Provable Speedups for SVP Approximation Under Random Local Blocks

Jianwei Li

Inria Paris and DIENS, PSL, France

Abstract. We point out if assuming every local block appearing in the slide reduction algorithms [ALNS20] is 'random' (as usual in the cryptographic background), then the combination of the slide reduction algorithms [ALNS20] and Pouly-Shen 's algorithm [PS24] yields exponentially faster provably correct algorithms for δ-approximate SVP for all approximation factors $n^{1/2+\varepsilon} \le \delta \le n^{O(1)}$, which is the regime most relevant for cryptography.

Keywords: Lattice Reduction · Slide Reduction · Approximating SVP.

1 Introduction

A lattice $\mathcal{L} \subset \mathbb{R}^m$ is the set of integer linear combinations

$$\mathcal{L} := \mathcal{L}(\mathbf{B}) = \{z_1\boldsymbol{b}_1 + \cdots + z_n\boldsymbol{b}_n \ : \ z_i \in \mathbb{Z}\}$$

of linearly independent basis vectors $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{R}^{m\times n}$. We call n the *rank* of the lattice.

The Shortest Vector Problem (SVP) is the computational search problem in which the input is (a basis for) a lattice $\mathcal{L} \subseteq \mathbb{Z}^m$, and the goal is to output a non-zero lattice vector $\boldsymbol{y} \in \mathcal{L}$ with minimal length, $\|\boldsymbol{y}\| = \lambda_1(\mathcal{L}) := \min_{\boldsymbol{x}\in\mathcal{L}_{\neq\mathbf{0}}} \|\boldsymbol{x}\|$. For $\delta \ge 1$, the δ-approximate variant of SVP (δ-SVP) is the relaxation of this problem in which any non-zero lattice vector $\boldsymbol{y} \in \mathcal{L}_{\neq\mathbf{0}}$ with $\|\boldsymbol{y}\| \le \delta \cdot \lambda_1(\mathcal{L})$ is a valid solution.

A closely related problem is δ-*Hermite SVP* (δ-HSVP), which asks to find a non-zero lattice vector $\boldsymbol{y} \in \mathcal{L}_{\neq\mathbf{0}}$ with $\|\boldsymbol{y}\| \le \delta \cdot \mathrm{vol}(\mathcal{L})^{1/n}$, where $\mathrm{vol}(\mathcal{L}) := \det(\mathbf{B}^T\mathbf{B})^{1/2}$ is the covolume of the lattice. *Hermite's constant* γ_n is (the square of) the minimal possible approximation factor that can be achieved in the worst case. I.e.,

$$\gamma_n := \max \frac{\lambda_1(\mathcal{L})^2}{\mathrm{vol}(\mathcal{L})^{2/n}} \ ,$$

where the maximum is over lattices $\mathcal{L} \subset \mathbb{R}^n$ with full rank n. Hermite's constant is only known exactly for $1 \le n \le 8$ and $n = 24$, but it is known to be asymptotically linear in n, i.e., $\gamma_n = \Theta(n)$. Hermite's constant plays a large role in algorithms for δ-SVP.

Starting with the celebrated work of Lenstra, Lenstra, and Lovász in 1982 [LLL82], algorithms for solving δ-(H)SVP for a wide range of parameters δ have found innumerable applications, including factoring polynomials over the rationals [LLL82], integer programming [Len83, Kan83, DPV11], cryptanalysis [Sha84, Odl90, JS98, NS01], etc.

Recently, many cryptographic primitives have been constructed whose security is based on the (worst-case) hardness of δ-SVP or closely related lattice problems [Ajt96, Reg09, GPV08, Pei09, Pei16]. Such lattice-based cryptographic constructions are likely to be used on massive scales (e.g., as part of the TLS protocol) in the not-too-distant future [NIS18], and in practice, the security of these constructions depends on the fastest algorithms for δ-(H)SVP, typically for $\delta = \mathrm{poly}(n)$.

This paper is a note on *blockwise basis reduction algorithms* [Sch87, SE91, GHKN06, HPS11, ABLR21, MW16, LN24] (more precisely, *slide reduction algorithms* [GN08a, ALNS20]) for solving δ-SVP. At a high level, these are reductions from δ-(H)SVP on lattices with rank n to exact/approximate SVP on lattices with rank $k \leq n$. More specifically, these algorithms divide a basis $\mathbf{B}$ into projected blocks $\mathbf{B}_{[i,i+k-1]}$ with *block size* k, where

$$\mathbf{B}_{[i,j]} = (\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), \ldots, \pi_i(\mathbf{b}_j))$$

and π_i is the orthogonal projection onto the subspace orthogonal to $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$. Blockwise basis reduction algorithms use their SVP oracle to find short vectors in these (low-rank) blocks and incorporate these short vectors into the lattice basis $\mathbf{B}$. By doing this repeatedly (at most $\mathrm{poly}(n)$ times (cf. [LW23, §3])) with a cleverly chosen sequence of blocks, such algorithms progressively improve the "quality" of the basis $\mathbf{B}$ until $\boldsymbol{b}_1$ is a solution to δ-(H)SVP for some $\delta \geq 1$. The goal, of course, is to take the block size k to be small enough that we can actually run an exact/approximate algorithm on lattices with rank k in reasonable time while still achieving a relatively good approximation factor δ.

We first recall the main results presented in [ALNS20, Theorems 1 and 2]:

Theorem 1 (Informal, slide reduction [ALNS20]). *For any approximation factor $\delta \geq 1$ and block size $k := k(n) \geq 2$, there is an efficient reduction from δ_S-SVP on lattices with rank $n \geq k \geq 2$ to δ-SVP on lattices with rank k, where*

$$\delta_S := \begin{cases} \delta(\delta^2\gamma_k)^{\frac{n-k}{k-1}} & \text{for } n \geq 2k, \\ \delta^2\sqrt{\gamma_k}(\delta^2\gamma_{n-k})^{\frac{n-k+1}{n-k-1}\cdot\frac{n-k}{2k}} \lesssim \delta(\delta^2\gamma_k)^{\frac{n}{2k}} & \text{for } k \leq n \leq 2k. \end{cases}$$

The starting point of this note is the intuition, based on [CN11, §4.3], that the first minimum of most local blocks in blcokwise basis reduction algorithms looks like that of a random lattice of rank the block size: this phenomenon does not hold in small block size ≤ 30 (as noted by Gama and Nguyen [GN08b]), but it becomes more and more true as the blocksize increases, as shown in [CN11, Fig. 2]. Intuitively, this was explained by a concentration phenomenon [CN11, §6.1]: *as the rank increases, random lattices dominate in the set of lattices, so unless there is a strong reason why a given lattice cannot be random, we may assume that it behaves like a random lattice.* This is particularly true for most cryptographic applications.

Recently, Pouly and Shen [PS24, Theorem 9 and Corollary 5] show the following interesting result for approximating SVP on random lattices:

Theorem 2 (Informal [PS24]). *For every $n \geq 1$, there is a randomized algorithm that on most lattices $\mathcal{L} \subset \mathbb{R}^n$ with full rank n, solves 1.123-SVP with probability at least $1/2$ in time and space $2^{n/2+o(n)}$.*

Our Results. The combination of Theorems 1 and 2 immediately implies the following result:

Theorem 3 (Informal). *Let $n > k \geq 2$ be integers and let $\delta = 1.123$. Given as input a block size k and an LLL-reduced basis B_0 of an n-rank lattice L in $\mathbb{R}^m$, if every projected block of rank k appearing in the slide reduction algorithms [ALNS20] is random (i.e., δ-SVP on such every projected lattice of rank k can be solved in time $2^{k/2+o(k)}$), then the slide reduction algorithms [ALNS20] solve δ_S-SVP on lattices with rank n in time $2^{\frac{n}{4c}+o(n)}$ if $n \leq 2k$ and in time $2^{\frac{n}{2c+2}+o(n)}$ otherwise where*

$$\delta_S := \begin{cases} \delta(\delta^2\gamma_k)^{\frac{n-k}{k-1}} & \text{for } n \geq 2k, \\ \delta^2\sqrt{\gamma_k}(\delta^2\gamma_{n-k})^{\frac{n-k+1}{n-k-1}\cdot\frac{n-k}{2k}} \lesssim \delta(\delta^2\gamma_k)^{\frac{n}{2k}} & \text{for } k < n \leq 2k. \end{cases}$$

This yields the asymptotically fastest proven running times for δ-SVP for all approximation factors $n^{1/2+\varepsilon} \leq \delta \leq n^{O(1)}$ 'in the cryptographic background'. Table 1 summarizes the current state of the art. For example, one can under random local blocks solve $O(n^{1.99})$-SVP in $2^{0.168n+o(n)}$-time and $O(n^{0.99})$-SVP in $2^{0.253n+o(n)}$ instead of the previously best $2^{0.192n+o(n)}$-time and $2^{0.406n+o(n)}$, respectively.

Approximation factor	Previous best without any assumption	This work under random local blocks
n^c for $c \in (0.5, 0.802]$	$2^{\frac{n}{2}}$ [ALSD21]	$2^{\frac{n}{4c}}$ [ALNS20]+[PS24]
n^c for $c \in (0.802, 1]$	$2^{\frac{0.401n}{c}}$ [ALNS20]	$2^{\frac{n}{4c}}$ [ALNS20]+[PS24]
n^c for $c > 1$	$2^{\frac{n}{2c+1.24}}$ [ALSD21]	$2^{\frac{n}{2c+2}}$ [ALNS20]+[PS24]

Table 1: Provable algorithms for solving SVP. We write [A]+[B] to denote the algorithm that uses basis reduction from [A] with the near-exact SVP algorithm from [B].

Theorem 3 just shows how to "recycle" one's favourite algorithm for near-exact SVP to equip the slide reduction algorithms in [ALNS20] for tackling higher dimension, provided that one is interested in approximating SVP rather than HSVP. Theorem 3 furthers our understanding of the hardness of SVP 'in the cryptographic background', but it does not impact usual security estimates, such as those of lattice-based candidates to NIST's post-quantum standardization: this is because current security estimates actually rely on HSVP estimates, following [GN08b].

1.1 Open question

Theorem 2 works for "most lattices" with full rank. It suggests an obvious open question: Is there an algorithm that provably solves $O(1)$-SVP for any lattice with rank n in time and space $2^{n/2+o(n)}$? If yes, the words "under random local blocks" in Table 1 can be removed.

References

[ABLR21] M. R. Albrecht, S. Bai, J. Li, and J. Rowell. Lattice reduction with approximate enumeration oracles: Practical algorithms and concrete performance. In *CRYPTO*, pages 732–759, 2021.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *STOC*, 1996.

[ALNS20] D. Aggarwal, J. Li, P. Q. Nguyen, and N. Stephens-Davidowitz. Slide reduction, revisited — filling the gaps in SVP approximation. In *CRYPTO*, pages 274–295, 2020.

[ALSD21] Divesh Aggarwal, Zeyong Li, and Noah Stephens-Davidowitz. A $2^{n/2}$-time algorithm for $\sqrt{n}$-SVP and $\sqrt{n}$-Hermite SVP, and an improved time-approximation tradeoff for (H)SVP. In *EUROCRYPT*, 2021.

[CN11] Y. Chen and P. Q. Nguyen. BKZ 2.0: better lattice security estimates. In *ASIACRYPT*, pages 1–20, 2011.

[DPV11] Daniel Dadush, Chris Peikert, and Santosh Vempala. Enumerative lattice algorithms in any norm via M-ellipsoid coverings. In *FOCS*, 2011.

[GHKN06] Nicolas Gama, Nick Howgrave-Graham, Henrik Koy, and Phong Q. Nguyen. Rankin's constant and blockwise lattice reduction. In *CRYPTO*, 2006.

[GN08a] Nicolas Gama and Phong Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *STOC*, 2008.

[GN08b] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT '08*, 2008.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008. https://eprint.iacr.org/2007/432.

[HPS11] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *CRYPTO*, 2011.

[JS98] Antoine Joux and Jacques Stern. Lattice reduction: A toolbox for the cryptanalyst. *J. Cryptology*, 11(3), 1998.

[Kan83] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *STOC*, 1983.

[Len83] Hendrik W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4), 1983.

[LLL82] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4), 1982.

[LN24] J. Li and P. Q. Nguyen. A complete analysis of the BKZ lattice reduction algorithm. *Journal of Cryptology*, 38, 2024.

[LW23] J. Li and M. Walter. Improving convergence and practicality of slide-type reductions. *Inf. Comput.*, 291:105012, 2023.

[MW16] Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In *Eurocrypt*, 2016. http://eprint.iacr.org/2015/1123.

[NIS18] Computer Security Division NIST. Post-quantum cryptography. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography, 2018.

[NS01] Phong Q. Nguyen and Jacques Stern. The two faces of lattices in cryptology. In *CaLC*, 2001.

[Odl90] Andrew M Odlyzko. The rise and fall of knapsack cryptosystems. *Cryptology and Computational Number Theory*, 42, 1990.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case Shortest Vector Problem. In *STOC*, 2009.

[Pei16] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science*, 10(4), 2016.

[PS24] Amaury Pouly and Yixin Shen. Solving the shortest vector problem in $2^{0.63269n+o(n)}$ time on random lattices. Cryptology ePrint Archive, Paper 2024/1805, 2024.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53(23), 1987.

[SE91] C. P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. In *FCT*, pages 68–85, 1991. Full version in *Math. Program.*, 1994.

[Sha84] Adi Shamir. A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Trans. Inform. Theory*, 30(5), 1984.