Faster Enumeration-based Lattice Reduction: Root Hermite Factor $k^{1/(2k)}$ in Time $k^{k/8+o(k)}$

Martin R. Albrecht[1], Shi Bai[2], Pierre-Alain Fouque[3], Paul Kirchner[3], Damien Stehlé[4] and **Weiqiang Wen**[3]

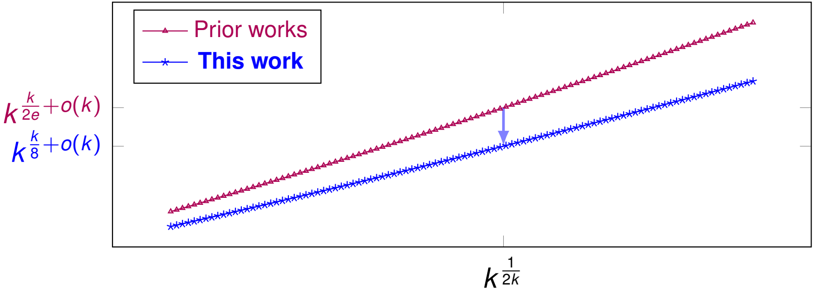
[1] Royal Holloway, University of London [2] Florida Atlantic University

[3] Rennes Univ [4] ENS de Lyon

CRYPTO 2020

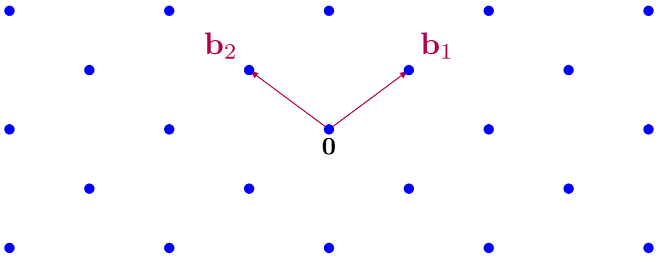
What is this work about?

Enumeration-based lattice reduction algorithms



- In case of input lattices of
  - **large** dimension: **proved** under a **heuristic** assumption;
  - **small** dimension: **simulation** still works for a variant algorithm.

Lattices

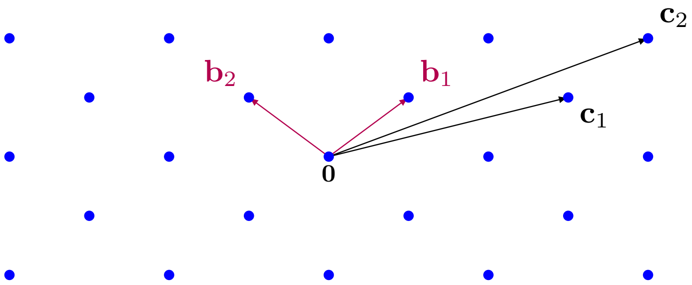


A definition of lattice

Given $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \subseteq \mathbb{Q}^m$ a set of linearly independent vectors, the lattice $\mathcal{L}$ spanned by the $\mathbf{b}_i$'s is

$$\mathcal{L}(\mathbf{B}) = \left\{ \sum_{i \in [n]} u_i \mathbf{b}_i \;:\; \mathbf{u} \in \mathbb{Z}^n \right\}.$$

Lattices

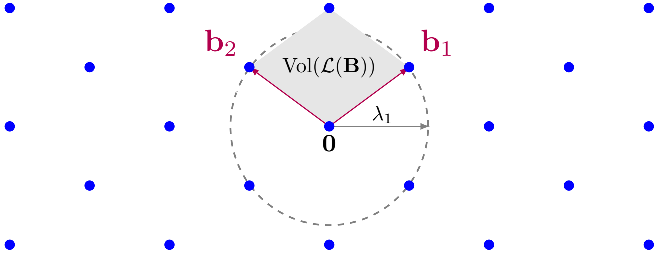


A definition of lattice

Given $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \subseteq \mathbb{Q}^m$ a set of linearly independent vectors, the lattice $\mathcal{L}$ spanned by the $\mathbf{b}_i$'s is

$$\mathcal{L}(\mathbf{B}) = \left\{ \sum_{i \in [n]} u_i \mathbf{b}_i \ : \ \mathbf{u} \in \mathbb{Z}^n \right\}.$$

Invariants in lattices



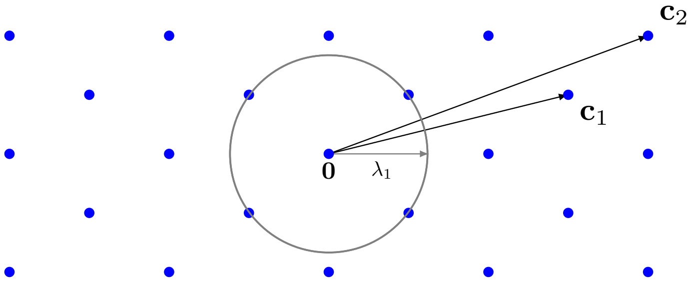
First minimum

$\lambda_1(\mathcal{L}) = \min\{\|\mathbf{b}\| : \mathbf{b} \in \mathcal{L} \backslash \{\mathbf{0}\}\}$.

Volume of lattice

$\mathrm{Vol}(\mathcal{L}(\mathbf{B})) = \sqrt{\det(\mathbf{B}^{\mathrm{T}}\mathbf{B})}$ for any basis $\mathbf{B}$.

Lattice problems

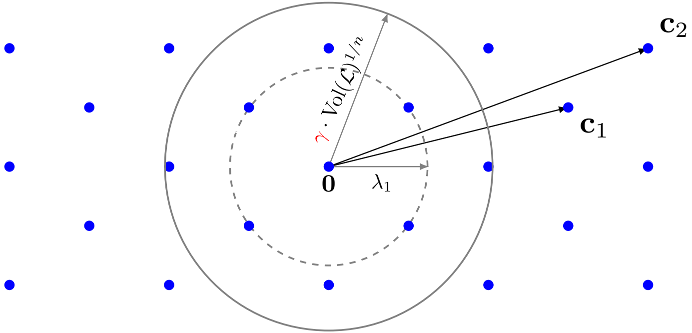


Shortest vector problem (SVP)

Given $\mathbf{B} \subseteq \mathbb{Q}^m$ a basis of the lattice $\mathcal{L}$, it asks to find a vector $\mathbf{s}$ in the lattice such that

$$\|\mathbf{s}\| = \lambda_1(\mathcal{L}).$$

Lattice problems



SVP

Given $\mathbf{B} \subseteq \mathbb{Q}^m$ a basis of the lattice $\mathcal{L}$, finds a vector $\mathbf{s}$ in the lattice such that

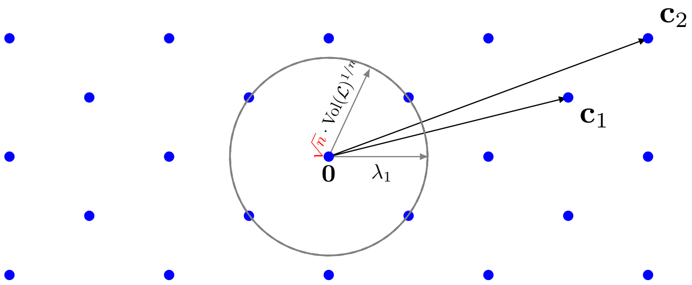
$$\|\mathbf{s}\| = \lambda_1(\mathcal{L}).$$

γ-Hermite SVP (γ-HSVP)

Given $\mathbf{B} \subseteq \mathbb{Q}^m$ a basis of the lattice $\mathcal{L}$, finds a non-zero vector $\mathbf{s}$ in the lattice such that

$$\|\mathbf{s}\| \leq \gamma \cdot \mathrm{Vol}(\mathcal{L})^{\frac{1}{n}}.$$

Lattice problems



Minkowski's theorem: $\mathrm{SVP} \Rightarrow \sqrt{n}$-$\mathrm{HSVP}$.

$(\lambda_1 \leq \sqrt{n} \cdot \mathrm{Vol}(\mathcal{L})^{1/n})$

SVP

Given **B** a basis of $\mathcal{L}$, finds a non-zero vector **s** in $\mathcal{L}$ such that

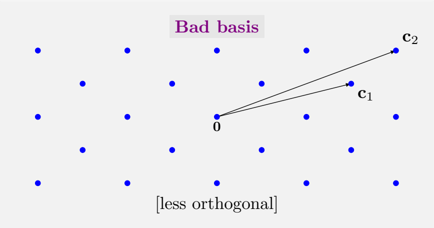
$$\|\mathbf{s}\| = \lambda_1(\mathcal{L}).$$

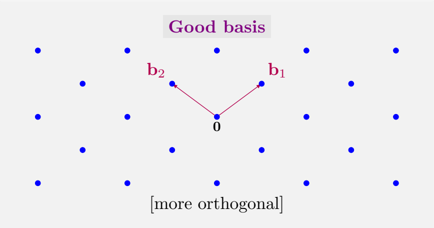
γ-Hermite SVP (γ-HSVP)

Given **B** a basis of $\mathcal{L}$, finds a non-zero vector **s** in $\mathcal{L}$ such that

$$\|\mathbf{s}\| \leq \gamma \cdot \mathrm{Vol}(\mathcal{L})^{\frac{1}{n}}.$$

Best known solution: reduce the basis



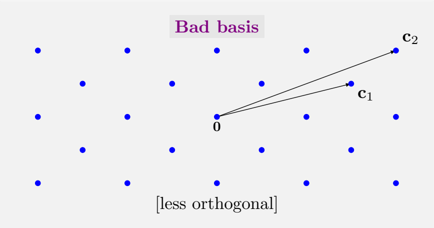


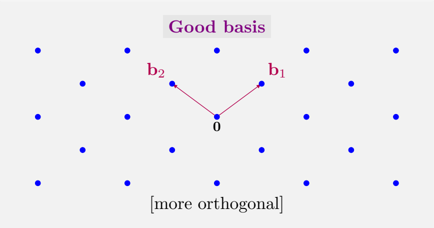
Hermite factor

Given $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \subseteq \mathbb{Q}^m$ a basis of the lattice $\mathcal{L}$, its Hermite factor is

$$\mathsf{HF}(\mathbf{B}) = \frac{\|\mathbf{b}_1\|}{\mathrm{Vol}(\mathcal{L})^{\frac{1}{n}}}.$$

Best known solution: reduce the basis





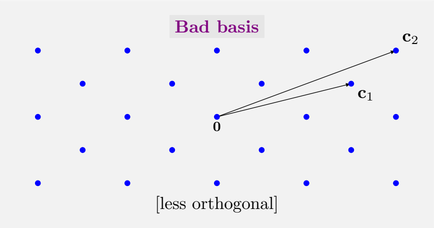
The BKZ lattice reduction is the most practical algorithm to achieve such task!

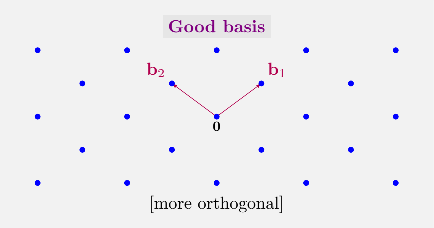
Hermite factor

Given $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \subseteq \mathbb{Q}^m$ a basis of the lattice $\mathcal{L}$, its Hermite factor is

$$\mathsf{HF}(\mathbf{B}) = \frac{\|\mathbf{b}_1\|}{\mathrm{Vol}(\mathcal{L})^{\frac{1}{n}}}.$$

Introduce root Hermite factor to quantify lattice reduction





The BKZ lattice reduction is the most practical algorithm to achieve such task!

Hermite factor

Given $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \subseteq \mathbb{Q}^m$ a basis of the lattice $\mathcal{L}$, its Hermite factor is

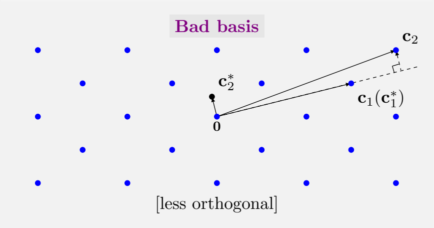
$$\mathsf{HF}(\mathbf{B}) = \frac{\|\mathbf{b}_1\|}{\mathrm{Vol}(\mathcal{L})^{\frac{1}{n}}}.$$

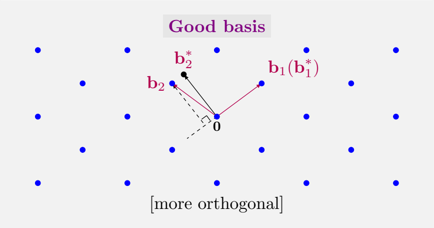
Root Hermite factor

Given $\mathbf{B} \subseteq \mathbb{Q}^m$ a basis of the lattice $\mathcal{L}$, its root Hermite factor is

$$\mathsf{RHF}(\mathbf{B}) = \mathsf{HF}(\mathbf{B})^{\frac{1}{n-1}}.$$

Gram-Schmidt orthogonalization



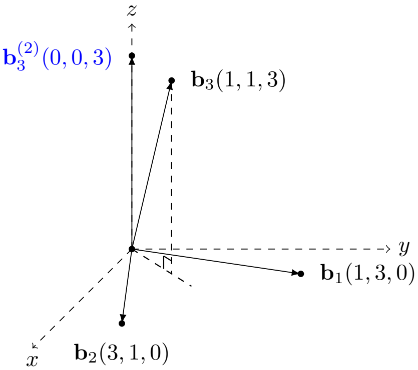


The BKZ lattice reduction is the most practical algorithm to achieve such task!

Gram-Schmidt orthogonalization

A matrix $\mathbf{B}^* = (\mathbf{b}_1^*, ..., \mathbf{b}_n^*)$ is the Gram-Schmidt orthogonalization of $\mathbf{B}$, if $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*$, where $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2}$.

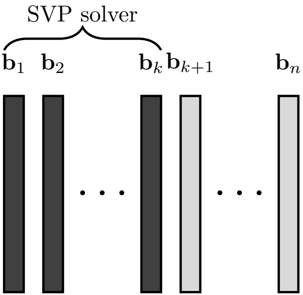
Orthogonal projection



Notation of projection

Given a basis $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n) \in \mathbb{Q}^m$, we let $\mathbf{b}_i^{(j)}$ denote the orthogonal projection over $(\mathbf{b}_1, \cdots, \mathbf{b}_j)^\perp$ of $\mathbf{b}_i$.

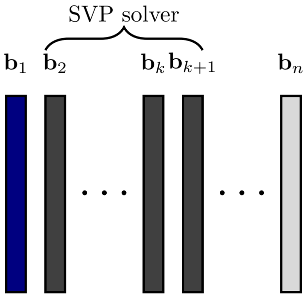
The BKZ algorithm [SE94]



Notation of projection

Given a basis $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n) \in \mathbb{Q}^m$, we let $\mathbf{b}_i^{(j)}$ denote the orthogonal projection over $(\mathbf{b}_1, \cdots, \mathbf{b}_j)^\perp$ of $\mathbf{b}_i$.

The BKZ algorithm [SE94]

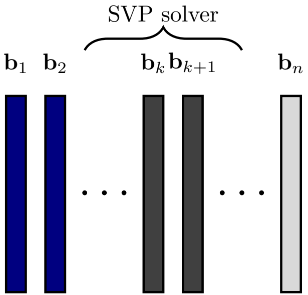


$$\|\mathbf{b}_1^*\| = \|\mathbf{b}_1\| = \lambda_1\left(\mathcal{L}\left(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_k\right)\right)$$

Notation of projection

Given a basis $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n) \in \mathbb{Q}^m$, we let $\mathbf{b}_i^{(j)}$ denote the orthogonal projection over $(\mathbf{b}_1, \cdots, \mathbf{b}_j)^\perp$ of $\mathbf{b}_i$.

The BKZ algorithm [SE94]

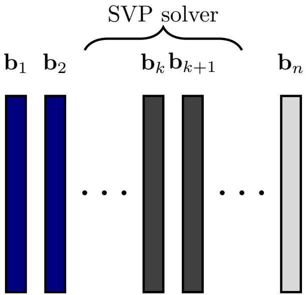


$$\|\mathbf{b}_2^*\| = \|\mathbf{b}_2^{(1)}\| = \lambda_1\left(\mathcal{L}\left(\mathbf{b}_2^{(1)}, \mathbf{b}_3^{(1)}, \cdots, \mathbf{b}_{k+1}^{(1)}\right)\right)$$

Notation of projection

Given a basis $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n) \in \mathbb{Q}^m$, we let $\mathbf{b}_i^{(j)}$ denote the orthogonal projection over $(\mathbf{b}_1, \cdots, \mathbf{b}_j)^\perp$ of $\mathbf{b}_i$.

The BKZ algorithm [SE94]

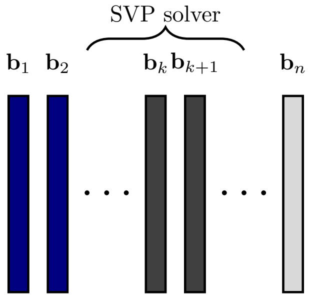


$$\|\mathbf{b}_2^*\| = \|\mathbf{b}_2^{(1)}\| = \lambda_1\left(\mathcal{L}\left(\mathbf{b}_2^{(1)}, \mathbf{b}_3^{(1)}, \cdots, \mathbf{b}_{k+1}^{(1)}\right)\right)$$

The two practical SVP solver families

	Sieve [BDGL16]	Enumeration [Kan83; FP83; HS07; GNR10]
Space	$\exp(k)$	$\mathrm{poly}(k)$
Time	$2^{0.292k+o(k)}$	$k^{k/(2e)+o(k)}$ ($\approx k^{0.184k}$)

The BKZ algorithm [SE94]

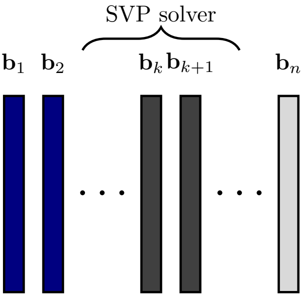


$$\|\mathbf{b}_2^*\| = \|\mathbf{b}_2^{(1)}\| = \lambda_1\left(\mathcal{L}\left(\mathbf{b}_2^{(1)}, \mathbf{b}_3^{(1)}, \cdots, \mathbf{b}_{k+1}^{(1)}\right)\right)$$

The two practical SVP solver families

	Sieve [BDGL16]	Enumeration [Kan83; FP83; HS07; GNR10]
Space	$\exp(k)$	$\mathrm{poly}(k)$
Time	$2^{0.292k+o(k)}$	$k^{k/(2e)+o(k)}$ $(\approx k^{0.184k})$

The prior results and our result (informal)

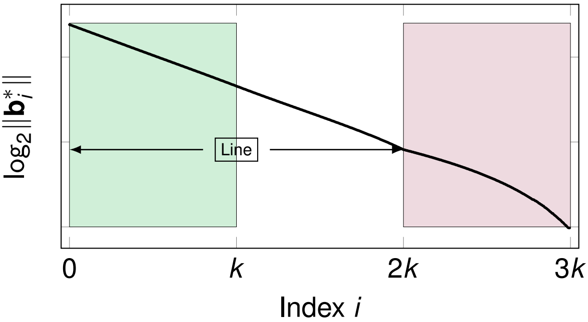


$$\|\mathbf{b}_2^*\| = \|\mathbf{b}_2^{(1)}\| = \lambda_1\left(\mathcal{L}\left(\mathbf{b}_2^{(1)}, \mathbf{b}_3^{(1)}, \cdots, \mathbf{b}_{k+1}^{(1)}\right)\right)$$

Performance of enumeration-based (SD)BKZ and ours

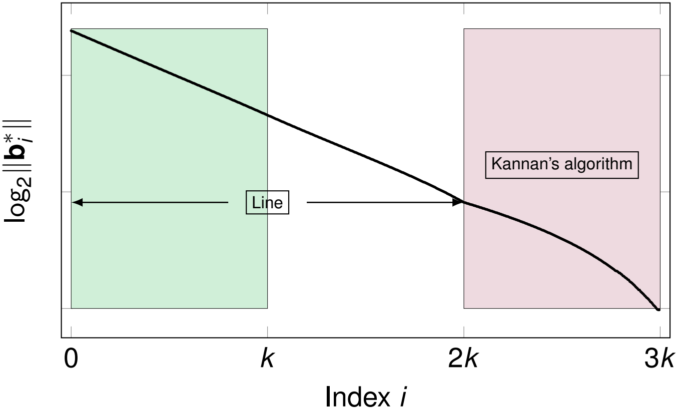
	(SD)BKZ [HPS11; MW16; Neu17]	This work (informally)
RHF	$k^{1/(2k)}$	$k^{1/(2k)}$
Time	$k^{k/(2e)+o(k)}$	$k^{k/8+o(k)}$

Observation on BKZ and SDBKZ reduced bases



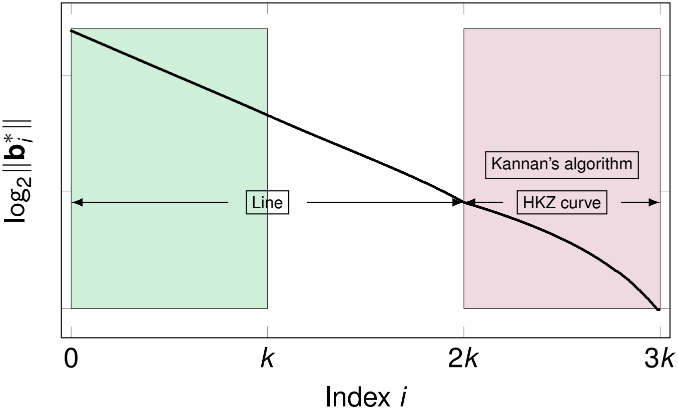
Study of $\delta_i = \|\mathbf{b}_i\| / \|\mathbf{b}_{i+1}\|$ for $i < n - k$	
BKZ	SDBKZ (in this work)
[This work, Appendix]: δ_i is not fixed. (E.g., it does not give a line.)	[MW16]*: fixed $\delta_i = \gamma^{2/(k-1)}$, given γ-HSVP on k-dim lattice.

The SDBKZ reduced basis



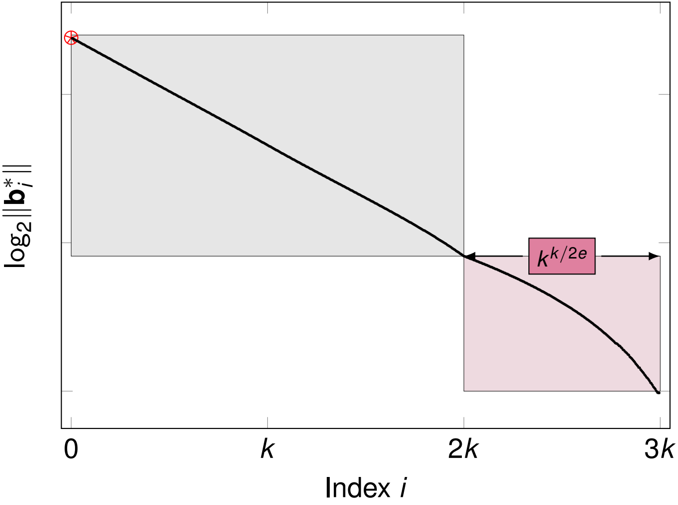
- $\mathrm{Enum\_Cost}$('first block') = $k^{k/8+o(k)}$;
- $\mathrm{Enum\_Cost}$('last block') = $k^{k/(2e)+o(k)}$.

The SDBKZ reduced basis

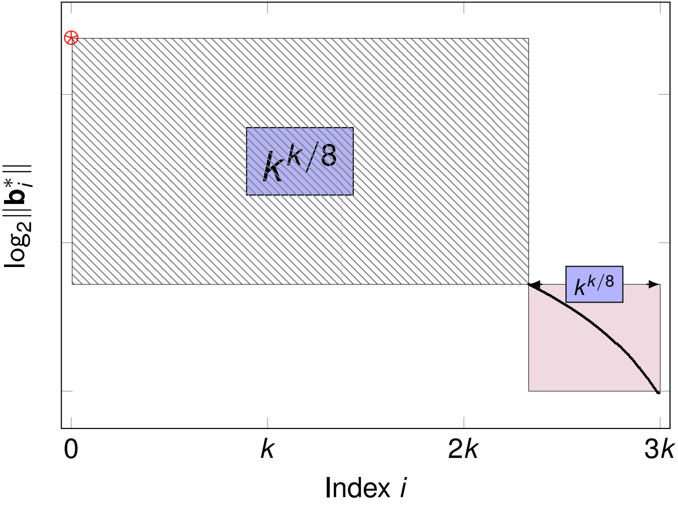


- Enum_Cost('first block') = $k^{k/8+o(k)}$;
- Enum_Cost('last block') = $k^{k/(2e)+o(k)}$.

How can we do better than $k^{k/(2e)}$?

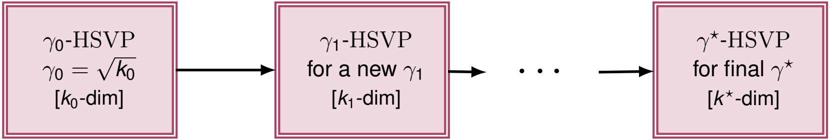


How can we do better than $k^{k/(2e)}$?



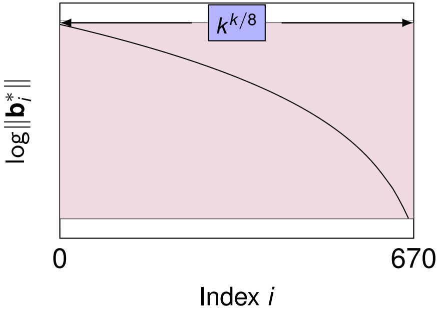
How can we do better than $k^{k/(2e)}$?

- Start from a smaller $k_0 = k \cdot 2e/8 (\approx 0.67k)$ as $k_0^{k_0/(2e)} \leq k^{k/8}$.
- k_0-dim $\mathrm{SVP} \Rightarrow \sqrt{k_0}$-HSVP
  $\Rightarrow$ For k_0-dim lattice, reach HF: $\sqrt{k_0}$ and RHF: $\sqrt{k_0}^{1/(k_0-1)} \approx k^{1/(1.36 \cdot k)}$.



RHF: $k^{1/(1.36 \cdot k)}$ Smaller and smaller RHF Target RHF: $k^{1/(2 \cdot k)}$

Targeting RHF : $k^{1/(2k)}$ $(k = 1000)$



- Starting block-size:

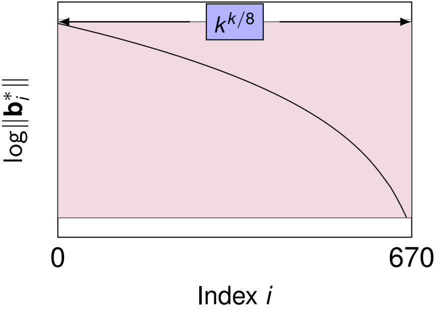
$$k_0 = k \cdot \frac{2e}{8} \approx 0.67k$$

$$\Rightarrow k_0^{k_0/2e} \approx k^{k/8}.$$

γ_0-HSVP
$\gamma_0 = \sqrt{k_0}$

SDBKZ oracle

Targeting RHF : $k^{1/(2k)}$ $(k = 1000)$

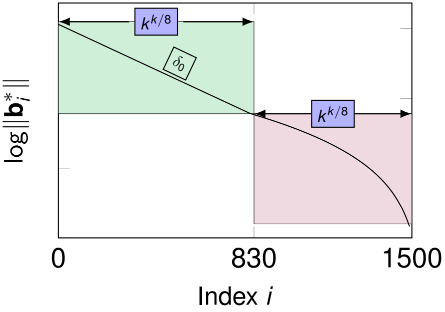


- RHF: $\gamma_0^{1/(k_0-1)}$;
  [0] $k^{1/(1.36k)}$;

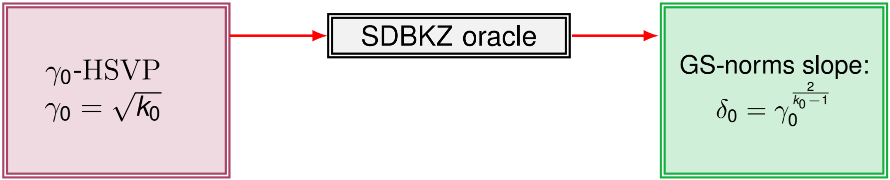
γ_0-HSVP
$\gamma_0 = \sqrt{k_0}$

SDBKZ oracle

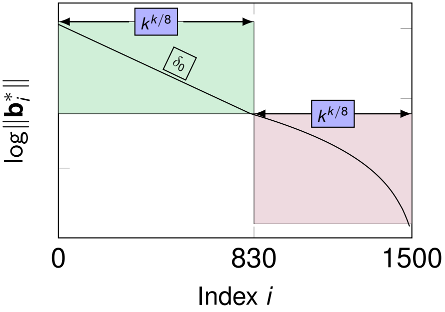
Boosting from bottom up [1st iteration]



- RHF: $\gamma_0^{1/(k_0-1)}$; [0] $k^{1/(1.36k)}$;

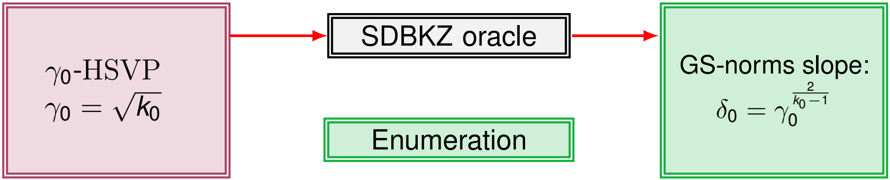


Boosting from bottom up [1st iteration]

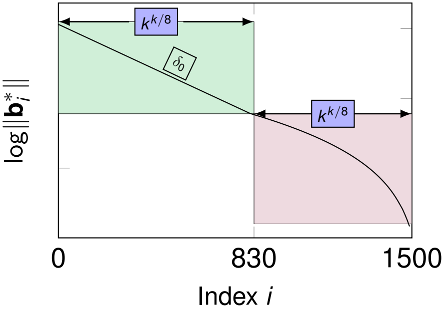


- Determine next dimension k_1:

$$\mathrm{Enum}(\delta_0, k_1 - k_0) \leq k^{\frac{k}{8}}.$$

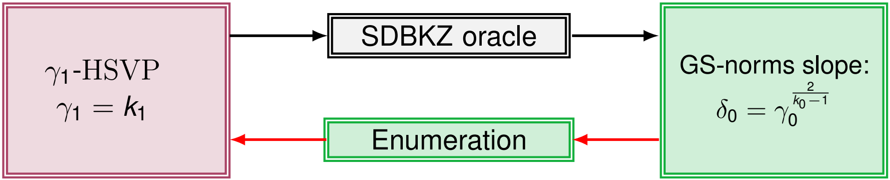


Boosting from bottom up [1st iteration]

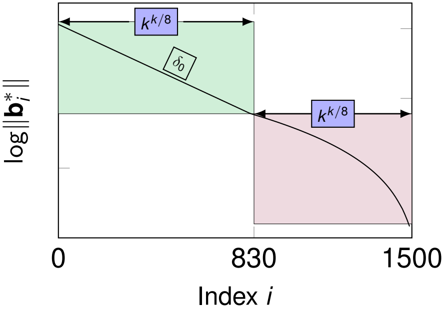


- Determine next dimension k_1:

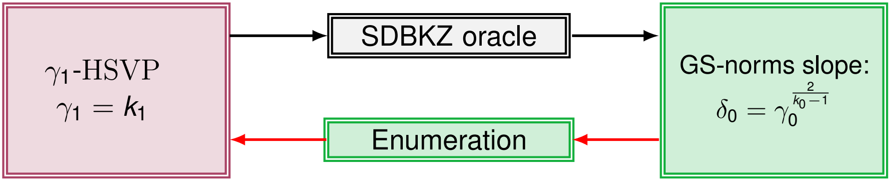
$$\mathrm{Enum}(\delta_0, k_1 - k_0) \leq k^{\frac{k}{8}}.$$



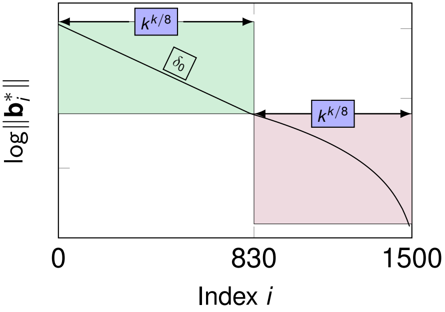
Boosting from bottom up [1st iteration]



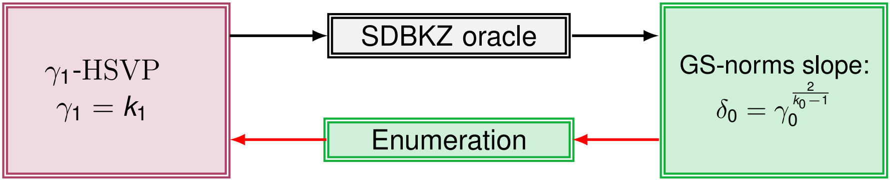
- RHF: $\gamma_1^{1/(k_1-1)}$;
  [0] $k^{1/(1.36k)}$;
  [1] $k^{1/(1.50k)}$;



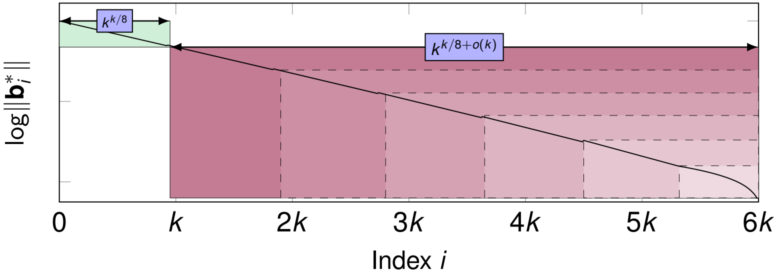
Boosting from bottom up [1st iteration]

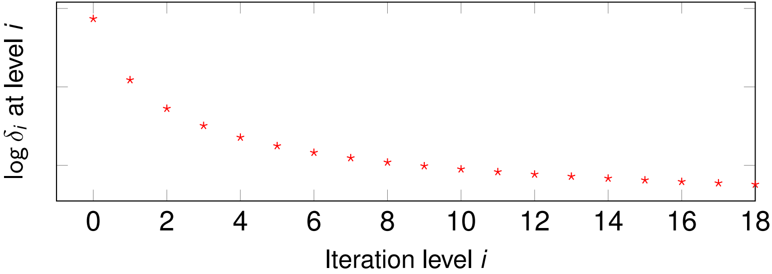


- RHF: $\gamma_1^{1/(k_1-1)}$;
  [0] $k^{1/(1.36k)}$;
  [1] $k^{1/(1.50k)}$;
  [2] $k^{1/(1.58k)}$;

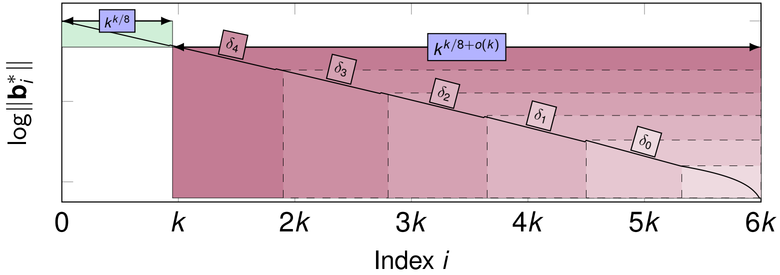


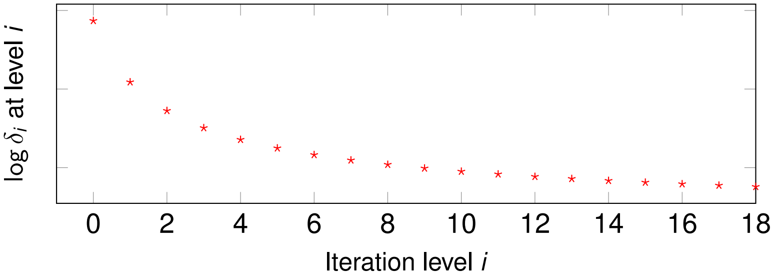
Overall complexity bound: $k^{k/8+o(k)}$



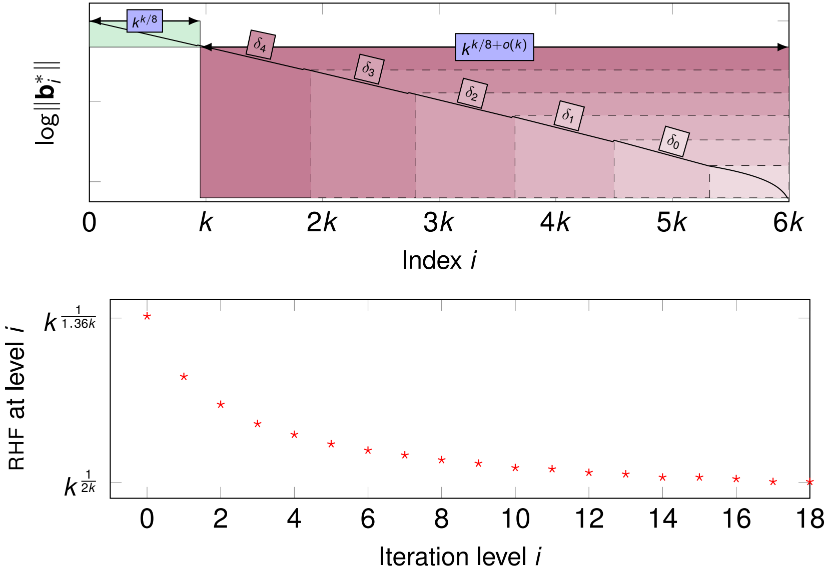


Fast convergence





Fast convergence



The FastEnum algorithm

Algorithm 1 The FastEnum algorithm (γ_i-HSVP solver).

Require: A cost parameter k, a basis $\mathbf{B}$ of dimension k_i, and a level $i \geq 0$.
Ensure: A solution of γ_i-HSVP on $\mathcal{L}(\mathbf{B})$.

```
if i = 0 then
    b ← Enum(B); // worst-case cost: k^{k/8} for size k · 2e/8
else
    C ← SDBKZ on B using γ_{i−1}-HSVP solver from last iteration;
    b ← Enum(C_[0:k_i − k_{i−1}]) with k_{i−1} from last iteration;
end if
return b;
```

Heuristic 1

During the SDBKZ execution, each call to γ-HSVP for a k-dimensional block $\mathbf{B}_{[i,i+k-1]}$ returns a vector of norm

$$\gamma \cdot \mathrm{Vol}(\mathcal{L}(\mathbf{B}_{[i,i+k-1]}))^{\frac{1}{k}}.$$

Main result

Theorem (Under Heuristic 1)

Given a basis **B** of a lattice and a parameter k, our new algorithm can reach root Hermite factor

$$k^{\frac{1}{2k}(1+o(1))} \text{ in time } k^{\frac{k}{8}+o(k)} \cdot \mathrm{poly}(\mathrm{size}(\mathbf{B})),$$

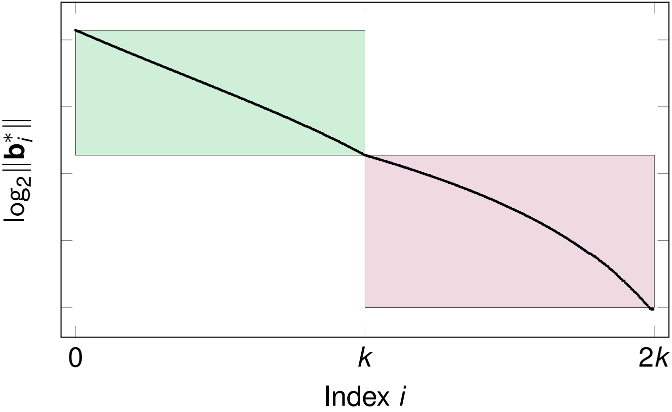
where the dimension of $\mathcal{L}(\mathbf{B})$ is $k \cdot \omega(1)$.

Heuristic 1

During the SDBKZ execution, each call to γ-HSVP for a k-dimensional block $\mathbf{B}_{[i,i+k-1]}$ returns a vector of norm around

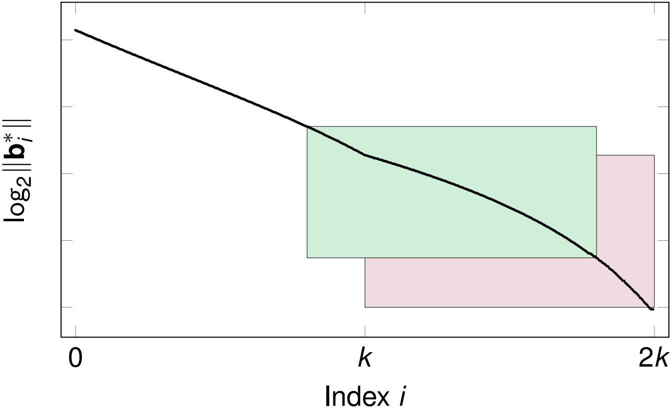
$$\gamma \cdot \mathrm{Vol}(\mathcal{L}(\mathbf{B}_{[i,i+k-1]}))^{\frac{1}{k}}.$$

Practical case: n is relatively close to k



- FastEnum: enumeration over 100%-line.

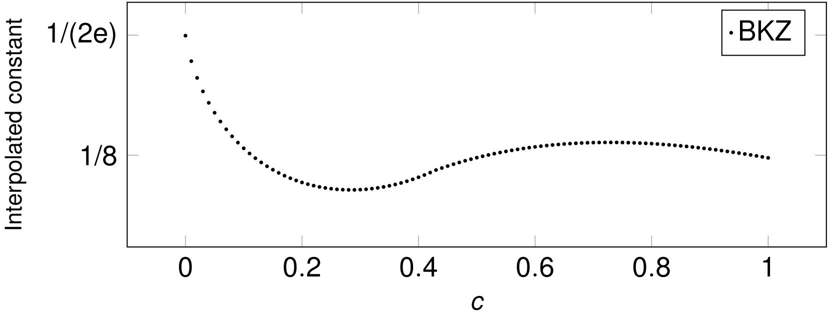
Make the enumeration zone cover the HKZ zone



- Enumeration over: c-line + (1-c)-HKZ curve for some $c \in [0, 1]$.

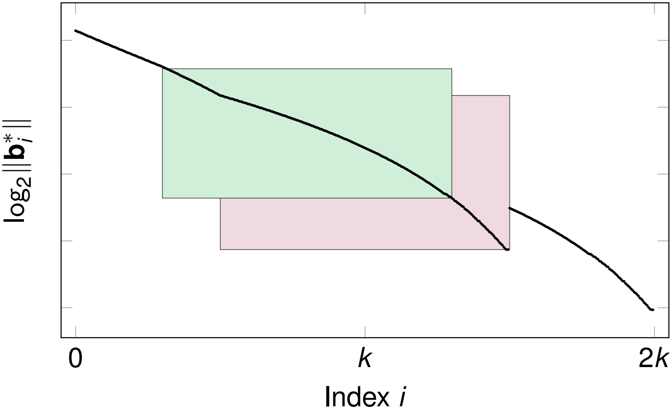
Determine concrete parameter c

Interpolated dominating constant u_0 on $k^{u_0 \cdot k + o(k)}$.

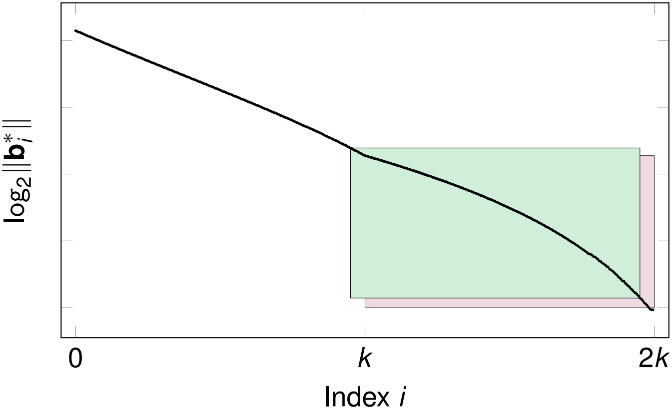


We choose $c = 0.25$!

Handling the tailing blocks

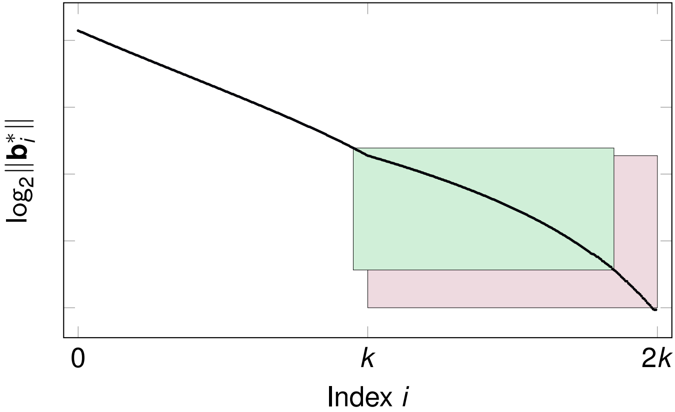


Handling the tailing blocks



- Decrease enumeration sizes for the tailing blocks.

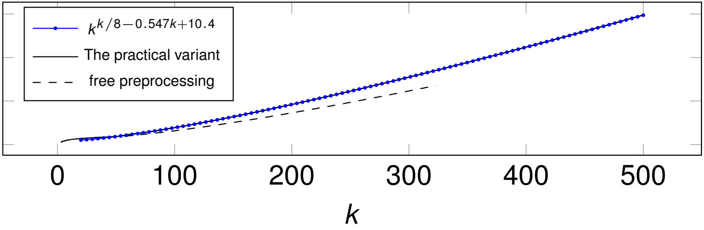
Handling the tailing blocks



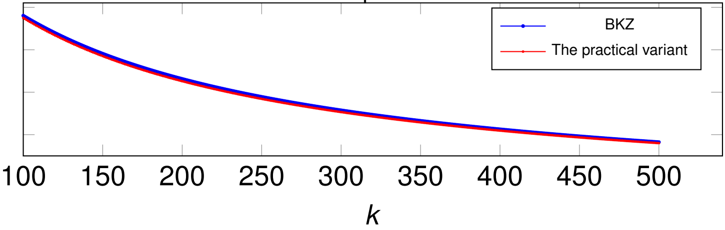
- Decrease enumeration sizes for the tailing blocks.

Experimental results ($n = 2k$)

Simulated cost of the practical variant when $c = 0.25$.



RHF: BKZ vs the practical variant.



Conclusion

Performance of enumeration-based (SD)BKZ and ours		
	(SD)BKZ	This work (informally)
RHF	$k^{1/(2k)}$	$k^{1/(2k)}$
Time	$k^{k/(2e)+o(k)}$	$k^{k/8+o(k)}$
Quantum acceleration		
Time	$k^{k/(4e)+o(k)}$ [ANS18]	$k^{k/16+o(k)}$ [ANS18]+[This work]

- Large $n/k = \omega(1)$: **heuristic** analysis of our $\mathrm{FastEnum}$ algorithm.
- Small $n/k = 2$: **simulation** analysis of our practical variant.

Future works and open questions

- [+] Remove the heuristic assumption;
  (e.g., follow the work of [HS07] + [HPS11; Neu17].)

Future works and open questions

- [+] Remove the heuristic assumption;
  (e.g., follow the work of [HS07] + [HPS11; Neu17].)
- [++] Extend to other lattice reduction algorithms;
  (e.g., BKZ reduction, slide reduction.)

Future works and open questions

- [+] Remove the heuristic assumption;
  (e.g., follow the work of [HS07] + [HPS11; Neu17].)
- [++] Extend to other lattice reduction algorithms;
  (e.g., BKZ reduction, slide reduction.)
- [++] Further investigation on cost below $k^{k/8}$.
  (e.g., the cost can be below $k^{k/8}$ for "free preprocessing".)

Future works and open questions

- [+] Remove the heuristic assumption;
  (e.g., follow the work of [HS07] + [HPS11; Neu17].)
- [++] Extend to other lattice reduction algorithms;
  (e.g., BKZ reduction, slide reduction.)
- [++] Further investigation on cost below $k^{k/8}$.
  (e.g., the cost can be below $k^{k/8}$ for "free preprocessing".)
- [+++] Study cryptographic relevance of this work;
  (e.g., give analysis for small n/k; concrete cross-over points with sieve-based algorithms classically and quantumly.)

References I

[ANS18] Yoshinori Aono, Phong Q. Nguyen, and Yixin Shen, Quantum lattice enumeration and tweaking discrete pruning, AISACRYPT, 2018, pp. 405–434.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven, New directions in nearest neighbor searching with applications to lattice sieving, SODA, 2016, pp. 10–24.

[FP83] Ulrich Fincke and Michael Pohst, A procedure for determining algebraic integers of given norm, EUROCAL (J. A. van Hulzen, ed.), LNCS, vol. 162, Springer, 1983, pp. 194–202.

[GNR10] Nicolas Gama, Phong Q. Nguyen, and Oded Regev, Lattice enumeration using extreme pruning, EUROCRYPT, 2010, pp. 257–278.

[HPS11] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé, Analyzing blockwise lattice algorithms using dynamical systems, CRYPTO, 2011, pp. 447–464.

[HS07] Guillaume Hanrot and Damien Stehlé, Improved analysis of kannan's shortest lattice vector algorithm, CRYPTO, 2007, pp. 170–186.

[Kan83] Ravi Kannan, Improved algorithms for integer programming and related lattice problems, STOC, 1983, pp. 193–206.

[MW16] Daniele Micciancio and Michael Walter, Practical, predictable lattice basis reduction, EUROCRYPT, 2016, pp. 820–849.

[Neu17] Arnold Neumaier, Bounding basis reduction properties, Des. Codes Cryptogr. **84** (2017), no. 1-2, 237–259.

[SE94] Claus-Peter Schnorr and Michael Euchner, Lattice basis reduction: Improved practical algorithms and solving subset sum problems, Math. Program. **66** (1994), 181–199.