AI Methods for Post-Quantum Cryptography

Supervisor: Keqin Liu
Members: Bo Wang, Yuan Cheng, Xingyue Fan, Yongrun Huang, Chang Liu, Jialun Luo, Yexi Ren

CODE: SURF-2025-0061

SURF
Summer Undergraduate Research Fellowship

Abstract

This poster presents the background and a series of possible evolutionary methods of sieving algorithm-bgj3). We begin with reproducing the original random filtering method, then introduce a deterministic Hybrid-Sieve to replace the randomness with the algebraic structure of the dual lattice. Finally, we explore a powerful shift using Reinforcement Learning (RL), where an intelligent agent learns to produce high quality center sieve vectors.

1. Introduction and Basic Knowledge

We begin by introducing three foundational elements that frame the methods and results that follow:

Lattice: A lattice in $\mathbb{R}^n$ is the discrete set of all integer linear combinations of basis vectors $b_1, \ldots, b_n$:

$$\mathcal{L} = \Big\{ \sum_{i=1}^{n} x_i b_i \;\Big|\; x_i \in \mathbb{Z} \Big\}.$$

The basis is not unique; its density is captured by the determinant $\det(\mathcal{L}) = |\det([b_1 \cdots b_n])|$. Lattice problems form the foundation of many post-quantum schemes.

Shortest Vector Problem: Given a basis of $\mathcal{L}$, the task is to find a nonzero vector of minimum Euclidean norm:

$$v^\star \in \arg\min_{v \in \mathcal{L}\setminus\{\mathbf{0}\}} \|v\|_2.$$

SVP is believed to be hard in high dimensions and underlies the security of lattice-based cryptography (e.g., SIS/LWE, NTRU). Related problems include the Closest Vector Problem (CVP) and Bounded Distance Decoding (BDD).

LLL Algorithm: The LenstraLenstraLovász algorithm runs in polynomial time to produce a reduced basis with shorter, nearly orthogonal vectors. It uses GramSchmidt coefficients $\mu_{i,j}$, size reduction ($|\mu_{i,j}| \le 1/2$), and the Lovász condition with $\delta \in (1/4, 1)$. A classical guarantee is $\|b_1\| \le 2^{\frac{(n-1)}{2}} \lambda_1(\mathcal{L})$, where b_1 is the first vector of the LLL-reduced basis. LLL is a standard preprocessing step for approximating SVP and for cryptanalysis (e.g., as a front-end to BKZ and sieving).

These concepts underpin modern lattice algorithms: SVP sets the target, and LLL conditions the basis for practical approximations. Building on these foundations, we study and refine sieving methods and sketch RL-based extensions.

3. Algorithmic Enhancements

We introduce two gradually deepening improvements to AllPairSearch(bgj3):

1. **Deterministic Voronoi Multi-Level Sieve:**
   - **Enumerated Dual Centers:** Precompute the top N_{enum} shortest vectors of the dual lattice $\mathcal{L}^\vee$ (via enumeration) as all sieve centers, replacing random picked center vectors.
   - **Algebraic Prefiltering.** Exploit the algebraic structure of $\mathcal{L}^\vee$ (innerproduct bounds, norm relations) to perform a coarse sieve that discards vectors provably far from any optimal center with
   - **Voronoi Partitioning:** Split centers into disjoint sets C_0, C_1, C_2 (by (B_0, B_1, B_2)) and assign each vector to exactly one bucket via its closest center: $min\|\mathbf{v} \pm \mathbf{c}_i\|_2$

   Key Benefits:
   - *Reproducible & Deterministic.*
   - *Zero Overlap.* Voronoi buckets are strictly disjoint.
   - *Controlled Complexity.* Cost per level $O\big(|L| \cdot |C_i| \cdot n\big)$, lower than random caps. Friendly to the requirements of computing time and RAM.
2. **Reinforcement LearningBased Center Selection:**
   - **RL Environment:** State $s_t = [c_1^{(t)}, \ldots, c_n^{(t)}] \in \mathbb{Z}^n$;
     action $a_t \in \{c_i^+, c_i^-\}$;
     reward $R_t = \|\mathbf{v}_t\|^2 - \|\mathbf{v}_{t+1}\|^2$.
   - **Policy Network:** A multilayer perceptron f_θ outputs $\pi_\theta(a_t \mid s_t)$ over the action set.
   - **Policy Optimization:** Use REINFORCE to maximize $J(\theta) = \mathbb{E}_{S\sim d}[v_\pi(S)]$,
     updating $\theta \leftarrow \theta + \alpha\, \nabla_\theta \ln \pi_\theta(a_t|s_t)\, q_t$.
   - **Application:** The trained agent proposes highquality sieve centers on the dual lattice.

   Key Benefits:
   - *Adaptive Selection.* Learns to pick centers that drive faster convergence.
   - *Memory Efficient.* Avoids storing large random cap tables by focusing on promising lattice directions.

2. Algorithms Re-implementation

We adopted a combination of Python and C++ in Kaggle to reproduce the results of Prof.Ding's paper under the constraint of limited memory. The algorithms include the classical **sieving algorithm**, a **refined BGJ15**, and **bgj3** with a three-stage filtering scheme. Our training data comes from websites that generate random n-dimensional Lattice: https://www.latticechallenge.org/lwe$_c$*hallenge/challenge.php*
The following is the pseudocode of the original bgj3-algorithm:

Algorithm 1: AllPairSearch bgj3 (Baseline)

Require: A list L of N_0 lattice vectors, repetitions (B_0, B_1, B_2), radius $(\alpha_0, \alpha_1, \alpha_2)$, goal norm l.
Ensure: A list of reducing pairs in L.

$\mathcal{N} \leftarrow \emptyset$
for $i = 0, \ldots, B_0 - 1$ **do**
  Pick a random center $c_0 \in S^{n-1}$
  $L_i \leftarrow \{v \in L \mid v \text{ passes } F_{c_0,\alpha_0}\}$
  for $j = 0, \ldots, B_1/B_0 - 1$ **do**
    Pick a random center $c_1 \in S^{n-1}$
    $L_{ij} \leftarrow \{v \in L_i \mid v \text{ passes } F_{c_1,\alpha_1}\}$
    for $k = 0, \ldots, B_2/B_1 - 1$ **do**
      Pick a random center $c_2 \in S^{n-1}$
      $L_{ijk} \leftarrow \{v \in L_{ij} \mid v \text{ passes } F_{c_2,\alpha_2}\}$
      $\mathcal{N} \leftarrow \mathcal{N} \cup \{(u, v) \in L_{ijk}^2 \mid \|u \pm v\| < l\}$
    end for
  end for
end for
return $\mathcal{N}$

4. Key Results

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

5. Future Work

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque.

References

[1] Chinberg, T., Kalbach, A. LLLAlgorithmforLatticeBasisReduction. Available from: arXiv:2410.22196v2 [math.NT] 20 Nov 2024.
[2] Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn* 8, 229256 (1992). https://doi.org/10.1007/BF00992696
[3] Zhao, Z., Ding, J. and Yang, B.-Y. (2025). Sieving with Streaming Memory Access. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(2), 362-384. Available from: https://doi.org/10.46586/tches.v2025.i2.362-384