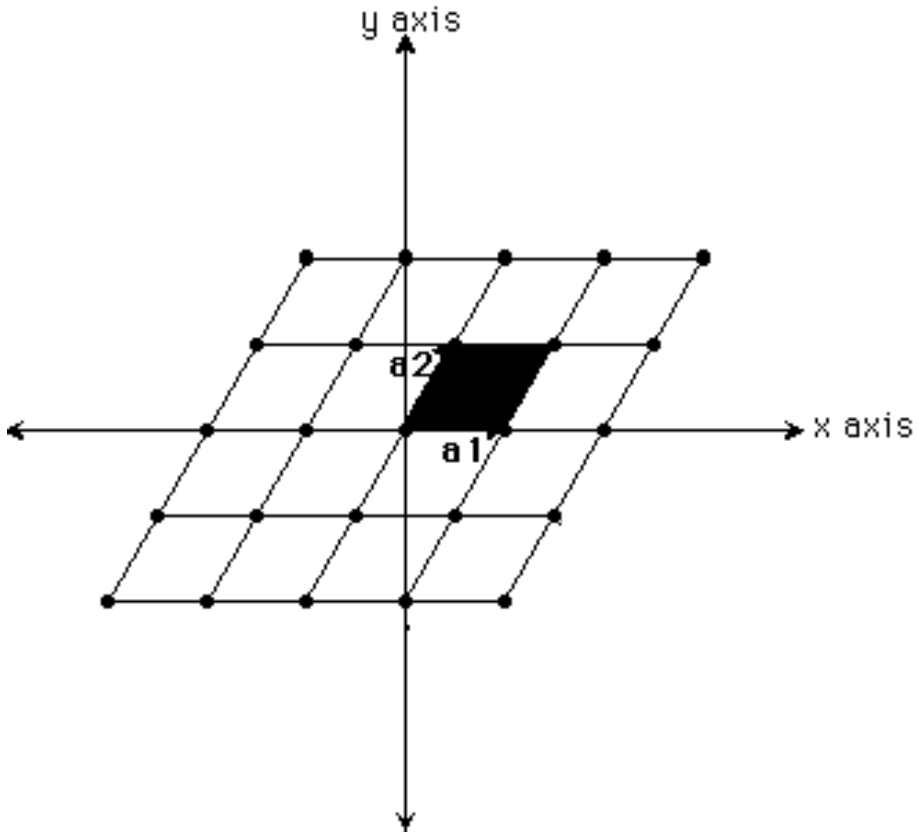
is a k-dimensional subspace of $\mathbb{R}^n$.

Example of a fundamental domain



Hexagonal lattice fundamental domain

Example of a fundamental domain



$$\text{Volume} = \det \begin{pmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix}$$

Successive minima

Let $\mathbb{B}_n$ be a unit ball centered at the origin in $\mathbb{R}^n$. If $\Lambda \subset \mathbb{R}^n$ is a lattice of rank k, then its **successive minima**

$$0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k$$

are real numbers such that

$$\lambda_i \mathbb{B}_n \cap \Lambda$$

contains at least i linearly independent vectors for each $1 \leq i \leq k$ – we call these the **vectors corresponding to successive minima**. They are not necessarily unique, but there are finitely many of them.

Important remark

Vectors corresponding to successive minima do not necessarily form a basis for the lattice. For instance, the 5-dimensional lattice

$$\Lambda = \begin{pmatrix} 1 & 0 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 \end{pmatrix} \mathbb{Z}^5$$

contains the standard basis vectors $\mathbf{e}_1, \ldots, \mathbf{e}_5$, and hence

$$\lambda_1 = \cdots = \lambda_5 = 1,$$

however these vectors do not span Λ over $\mathbb{Z}$.

Lattice problems

This is a class of algorithmic optimization problems on lattices. We will consider two famous examples.

Lattice problems

This is a class of algorithmic optimization problems on lattices. We will consider two famous examples.

Definition 1 (Shortest Vector Problem – SVP)

Input: An $n \times n$ basis matrix A for a lattice $\Lambda = A\mathbb{Z}^n \subset \mathbb{R}^n$.
Output: A shortest nonzero vector in Λ, i.e. $\mathbf{x} \in \Lambda$ such that

$$\|\mathbf{x}\| = \min\{\|\mathbf{y}\| : \mathbf{y} \in \Lambda \setminus \{\mathbf{0}\}\},$$

where $\| \ \|$ is Euclidean norm.

Lattice problems

This is a class of algorithmic optimization problems on lattices. We will consider two famous examples.

Definition 1 (Shortest Vector Problem – SVP)

Input: An $n \times n$ basis matrix A for a lattice $\Lambda = A\mathbb{Z}^n \subset \mathbb{R}^n$.
Output: A shortest nonzero vector in Λ, i.e. $\mathbf{x} \in \Lambda$ such that

$$\|\mathbf{x}\| = \min\{\|\mathbf{y}\| : \mathbf{y} \in \Lambda \setminus \{\mathbf{0}\}\},$$

where $\| \ \|$ is Euclidean norm.

Remark 1

This is precisely a vector corresponding to λ_1, the first successive minimum.

Lattice problems

Definition 2 (Shortest Independent Vector Problem – SIVP)

Input: An $n \times n$ basis matrix A for a lattice $\Lambda = A\mathbb{Z}^n \subset \mathbb{R}^n$.
Output: A collection of n shortest linearly independent vectors in Λ, i.e. linearly independent $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \Lambda$ such that

$$\|\mathbf{x}_i\| = \lambda_i.$$

Lattice problems

Definition 2 (Shortest Independent Vector Problem – SIVP)

Input: An $n \times n$ basis matrix A for a lattice $\Lambda = A\mathbb{Z}^n \subset \mathbb{R}^n$.
Output: A collection of n shortest linearly independent vectors in Λ, i.e. linearly independent $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \Lambda$ such that

$$\|\mathbf{x}_i\| = \lambda_i.$$

Clearly SIVP should generally be harder than SVP.

Lattice problems

Definition 2 (Shortest Independent Vector Problem – SIVP)

Input: An $n \times n$ basis matrix A for a lattice $\Lambda = A\mathbb{Z}^n \subset \mathbb{R}^n$.
Output: A collection of n shortest linearly independent vectors in Λ, i.e. linearly independent $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \Lambda$ such that

$$\|\mathbf{x}_i\| = \lambda_i.$$

Clearly SIVP should generally be harder than SVP.

Question 1

How much harder?

Lattice problems

Definition 2 (Shortest Independent Vector Problem – SIVP)

Input: An $n \times n$ basis matrix A for a lattice $\Lambda = A\mathbb{Z}^n \subset \mathbb{R}^n$.
Output: A collection of n shortest linearly independent vectors in Λ, i.e. linearly independent $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \Lambda$ such that

$$\|\mathbf{x}_i\| = \lambda_i.$$

Clearly SIVP should generally be harder than SVP.

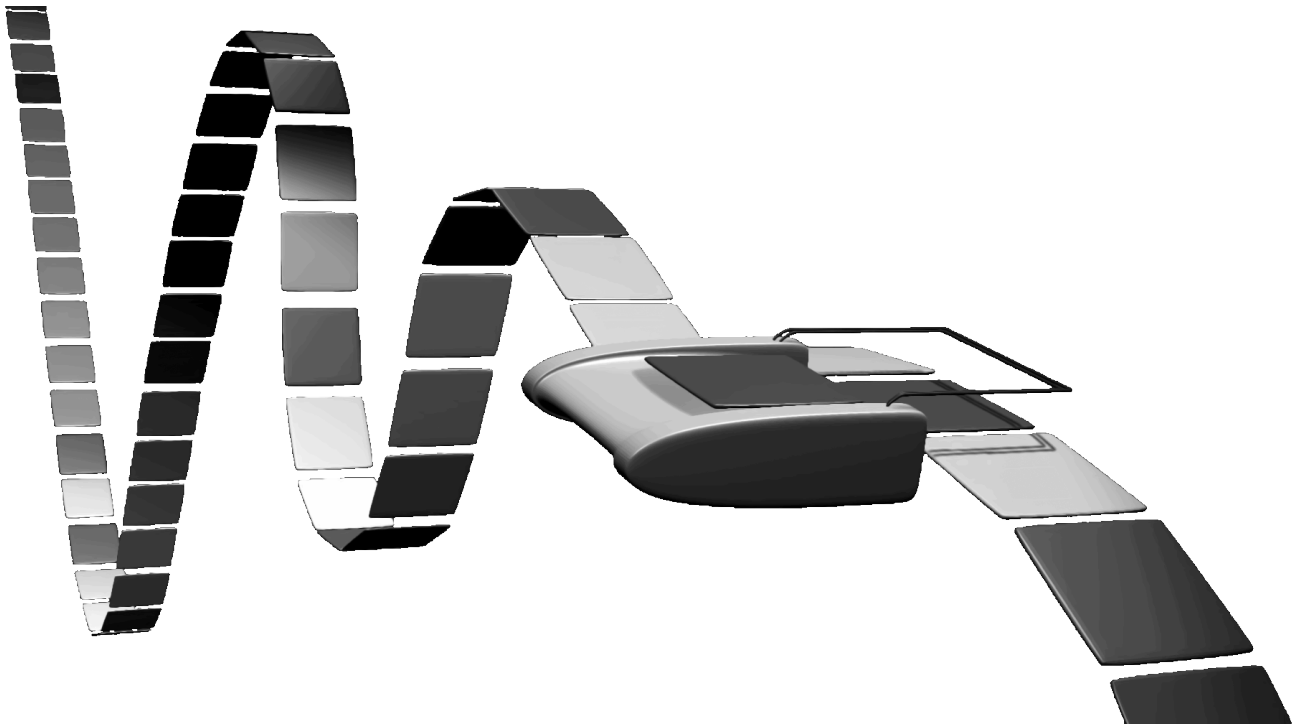
Question 1

How much harder?

To answer this question, we need to explain how we measure "hardness".

Turing machine

Device with a **head** and an infinite **tape** going through it:



Elementary operations: read 1 cell, write 1 cell, move tape left 1 cell, move tape right 1 cell.

Example: a modern computer

Complexity classes: **P** and **NP**

Given an algorithmic problem, we can measure the size of its input in number of bits of memory it takes to store it.

Complexity classes: **P** and **NP**

Given an algorithmic problem, we can measure the size of its input in number of bits of memory it takes to store it.

Definition 3

A problem is called **polynomial** if the number of elementary operations required to solve it on a Turing machine is polynomial in the size of the input. If this is the case, we say that the problem can be **solved in polynomial time**. The class of all such problems is denoted by **P**.

Complexity classes: **P** and **NP**

Given an algorithmic problem, we can measure the size of its input in number of bits of memory it takes to store it.

Definition 3

A problem is called **polynomial** if the number of elementary operations required to solve it on a Turing machine is polynomial in the size of the input. If this is the case, we say that the problem can be **solved in polynomial time**. The class of all such problems is denoted by **P**.

Definition 4

A problem is called **non-deterministic polynomial** if the number of elementary operations required to verify a potential answer for it on a Turing machine is polynomial in the size of the input. If this is the case, we say that the problem can be **verified in polynomial time**. The class of all such problems is denoted by **NP**.

More complexity: **NP**-hard and **NP**-complete

It is clear that every problem which can be solved in polynomial time, can be verified in polynomial time, and so

$$\mathbf{P} \subseteq \mathbf{NP}.$$

More complexity: **NP**-hard and **NP**-complete

It is clear that every problem which can be solved in polynomial time, can be verified in polynomial time, and so

$$\mathbf{P} \subseteq \mathbf{NP}.$$

Definition 5

Informally speaking, a problem is called **NP-hard** if it is at least as hard as the hardest problem in **NP**. An **NP**-hard problem does not need to be in **NP**.

More complexity: **NP**-hard and **NP**-complete

It is clear that every problem which can be solved in polynomial time, can be verified in polynomial time, and so

$$\mathbf{P} \subseteq \mathbf{NP}.$$

Definition 5

Informally speaking, a problem is called **NP-hard** if it is at least as hard as the hardest problem in **NP**. An **NP**-hard problem does not need to be in **NP**.

Definition 6

A problem is called **NP-complete** if it is in **NP** and is **NP**-hard.

P vs **NP**: a million dollar problem

One of the seven Clay Millenium Prize Problems is the question whether

$$\mathbf{P} = \mathbf{NP}?$$

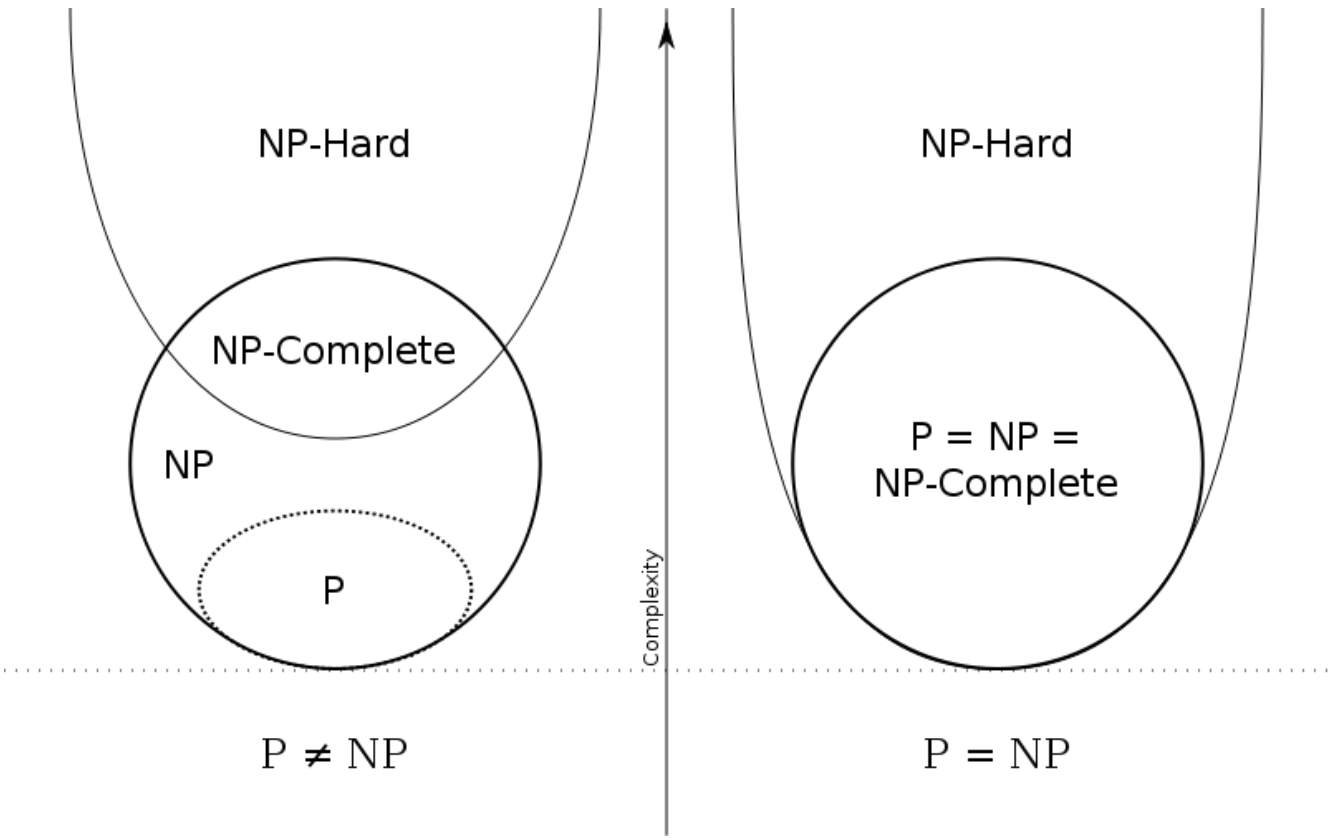
The problem was first posed in 1971 independently by Stephen Cook and Leonid Levin.

P vs **NP**: a million dollar problem

One of the seven Clay Millenium Prize Problems is the question whether

$$\mathbf{P} = \mathbf{NP}?$$

The problem was first posed in 1971 independently by Stephen Cook and Leonid Levin.



Complexity of lattice problems

SVP and SIVP are both known to be **NP**-hard. In fact, even the problem of finding the first successive minimum λ_1 (respectively, all successive minima $\lambda_1, \ldots, \lambda_n$) of a given lattice is **NP**-hard: it is as hard as SVP (respectively, SIVP).

Complexity of lattice problems

SVP and SIVP are both known to be **NP**-hard. In fact, even the problem of finding the first successive minimum λ_1 (respectively, all successive minima $\lambda_1, \ldots, \lambda_n$) of a given lattice is **NP**-hard: it is as hard as SVP (respectively, SIVP). Moreover –

Theorem 1 (SIVP to SVP reduction)

For lattices of rank n, there exists a polynomial time reduction algorithm that, given an oracle for SVP, produces an approximate solution to SIVP within an approximation factor of $\sqrt{n}$ – that is, a collection of linearly independent vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \in \Lambda$ with

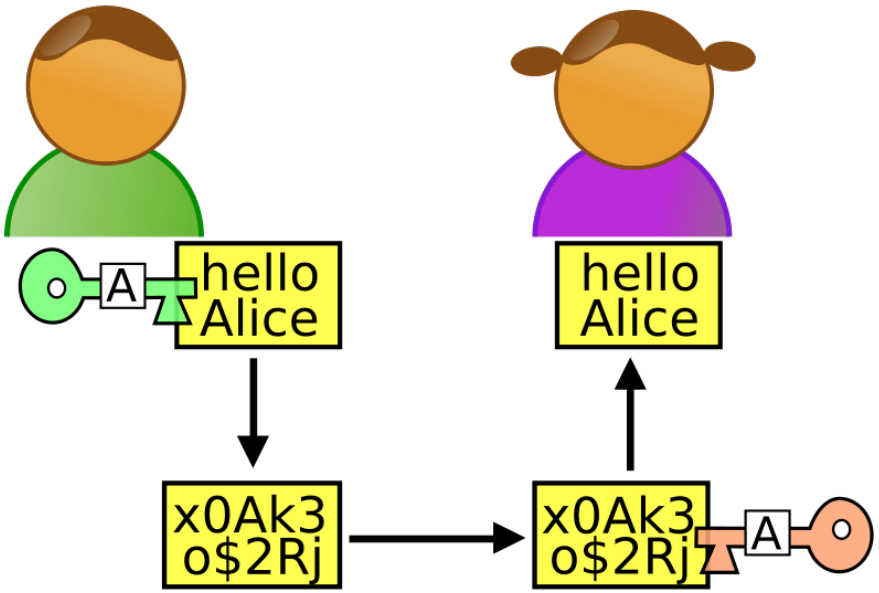
$$\|\mathbf{a}_1\| \leq \|\mathbf{a}_2\| \leq \cdots \leq \|\mathbf{a}_n\| \leq \sqrt{n}\lambda_n.$$

Hard is good: cryptography connection

One of the main applications of lattice problems is **cryptography**.

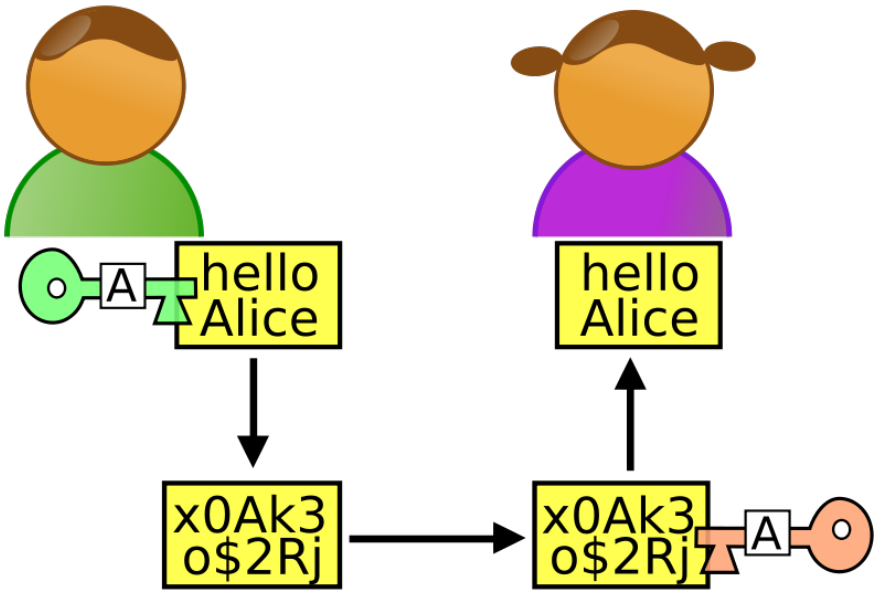
Hard is good: cryptography connection

One of the main applications of lattice problems is **cryptography**.



Hard is good: cryptography connection

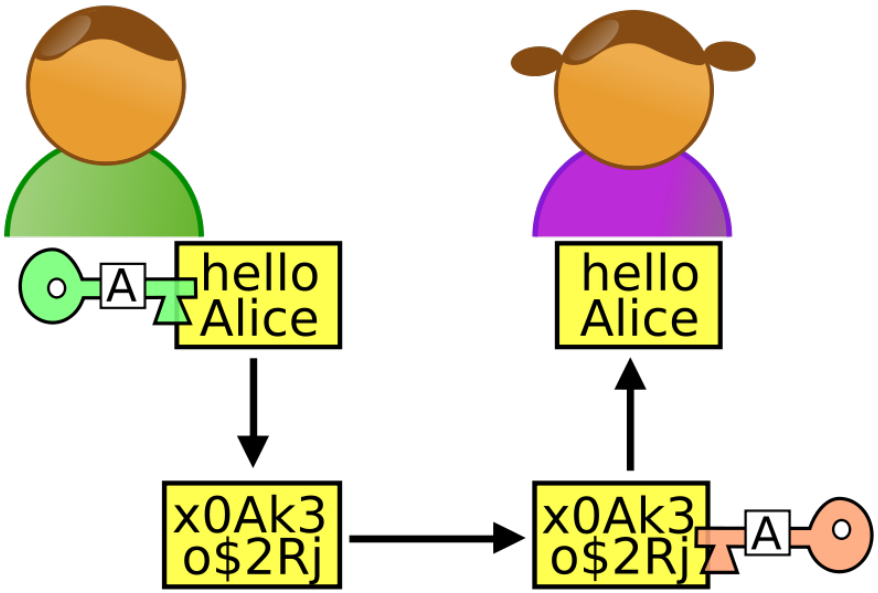
One of the main applications of lattice problems is **cryptography**.



Encryption algorithm is usually based on a very hard problem.

Hard is good: cryptography connection

One of the main applications of lattice problems is **cryptography**.



Encryption algorithm is usually based on a very hard problem.

Some possible choices: SVP, SIVP.

Encryption challenge

A lattice-based cryptographic algorithm takes a basis matrix for a lattice on the input.

Encryption challenge

A lattice-based cryptographic algorithm takes a basis matrix for a lattice on the input. If $\Lambda \subset \mathbb{R}^n$ has rank n, then the input size is n^2.

Encryption challenge

A lattice-based cryptographic algorithm takes a basis matrix for a lattice on the input. If $\Lambda \subset \mathbb{R}^n$ has rank n, then the input size is n^2. In order to make the message hard to decrypt for a hostile attacker, n should be large.

Encryption challenge

A lattice-based cryptographic algorithm takes a basis matrix for a lattice on the input. If $\Lambda \subset \mathbb{R}^n$ has rank n, then the input size is n^2. In order to make the message hard to decrypt for a hostile attacker, n should be large. But large size input slows down the algorithm.

Encryption challenge

A lattice-based cryptographic algorithm takes a basis matrix for a lattice on the input. If $\Lambda \subset \mathbb{R}^n$ has rank n, then the input size is n^2. In order to make the message hard to decrypt for a hostile attacker, n should be large. But large size input slows down the algorithm.

Question 2

Are there lattices which can be described by the input data of size less than n^2?

Cyclic lattices: definition

Define the **rotational shift operator** on $\mathbb{R}^n$, $n \geq 2$, by

$$\mathrm{rot}(x_1, x_2, \ldots, x_{n-1}, x_n) = (x_n, x_1, x_2, \ldots, x_{n-1})$$

for every $\mathbf{x} = (x_1, x_2, \ldots, x_{n-1}, x_n) \in \mathbb{R}^n$. We will write rot^k for iterated application of rot k times for each $k \in \mathbb{Z}_{>0}$ (then rot^0 is just the identity map, and $\mathrm{rot}^k = \mathrm{rot}^{n+k}$). It is also easy to see that rot (and hence each iteration rot^k) is a linear operator. A sublattice Γ of $\mathbb{Z}^n$ is called **cyclic** if $\mathrm{rot}(\Gamma) = \Gamma$, i.e. if for every $\mathbf{x} \in \Gamma$, $\mathrm{rot}(\mathbf{x}) \in \Gamma$. Clearly, $\mathbb{Z}^n$ itself is a cyclic lattice.

Cyclic lattices from ideals in $\mathbb{Z}[x]/(x^n - 1)$

Let

$$p(x) = \sum_{k=0}^{n-1} a_k x^k \in \mathbb{Z}[x]/(x^n - 1).$$

Define a map $\rho : \mathbb{Z}[x]/(x^n - 1) \to \mathbb{Z}^n$ by

$$\rho(p(x)) = (a_0, \ldots, a_{n-1}) \in \mathbb{Z}^n,$$

then for any ideal $I \subseteq \mathbb{Z}[x]/(x^n - 1)$, $\rho(I)$ is a sublattice of $\mathbb{Z}^n$ of full rank. Notice that for every $p(x) \in I$,

$$xp(x) = a_{n-1} + a_0 x + a_1 x^2 + \cdots + a_{n-2} x^{n-1} \in I,$$

and so

$$\rho(xp(x)) = (a_{n-1}, a_0, a_1, \ldots, a_{n-2}) = \mathrm{rot}(\rho(p(x))) \in \rho(I).$$

In other words, $\Gamma \subseteq \mathbb{Z}^n$ is a cyclic lattice if and only if $\Gamma = \rho(I)$ for some ideal $I \subseteq \mathbb{Z}[x]/(x^n - 1)$.

Cyclic lattices in cryptosystems

Cyclic lattices were formally introduced for cryptographic use by D. Micciancio in 2002, but "in disguise" they were already used earlier.

Cyclic lattices in cryptosystems

Cyclic lattices were formally introduced for cryptographic use by D. Micciancio in 2002, but "in disguise" they were already used earlier.

The **NTRUEncrypt** public key cryptosystem was introduced in 1996 by J. Hoffstein, J. Pipher, and J. H. Silverman at Brown University.

Cyclic lattices in cryptosystems

Cyclic lattices were formally introduced for cryptographic use by D. Micciancio in 2002, but "in disguise" they were already used earlier.

The **NTRUEncrypt** public key cryptosystem was introduced in 1996 by J. Hoffstein, J. Pipher, and J. H. Silverman at Brown University.

NTRUE is based on difficulty of factoring polynomials in the ring $\mathbb{Z}[x]/(x^n - 1)$, which is closely related to **lattice reduction**, i.e., solving SVP, SIVP on cyclic lattices.

Cyclic lattices in cryptosystems

Cyclic lattices were formally introduced for cryptographic use by D. Micciancio in 2002, but "in disguise" they were already used earlier.

The **NTRUEncrypt** public key cryptosystem was introduced in 1996 by J. Hoffstein, J. Pipher, and J. H. Silverman at Brown University.

NTRUE is based on difficulty of factoring polynomials in the ring $\mathbb{Z}[x]/(x^n - 1)$, which is closely related to **lattice reduction**, i.e., solving SVP, SIVP on cyclic lattices.

This motivates studying cyclic lattices more in depth.

Cyclic lattices: basic properties - 1

Definition 7

For a vector $\mathbf{a} \in \mathbb{Z}^n$, define

$$\Lambda(\mathbf{a}) = \text{span}_{\mathbb{Z}} \left\{ \mathbf{a}, \text{rot}(\mathbf{a}), \ldots, \text{rot}^{n-1}(\mathbf{a}) \right\}.$$

This is always a cyclic lattice.

Cyclic lattices: basic properties - 1

Definition 7

For a vector $\mathbf{a} \in \mathbb{Z}^n$, define

$$\Lambda(\mathbf{a}) = \text{span}_{\mathbb{Z}} \left\{ \mathbf{a}, \text{rot}(\mathbf{a}), \ldots, \text{rot}^{n-1}(\mathbf{a}) \right\}.$$

This is always a cyclic lattice.

Question 3

What is the rank of $\Lambda(\mathbf{a})$*?*

Cyclic lattices: basic properties - 1

Definition 7

For a vector $\mathbf{a} \in \mathbb{Z}^n$, define

$$\Lambda(\mathbf{a}) = \text{span}_{\mathbb{Z}} \left\{ \mathbf{a}, \text{rot}(\mathbf{a}), \ldots, \text{rot}^{n-1}(\mathbf{a}) \right\}.$$

This is always a cyclic lattice.

Question 3

What is the rank of $\Lambda(\mathbf{a})$?

Lemma 2

Let $\mathbf{a} \in \mathbb{Z}^n$ and let $p_{\mathbf{a}}(x) \in \mathbb{Z}[x]/(x^n - 1)$ be a polynomial with coefficient vector $\mathbf{a}$. Then $\mathbf{a}, \text{rot}(\mathbf{a}), \ldots, \text{rot}^{n-1}(\mathbf{a})$ are linearly dependent if and only if $p_{\mathbf{a}}(x)$ is divisible by some cyclotomic polynomial divisor of $x^n - 1$.

Cyclic lattices: basic properties - 2

Let

$$C_R^n = \{\mathbf{x} \in \mathbb{R}^n : |\mathbf{x}| := \max\{|x_1|, \ldots, |x_n|\} \leq R\}$$

for every $R \in \mathbb{R}_{>0}$, i.e. C_R^n is a cube of side-length $2R$ centered at the origin in $\mathbb{R}^n$.

Cyclic lattices: basic properties - 2

Let

$$C_R^n = \{\mathbf{x} \in \mathbb{R}^n : |\mathbf{x}| := \max\{|x_1|, \ldots, |x_n|\} \leq R\}$$

for every $R \in \mathbb{R}_{>0}$, i.e. C_R^n is a cube of side-length $2R$ centered at the origin in $\mathbb{R}^n$.

Lemma 3

Let $R > \frac{n-1}{2}$, *then*

$$Prob_{\infty,R}\left(\mathrm{rk}(\Lambda(\mathbf{a})) = n\right) \geq 1 - \frac{n}{2R+1},$$

where probability $\mathrm{Prob}_{\infty,R}(\cdot)$ *is with respect to the uniform distribution among all points* $\mathbf{a}$ *in the set* $C_R^n \cap \mathbb{Z}^n$.

Cyclic lattices: cryptographic use

Hence if we pick $\mathbf{a} \in \mathbb{Z}^n$ with large $|\mathbf{a}|$, the probability that

$$\mathrm{rk}(\Lambda(\mathbf{a})) = n$$

is high, and the size of the input data necessary to describe this lattice is only n. This observation makes cyclic lattices very attractive for cryptographic purposes.

Cyclic lattices: cryptographic use

Hence if we pick $\mathbf{a} \in \mathbb{Z}^n$ with large $|\mathbf{a}|$, the probability that

$$\mathrm{rk}(\Lambda(\mathbf{a})) = n$$

is high, and the size of the input data necessary to describe this lattice is only n. This observation makes cyclic lattices very attractive for cryptographic purposes.

Question 4

But are cyclic lattices hard enough? In other words, are SVP, SIVP still **NP***-hard on cyclic lattices?*

Cyclic lattices: cryptographic use

Hence if we pick $\mathbf{a} \in \mathbb{Z}^n$ with large $|\mathbf{a}|$, the probability that

$$\mathrm{rk}(\Lambda(\mathbf{a})) = n$$

is high, and the size of the input data necessary to describe this lattice is only n. This observation makes cyclic lattices very attractive for cryptographic purposes.

Question 4

But are cyclic lattices hard enough? In other words, are SVP, SIVP still **NP***-hard on cyclic lattices?*

This is an open question, but many people believe that the answer is **yes**, at least in the worst case.

SIVP to SVP on cyclic lattices

On the other hand, there is some indication that SIVP is at least easier on cyclic lattices than on generic lattices.

SIVP to SVP on cyclic lattices

On the other hand, there is some indication that SIVP is at least easier on cyclic lattices than on generic lattices.

Theorem 4 (Peikert, Rosen (2005))

Let n be a **prime** *and let $\Lambda \subset \mathbb{R}^n$ be a lattice of rank n. There exists a polynomial time algorithm that, given an oracle for SVP, produces an approximate solution to SIVP on Λ within an approximation factor of 2. In other words, given $\mathbf{a}_1 \in \Lambda$ with $\|\mathbf{a}_1\| = \lambda_1$ we can find a collection of linearly independent vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \in \Lambda$ with*

$$\|\mathbf{a}_1\| \leq \|\mathbf{a}_2\| \leq \cdots \leq \|\mathbf{a}_n\| \leq 2\lambda_n$$

polynomial time. Moreover, only one call to the oracle is necessary.

Well-rounded lattices

More generally, we can show that for **every** n, SIVP is **equivalent** to SVP on a positive proportion of cyclic lattices. To explain what this means, we need more notation.

Well-rounded lattices

More generally, we can show that for **every** n, SIVP is **equivalent** to SVP on a positive proportion of cyclic lattices. To explain what this means, we need more notation.

A lattice $\Gamma \subset \mathbb{R}^n$ of rank n is called **well-rounded** (abbreviated WR) if

$$\lambda_1(\Gamma) = \cdots = \lambda_n(\Gamma).$$

Well-rounded lattices

More generally, we can show that for **every** n, SIVP is **equivalent** to SVP on a positive proportion of cyclic lattices. To explain what this means, we need more notation.

A lattice $\Gamma \subset \mathbb{R}^n$ of rank n is called **well-rounded** (abbreviated WR) if

$$\lambda_1(\Gamma) = \cdots = \lambda_n(\Gamma).$$

Notice that for a WR lattice, finding λ_1 is equivalent to finding all successive minima.

WR cyclic lattices

Let $\mathcal{C}_n$ be the set of all full rank cyclic sublattices of $\mathbb{Z}^n$.

WR cyclic lattices

Let $\mathcal{C}_n$ be the set of all full rank cyclic sublattices of $\mathbb{Z}^n$.

Question 5

Which lattices in $\mathcal{C}_n$ are WR?

WR cyclic lattices

Let $\mathcal{C}_n$ be the set of all full rank cyclic sublattices of $\mathbb{Z}^n$.

Question 5

Which lattices in $\mathcal{C}_n$ are WR?

Theorem 5 (F., Sun (2013))

For each dimension $n \geq 2$, there exist real constants

$$0 < \alpha_n \leq \beta_n \leq 1,$$

depending only on n, such that

$$\alpha_n \leq \frac{\#\{\Gamma \in \mathcal{C}_n : \lambda_n(\Gamma) \leq R,\ \Gamma \text{ is WR}\}}{\#\{\Gamma \in \mathcal{C}_n : \lambda_n(\Gamma) \leq R\}} \leq \beta_n \text{ as } R \to \infty. \quad (1)$$

For instance, one can take $\alpha_2 = 0.261386...$ and $\beta_2 = 0.348652...$, meaning that between 26% and 35% of full rank cyclic sublattices of $\mathbb{Z}^2$ are WR.

SVP - SIVP equivalence

We prove that SVP and SIVP are equivalent on a positive proportion of WR cyclic lattices in every dimension, hence -

SVP - SIVP equivalence

We prove that SVP and SIVP are equivalent on a positive proportion of WR cyclic lattices in every dimension, hence -

Corollary 6 (F., Sun (2013))

Let $R \in \mathbb{R}_{>0}$, *then*

$$\frac{\#\{\Gamma \in \mathcal{C}_n : \lambda_n(\Gamma) \leq R,\ SVP \equiv SIVP \text{ on } \Gamma\}}{\#\{\Gamma \in \mathcal{C}_n : \lambda_n(\Gamma) \leq R\}} \gg_n 1 \text{ as } R \to \infty.$$

SVP - SIVP equivalence

We prove that SVP and SIVP are equivalent on a positive proportion of WR cyclic lattices in every dimension, hence -

Corollary 6 (F., Sun (2013))

Let $R \in \mathbb{R}_{>0}$, *then*

$$\frac{\#\{\Gamma \in \mathcal{C}_n : \lambda_n(\Gamma) \leq R,\ SVP \equiv SIVP \text{ on } \Gamma\}}{\#\{\Gamma \in \mathcal{C}_n : \lambda_n(\Gamma) \leq R\}} \gg_n 1 \text{ as } R \to \infty.$$

Corollary 7 (F., Sun (2013))

Let $k_1, \ldots, k_{n-1} \in \mathbb{Z}$ *be nonzero integers,* $m = \mathrm{lcm}(k_1, \ldots, k_{n-1})$, *and*

$$\mathbf{a} = \left(m, \frac{m}{k_1}, \ldots, \frac{m}{k_{n-1}}\right)^t \in \mathbb{Z}^n.$$

There exists an integer l, *depending only on* n, *such that whenever* $|k_1|, \ldots, |k_{n-1}| \geq l$, $SVP \equiv SIVP$ *on* $\Lambda(\mathbf{a})$.

Some of my work with students on WR lattices

WR lattices are important in discrete optimization, algebraic number theory, coding theory, cohomology computations of arithmetic groups, etc. Some of my additional recent work with graduate and undergraduate students on WR lattices includes:

Some of my work with students on WR lattices

WR lattices are important in discrete optimization, algebraic number theory, coding theory, cohomology computations of arithmetic groups, etc. Some of my additional recent work with graduate and undergraduate students on WR lattices includes:

Claremont Colleges NSF REU - 2009

- L. F., D. Moore, R. A. Ohana, W. Zeldow. **On well-rounded sublattices of the hexagonal lattice**, Discrete Mathematics 310 (2010), no. 23, 3287–3302.

Some of my work with students on WR lattices

WR lattices are important in discrete optimization, algebraic number theory, coding theory, cohomology computations of arithmetic groups, etc. Some of my additional recent work with graduate and undergraduate students on WR lattices includes:

Claremont Colleges NSF REU - 2009

- L. F., D. Moore, R. A. Ohana, W. Zeldow. **On well-rounded sublattices of the hexagonal lattice**, Discrete Mathematics 310 (2010), no. 23, 3287–3302.

Claremont Fletcher Jones Fellowship Program - 2011

- L. F., G. Henshaw, P. Liao, M. Prince, X. Sun, S. Whitehead. **On integral well-rounded lattices in the plane**, Discrete and Computational Geometry, vol. 48 no. 3 (2012), pg. 735–748.
- L. F., G. Henshaw, P. Liao, M. Prince, X. Sun, S. Whitehead. **On well-rounded ideal lattices - II**, International Journal of Number Theory, vol. 9 no. 1 (2013) pg. 139–154.

Thank you!