AI Methods for Post-Quantum Cryptography

Supervisors: Keqin Liu and Jintai Ding
Members: Yuan Cheng, Xingyue Fan, Yongrun Huang, Chang Liu, Jialun Luo, Yexi Ren, Bo Wang

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 shift using Reinforcement Learning (RL), where an intelligent agent learns to produce high quality center sieve vectors. We also provided an illustrative presentation of the results and made some feasible plans for future development.

1. Introduction and Basic Knowledge

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

1. **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.

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

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

   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).

3. **LLL Algorithm:** The Lenstra-Lenstra-Lovász algorithm runs in polynomial time to produce a reduced basis with shorter, nearly orthogonal vectors. It uses Gram-Schmidt coefficients $\mu_{i,j}$, size reduction ($|\mu_{i,j}| \leq 1/2$), and the Lovász condition with $\delta \in (1/4, 1)$. A classical guarantee is

$$\|b_1\| \leq 2^{\frac{(n-1)}{2}} \lambda_1(\mathcal{L}),\ where: \lambda_1(\mathcal{L}) \approx \sqrt{\frac{n}{2\pi e}} Vol(\mathcal{L})^{\frac{1}{n}} =: \mathbf{GH}(\mathcal{L})$$

   and b_1 is the first vector of the LLL-reduced basis. LLL is a standard pre-processing step for approximating SVP and for cryptanalysis (e.g., as a front-end to BKZ and sieving).

4. **Evaluation:** To evaluate our implementation's performance, we adopt the core quality metric from the target paper[3]: the ratio of the found vector's norm to the Gaussian Heuristic(GH)[1]. The **norm** is the length of the vector our algorithm outputs, while the **GH** is a theoretical estimate of the shortest possible vector's length. **Norm/GH $\to$ 1** is the primary indicator of a successful SVP solution.

   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.

2. Algorithms Re-implementation

We adopted a combination of Python and C++ in Kaggle to reproduce the results of Prof.Ding's paper[3] 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 a website that generate random n-dimensional Lattice: https://www.latticechallenge.org/lwe_challenge/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 $\mathcal{N}$

This algorithm is already highly efficient. However, it suffers from two fundamental pain-points:

- *Randomness & Non-Reproducibility:* Centers are drawn *randomly* on the unit sphere at each run, so results vary across executions. This hinders both theoretical analysis and practical debugging.
- *Computational Redundancy & Inefficiency:* Random spherical caps inevitably *overlap*. Consequently the same lattice vector (or pair) may be tested multiple times, wasting both CPU and RAM resources.

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 Pre-filtering:** Exploit the algebraic structure of $\mathcal{L}^\vee$ (inner-product 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: $min\|\mathbf{v} \pm \mathbf{c}_i\|_2$

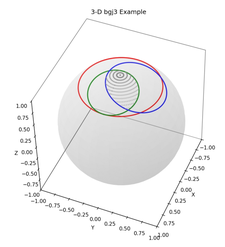
   Key Benefits:
   - *Reproducible & Deterministic.*
   - *Zero Overlap.* Voronoi buckets are strictly disjoint.
   - *Controlled Complexity.* Cost per level $O\big(|L| \cdot |C_i| \cdot n\big)$, lower than random caps. Friendly to the requirements of computing time and RAM.
2. **Reinforcement Learning-Based 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 multi-layer perceptron f_θ outputs $\pi_\theta(a_t \mid s_t)$ over the action set.
   - **Policy Optimization:** Use REINFORCE[2] 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 high-quality sieve centers on the dual lattice.

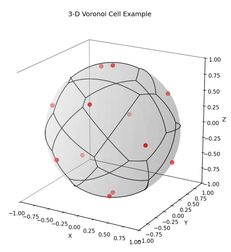
   Key Benefits:
   - *Adaptive Selection:* Learns to pick centers that drive faster convergence.
   - *Memory Efficient:* Avoids storing large random cap tables by focusing on promising lattice directions.

4. Visualization and Conceptual Innovation

The following are two simplified schematic diagrams on the 3-D sphere regarding the **first two** sieving methods.

- **Left (bgj3):** The search space is covered by **overlapping spherical caps** centered on *randomly* chosen points.
- **Right (Voronoi improved):**We introduce a **deterministic and non-overlapping partition** of the space using Voronoi cells. These cells are centered on points derived from the dual lattice.





Advertisement of the two improvements:

- **The Hybrid Sieve:** Achieves **full reproducibility** and **eliminates computational redundancy** by ensuring each vector been filtered exactly once.
- **RL-Powered Center Selection:** Transforms the static selection process into a **dynamic strategy** that the intelegence learns to **accelerate convergence** to the shortest vector.

5. Future Work

- Due to computational resource constraints, we are temporarily not able to fully reproduce the results of Professor Ding's paper and verify our conjectures regarding potential algorithmic improvements during the few months. In the coming future, we will apply for access to the SMP's servers to conduct more in-depth research.
- Additionally, we will expand our analysis to a deeper theoretical framework, rigorously investigating algorithms and theorems through both conceptual discussions and empirical research.

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, 229-256 (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