Slide Reduction, Revisited—Filling the Gaps in SVP Approximation

Divesh Aggarwal
NUS

Jianwei Li
RHUL

Phong Q. Nguyen
ENS

Noah Stephens-Davidowitz
Cornell University

Crypto 2020

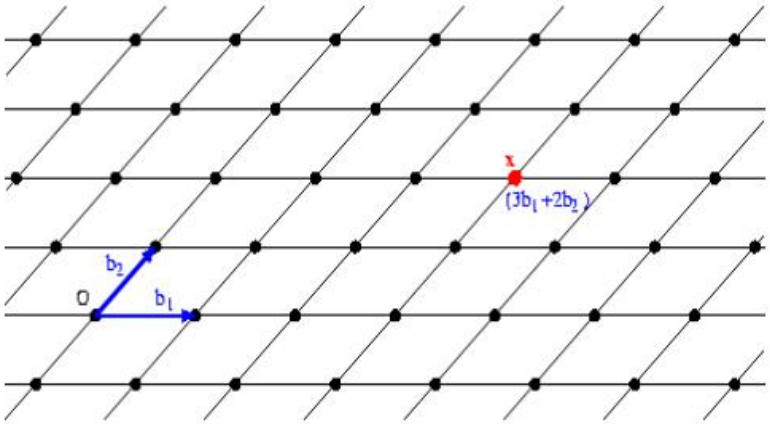
Outline

Lattice and basis

- An n-rank lattice L is a set of all integer linear combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$:

$$L = \{z_1\mathbf{b}_1 + \cdots + z_n\mathbf{b}_n, z_i \in \mathbb{Z}\}.$$

- $B := (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called a basis of L.



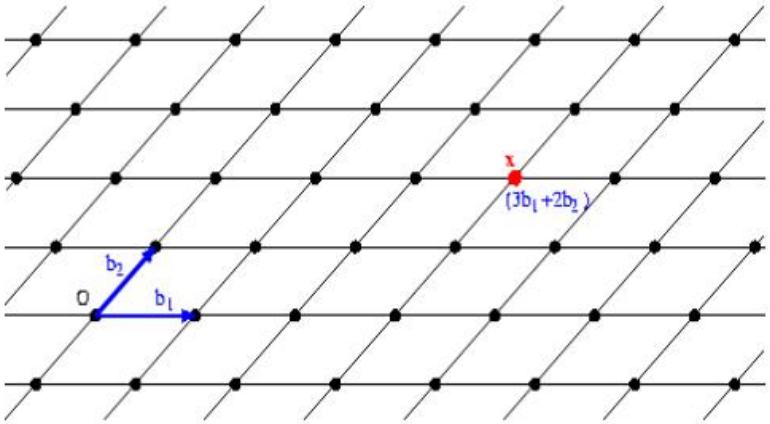
A Lattice of rank 2

Lattice and basis

- An n-rank lattice L is a set of all integer linear combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$:

$$L = \{z_1\mathbf{b}_1 + \cdots + z_n\mathbf{b}_n, z_i \in \mathbb{Z}\}.$$

- $B := (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called a basis of L.



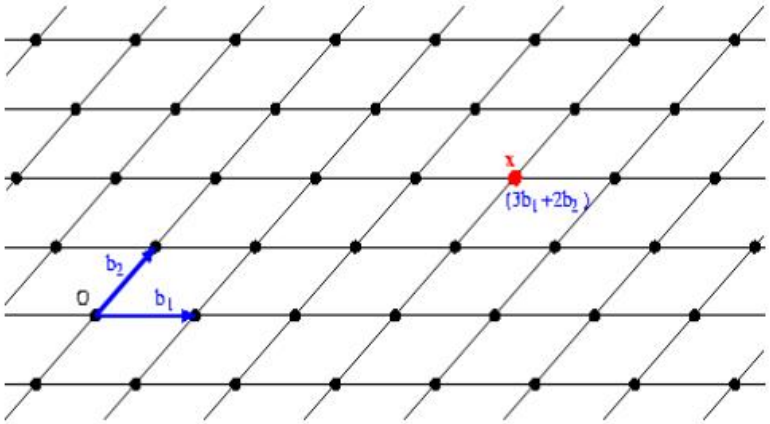
A Lattice of rank 2

Lattice and basis

- An n-rank lattice L is a set of all integer linear combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$:

$$L = \{z_1\mathbf{b}_1 + \cdots + z_n\mathbf{b}_n, z_i \in \mathbb{Z}\}.$$

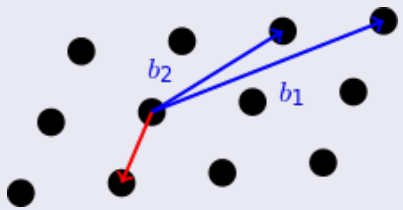
- $B := (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called a basis of L.



A Lattice of rank 2

The most important lattice problem is the shortest vector problem (SVP)

Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x} \in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.

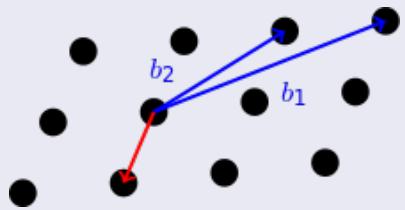


Two natural relaxations

- f-approximate SVP (f-SVP): Given a lattice L, find a non-zero vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given an n-rank lattice L, find a non-zero vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L)$ is the determinant of L.

The most important lattice problem is the shortest vector problem (SVP)

Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x} \in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.

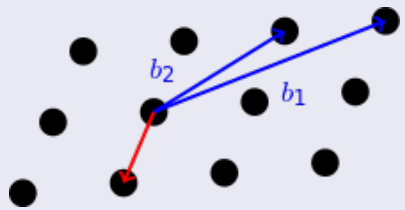


Two natural relaxations

- f-approximate SVP (f-SVP): Given a lattice L, find a non-zero vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given an n-rank lattice L, find a non-zero vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L)$ is the determinant of L.

The most important lattice problem is the shortest vector problem (SVP)

Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x} \in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.

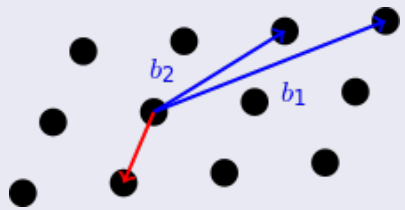


Two natural relaxations

- f-approximate SVP (f-SVP): Given a lattice L, find a non-zero vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given an n-rank lattice L, find a non-zero vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L)$ is the determinant of L.

The most important lattice problem is the shortest vector problem (SVP)

Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x} \in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.



Two natural relaxations

- f-approximate SVP (f-SVP): Given a lattice L, find a non-zero vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given an n-rank lattice L, find a non-zero vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L)$ is the determinant of L.

Hardness of SVP

There is some constant $c > 0$ s.t. $n^{c/\log\log n}$-SVP on n-rank lattices is NP-hard under reasonable complexity theoretic assumptions.[a] [b] [c] [d]

---

[a] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. STOC 1998.

[b] D. Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. SIAM J. Comput 2000 and FOCS 1998.

[c] S. Khot. Hardness of approximating the shortest vector problem in lattices. JACM 2005 and FOCS 2004.

[d] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polyno-mial factors.Theory of Computing 2012 and STOC 2007.

Cryptography VS Cryptanalysis

Lattice cryptography

From Ajtai 1996's beginning, many cryptographic primitives have been constructed whose security is based on the (worst-case) hardness of n^c-SVP for some constant c. [a]

- NIST PQC Round 3 submission

[a] M. Ajtai.Generating Hard Instances of Lattice Problems. STOC 1996.

Lattice cryptanalysis

How to do lattice cryptanalysis?

- How to estimate the concrete security of lattice cryptographic schemes?
  - Solve γ-(H)SVP
  - Lattice reduction

Cryptography VS Cryptanalysis

Lattice cryptography

From Ajtai 1996's beginning, many cryptographic primitives have been constructed whose security is based on the (worst-case) hardness of n^c-SVP for some constant c. [a]

- NIST PQC Round 3 submission

[a] M. Ajtai.Generating Hard Instances of Lattice Problems. STOC 1996.

Lattice cryptanalysis

How to do lattice cryptanalysis?

- How to estimate the concrete security of lattice cryptographic schemes?
  - Solve γ-(H)SVP
  - Lattice reduction

Cryptography VS Cryptanalysis

Lattice cryptography

From Ajtai 1996's beginning, many cryptographic primitives have been constructed whose security is based on the (worst-case) hardness of n^c-SVP for some constant c. [a]

- NIST PQC Round 3 submission

[a] M. Ajtai.Generating Hard Instances of Lattice Problems. STOC 1996.

Lattice cryptanalysis

How to do lattice cryptanalysis?

- How to estimate the concrete security of lattice cryptographic schemes?
  - $\Rightarrow$ Solve n^c-(H)SVP
  - $\Rightarrow$ Lattice reduction

Cryptography VS Cryptanalysis

Lattice cryptography

From Ajtai 1996's beginning, many cryptographic primitives have been constructed whose security is based on the (worst-case) hardness of n^c-SVP for some constant c. [a]

- NIST PQC Round 3 submission

[a] M. Ajtai.Generating Hard Instances of Lattice Problems. STOC 1996.

Lattice cryptanalysis

How to do lattice cryptanalysis?

- How to estimate the concrete security of lattice cryptographic schemes?
  - $\Rightarrow$ Solve n^c-(H)SVP
  - $\Rightarrow$ Lattice reduction

Cryptography VS Cryptanalysis

Lattice cryptography

From Ajtai 1996's beginning, many cryptographic primitives have been constructed whose security is based on the (worst-case) hardness of n^c-SVP for some constant c. [a]

- NIST PQC Round 3 submission

[a] M. Ajtai.Generating Hard Instances of Lattice Problems. STOC 1996.

Lattice cryptanalysis

How to do lattice cryptanalysis?

- How to estimate the concrete security of lattice cryptographic schemes?
  - $\Rightarrow$ Solve n^c-(H)SVP
  - $\Rightarrow$ Lattice reduction

Cryptography VS Cryptanalysis

Lattice cryptography

From Ajtai 1996's beginning, many cryptographic primitives have been constructed whose security is based on the (worst-case) hardness of n^c-SVP for some constant c. [a]

- NIST PQC Round 3 submission

[a] M. Ajtai.Generating Hard Instances of Lattice Problems. STOC 1996.

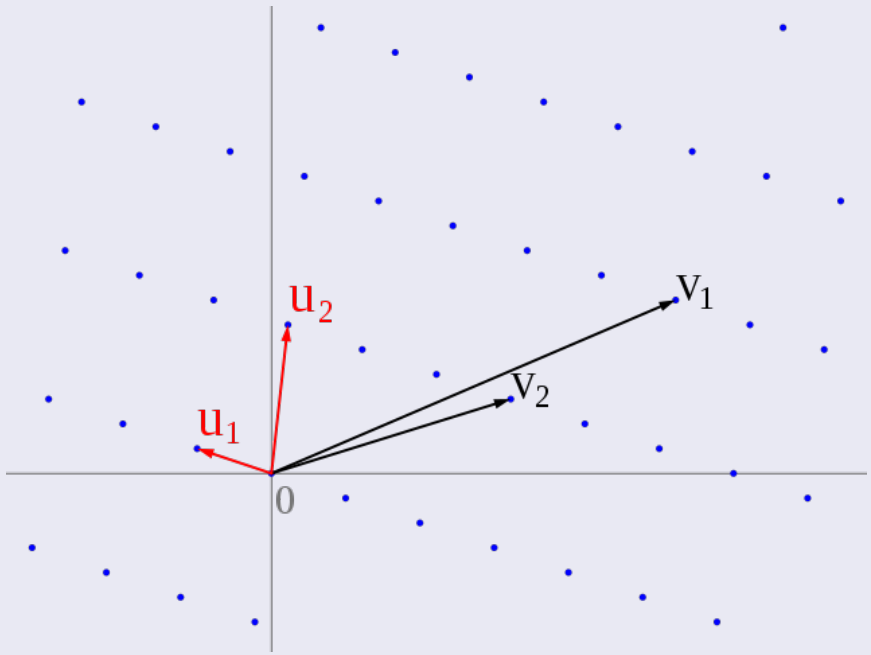
Lattice cryptanalysis

How to do lattice cryptanalysis?

- How to estimate the concrete security of lattice cryptographic schemes?
  - $\Rightarrow$ Solve n^c-(H)SVP
  - $\Rightarrow$ Lattice reduction

Lattice reduction

Given a lattice, find a good basis consisting of reasonably short and almost orthogonal vectors.



Importance

Lattice reduction is the classical approach for solving f-(H)SVP:

- It has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology:
  - It is a popular tool to both public-key cryptography and cryptanalysis;
  - Its importance is growing as lattice-based cryptography becomes the most popular candidate for PQC.

Importance

Lattice reduction is the classical approach for solving f-(H)SVP:

- It has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology:
  - It is a popular tool to both public-key cryptography and cryptanalysis;
  - Its importance is growing as lattice-based cryptography becomes the most popular candidate for PQC.

Importance

Lattice reduction is the classical approach for solving f-(H)SVP:

- It has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology:
  - It is a popular tool to both public-key cryptography and cryptanalysis;
  - Its importance is growing as lattice-based cryptography becomes the most popular candidate for PQC.

Importance

Lattice reduction is the classical approach for solving f-(H)SVP:

- It has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology:
  - It is a popular tool to both public-key cryptography and cryptanalysis;
  - Its importance is growing as lattice-based cryptography becomes the most popular candidate for PQC.

Importance

Lattice reduction is the classical approach for solving f-(H)SVP:

- It has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology:
  - It is a popular tool to both public-key cryptography and cryptanalysis;
  - Its importance is growing as lattice-based cryptography becomes the most popular candidate for PQC.

- SVP: $\lambda_1(L) = \min_{\mathbf{v} \in L \setminus \{\mathbf{0}\}} \|\mathbf{v}\|$.
- Hermite's constant: $\gamma_n = \max \lambda_1(L)^2$ over all n-rank lattices L of unit determinant.
- $\gamma_n = \Theta(n)$ measures the output quality of lattice reduction algorithms.

- SVP: $\lambda_1(L) = \min_{\mathbf{v} \in L \setminus \{\mathbf{0}\}} \|\mathbf{v}\|$.
- Hermite's constant: $\gamma_n = \max \lambda_1(L)^2$ over all n-rank lattices L of unit determinant.
- $\gamma_n = \Theta(n)$ measures the output quality of lattice reduction algorithms.

- SVP: $\lambda_1(L) = \min_{\mathbf{v} \in L \setminus \{\mathbf{0}\}} \|\mathbf{v}\|$.
- Hermite's constant: $\gamma_n = \max \lambda_1(L)^2$ over all n-rank lattices L of unit determinant.
- $\gamma_n = \Theta(n)$ measures the output quality of lattice reduction algorithms.

Prior work 1

Previous best algorithms for $n^{c\geq 1}$-SVP in theory

GN-slide-reduction is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq 1}$-SVP in theory: [a]

- For $n = pk \geq 2k$, with polynomially many calls to exact SVP-oracle in rank k, it outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$\begin{aligned} \|\mathbf{b}_1\| &\leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq 2\gamma_k^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

[a] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Prior work 2

Previous best algorithms for $n^{c\geq\frac{1}{2}}$-HSVP in theory

DBKZ is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq\frac{1}{2}}$-HSVP in theory: [a]

- For $n \geq k \geq 2$, with polynomially many calls to exact SVP-oracle in rank k, it outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$\begin{aligned} \|\mathbf{b}_1\| &\leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq 2\gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

[a] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. EUROCRYPT 2016.

Slide-reduction VS DBKZ

- For $n = pk \geq 2k$, GN-slide-reduction achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-k}{k-1}} \cdot \lambda_1(L).$$

- For $n \geq k \geq 2$, DBKZ achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L).$$

Two natural questions

- Can we extend GN-sldie-reduction algorithm into the case that k might not divide n?
  $\Rightarrow$ Exponential speedup for solving n^c-SVP over $1 < c \notin \mathbb{Z}$.
- The best (proven) approximation factor for appximating SVP and HSVP is now achieved by a single algorithm:
  - Can we get the best of both [GN08] and [MW16]?

Slide-reduction VS DBKZ

- For $n = pk \geq 2k$, GN-slide-reduction achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-k}{k-1}} \cdot \lambda_1(L).$$

- For $n \geq k \geq 2$, DBKZ achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L).$$

Two natural questions

- Can we extend GN-sldie-reduction algorithm into the case that k might not divide n?
  $\Rightarrow$ Exponential speedup for solving n^c-SVP over $1 < c \notin \mathbb{Z}$.
- The best (proven) approximation factor for appximating SVP and HSVP is now achieved by a single algorithm:
  - Can we get the best of both [GN08] and [MW16]?

Slide-reduction VS DBKZ

- For $n = pk \geq 2k$, GN-slide-reduction achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-k}{k-1}} \cdot \lambda_1(L).$$

- For $n \geq k \geq 2$, DBKZ achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L).$$

Two natural questions

- Can we extend GN-sldie-reduction algorithm into the case that k might not divide n?
  $\Rightarrow$ Exponential speedup for solving n^c-SVP over $1 < c \notin \mathbb{Z}$.
- The best (proven) approximation factor for appximating SVP and HSVP is now achieved by a single algorithm:
  - Can we get the best of both [GN08] and [MW16]?

Slide-reduction VS DBKZ

- For $n = pk \geq 2k$, GN-slide-reduction achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-k}{k-1}} \cdot \lambda_1(L).$$

- For $n \geq k \geq 2$, DBKZ achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L).$$

Two natural questions

- Can we extend GN-sldie-reduction algorithm into the case that k might not divide n?
  $\Rightarrow$ Exponential speedup for solving n^c-SVP over $1 < c \notin \mathbb{Z}$.
- The best (proven) approximation factor for appximating SVP and HSVP is now achieved by a single algorithm:
  - Can we get the best of both [GN08] and [MW16]?

Slide-reduction VS DBKZ

- For $n = pk \geq 2k$, GN-slide-reduction achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-k}{k-1}} \cdot \lambda_1(L).$$

- For $n \geq k \geq 2$, DBKZ achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L).$$

Two natural questions

- Can we extend GN-sldie-reduction algorithm into the case that k might not divide n?
  $\Rightarrow$ Exponential speedup for solving n^c-SVP over $1 < c \notin \mathbb{Z}$.
- The best (proven) approximation factor for appximating SVP and HSVP is now achieved by a single algorithm:
  - Can we get the best of both [GN08] and [MW16]?

Slide-reduction VS DBKZ

- For $n = pk \geq 2k$, GN-slide-reduction achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-k}{k-1}} \cdot \lambda_1(L).$$

- For $n \geq k \geq 2$, DBKZ achieves:

$$\|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\| \leq 2\gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L).$$

Two natural questions

- Can we extend GN-sldie-reduction algorithm into the case that k might not divide n?
  $\Rightarrow$ Exponential speedup for solving n^c-SVP over $1 < c \notin \mathbb{Z}$.
- The best (proven) approximation factor for appximating SVP and HSVP is now achieved by a single algorithm:
  - Can we get the best of both [GN08] and [MW16]?

Approximating SVP with sublinear factors

- The security of many lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \in [\frac{1}{2}, 1]$.
- Awkward: There is no known non-trivial algorithm for approximating SVP with sublinear factors.
- Question: Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?
  $\Rightarrow$ At least exponential speedup.

Approximating SVP with sublinear factors

- The security of many lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \in [\frac{1}{2}, 1]$.
- Awkward: There is no known non-trivial algorithm for approximating SVP with sublinear factors.
- Question: Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?
  $\Rightarrow$ At least exponential speedup.

Approximating SVP with sublinear factors

- The security of many lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \in [\frac{1}{2}, 1]$.
- Awkward: There is no known non-trivial algorithm for approximating SVP with sublinear factors.
- Question: Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?
  $\Rightarrow$ At least exponential speedup.

Approximating SVP with sublinear factors

- The security of many lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \in [\frac{1}{2}, 1]$.
- Awkward: There is no known non-trivial algorithm for approximating SVP with sublinear factors.
- Question: Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?
  $\Rightarrow$ At least exponential speedup.

Our paper solves the three questions:

Q1 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Q2 Can we extend GN-slide-reduction algorithm into the case that k does not divide n exactly, so that it can faster solve n^c-SVP over any fractional constant $c \geq 1$?

Q3 Is there a single algorithm which is the best in theory for solving both $n^{c \geq 1}$-SVP and $n^{c \geq \frac{1}{2}}$-HSVP?

Our paper solves the three questions:

Q1 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Q2 Can we extend GN-slide-reduction algorithm into the case that k does not divide n exactly, so that it can faster solve n^c-SVP over any fractional constant $c \geq 1$?

Q3 Is there a single algorithm which is the best in theory for solving both $n^{c \geq 1}$-SVP and $n^{c \geq \frac{1}{2}}$-HSVP?

Our paper solves the three questions:

Q1 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Q2 Can we extend GN-slide-reduction algorithm into the case that k does not divide n exactly, so that it can faster solve n^c-SVP over any fractional constant $c \geq 1$?

Q3 Is there a single algorithm which is the best in theory for solving both $n^{c \geq 1}$-SVP and $n^{c \geq \frac{1}{2}}$-HSVP?

Our paper solves the three questions:

Q1 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Q2 Can we extend GN-slide-reduction algorithm into the case that k does not divide n exactly, so that it can faster solve n^c-SVP over any fractional constant $c \geq 1$?

Q3 Is there a single algorithm which is the best in theory for solving both $n^{c \geq 1}$-SVP and $n^{c \geq \frac{1}{2}}$-HSVP?

Our first result

Theorem (Approximating SVP with sublinear factor)

Let $2k > n \geq k \geq 2$ be integers and $\delta \geq 1$. There is an algorithm that with polynomially many calls to δ-SVP-oracle in rank k, it outputs an nonzero vector $\mathbf{b}$ of the input lattice L s.t.

$$\|\mathbf{b}\| \leq O(\delta(\delta^2\gamma_k)^{\frac{n}{2k}}) \cdot \lambda_1(L).$$

★ This is the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$.

Corollary

For any constant $c \in (1/2, 1)$ and any factor $\delta \geq 1$, there is an efficient reduction from $O(\delta^{2c+1}n^c)$-SVP in rank n to δ-SVP in rank $\lceil \frac{n}{2c} \rceil$.

Our first result

Theorem (Approximating SVP with sublinear factor)

Let $2k > n \geq k \geq 2$ be integers and $\delta \geq 1$. There is an algorithm that with polynomially many calls to δ-SVP-oracle in rank k, it outputs an nonzero vector $\mathbf{b}$ of the input lattice L s.t.

$$\|\mathbf{b}\| \leq O(\delta(\delta^2\gamma_k)^{\frac{n}{2k}}) \cdot \lambda_1(L).$$

★ This is the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$.

Corollary

For any constant $c \in (1/2, 1)$ and any factor $\delta \geq 1$, there is an efficient reduction from $O(\delta^{2c+1}n^c)$-SVP in rank n to δ-SVP in rank $\lceil \frac{n}{2c} \rceil$.

Our first result

Theorem (Approximating SVP with sublinear factor)

Let $2k > n \geq k \geq 2$ be integers and $\delta \geq 1$. There is an algorithm that with polynomially many calls to δ-SVP-oracle in rank k, it outputs an nonzero vector $\mathbf{b}$ of the input lattice L s.t.

$$\|\mathbf{b}\| \leq O(\delta(\delta^2\gamma_k)^{\frac{n}{2k}}) \cdot \lambda_1(L).$$

★ This is the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$.

Corollary

For any constant $c \in (1/2, 1)$ and any factor $\delta \geq 1$, there is an efficient reduction from $O(\delta^{2c+1}n^c)$-SVP in rank n to δ-SVP in rank $\lceil \frac{n}{2c} \rceil$.

Our first result

Theorem (Approximating SVP with sublinear factor)

Let $2k > n \geq k \geq 2$ be integers and $\delta \geq 1$. There is an algorithm that with polynomially many calls to δ-SVP-oracle in rank k, it outputs an nonzero vector $\mathbf{b}$ of the input lattice L s.t.

$$\|\mathbf{b}\| \leq O(\delta(\delta^2\gamma_k)^{\frac{n}{2k}}) \cdot \lambda_1(L).$$

★ This is the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$.

Corollary

For any constant $c \in (1/2, 1)$ and any factor $\delta \geq 1$, there is an efficient reduction from $O(\delta^{2c+1}n^c)$-SVP in rank n to δ-SVP in rank $\lceil \frac{n}{2c} \rceil$.

Our second result

Theorem (Approximating SVP with (at least) polynomial factor)

Let $n \geq 2k \geq 4$ be integers and $\delta \geq 1$. There is an algorithm that with polynomially many calls to δ-SVP-oracle in rank k, it outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$\begin{aligned} \|\mathbf{b}_1\| &\leq 2(\delta^2\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq 2(\delta^2\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

Corollary

For any constant $c \geq 1$ and any factor $\delta \geq 1$, there is an efficient reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $\lfloor \frac{n}{c+1} \rfloor$.

Our second result

Theorem (Approximating SVP with (at least) polynomial factor)

Let $n \geq 2k \geq 4$ be integers and $\delta \geq 1$. There is an algorithm that with polynomially many calls to δ-SVP-oracle in rank k, it outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$\begin{aligned} \|\mathbf{b}_1\| &\leq 2(\delta^2\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq 2(\delta^2\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

Corollary

For any constant $c \geq 1$ and any factor $\delta \geq 1$, there is an efficient reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $\lfloor \frac{n}{c+1} \rfloor$.

Our second result

Theorem (Approximating SVP with (at least) polynomial factor)

Let $n \geq 2k \geq 4$ be integers and $\delta \geq 1$. There is an algorithm that with polynomially many calls to δ-SVP-oracle in rank k, it outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$\begin{aligned} \|\mathbf{b}_1\| &\leq 2(\delta^2\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq 2(\delta^2\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

Corollary

For any constant $c \geq 1$ and any factor $\delta \geq 1$, there is an efficient reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $\lfloor \frac{n}{c+1} \rfloor$.

Impact

- Our two algorithms provide currently the best polynomial-time lattice reduction algorithm:
  $\Rightarrow$ Achieve the best time/quality trade-off in theory.
- With well-chosen SVP-oracles in lower rank, our work implies the exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$:
  $\Rightarrow$ This is the regime most relevant for cryptography.

Impact

- Our two algorithms provide currently the best polynomial-time lattice reduction algorithm:
  $\Rightarrow$ Achieve the best time/quality trade-off in theory.
- With well-chosen SVP-oracles in lower rank, our work implies the exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$:
  $\Rightarrow$ This is the regime most relevant for cryptography.

Impact: the faster provable/heuristic algorithm

Table: Provable algorithms for approximating SVP.[1]

Approx-factor	Previous best	This work
n^c for $c \in [\frac{1}{2}, 1)$	$2^{0.802n}$ [WLW15]	$2^{\frac{0.802n}{2c}}$
n^c for $c \geq 1$	$2^{\frac{n}{\lfloor c+1 \rfloor}}$ [GN08]+[ADRS15]	$2^{\frac{0.802n}{c+1}}$

Table: Heuristic algorithms for approximating SVP.[2]

Approx-factor	Previous best	This work
n^c for $c \in [\frac{1}{2}, 1)$	$2^{0.292n}$ [BDGL16]	$2^{\frac{0.292n}{2c}}$
n^c for $c \geq 1$	$2^{\frac{0.292n}{\lfloor c+1 \rfloor}}$ [GN08]+[BDGL16]	$2^{\frac{0.292n}{c+1}}$

[1] W. Wei, M. Liu, and X. Wang. Finding shortest latticevectors in the presence of gaps. CT-RSA 2015.

[2] A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. SODA 2016.

Warning

- For simplicity, we describe our ideas with exact SVP-oracle;
- It is easy to replace exact SVP-oracle with approximate-SVP-oracle.

Warning

- For simplicity, we describe our ideas with exact SVP-oracle;
- It is easy to replace exact SVP-oracle with approximate-SVP-oracle.

Warning

- For simplicity, we describe our ideas with exact SVP-oracle;
- It is easy to replace exact SVP-oracle with approximate-SVP-oracle.

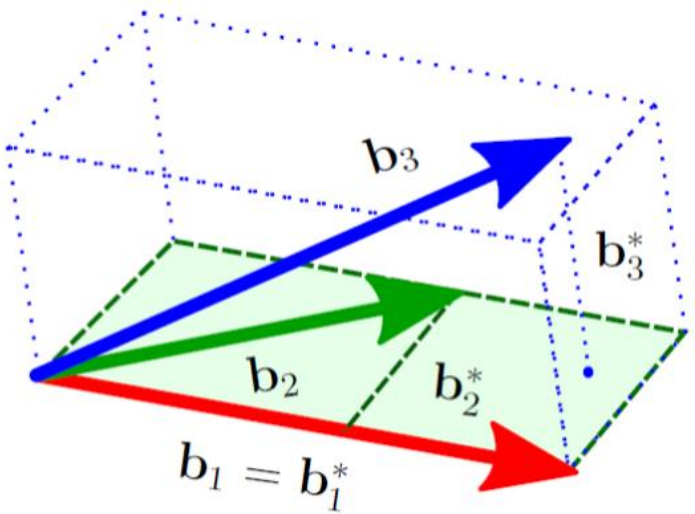
Preliminaries

GSO

Given a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, define the orthogonal projection:

$$\pi_i : \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \mapsto \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^{\perp}.$$

- Each vector $\mathbf{b}_i^* = \pi_i(\mathbf{b}_i)$ is the Gram-Schmidt vector of B.
- The projected block $B_{[i,j]} = (\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), \ldots, \pi_i(\mathbf{b}_j))$.



Preliminaries

Several reduction notions

Let B be a basis of a lattice L.

- B is *SVP-reduced* if its first basis vector is a shortest nonzero vector of L.
- B is *DSVP-reduced* if its dual basis is SVP-reduced.
- B is *DBKZ-reduced* if it is produced by the DBKZ algorithm with blocksize k.[a]
- B is *GN-slide-reduced* if it is produced by the GN-slide reduction algorithm with blocksize k.[b]

---

[a] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. EUROCRYPT 2016.

[b] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Preliminaries

Several reduction notions

Let B be a basis of a lattice L.

- B is *SVP-reduced* if its first basis vector is a shortest nonzero vector of L.
- B is *DSVP-reduced* if its dual basis is SVP-reduced.
- B is *DBKZ-reduced* if it is produced by the DBKZ algorithm with blocksize k.[a]
- B is *GN-slide-reduced* if it is produced by the GN-slide reduction algorithm with blocksize k.[b]

---

[a] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. EUROCRYPT 2016.

[b] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Preliminaries

Several reduction notions

Let B be a basis of a lattice L.

- B is *SVP-reduced* if its first basis vector is a shortest nonzero vector of L.
- B is *DSVP-reduced* if its dual basis is SVP-reduced.
- B is *DBKZ-reduced* if it is produced by the DBKZ algorithm with blocksize k.[a]
- B is *GN-slide-reduced* if it is produced by the GN-slide reduction algorithm with blocksize k.[b]

[a] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. EUROCRYPT 2016.

[b] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Preliminaries

Several reduction notions

Let B be a basis of a lattice L.

- B is *SVP-reduced* if its first basis vector is a shortest nonzero vector of L.
- B is *DSVP-reduced* if its dual basis is SVP-reduced.
- B is *DBKZ-reduced* if it is produced by the DBKZ algorithm with blocksize k.[a]
- B is *GN-slide-reduced* if it is produced by the GN-slide reduction algorithm with blocksize k.[b]

[a] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. EUROCRYPT 2016.

[b] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Preliminaries

Several reduction notions

Let B be a basis of a lattice L.

- B is *SVP-reduced* if its first basis vector is a shortest nonzero vector of L.
- B is *DSVP-reduced* if its dual basis is SVP-reduced.
- B is *DBKZ-reduced* if it is produced by the DBKZ algorithm with blocksize k.[a]
- B is *GN-slide-reduced* if it is produced by the GN-slide reduction algorithm with blocksize k.[b]

---

[a] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. EUROCRYPT 2016.

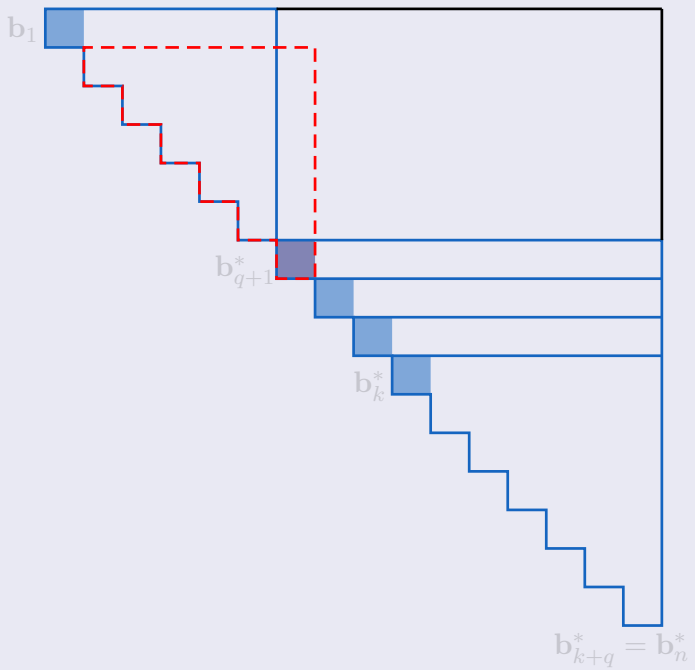
[b] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Approximating SVP with sublinear factor

Given a lattice L of rank $n \in (k, 2k)$ and a SVP-oracle in rank k.

Ideas

- Partition the input basis into two blocks s.t. the first block has smaller rank $n - k$ and the second block has rank k:

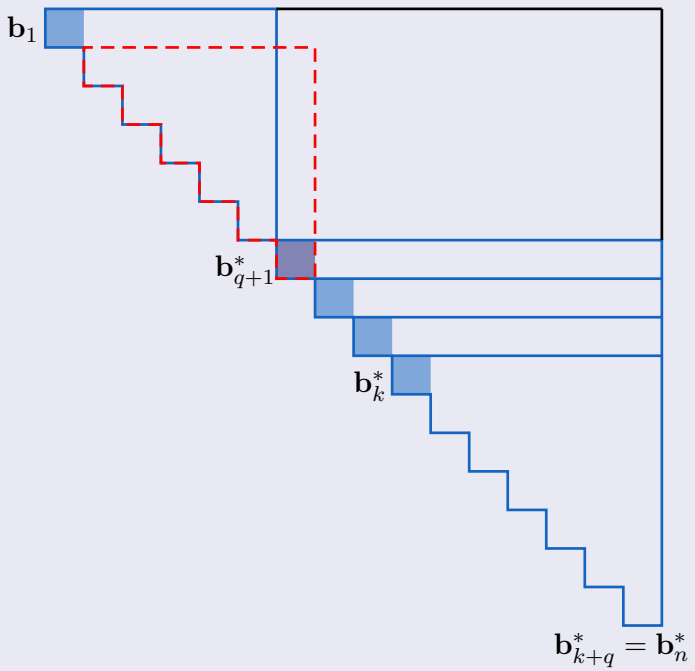


Approximating SVP with sublinear factor

Given a lattice L of rank $n \in (k, 2k)$ and a SVP-oracle in rank k.

Ideas

- Partition the input basis into two blocks s.t. the first block has smaller rank $n - k$ and the second block has rank k:



Approximating SVP with sublinear factor

Given a lattice L of rank $n \in (k, 2k)$ and a SVP-oracle in rank k.

Ideas

- Partition the input basis into two blocks s.t. the first block has smaller rank $n-k$ and the second block has rank k;
- Alternately SVP-reduce and DSVP-reduce some projected blocks on the input basis s.t. the head basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k$ become:

$$\min\left\{\lambda_1(L(\mathbf{b}_1,\ldots,\mathbf{b}_k)), \frac{\mathrm{vol}(\mathbf{b}_1,\ldots,\mathbf{b}_k)^{1/k}}{\gamma_{n-k}^{(n-k)/(2k)}})\right\} \lesssim \lambda_1(L).$$

- Extra SVP-reduce $\mathbf{b}_1, \ldots, \mathbf{b}_k$ to find:

$$\|\mathbf{b}\| \lesssim \gamma_k^{\frac{n}{2k}} \cdot \lambda_1(L).$$

Approximating SVP with sublinear factor

Given a lattice L of rank $n \in (k, 2k)$ and a SVP-oracle in rank k.

Ideas

- Partition the input basis into two blocks s.t. the first block has smaller rank $n-k$ and the second block has rank k;
- Alternately SVP-reduce and DSVP-reduce some projected blocks on the input basis s.t. the head basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k$ become:

$$\min \left\{ \lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)), \frac{\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)^{1/k}}{\gamma_{n-k}^{(n-k)/(2k)}} \right\} \lesssim \lambda_1(L).$$

- Extra SVP-reduce $\mathbf{b}_1, \ldots, \mathbf{b}_k$ to find:

$$\|\mathbf{b}\| \lesssim \gamma_k^{\frac{n}{2k}} \cdot \lambda_1(L).$$

Approximating SVP with sublinear factor

Given a lattice L of rank $n \in (k, 2k)$ and a SVP-oracle in rank k.

Ideas

- Partition the input basis into two blocks s.t. the first block has smaller rank $n-k$ and the second block has rank k;
- Alternately SVP-reduce and DSVP-reduce some projected blocks on the input basis s.t. the head basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k$ become:

$$\min \left\{ \lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)), \frac{\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)^{1/k}}{\gamma_{n-k}^{(n-k)/(2k)}} \right\} \lesssim \lambda_1(L).$$

- Extra SVP-reduce $\mathbf{b}_1, \ldots, \mathbf{b}_k$ to find:

$$\|\mathbf{b}\| \lesssim \gamma_k^{\frac{n}{2k}} \cdot \lambda_1(L).$$

Approximating SVP with sublinear factor

Given a lattice L of rank $n \in (k, 2k)$ and a SVP-oracle in rank k.

Ideas

- Partition the input basis into two blocks s.t. the first block has smaller rank $n-k$ and the second block has rank k;
- Alternately SVP-reduce and DSVP-reduce some projected blocks on the input basis s.t. the head basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k$ become:

$$\min \left\{ \lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)), \frac{\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)^{1/k}}{\gamma_{n-k}^{(n-k)/(2k)}}) \right\} \lesssim \lambda_1(L).$$

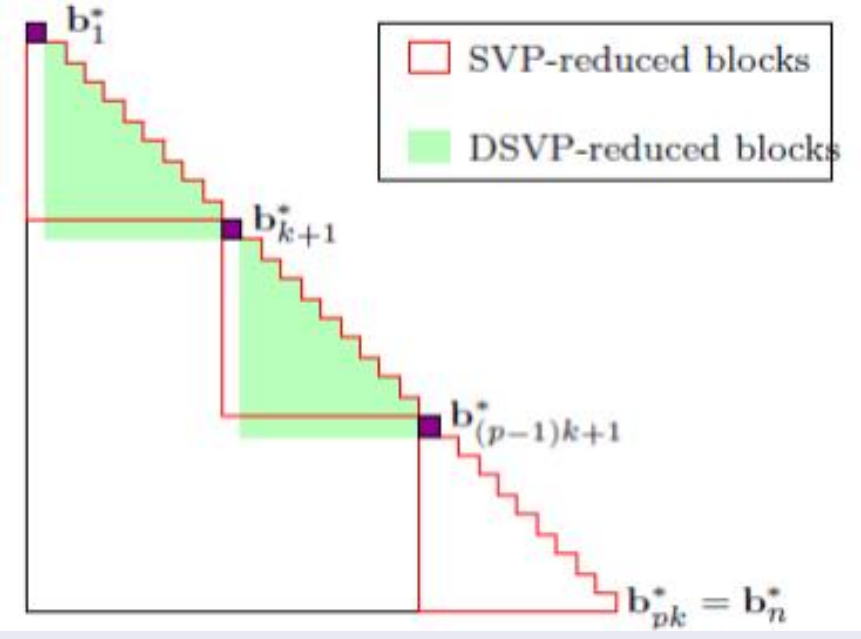
- Extra SVP-reduce $\mathbf{b}_1, \ldots, \mathbf{b}_k$ to find:

$$\|\mathbf{b}\| \lesssim \gamma_k^{\frac{n}{2k}} \cdot \lambda_1(L).$$

Approximating SVP with (at least) polynomial factor

Recall ideas of GN-slide-reduction

Given a basis B of rank $n = pk \geq 2k$, GN partitions the basis into p blocks of equal rank k:

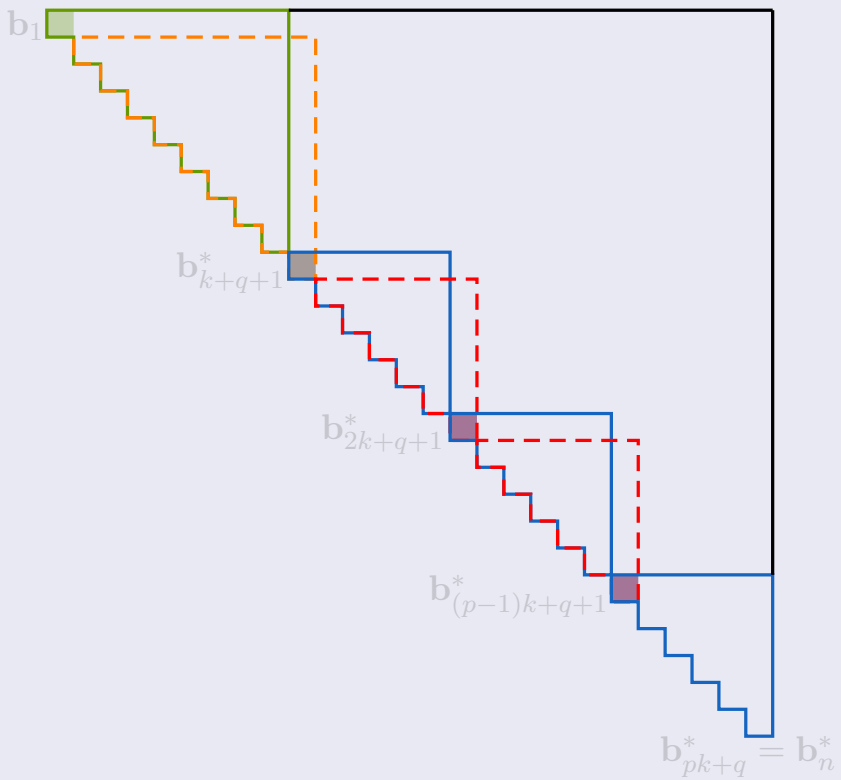


Approximating SVP with (at least) polynomial factor

Given as input a basis B of a lattice L of rank $n = pk + q$ with $p, k \geq 2$ and $0 \leq q < k$ and a SVP-oracle in rank k.

Ideas

- Partition B into p blocks s.t. the first block has larger rank $k + q$ and the other block has the same rank k;

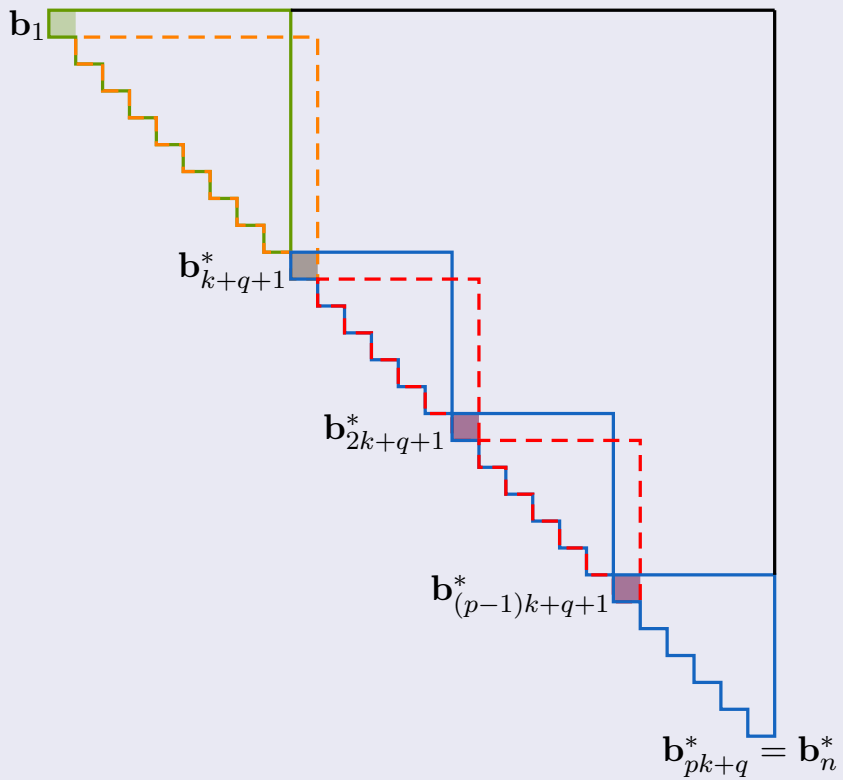


Approximating SVP with (at least) polynomial factor

Given as input a basis B of a lattice L of rank $n = pk + q$ with $p, k \geq 2$ and $0 \leq q < k$ and a SVP-oracle in rank k.

Ideas

- Partition B into p blocks s.t. the first block has larger rank $k + q$ and the other block has the same rank k;



Approximating SVP with (at least) polynomial factor

Given as input a basis B of a lattice L of rank $n = pk + q$ with $p, k \geq 2$ and $0 \leq q < k$ and a SVP-oracle in rank k.

Ideas

- Partition B into p blocks s.t. the first block has larger rank $k + q$ and the other block has the same rank k;
- Alternately SVP-reduce and DSVP-reduce some projected blocks on B s.t. $B_{[1,k+q]}$ becomes DBKZ-reduced, $B_{[k+q+1,n]}$ becomes GN-slide-reduced, and both blocks are glued by DBKZ-reducedness of $B_{[2,k+q+1]}$
- The output basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ satisfies:

$$\|\mathbf{b}_1\| \lesssim \gamma_k^{\frac{n-1}{2(k-1)}} \mathrm{vol}(L)^{1/n},$$

$$\|\mathbf{b}_1\| \lesssim \gamma_k^{\frac{n-k}{k-1}} \lambda_1(L).$$

Approximating SVP with (at least) polynomial factor

Given as input a basis B of a lattice L of rank $n = pk + q$ with $p, k \geq 2$ and $0 \leq q < k$ and a SVP-oracle in rank k.

Ideas

- Partition B into p blocks s.t. the first block has larger rank $k + q$ and the other block has the same rank k;
- Alternately SVP-reduce and DSVP-reduce some projected blocks on B s.t. $B_{[1,k+q]}$ becomes DBKZ-reduced, $B_{[k+q+1,n]}$ becomes GN-slide-reduced, and both blocks are glued by DBKZ-reducedness of $B_{[2,k+q+1]}$
- The output basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ satisfies:

$$\|\mathbf{b}_1\| \lesssim \gamma_k^{\frac{n-1}{2(k-1)}} \mathrm{vol}(L)^{1/n},$$

$$\|\mathbf{b}_1\| \lesssim \gamma_k^{\frac{n-k}{k-1}} \lambda_1(L).$$

Approximating SVP with (at least) polynomial factor

Given as input a basis B of a lattice L of rank $n = pk + q$ with $p, k \geq 2$ and $0 \leq q < k$ and a SVP-oracle in rank k.

Ideas

- Partition B into p blocks s.t. the first block has larger rank $k + q$ and the other block has the same rank k;
- Alternately SVP-reduce and DSVP-reduce some projected blocks on B s.t. $B_{[1,k+q]}$ becomes DBKZ-reduced, $B_{[k+q+1,n]}$ becomes GN-slide-reduced, and both blocks are glued by DBKZ-reducedness of $B_{[2,k+q+1]}$
- The output basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ satisfies:

$$\|\mathbf{b}_1\| \lesssim \gamma_k^{\frac{n-1}{2(k-1)}} \mathrm{vol}(L)^{1/n},$$

$$\|\mathbf{b}_1\| \lesssim \gamma_k^{\frac{n-k}{k-1}} \lambda_1(L).$$

Approximating SVP with (at least) polynomial factor

Given as input a basis B of a lattice L of rank $n = pk + q$ with $p, k \geq 2$ and $0 \leq q < k$ and a SVP-oracle in rank k.

Ideas

- Partition B into p blocks s.t. the first block has larger rank $k + q$ and the other block has the same rank k;
- Alternately SVP-reduce and DSVP-reduce some projected blocks on B s.t. $B_{[1,k+q]}$ becomes DBKZ-reduced, $B_{[k+q+1,n]}$ becomes GN-slide-reduced, and both blocks are glued by DBKZ-reducedness of $B_{[2,k+q+1]}$
- The output basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ satisfies:

$$\|\mathbf{b}_1\| \lesssim \gamma_k^{\frac{n-1}{2(k-1)}} \mathrm{vol}(L)^{1/n},$$
$$\|\mathbf{b}_1\| \lesssim \gamma_k^{\frac{n-k}{k-1}} \lambda_1(L).$$

Conclusion

The best polynomial-time lattice reduction in theory, including the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$:

- The exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$;
  $\Rightarrow$ The regime most relevant for cryptography.
  $\Rightarrow$ Lattice security estimates.

$n^{0.99}$-SVP	Provable: $2^{0.802n}$	$\rightarrow$ $2^{0.405n}$
	Heuristic: $2^{0.292n}$	$\rightarrow$ $2^{0.148n}$
$n^{1.99}$-SVP	Provable: $2^{0.401n}$	$\rightarrow$ $2^{0.269n}$
	Heuristic: $2^{0.146n}$	$\rightarrow$ $2^{0.098n}$

- For more details please refer to our paper.

Conclusion

The best polynomial-time lattice reduction in theory, including the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$:

- The exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$;
  $\Rightarrow$ The regime most relevant for cryptography.
  $\Rightarrow$ Lattice security estimates.

$n^{0.99}$-SVP	Provable: $2^{0.802n}$	$\rightarrow$ $2^{0.405n}$
	Heuristic: $2^{0.292n}$	$\rightarrow$ $2^{0.148n}$
$n^{1.99}$-SVP	Provable: $2^{0.401n}$	$\rightarrow$ $2^{0.269n}$
	Heuristic: $2^{0.146n}$	$\rightarrow$ $2^{0.098n}$

- For more details please refer to our paper.

Conclusion

The best polynomial-time lattice reduction in theory, including the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$:

- The exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$;
  $\Rightarrow$ The regime most relevant for cryptography.
  $\Rightarrow$ Lattice security estimates.

$n^{0.99}$-SVP	Provable: $2^{0.802n} \rightarrow 2^{0.405n}$
	Heuristic: $2^{0.292n} \rightarrow 2^{0.148n}$
$n^{1.99}$-SVP	Provable: $2^{0.401n} \rightarrow 2^{0.269n}$
	Heuristic: $2^{0.146n} \rightarrow 2^{0.098n}$

- For more details please refer to our paper.