Provable Sieving Algorithms for the Shortest Vector Problem and the Closest Vector Problem in the ℓ_p norm

Priyanka Mukhopadhyay

Institute for Quantum Computing
and
Department of Combinatorics and Optimization

University of Waterloo

February 2020

Overview

Lattice

$\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \in \mathbb{R}^d$: n linearly independent vectors

Lattice

$\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \in \mathbb{R}^d$: n linearly independent vectors

Lattice $\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$

Lattice

$\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \in \mathbb{R}^d$: n linearly independent vectors

Lattice $\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$

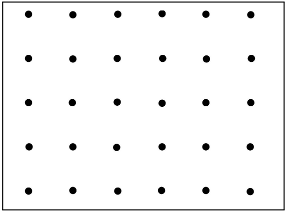


Figure: A lattice in $\mathbb{R}^2$

Lattice

$\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \in \mathbb{R}^d$: n linearly independent vectors

Lattice $\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$

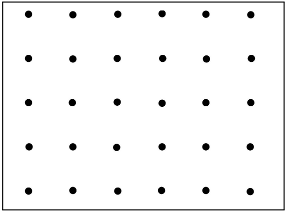


Figure: A lattice in $\mathbb{R}^2$

- n : rank of the lattice

Lattice

$\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \in \mathbb{R}^d$: n linearly independent vectors

Lattice $\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$

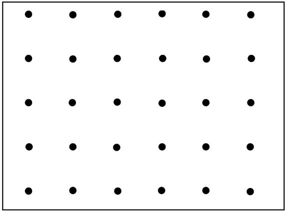


Figure: A lattice in $\mathbb{R}^2$

- n : rank of the lattice
- d : dimension of the lattice

Lattice

$\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \in \mathbb{R}^d$: n linearly independent vectors

Lattice $\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$

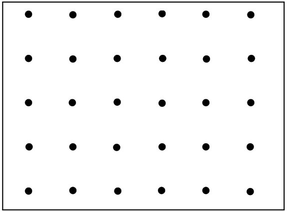


Figure: A lattice in $\mathbb{R}^2$

- n : rank of the lattice
- d : dimension of the lattice
- $n = d$: Full-rank lattice

Lattice

$\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \in \mathbb{R}^d$: n linearly independent vectors

Lattice $\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$

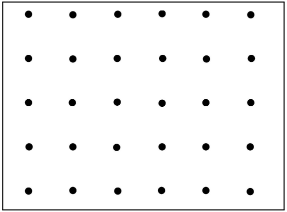


Figure: A lattice in $\mathbb{R}^2$

- n : rank of the lattice
- d : dimension of the lattice
- $n = d$: Full-rank lattice

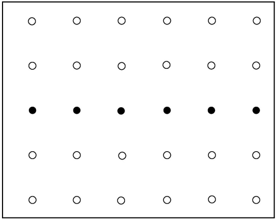


Figure: Not a full-rank lattice

Lattice basis

$$\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$$

Lattice basis

$$\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$$

- $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$: Basis of $\mathscr{L}$

Lattice basis

$$\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$$

- $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$: Basis of $\mathscr{L}$
- Not unique

Lattice basis

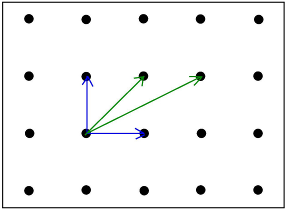


Figure: Bases of $\mathbb{Z}^2$

$$\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$$

- $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$: Basis of $\mathscr{L}$
- Not unique

Lattice basis

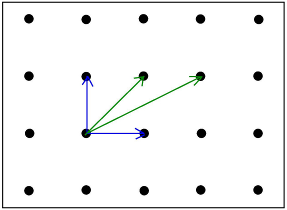


Figure: Bases of $\mathbb{Z}^2$

$$\mathscr{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$$

- $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$: Basis of $\mathscr{L}$
- Not unique

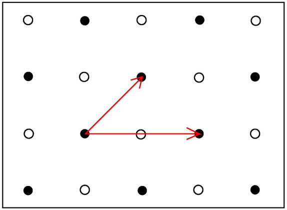


Figure: Not a basis of $\mathbb{Z}^2$

Fundamental Parallelepiped

$$\mathscr{P}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{R}^n, \quad \forall i \quad 0 \leq x_i < 1\}$$

Fundamental Parallelepiped

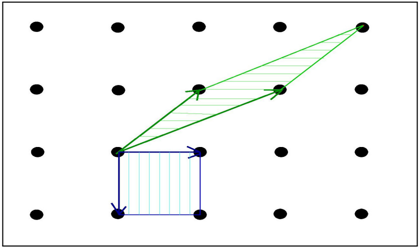
$$\mathscr{P}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{R}^n, \quad \forall i \quad 0 \leq x_i < 1\}$$

- Depends on the basis

Fundamental Parallelepiped

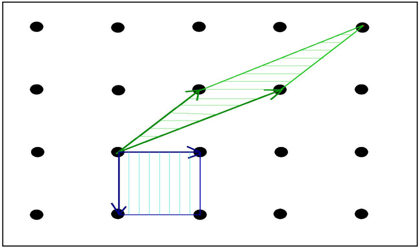
$$\mathscr{P}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{R}^n, \quad \forall i \quad 0 \leq x_i < 1\}$$

- Depends on the basis



Fundamental Parallelepiped

$$\mathscr{P}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{R}^n, \quad \forall i \quad 0 \le x_i < 1\}$$

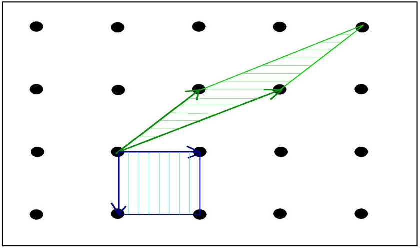


- Depends on the basis
- For any $\mathbf{z} \in \mathbb{R}^n$, there exists a unique $\mathbf{y} \in \mathscr{P}(\mathbf{B})$ such that $\mathbf{z} - \mathbf{y} \in \mathscr{L}(\mathbf{B})$.
  $\mathbf{y} \equiv \mathbf{z} \mod \mathbf{B}$.

Fundamental Parallelepiped

$$\mathscr{P}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{R}^n, \quad \forall i \quad 0 \leq x_i < 1\}$$

- Depends on the basis



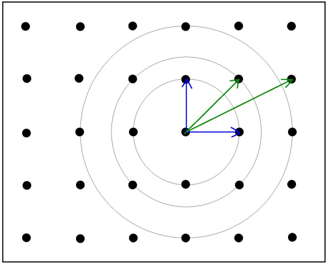
- For any $\mathbf{z} \in \mathbb{R}^n$, there exists a unique $\mathbf{y} \in \mathscr{P}(\mathbf{B})$ such that $\mathbf{z} - \mathbf{y} \in \mathscr{L}(\mathbf{B})$. $\mathbf{y} \equiv \mathbf{z} \mod \mathbf{B}$.
- Translates $\mathscr{P}(\mathbf{B}) + \mathbf{v}$ where $\mathbf{v} \in \mathscr{L}$ form a partition of $\text{span}(\mathbf{B})$.

Successive Minimum

i^{th} successive minimum $= \lambda_i(\mathscr{L}) =$ Smallest $r > 0$ such that $\mathscr{L}$ contains at least i linearly independent vectors of length at most r.

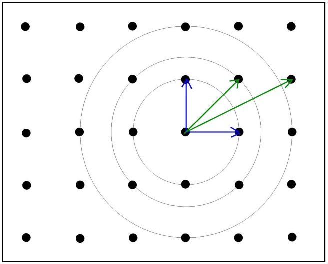
Successive Minimum

i^{th} successive minimum $= \lambda_i(\mathscr{L}) =$ Smallest $r > 0$ such that $\mathscr{L}$ contains at least i linearly independent vectors of length at most r.



Successive Minimum

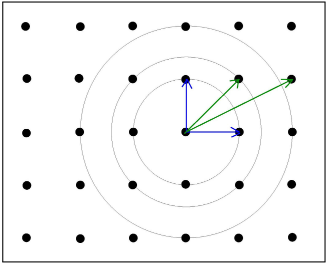
i^{th} successive minimum $= \lambda_i(\mathscr{L}) =$ Smallest $r > 0$ such that $\mathscr{L}$ contains at least i linearly independent vectors of length at most r.



First minimum $= \lambda_1(\mathscr{L}) =$ Length of the shortest non-zero lattice vector

Successive Minimum

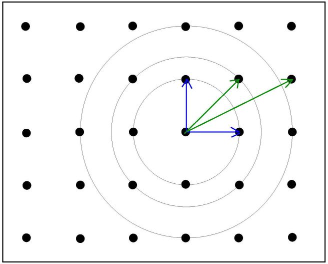
i^{th} successive minimum $= \lambda_i(\mathscr{L}) =$ Smallest $r > 0$ such that $\mathscr{L}$ contains at least i linearly independent vectors of length at most r.



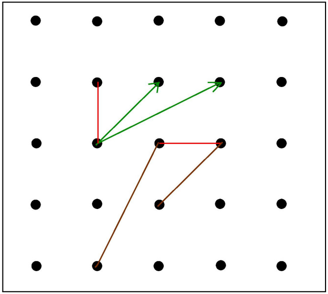
First minimum $= \lambda_1(\mathscr{L}) =$ Length of the shortest non-zero lattice vector $=$ Smallest distance between any two lattice vectors

Successive Minimum

i^{th} successive minimum $= \lambda_i(\mathscr{L}) =$ Smallest $r > 0$ such that $\mathscr{L}$ contains at least i linearly independent vectors of length at most r.



First minimum $= \lambda_1(\mathscr{L}) =$ Length of the shortest non-zero lattice vector $=$ Smallest distance between any two lattice vectors



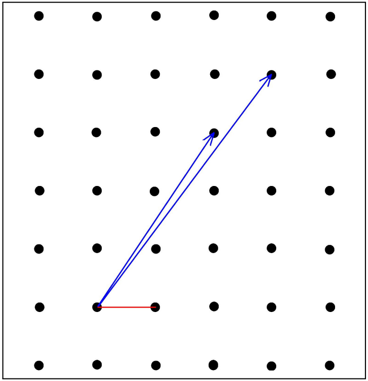
Overview

- Preliminary definitions
- Shortest Vector Problem and Closest Vector Problem
- Sieving algorithms for SVP and CVP

Shortest Vector Problem ($SVP_c^{(p)}$)

Input : A lattice specified by a basis $\mathbf{B}$

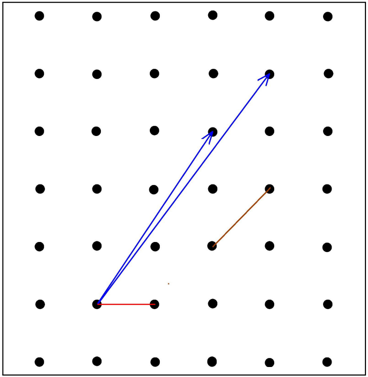
Output : Find a non-zero lattice vector of smallest norm upto some approximation factor c.
i.e. Find $\mathbf{v} \in \mathscr{L} \setminus \{\mathbf{0}\}$ such that $\|\mathbf{v}\| \leq c\|\mathbf{u}\|$ for any other $\mathbf{u} \in \mathscr{L} \setminus \{\mathbf{0}\}$.



Shortest Vector Problem ($SVP_c^{(p)}$)

Input : A lattice specified by a basis $\mathbf{B}$

Output : Find a non-zero lattice vector of smallest norm upto some approximation factor c.
i.e. Find $\mathbf{v} \in \mathscr{L} \setminus \{\mathbf{0}\}$ such that $\|\mathbf{v}\| \leq c\|\mathbf{u}\|$ for any other $\mathbf{u} \in \mathscr{L} \setminus \{\mathbf{0}\}$.

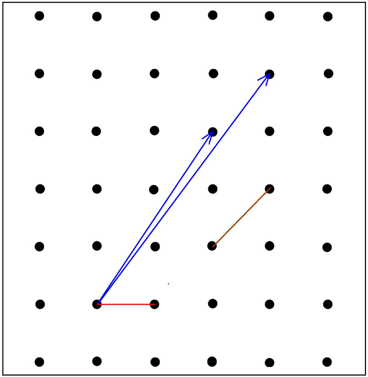


- c : approximation factor.

Shortest Vector Problem ($SVP_c^{(p)}$)

Input : A lattice specified by a basis $\mathbf{B}$

Output : Find a non-zero lattice vector of smallest norm upto some approximation factor c.
i.e. Find $\mathbf{v} \in \mathscr{L} \setminus \{\mathbf{0}\}$ such that
$\|\mathbf{v}\| \leq c\|\mathbf{u}\|$ for any other $\mathbf{u} \in \mathscr{L} \setminus \{\mathbf{0}\}$.

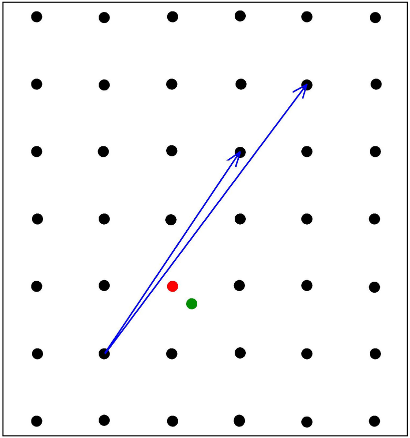


- c : approximation factor.
- $c = 1$: exact version.

Closest Vector Problem ($\mathrm{CVP}_c^{(p)}$)

Input : (i) A lattice specified by a basis **B**, (ii) Target vector **t**

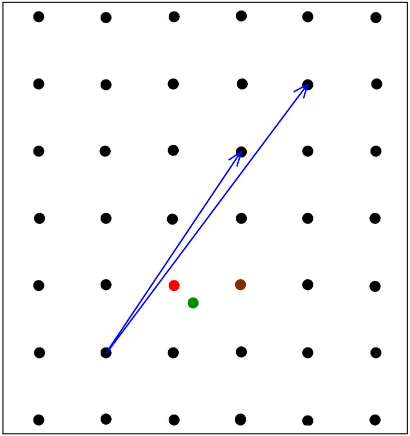
Output : Find a lattice vector closest to **t** upto some approximation factor c.
i.e. Find $\mathbf{v} \in \mathscr{L}$ such that $\|\mathbf{v} - \mathbf{t}\|_p \leq c\|\mathbf{w} - \mathbf{t}\|_p$ for any other $\mathbf{w} \in \mathscr{L}$.



Closest Vector Problem ($CVP_c^{(p)}$)

Input : (i) A lattice specified by a basis **B**, (ii) Target vector **t**

Output : Find a lattice vector closest to **t** upto some approximation factor c.
i.e. Find $\mathbf{v} \in \mathscr{L}$ such that $\|\mathbf{v} - \mathbf{t}\|_p \leq c\|\mathbf{w} - \mathbf{t}\|_p$ for any other $\mathbf{w} \in \mathscr{L}$.

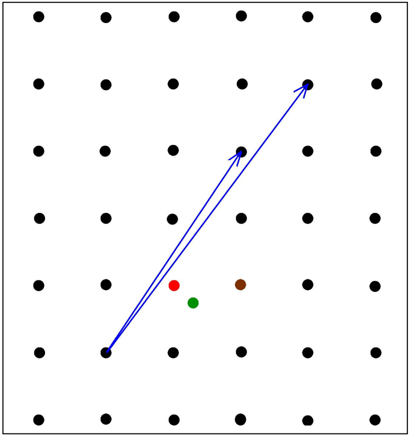


- c : approximation factor.

Closest Vector Problem ($CVP_c^{(p)}$)

Input : (i) A lattice specified by a basis **B**, (ii) Target vector **t**

Output : Find a lattice vector closest to **t** upto some approximation factor c.
i.e. Find $\mathbf{v} \in \mathscr{L}$ such that $\|\mathbf{v} - \mathbf{t}\|_p \leq c\|\mathbf{w} - \mathbf{t}\|_p$ for any other $\mathbf{w} \in \mathscr{L}$.



- c : approximation factor.
- $c = 1$: exact version.

ℓ_p norm and ℓ_p ball

ℓ_p norm and ℓ_p ball

ℓ_p norm of a vector $\mathbf{x} \in \mathbb{R}^n = \|\mathbf{x}\|_p$

$$= \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \text{ for } 1 \le p < \infty$$

$$= \max\{|x_i| : i = 1, \ldots, n\} \text{ for } p = \infty$$

ℓ_p norm and ℓ_p ball

ℓ_p norm of a vector $\mathbf{x} \in \mathbb{R}^n = \|\mathbf{x}\|_p$

$$= \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} \text{ for } 1 \leq p < \infty$$

$$= \max\{|x_i| : i = 1, \ldots, n\} \text{ for } p = \infty$$

Ball : Set of all points within a fixed distance or radius (r) (defined by a metric) from a fixed point or centre ($\mathbf{v}$).

- Closed ball $B_n^{(p)}(\mathbf{v}, r)$
  $= \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{v}\|_p \leq r\}$

ℓ_p norm and ℓ_p ball

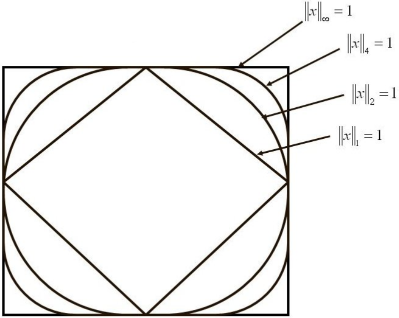
ℓ_p norm of a vector $\mathbf{x} \in \mathbb{R}^n = \|\mathbf{x}\|_p$

$$= \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} \text{ for } 1 \leq p < \infty$$

$$= \max\{|x_i| : i = 1, \ldots, n\} \text{ for } p = \infty$$

Ball : Set of all points within a fixed distance or radius (r) (defined by a metric) from a fixed point or centre ($\mathbf{v}$).

- Closed ball $B_n^{(p)}(\mathbf{v}, r)$
  $= \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{v}\|_p \leq r\}$



Applications of SVP and CVP

Applications of SVP and CVP

- Factoring polynomials over rationals.

Applications of SVP and CVP

- Factoring polynomials over rationals.
- Checking the solvability by radicals.

Applications of SVP and CVP

- Factoring polynomials over rationals.
- Checking the solvability by radicals.
- Solving low-density subset-sum problems.

Applications of SVP and CVP

- Factoring polynomials over rationals.
- Checking the solvability by radicals.
- Solving low-density subset-sum problems.
- Cryptanalysis.

Applications of SVP and CVP

- Factoring polynomials over rationals.
- Checking the solvability by radicals.
- Solving low-density subset-sum problems.
- Cryptanalysis.
- Security of some powerful cryptographic primitives based on the *worst-case* hardness of these or related lattice problems.

Applications of SVP and CVP

- Factoring polynomials over rationals.
- Checking the solvability by radicals.
- Solving low-density subset-sum problems.
- Cryptanalysis.
- Security of some powerful cryptographic primitives based on the *worst-case* hardness of these or related lattice problems.
- CVP in the ℓ_∞ norm is equivalent to integer programming.

Overview

- Preliminary definitions
- Shortest Vector Problem and Closest Vector Problem
- Sieving algorithms for SVP and CVP

Overview

- Preliminary definitions
- Shortest Vector Problem and Closest Vector Problem
- Sieving algorithms for SVP and CVP
  - Prior works
  - AKS sieving algorithm in the ℓ_p norm
  - Linear Sieve
  - Mixed Sieve

Prior Works : Sieving algorithms for SVP and CVP

Euclidean norm

[1] M.Ajtai,R.Kumar and D.Sivakumar, *A sieve algorithm for the shortest lattice vector problem*,STOC,2001.

[2] M.Ajtai,R.Kumar and D.Sivakumar, *Sampling short lattice vectors and the closest vector problem*,CCC,2002.

[3] M.Liu,X.Wang,G.Xu and X.Zheng, *Shortest lattice vectors in the presence of gaps*, IACR Cryptology ePrint Archive,2011.

[4] D.Aggarwal, D.Dadush, O.Regev and N.Stephens-Davidowitz,*Solving the shortest vector problem in 2^n time using Discrete Gaussian sampling*, STOC, 2015.

[5] A.Becker,L.Ducas,N.Gama and T.Laarhoven,*New directions in nearest neighbor searching with applications to lattice sieving*,ACM Symp. on Discrete Algo.,2016.

Prior Works : Sieving algorithms for SVP and CVP

Euclidean norm

- Ajtai,Kumar,Sivakumar(2000[1], 2002[2]) solved SVP and approximate CVP in 2^{cn} time using *randomized sieving*.

---

[1] M.Ajtai,R.Kumar and D.Sivakumar, *A sieve algorithm for the shortest lattice vector problem*,STOC,2001.

[2] M.Ajtai,R.Kumar and D.Sivakumar, *Sampling short lattice vectors and the closest vector problem*,CCC,2002.

[3] M.Liu,X.Wang,G.Xu and X.Zheng, *Shortest lattice vectors in the presence of gaps*, IACR Cryptology ePrint Archive,2011.

[4] D.Aggarwal, D.Dadush, O.Regev and N.Stephens-Davidowitz,*Solving the shortest vector problem in 2^n time using Discrete Gaussian sampling*, STOC, 2015.

[5] A.Becker,L.Ducas,N.Gama and T.Laarhoven,*New directions in nearest neighbor searching with applications to lattice sieving*,ACM Symp. on Discrete Algo.,2016.

Prior Works : Sieving algorithms for SVP and CVP

Euclidean norm

- Ajtai,Kumar,Sivakumar(2000[1], 2002[2]) solved SVP and approximate CVP in 2^{cn} time using *randomized sieving*.
- Fastest algorithm[3] for SVP_c (c a constant) runs in time $2^{0.802n+o(n)}$ (Liu, Wang, Xu, Zheng, 2011).

[1] M.Ajtai,R.Kumar and D.Sivakumar, *A sieve algorithm for the shortest lattice vector problem*,STOC,2001.

[2] M.Ajtai,R.Kumar and D.Sivakumar, *Sampling short lattice vectors and the closest vector problem*,CCC,2002.

[3] M.Liu,X.Wang,G.Xu and X.Zheng, *Shortest lattice vectors in the presence of gaps*, IACR Cryptology ePrint Archive,2011.

[4] D.Aggarwal, D.Dadush, O.Regev and N.Stephens-Davidowitz,*Solving the shortest vector problem in 2^n time using Discrete Gaussian sampling*, STOC, 2015.

[5] A.Becker,L.Ducas,N.Gama and T.Laarhoven,*New directions in nearest neighbor searching with applications to lattice sieving*,ACM Symp. on Discrete Algo.,2016.

Prior Works : Sieving algorithms for SVP and CVP

Euclidean norm

- Ajtai,Kumar,Sivakumar(2000[1], 2002[2]) solved SVP and approximate CVP in 2^{cn} time using *randomized sieving*.
- Fastest algorithm[3] for SVP_c (c a constant) runs in time $2^{0.802n+o(n)}$ (Liu, Wang, Xu, Zheng, 2011).
- Provable algorithms for SVP and CVP based on Discrete Gaussian sampling[4] run in time $2^{n+o(n)}$ (Aggarwal, Dadush, Regev, Stephens-Davidowitz, 2015).

---

[1] M.Ajtai,R.Kumar and D.Sivakumar, *A sieve algorithm for the shortest lattice vector problem*,STOC,2001.

[2] M.Ajtai,R.Kumar and D.Sivakumar, *Sampling short lattice vectors and the closest vector problem*,CCC,2002.

[3] M.Liu,X.Wang,G.Xu and X.Zheng, *Shortest lattice vectors in the presence of gaps*, IACR Cryptology ePrint Archive,2011.

[4] D.Aggarwal, D.Dadush, O.Regev and N.Stephens-Davidowitz,*Solving the shortest vector problem in 2^n time using Discrete Gaussian sampling*, STOC, 2015.

[5] A.Becker,L.Ducas,N.Gama and T.Laarhoven,*New directions in nearest neighbor searching with applications to lattice sieving*,ACM Symp. on Discrete Algo.,2016.

Prior Works : Sieving algorithms for SVP and CVP

Euclidean norm

- Ajtai,Kumar,Sivakumar(2000[1], 2002[2]) solved SVP and approximate CVP in 2^{cn} time using *randomized sieving*.
- Fastest algorithm[3] for SVP_c (c a constant) runs in time $2^{0.802n+o(n)}$ (Liu, Wang, Xu, Zheng, 2011).
- Provable algorithms for SVP and CVP based on Discrete Gaussian sampling[4] run in time $2^{n+o(n)}$ (Aggarwal, Dadush, Regev, Stephens-Davidowitz, 2015).
- Heuristic algorithms[5] for SVP run in time $(3/2)^{n/2}$ (Becker, Ducas, Gama, Laarhoven, 2016).

---

[1] M.Ajtai,R.Kumar and D.Sivakumar, *A sieve algorithm for the shortest lattice vector problem*,STOC,2001.

[2] M.Ajtai,R.Kumar and D.Sivakumar, *Sampling short lattice vectors and the closest vector problem*,CCC,2002.

[3] M.Liu,X.Wang,G.Xu and X.Zheng, *Shortest lattice vectors in the presence of gaps*, IACR Cryptology ePrint Archive,2011.

[4] D.Aggarwal, D.Dadush, O.Regev and N.Stephens-Davidowitz,*Solving the shortest vector problem in* 2^n *time using Discrete Gaussian sampling*, STOC, 2015.

[5] A.Becker,L.Ducas,N.Gama and T.Laarhoven,*New directions in nearest neighbor searching with applications to lattice sieving*,ACM Symp. on Discrete Algo.,2016.

Prior Works : Sieving algorithms for SVP and CVP

Other norms

[1] J.Blömer,S.Naewe,*Sampling methods for shortest vectors,closest vectors and successive minima*,Theoretical Computer Science, 2009.

[2] V.Arvind and P.S.Joglekar, *Some sieving algorithms for lattice problems*,FSTTCS, 2008.

[3] F.Eisenbrand,N.Hähnle and M.Niemeier, *Covering cubes and the closest vector problem*, Annual.Symp. on Computational Geometry, 2011.

[4] D.Aggarwal and P.Mukhopadhyay, *Improved algorithms for the shortest vector problem and the closest vector problem in the infinity norm*, ISAAC, 2018.

Prior Works : Sieving algorithms for SVP and CVP

Other norms

- (Blömer and Naewe,2009)[1] and (Arvind and Joglekar,2008)[2] generalized the AKS algorithm to give exact and approximate algorithms for $\mathrm{SVP}^{(p)}$ and $\mathrm{CVP}^{(p)}_{1+\epsilon}$ with running time $2^{O(n)}$.

[1] J.Blömer,S.Naewe,*Sampling methods for shortest vectors,closest vectors and successive minima*,Theoretical Computer Science, 2009.

[2] V.Arvind and P.S.Joglekar, *Some sieving algorithms for lattice problems*,FSTTCS, 2008.

[3] F.Eisenbrand,N.Hähnle and M.Niemeier, *Covering cubes and the closest vector problem*, Annual.Symp. on Computational Geometry, 2011.

[4] D.Aggarwal and P.Mukhopadhyay, *Improved algorithms for the shortest vector problem and the closest vector problem in the infinity norm*, ISAAC, 2018.

Prior Works : Sieving algorithms for SVP and CVP

Other norms

- (Blömer and Naewe,2009)[1] and (Arvind and Joglekar,2008)[2] generalized the AKS algorithm to give exact and approximate algorithms for $SVP^{(p)}$ and $CVP^{(p)}_{1+\epsilon}$ with running time $2^{O(n)}$.
- Eisenbrand et.al.(2011)[3] gave a $2^{O(n)}\dot{(}\log(1/\epsilon))^n$ algorithm for $CVP^{(\infty)}_{1+\epsilon}$.

[1] J.Blömer,S.Naewe,*Sampling methods for shortest vectors,closest vectors and successive minima*,Theoretical Computer Science, 2009.

[2] V.Arvind and P.S.Joglekar, *Some sieving algorithms for lattice problems*,FSTTCS, 2008.

[3] F.Eisenbrand,N.Hähnle and M.Niemeier, *Covering cubes and the closest vector problem*, Annual.Symp. on Computational Geometry, 2011.

[4] D.Aggarwal and P.Mukhopadhyay, *Improved algorithms for the shortest vector problem and the closest vector problem in the infinity norm*, ISAAC, 2018.

Prior Works : Sieving algorithms for SVP and CVP

Other norms

- (Blömer and Naewe,2009)[1] and (Arvind and Joglekar,2008)[2] generalized the AKS algorithm to give exact and approximate algorithms for $\text{SVP}^{(p)}$ and $\text{CVP}^{(p)}_{1+\epsilon}$ with running time $2^{O(n)}$.
- Eisenbrand et.al.(2011)[3] gave a $2^{O(n)}\dot{(}\log(1/\epsilon))^n$ algorithm for $\text{CVP}^{(\infty)}_{1+\epsilon}$.
- Aggarwal and Mukhopadhyay (2018)[4] improved the running time for exact and approximate $\text{SVP}^{(\infty)}$ and $\text{CVP}^{(\infty)}_c$.

[1] J.Blömer,S.Naewe,*Sampling methods for shortest vectors,closest vectors and successive minima*,Theoretical Computer Science, 2009.

[2] V.Arvind and P.S.Joglekar, *Some sieving algorithms for lattice problems*,FSTTCS, 2008.

[3] F.Eisenbrand,N.Hähnle and M.Niemeier, *Covering cubes and the closest vector problem*, Annual.Symp. on Computational Geometry, 2011.

[4] D.Aggarwal and P.Mukhopadhyay, *Improved algorithms for the shortest vector problem and the closest vector problem in the infinity norm*, ISAAC, 2018.

Hardness results for SVP and CVP

[1] P. van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Technical report, 1981.

[2] I.Dinur,G.Kindler,R.Raz,S.Safra,*Approximating CVP to within almost-polynomial factors is NP-hard*,Combinatorica,2003.

[3] I.Dinur, *Approximating* $SVP^{(\infty)}$ *to within almost-polynomial factors is NP-hard*, Theoretical Computer Science, 2002.

[4] I.Haviv and O.Regev, *Tensor-based hardness of the shortest vector problem to within almost polynomial factors*, STOC, 2007.

[5] P.Mukhopadhyay, *The projection games conjecture and the hardness of approximation of SSAT and related problems*, arXiv:1907.05548, 2019.

Hardness results for SVP and CVP

[1] P. van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Technical report, 1981.

[2] I.Dinur,G.Kindler,R.Raz,S.Safra,*Approximating CVP to within almost-polynomial factors is NP-hard*,Combinatorica,2003.

[3] I.Dinur, *Approximating* $SVP^{(\infty)}$ *to within almost-polynomial factors is NP-hard*, Theoretical Computer Science, 2002.

[4] I.Haviv and O.Regev, *Tensor-based hardness of the shortest vector problem to within almost polynomial factors*, STOC, 2007.

[5] P.Mukhopadhyay, *The projection games conjecture and the hardness of approximation of SSAT and related problems*, arXiv:1907.05548, 2019.

Hardness results for SVP and CVP

- The first NP hardness result for $\mathrm{CVP}^{(p)}$ and $\mathrm{SVP}^{(\infty)}$ was given by van Emde Boas (1981)[1].

[1] P. van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Technical report, 1981.

[2] I.Dinur,G.Kindler,R.Raz,S.Safra,*Approximating CVP to within almost-polynomial factors is NP-hard*,Combinatorica,2003.

[3] I.Dinur, *Approximating* $\mathit{SVP}^{(\infty)}$ *to within almost-polynomial factors is NP-hard*, Theoretical Computer Science, 2002.

[4] I.Haviv and O.Regev, *Tensor-based hardness of the shortest vector problem to within almost polynomial factors*, STOC, 2007.

[5] P.Mukhopadhyay, *The projection games conjecture and the hardness of approximation of SSAT and related problems*, arXiv:1907.05548, 2019.

Hardness results for SVP and CVP

- The first NP hardness result for $\mathrm{CVP}^{(p)}$ and $\mathrm{SVP}^{(\infty)}$ was given by van Emde Boas (1981)[1].
- NP hardness of $\mathrm{CVP}^{(p)}$ and $\mathrm{SVP}^{(\infty)}$ upto a factor of $n^{c/\log\log n}$ (Dinur et al.,2003)[2], (Dinur,2002)[3].

---

[1] P. van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Technical report, 1981.

[2] I.Dinur,G.Kindler,R.Raz,S.Safra,*Approximating CVP to within almost-polynomial factors is NP-hard*,Combinatorica,2003.

[3] I.Dinur, *Approximating* $SVP^{(\infty)}$ *to within almost-polynomial factors is NP-hard*, Theoretical Computer Science, 2002.

[4] I.Haviv and O.Regev, *Tensor-based hardness of the shortest vector problem to within almost polynomial factors*, STOC, 2007.

[5] P.Mukhopadhyay, *The projection games conjecture and the hardness of approximation of SSAT and related problems*, arXiv:1907.05548, 2019.

Hardness results for SVP and CVP

- The first NP hardness result for $\mathrm{CVP}^{(p)}$ and $\mathrm{SVP}^{(\infty)}$ was given by van Emde Boas (1981)[1].
- NP hardness of $\mathrm{CVP}^{(p)}$ and $\mathrm{SVP}^{(\infty)}$ upto a factor of $n^{c/\log\log n}$ (Dinur et al.,2003)[2], (Dinur,2002)[3].
- Hardness of $\mathrm{SVP}^{(p)}$ upto a factor of $2^{(\log n)^{1-\epsilon}}$ assuming $\mathrm{NP} \not\subseteq \mathrm{RTIME}(n^{\mathrm{poly}(\log n)})$ (Haviv and Regev, 2007)[4].

---

[1] P. van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Technical report, 1981.

[2] I.Dinur,G.Kindler,R.Raz,S.Safra,*Approximating CVP to within almost-polynomial factors is NP-hard*,Combinatorica,2003.

[3] I.Dinur, *Approximating* $SVP^{(\infty)}$ *to within almost-polynomial factors is NP-hard*, Theoretical Computer Science, 2002.

[4] I.Haviv and O.Regev, *Tensor-based hardness of the shortest vector problem to within almost polynomial factors*, STOC, 2007.

[5] P.Mukhopadhyay, *The projection games conjecture and the hardness of approximation of SSAT and related problems*, arXiv:1907.05548, 2019.

Hardness results for SVP and CVP

- The first NP hardness result for $\mathrm{CVP}^{(p)}$ and $\mathrm{SVP}^{(\infty)}$ was given by van Emde Boas (1981)[1].
- NP hardness of $\mathrm{CVP}^{(p)}$ and $\mathrm{SVP}^{(\infty)}$ upto a factor of $n^{c/\log\log n}$ (Dinur et al.,2003)[2], (Dinur,2002)[3].
- Hardness of $\mathrm{SVP}^{(p)}$ upto a factor of $2^{(\log n)^{1-\epsilon}}$ assuming $\mathrm{NP} \not\subseteq \mathrm{RTIME}(n^{\mathrm{poly}(\log n)})$ (Haviv and Regev, 2007)[4].
- Hardness of $\mathrm{CVP}^{(p)}$ and $\mathrm{SVP}^{(\infty)}$ upto factor n^c ($c < \frac{1}{2}$) assuming the Projection Games Conjecture (Mukhopadhyay, 2019)[5].

---

[1] P. van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Technical report, 1981.

[2] I.Dinur,G.Kindler,R.Raz,S.Safra,*Approximating CVP to within almost-polynomial factors is NP-hard*,Combinatorica,2003.

[3] I.Dinur, *Approximating* $SVP^{(\infty)}$ *to within almost-polynomial factors is NP-hard*, Theoretical Computer Science, 2002.

[4] I.Haviv and O.Regev, *Tensor-based hardness of the shortest vector problem to within almost polynomial factors*, STOC, 2007.

[5] P.Mukhopadhyay, *The projection games conjecture and the hardness of approximation of SSAT and related problems*, arXiv:1907.05548, 2019.

Sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

Sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

- *S* : Set of N lattice vectors sampled in a ball of radius R.

Sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

- S : Set of N lattice vectors sampled in a ball of radius R.

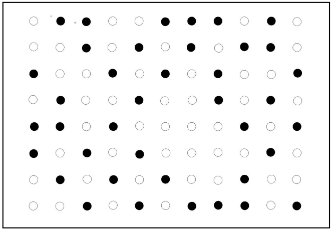


Figure: S

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

- S : Set of N lattice vectors sampled in a ball of radius R.
- **Sieve** : Select a subset C (*centre*) such that
  – $|C|$ is not too large.
  – $\forall \mathbf{u} \in S \setminus C$, there exists $\mathbf{v} \in C$ such that $\|\mathbf{u} - \mathbf{v}\| \leq \gamma R$.

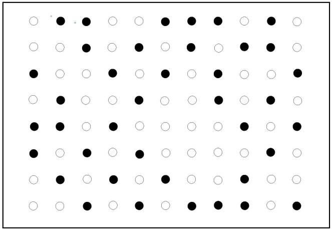


Figure: S

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

- S : Set of N lattice vectors sampled in a ball of radius R.
- **Sieve** : Select a subset C (*centre*) such that
  – $|C|$ is not too large.
  – $\forall \mathbf{u} \in S \setminus C$, there exists $\mathbf{v} \in C$ such that $\|\mathbf{u} - \mathbf{v}\| \leq \gamma R$.

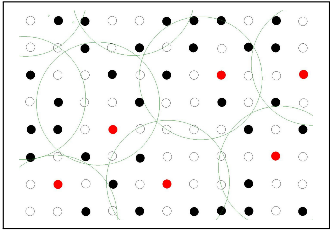


Figure: S during sieving step

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

- *S* : Set of *N* lattice vectors sampled in a ball of radius *R*.
- **Sieve** : Select a subset *C* (*centre*) such that
  – $|C|$ is not too large.
  – $\forall \mathbf{u} \in S \setminus C$, there exists $\mathbf{v} \in C$ such that $\|\mathbf{u} - \mathbf{v}\| \leq \gamma R$

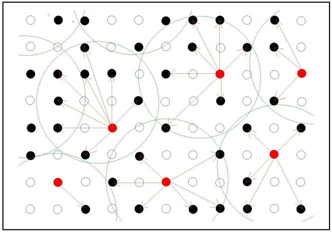


Figure: *S* during sieving step

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

- S : Set of N lattice vectors sampled in a ball of radius R.
- **Sieve** : Select a subset C (*centre*) such that
  – $|C|$ is not too large
  – $\forall \mathbf{u} \in S \setminus C$, there exists $\mathbf{v} \in C$ such that $\|\mathbf{u} - \mathbf{v}\| \leq \gamma R$

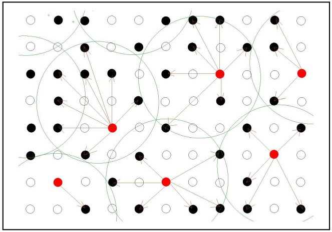


Figure: S during sieving step

- After one sieving step : $S' \subseteq \mathscr{L} \cap B(\gamma R)$ with $|S'| = |S| - |C|$.

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

- S : Set of N lattice vectors sampled in a ball of radius R.
- **Sieve** : Select a subset C (*centre*) such that
  – $|C|$ is not too large
  – $\forall \mathbf{u} \in S \setminus C$, there exists $\mathbf{v} \in C$ such that $\|\mathbf{u} - \mathbf{v}\| \leq \gamma R$

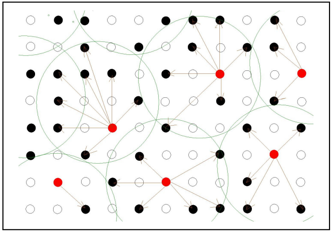


Figure: S during sieving step

- After one sieving step : $S' \subseteq \mathscr{L} \cap B(\gamma R)$ with $|S'| = |S| - |C|$.
- Polynomial number of sieve operations gives lattice vectors of norm at most $r_0\lambda_1(\mathscr{L})$ for some constant r_0.

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

Issues !!

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

Issues !!

- Cannot ensure the distribution of the vectors after sieving step.

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

Issues !!

- Cannot ensure the distribution of the vectors after sieving step.
- May end up with all zero vectors.

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

Issues !!

- Cannot ensure the distribution of the vectors after sieving step.
- May end up with all zero vectors.

Solution

AKS sieving algorithm in the ℓ_p norm : AKS sieve

(Blömer and Naewe, 2009)

Issues !!

- Cannot ensure the distribution of the vectors after sieving step.
- May end up with all zero vectors.

Solution

- For each sampled vector, add a randomly chosen perturbation vector.

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

I. Initial Sampling

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

I. Initial Sampling

- Sample N perturbation vectors $\{\mathbf{e}_i\}$ uniformly from a ball of radius R_0.

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

I. Initial Sampling

- Sample N perturbation vectors $\{\mathbf{e}_i\}$ uniformly from a ball of radius R_0.
- Calculate $\mathbf{y}_i \equiv \mathbf{e}_i \mod \mathscr{P}(\mathbf{B})$ (perturbed vectors).

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

I. Initial Sampling

- Sample N perturbation vectors $\{\mathbf{e}_i\}$ uniformly from a ball of radius R_0.
- Calculate $\mathbf{y}_i \equiv \mathbf{e}_i \mod \mathscr{P}(\mathbf{B})$ (perturbed vectors).

II. AKS Sieve

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

I. Initial Sampling

- Sample N perturbation vectors $\{\mathbf{e}_i\}$ uniformly from a ball of radius R_0.
- Calculate $\mathbf{y}_i \equiv \mathbf{e}_i \mod \mathscr{P}(\mathbf{B})$ (perturbed vectors).

II. AKS Sieve

- Polynomial number of sieving operations.

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

I. Initial Sampling

- Sample N perturbation vectors $\{\mathbf{e}_i\}$ uniformly from a ball of radius R_0.
- Calculate $\mathbf{y}_i \equiv \mathbf{e}_i \mod \mathscr{P}(\mathbf{B})$ (perturbed vectors).

II. AKS Sieve

- Polynomial number of sieving operations.
- Sieve function makes test only on $\mathbf{y}_i$.

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

I. Initial Sampling

- Sample N perturbation vectors $\{\mathbf{e}_i\}$ uniformly from a ball of radius R_0.
- Calculate $\mathbf{y}_i \equiv \mathbf{e}_i \mod \mathscr{P}(\mathbf{B})$ (perturbed vectors).

II. AKS Sieve

- Polynomial number of sieving operations.
- Sieve function makes test only on $\mathbf{y}_i$.
- Same operations get reflected on the corresponding lattice vectors.

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

I. Initial Sampling

- Sample N perturbation vectors $\{\mathbf{e}_i\}$ uniformly from a ball of radius R_0.
- Calculate $\mathbf{y}_i \equiv \mathbf{e}_i \mod \mathscr{P}(\mathbf{B})$ (perturbed vectors).

II. AKS Sieve

- Polynomial number of sieving operations.
- Sieve function makes test only on $\mathbf{y}_i$.
- Same operations get reflected on the corresponding lattice vectors.

III. Pair-wise difference

AKS sieving algorithm in the ℓ_p norm

(Blömer and Naewe, 2009)

I. Initial Sampling

- Sample N perturbation vectors $\{\mathbf{e}_i\}$ uniformly from a ball of radius R_0.
- Calculate $\mathbf{y}_i \equiv \mathbf{e}_i \mod \mathscr{P}(\mathbf{B})$ (perturbed vectors).

II. AKS Sieve

- Polynomial number of sieving operations.
- Sieve function makes test only on $\mathbf{y}_i$.
- Same operations get reflected on the corresponding lattice vectors.

III. Pair-wise difference

- Take pair-wise difference of the vectors in the final set and **output the one with the smallest norm**.

Complexity of AKS sieve

(Blömer and Naewe, 2009)

Complexity of AKS sieve

(Blömer and Naewe, 2009)

- Quadratic sieve : Usually the most expensive part in the algorithm.

Complexity of AKS sieve

(Blömer and Naewe, 2009)

- Quadratic sieve : Usually the most expensive part in the algorithm.
- Space complexity : $O(N)$ where $N = 2^{cn}$, for some constant c.

Complexity of AKS sieve

(Blömer and Naewe, 2009)

- Quadratic sieve : Usually the most expensive part in the algorithm.
- Space complexity : $O(N)$ where $N = 2^{cn}$, for some constant c.
- Time complexity : $O(N^2)$, i.e. 2^{2cn}.

Faster sieving algorithms in the ℓ_p norm : Linear sieve

(Mukhopadhyay, 2019)

P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Faster sieving algorithms in the ℓ_p norm : Linear sieve

(Mukhopadhyay, 2019)

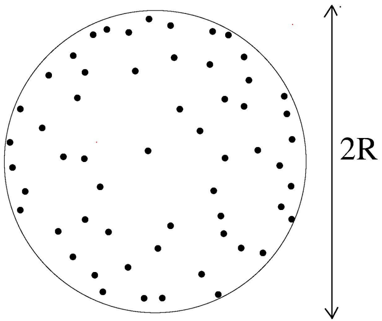
- Partition $B(R)$ into hypercubes such that their longest diagonal has length r.

P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Faster sieving algorithms in the ℓ_p norm : Linear sieve

(Mukhopadhyay, 2019)

- Partition $B(R)$ into hypercubes such that their longest diagonal has length r.

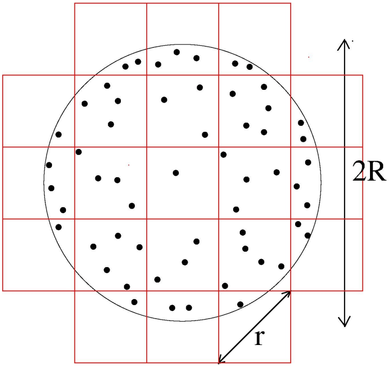


P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Faster sieving algorithms in the ℓ_p norm : Linear sieve

(Mukhopadhyay, 2019)

- Partition $B(R)$ into hypercubes such that their longest diagonal has length r.

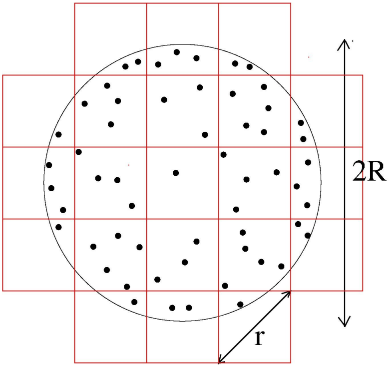


P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Faster sieving algorithms in the ℓ_p norm : Linear sieve

(Mukhopadhyay, 2019)

- Partition $B(R)$ into hypercubes such that their longest diagonal has length r.
  - $\|\mathbf{u} - \mathbf{v}\| \leq r$ for any $\mathbf{u}, \mathbf{v}$ in same region.

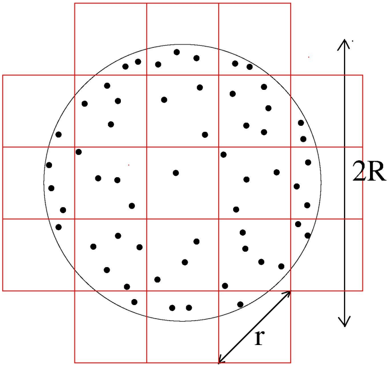


P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Faster sieving algorithms in the ℓ_p norm : Linear sieve

(Mukhopadhyay, 2019)

- Partition $B(R)$ into hypercubes such that their longest diagonal has length r.
  - $\|\mathbf{u} - \mathbf{v}\| \leq r$ for any $\mathbf{u}, \mathbf{v}$ in same region.
  - Map each vector to a region by looking at the co-ordinates : $n + o(1)$ time.

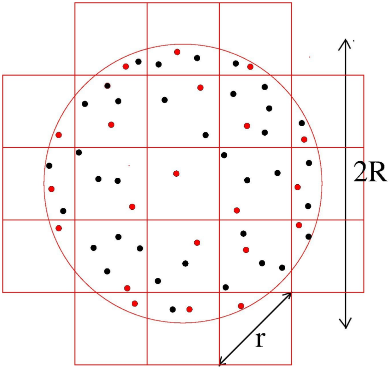


P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Faster sieving algorithms in the ℓ_p norm : Linear sieve

(Mukhopadhyay, 2019)

- Partition $B(R)$ into hypercubes such that their longest diagonal has length r.
  - $\|\mathbf{u} - \mathbf{v}\| \leq r$ for any $\mathbf{u}, \mathbf{v}$ in same region.
  - Map each vector to a region by looking at the co-ordinates : $n + o(1)$ time.
- At most one centre in each hypercube.

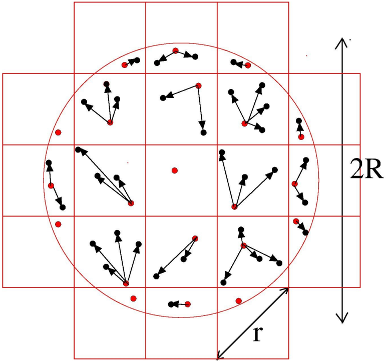


P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Faster sieving algorithms in the ℓ_p norm : Linear sieve

(Mukhopadhyay, 2019)

- Partition $B(R)$ into hypercubes such that their longest diagonal has length r.
  - $\|\mathbf{u}-\mathbf{v}\| \leq r$ for any $\mathbf{u}, \mathbf{v}$ in same region.
  - Map each vector to a region by looking at the co-ordinates : $n + o(1)$ time.
- At most one centre in each hypercube.
- Take difference.



P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Number of partitions (hypercubes)

Number of partitions (hypercubes)

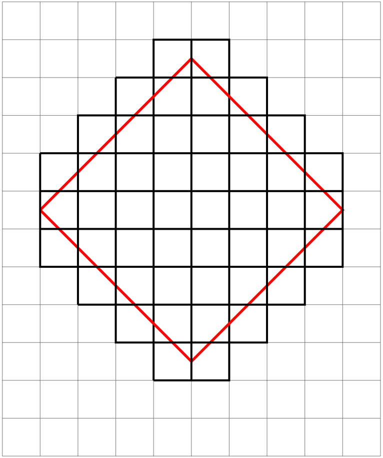
- Determines number of centres and number of sampled vectors.

Number of partitions (hypercubes)

- Determines number of centres and number of sampled vectors.
- $O\left(\left(2+\frac{2}{\gamma}\right)^n\right)$ if $r = \gamma R$.

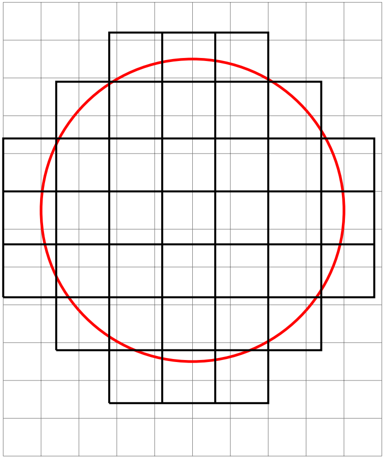
Number of partitions (hypercubes)

- Determines number of centres and number of sampled vectors.
- $O\left(\left(2 + \frac{2}{\gamma}\right)^n\right)$ if $r = \gamma R$.



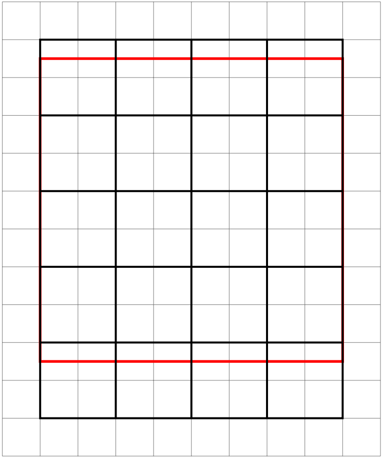
Number of partitions (hypercubes)

- Determines number of centres and number of sampled vectors.
- $O\left(\left(2+\frac{2}{\gamma}\right)^n\right)$ if $r = \gamma R$.



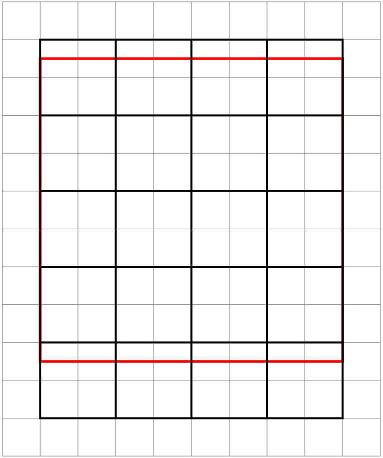
Number of partitions (hypercubes)

- Determines number of centres and number of sampled vectors.
- $O\left(\left(2+\frac{2}{\gamma}\right)^n\right)$ if $r = \gamma R$.



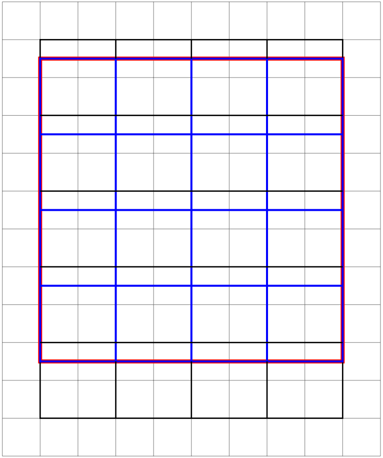
Number of partitions (hypercubes)

- Determines number of centres and number of sampled vectors.
- $O\left(\left(2+\frac{2}{\gamma}\right)^n\right)$ if $r = \gamma R$.
- Depends on how each axis is divided into intervals.



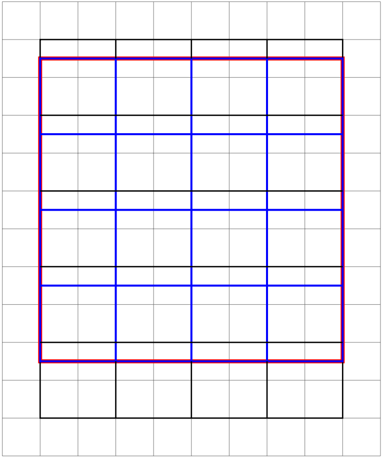
Number of partitions (hypercubes)

- Determines number of centres and number of sampled vectors.
- $O\left(\left(2+\frac{2}{\gamma}\right)^n\right)$ if $r = \gamma R$.
- Depends on how each axis is divided into intervals.



Number of partitions (hypercubes)

- Determines number of centres and number of sampled vectors.
- $O\left(\left(2+\frac{2}{\gamma}\right)^n\right)$ if $r = \gamma R$.
- Depends on how each axis is divided into intervals.
- $O\left(\left\lceil \frac{2}{\gamma} \right\rceil^n\right)$ [1].



[1] D.Aggarwal and P.Mukhopadhyay, *Improved algorithms for the shortest vector problem and the closest vector problem in the infinity norm*, ISAAC, 2018.

Comparison of space and time complexity

(Linear sieve)

Comparison of space and time complexity

(Linear sieve)

$$1 \leq p \leq \infty$$

ALGORITHM	SPACE	TIME
Blömer and Naewe,2009	$2^{2.023n+o(n)}$	$2^{3.849n+o(n)}$
Mukhopadhyay,2019	$2^{2.751n+o(n)}$	$2^{2.751n+o(n)}$

Comparison of space and time complexity

(Linear sieve)

$1 \leq p \leq \infty$

ALGORITHM	SPACE	TIME
Blömer and Naewe,2009	$2^{2.023n+o(n)}$	$2^{3.849n+o(n)}$
Mukhopadhyay,2019	$2^{2.751n+o(n)}$	$2^{2.751n+o(n)}$

$p = 2$

ALGORITHM	SPACE	TIME
Hanrot,Pujol,Stehle,2011[1]	$2^{1.407n+o(n)}$	$2^{2.571n+o(n)}$
Mukhopadhyay,2019	$2^{2.49n+o(n)}$	$2^{2.49n+o(n)}$

[1] G.Hanrot,X.Pujol,D.Stehle,*Algorithms for the shortest and closest lattice vector problems.*,International Conference on Coding and Cryptology,2011.

Comparison of space and time complexity

(Linear sieve)

$1 \leq p \leq \infty$

ALGORITHM	SPACE	TIME
Blömer and Naewe,2009	$2^{2.023n+o(n)}$	$2^{3.849n+o(n)}$
Mukhopadhyay,2019	$2^{2.751n+o(n)}$	$2^{2.751n+o(n)}$

$p = 2$

ALGORITHM	SPACE	TIME
Hanrot,Pujol,Stehle,2011[1]	$2^{1.407n+o(n)}$	$2^{2.571n+o(n)}$
Mukhopadhyay,2019	$2^{2.49n+o(n)}$	$2^{2.49n+o(n)}$

$p = \infty$

ALGORITHM	SPACE	TIME
Mukhopadhyay,2019	$2^{2.443n+o(n)}$	$2^{2.443n+o(n)}$

[1] G.Hanrot,X.Pujol,D.Stehle,*Algorithms for the shortest and closest lattice vector problems.*,International Conference on Coding and Cryptology,2011.

Mixed sieving

(Mukhopadhyay, 2019)

P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Mixed sieving

(Mukhopadhyay, 2019)

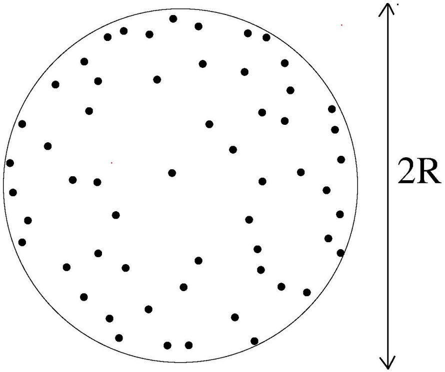
Linear sieve + Quadratic sieve

P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Mixed sieving

(Mukhopadhyay, 2019)

Linear sieve + Quadratic sieve

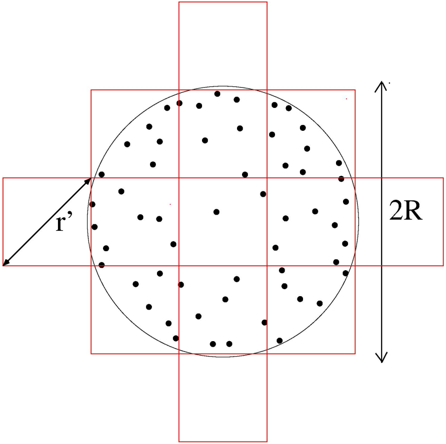


P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Mixed sieving

(Mukhopadhyay, 2019)

Linear sieve + Quadratic sieve

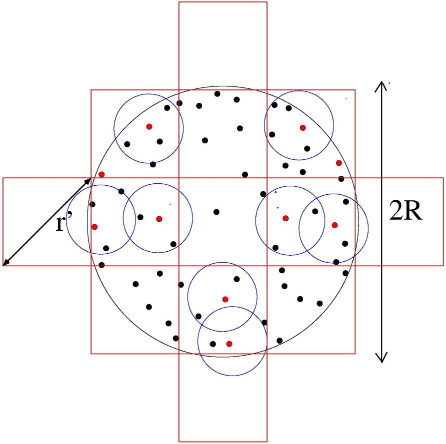


P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Mixed sieving

(Mukhopadhyay, 2019)

Linear sieve + Quadratic sieve



P.Mukhopadhyay, *Faster provable sieving algorithms for the Shortest Vector Problem and the Closest Vector Problem on lattices in ell_p norm*,arXiv:1907.04406, 2019.

Comparison of space and time complexity

(Mixed sieve)

$p = 2$

ALGORITHM	SPACE	TIME
List sieve,2011[2]	$2^{1.233n+o(n)}$	$2^{2.465n+o(n)}$
Mukhopadhyay,2019	$2^{2.25n+o(n)}$	$2^{2.25n+o(n)}$
Aggarwal et al.,2015[3]	2^n	2^n

[1] D.Micciancio, P.Voulgaris, *Faster exponential time algorithms for the shortest vector problem.*, SODA, 2010.

[2] G.Hanrot,X.Pujol,D.Stehle,*Algorithms for the shortest and closest lattice vector problems.*,International Conference on Coding and Cryptology,2011.

[3] D.Aggarwal, D.Dadush, O.Regev and N.Stephens-Davidowitz,*Solving the shortest vector problem in 2^n time using Discrete Gaussian sampling*, STOC, 2015.

Approximation algorithms for large constant approximation factor

Approximation algorithms for large constant approximation factor

- SVP

Approximation algorithms for large constant approximation factor

- SVP
  - Skip the last step of exact algorithm.

Approximation algorithms for large constant approximation factor

- SVP
  - Skip the last step of exact algorithm.
  - Sample – Sieve – Return a non-zero vector.

Approximation algorithms for large constant approximation factor

- SVP
  - Skip the last step of exact algorithm.
  - Sample – Sieve – Return a non-zero vector.
- CVP

Approximation algorithms for large constant approximation factor

- SVP
  - Skip the last step of exact algorithm.
  - Sample – Sieve – Return a non-zero vector.
- CVP
  - Reduction from approximate CVP to approximate SVP (Blömer and Naewe,2009).

Comparison of space and time complexity

(Approximation algorithm)

Comparison of space and time complexity

(Approximation algorithm)

$$1 \leq p \leq \infty$$

ALGORITHM	SPACE	TIME
Blömer and Naewe,2009	$2^{1.586n+o(n)}$	$2^{3.169n+o(n)}$
Mukhopadhyay,2019	$2^{2.001n+o(n)}$	$2^{2.001n+o(n)}$

Comparison of space and time complexity

(Approximation algorithm)

$1 \leq p \leq \infty$

ALGORITHM	SPACE	TIME
Blömer and Naewe,2009	$2^{1.586n+o(n)}$	$2^{3.169n+o(n)}$
Mukhopadhyay,2019	$2^{2.001n+o(n)}$	$2^{2.001n+o(n)}$

$p = 2$

ALGORITHM	SPACE	TIME
Liu,Wang,Xu and Zheng,2011[1]	$2^{0.401n+o(n)}$	$2^{0.802n+o(n)}$
Mukhopadhyay,2019	$2^{1.73n+o(n)}$	$2^{1.73n+o(n)}$

[1] M.Liu,X.Wang,G.Xu and X.Zheng, *Shortest lattice vectors in the presence of gaps*, IACR Cryptology ePrint Archive,2011.

Comparison of space and time complexity

(Approximation algorithm)

$1 \leq p \leq \infty$

ALGORITHM	SPACE	TIME
Blömer and Naewe,2009	$2^{1.586n+o(n)}$	$2^{3.169n+o(n)}$
Mukhopadhyay,2019	$2^{2.001n+o(n)}$	$2^{2.001n+o(n)}$

$p = 2$

ALGORITHM	SPACE	TIME
Liu,Wang,Xu and Zheng,2011[1]	$2^{0.401n+o(n)}$	$2^{0.802n+o(n)}$
Mukhopadhyay,2019	$2^{1.73n+o(n)}$	$2^{1.73n+o(n)}$

$p = \infty$

ALGORITHM	SPACE	TIME
Aggarwal and Mukhopadhyay,2018[2]	$2^{1.585n+o(n)}$	$2^{1.585n+o(n)}$

[1] M.Liu,X.Wang,G.Xu and X.Zheng, *Shortest lattice vectors in the presence of gaps*, IACR Cryptology ePrint Archive,2011.

[2] D.Aggarwal and P.Mukhopadhyay, *Improved algorithms for the shortest vector problem and the closest vector problem in the infinity norm*, ISAAC, 2018.

Thank You