POEM

POEM: Parallel mathematical Optimization based EnuM

***Yuji Shinano**, Zuse Institute Berlin*

In Short

- POEM is a novel massively parallel solver for Shortest Vector Problem (SVP) .
- POEM is a asynchronous distributed-memory solver, which uses powerful mathematical optimization technology.

A (full-rank) *lattice* of dimension n is the set of all integral linear combinations

$$L = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) := \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_1, \ldots, x_n \in \mathbb{Z} \right\},$$

where $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are n linearly independent vectors in $\mathbb{R}^n$ for a positive integer n. The set of the n vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is called a *basis* of L. When another set of vectors $\{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$ spans the same lattice L, it is also called a basis of L. Furthermore, $\mathcal{L}(\mathbf{B})$ denotes the lattice spanned by the row vectors of an invertible matrix $\mathbf{B}$. The $n \times n$ matrix $\mathbf{B}$ is called a *basis matrix* of L. Two matrices $\mathbf{B}$ and $\mathbf{C}$ span the same lattice if and only if there exists a unimodular matrix $\mathbf{T}$ satisfying $\mathbf{C} = \mathbf{TB}$. (An integral square matrix is called *unimodular* if its determinant equals ± 1.) Given a basis matrix $\mathbf{B}$ of L, the volume of L is defined as $\mathrm{vol}(L) := |\det(\mathbf{B})|$, which is independent of the choice of basis matrices.

Lattice problems are algorithmic problems that involve lattices. Among lattice problems, the following is of fundamental importance:

Definition 1 (Shortest Vector Problem (SVP))**.** Find the shortest non-zero vector with respect to the ℓ_2-norm in the lattice $\mathcal{L}(\mathbf{B})$, given a basis matrix $\mathbf{B}$.

SVP is a discrete optimization problem for finding x_i's in () and is shown to be NP-hard under randomized reductions [1]. (That is, there exists a probabilistic Turing-machine that reduces any problem in NP to SVP instances in polynomial-time.) Note that the shortest vectors are not unique, and SVP asks to find one of them. The length of the shortest non-zero vector in L is denoted by $\lambda_1(L)$. SVP is the problem of finding a lattice vector $\mathbf{s} \in L$ with $\|\mathbf{s}\| = \lambda_1(L)$. It should be emphasized that there is no known NP algorithm to check if $\|\mathbf{v}\| = \lambda_1(L)$ for a given $\mathbf{v} \in L$. Therefore, we rely on *Gaussian Heuristic*, which assumes that the number of vectors in $L \cap S$ is roughly equal to $\mathrm{vol}(S)/\mathrm{vol}(L)$ for a measurable set S in $\mathbb{R}^n$.

By taking S to be the ball of radius $\lambda_1(L)$ centered at the origin $\mathbf{0}$ in $\mathbb{R}^n$, the Gaussian Heuristic leads to an estimation of $\lambda_1(L)$ as

$$\lambda_1(L) \approx \left(\frac{\mathrm{vol}(L)}{\omega_n} \right)^{1/n},$$

where ω_n denotes the volume of the n-dimensional unit ball. By Stirling's formula, we have $\omega_n \approx \left(\frac{2\pi e}{n}\right)^{n/2}$ as $n \to \infty$, and define

$$\mathrm{GH}(L) := \sqrt{\frac{n}{2\pi e}} \mathrm{vol}(L)^{1/n}. \tag{0.1}$$

Then, $\lambda_1(L) \approx \mathrm{GH}(L)$ holds for random lattices L in high dimensions $n \geq 40$. (Gaussian Heuristic does not hold in low dimensions.) For a vector $\mathbf{v} \in L$, the value $\|\mathbf{v}\|/\mathrm{GH}(L)$ is called the *approximation factor* of $\mathbf{v}$. Similarly, for a basis matrix $\mathbf{B}$, the value $\min_{1 \leq i \leq n} \|\mathbf{b}_i\|/\mathrm{GH}(L)$ is called the approximation factor of $\mathbf{B}$. They are evaluation metrics for the lattice vector and the basis. Based on this observation, an approximate variant of SVP is defined:

Definition 2 (Hermite Shortest Vector Problem (HSVP))**.** Given a basis matrix $\mathbf{B}$ and an approximation factor $\gamma > 0$, find a non-zero vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}\| \leq \gamma \cdot \mathrm{vol}(\mathcal{L}(\mathbf{B}))^{1/n}$.

Another important lattice problem is:

Definition 3 (Closest Vector Problem (CVP))**.** Given a basis of a lattice L and a target vector $\mathbf{t}$, find a vector in L that is closest to $\mathbf{t}$.

CVP is a generalization of SVP because we can easily convert an instance of SVP to one of CVP. This implies that CVP is at least as hard as SVP. From a practical point of view, however, both problems are considered equally hard due to Kannan's embedding technique that can transform CVP into SVP.

Lattice problems are believed to be computationally hard with both classical and quantum algorithms and have been used to construct various cryptosystems, including post-quantum cryptography. Therefore, developing a framework for lattice problems is an important task both in large-scale optimization and cryptanalysis. More specifically, the security of many cryptosystems is based on the hardness of an approximate variant of SVP. Lattice problem solvers have been extensively tested at the Darmstadt SVP challenge [2], which asks to find a lattice vector shorter than 1.05 times the expected length of a non-zero shortest lattice vector.

There are three basic families of lattice algorithms that have been developed to solve practical lattice problems: basis reduction, enumeration (ENUM), and sieve. These algorithms have advantages and disadvantages, and there is no single definite algorithm for lattice problems. Therefore, practical lattice-problem solvers generally rely on two or more algorithms. G6K [3] implements a variety of basis reduction and sieve algorithms, and it is considered the state-of-the-art SVP solver. G6K is equipped with both CPU and GPU highly parallelized implementations, but it runs only on a single machine. Furthermore, the memory requirement is exponential with respect to the dimension of the lattice, which is inevitable for sieve algorithms. On the other hand, MAP-SVP [4] is based on basis reduction and ENUM, which showed efficient scalability above $100,000$ MPI processes.

There are two main obstacles in developing a large-scale multi-paradigm solver: the need for an efficient high-level information-sharing scheme across different algorithms, and an adaptive task selection and distribution strategy for hundreds of thousands of processes. To provide solutions to overcome these obstacles, by exploiting the mathematical properties of lattice, a clever vector pooling scheme is introduced to minimize the amount of information communicated among processes, a flexible framework to make various algorithms work cooperatively on a large-scale distributed computing platform, named CMAP-LAP(Configurable Massively Parallel Solver for Lattice Problems) [5], has been developed. By extending the well-recognized Ubiquity Generator (UG) framework [6] for Branch-and-Bound (B&B) algorithms, CMAP-LAP was built as a solid backbone to manage hundreds of thousands of processes running heterogeneous algorithms in parallel, where the assignment of algorithms and their parameters can be adaptively tuned according to the available resources and the progress of the whole system.

This project aims to develop the strongest parallel solver, which has a novel algorithm implementation, to solve SVP. The SVP can be formulated as an IQP (Integer Quadratic Problem) from a mathematical optimization point of view. However, the formulated IQP instances can not be solved directly in a reasonable amount of time by using the latest commercial solvers, even if its dimension is about 50 (the highest dimension in the current SVP challenge page is 180). Therefore, solving an instance of the naive IQP formulation has no chance to beat the record. In this project, we would like to develop a parallel solver, which is referred to as POEM (Parallel mathematical Optimization based EnuM). POEM uses a mathematical optimization solver inside of the ENUM algorithm to find a good feasible solution faster and is parallelized by using the CMAP-LAP framework. This project includes new algorithm development and the parallel implementation of the new algorithms. Therefore, if the solver can find a good feasible solution in the highest dimension of the SVP challenge page, it would have a big impact on the research field.

WWW

https://www.zib.de/members/shinano

More Information

[1] M. Ajtai, *Generating hard instances of lattice problems*, in Symposium on Theory of Computing (STOC 1996), ACM, 1996, pp. 99–108.

[2] M. Schneider, N. Gama, P. Baumann, and L. Nobach, *SVP challenge (2010)*, URL: http://latticechallenge.org/svp-challenge.

[3] M. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, and M. Stevens, *The general sieve kernel and new records in lattice reduction*, in Advances in Cryptology–EUROCRYPT 2019, vol. 11477 of Lecture Notes in Computer Science, Springer, 2019, pp. 717–746.

[4] N. Tateiwa, Y. Shinano, S. Nakamura, A. Yoshida, S. Kaji, M. Yasuda, and K. Fujisawa, *Massive parallelization for finding shortest lattice vectors based on ubiquity generator framework*, in SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2020, pp. 1–15.

[5] N. Tateiwa, Y. Shinano, K. Yamamura, A. Yoshida, S. Kaji, M. Yasuda, and K. Fujisawa, *CMAP-LAP: Configurable Massively Parallel Solver for Lattice Problems*, in: 2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC), pp. 42–52, 2021.

[6] *UG: Ubiquity Generator framework*. http://ug.zib.de/.

Project Partners

Institute of Science Tokyo, IMI(Kyushu University)

Funding

Forschungscampus Modal

DFG Subject Area

409-02