Shortest Vector Problem (1982; Lenstra, Lenstra, Lovasz)

Daniele Micciancio, University of California at San Diego, www.cs.ucsd.edu/~daniele
entry editor: Sanjeev Khanna

INDEX TERMS: Point lattices. Algorithmic geometry of numbers. Quadratic forms.
SYNONYMS: Lattice basis reduction. LLL algorithm. Closest vector problem. Nearest vector problem. Minimum distance problem.

1 PROBLEM DEFINITION

A *point lattice* is the set of all integer linear combinations

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

of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ in m-dimensional Euclidean space. For computational purposes, the lattice vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are often assumed to have integer (or rational) entries, so that the lattice can be represented by an integer matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^{m \times n}$ (called *basis*) having the generating vectors as columns. Using matrix notation, lattice points in $\mathcal{L}(\mathbf{B})$ can be conveniently represented as $\mathbf{Bx}$ where $\mathbf{x}$ is an integer vector. The integers m and n are called the *dimension* and *rank* of the lattice respectively. Notice that any lattice admits multiple bases, but they all have the same rank and dimension.

The main computational problems on lattices are the *Shortest Vector Problem*, which asks to find the shortest nonzero vector in a given lattice, and the *Closest Vector Problem*, which asks to find the lattice point closest to a given target. Both problems can be defined with respect to any norm, but the Euclidean norm $\|\mathbf{v}\| = \sqrt{\sum_i v_i^2}$ is the most common. Other norms typically found in computer science applications are the ℓ_1 norm $\|\mathbf{v}\|_1 = \sum_i |v_i|$ and the *max* norm $\|\mathbf{v}\|_\infty = \max_i |v_i|$. This entry focuses on the Euclidean norm.

Since no efficient algorithm is known to solve SVP and CVP exactly in arbitrary high dimension, the problems are usually defined in their approximation version, where the approximation factor $\gamma \geq 1$ can be a function of the dimension or rank of the lattice.

Definition 1 (Shortest Vector Problem, SVP_γ)**.** *Given a lattice $\mathcal{L}(\mathbf{B})$, find a nonzero lattice vector $\mathbf{Bx}$ (where $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$) such that $\|\mathbf{Bx}\| \leq \gamma \cdot \|\mathbf{By}\|$ for any $\mathbf{y} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$.*

Definition 2 (Closest Vector Problem, CVP_γ)**.** *Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ (where $\mathbf{x} \in \mathbb{Z}^n$) such that $\|\mathbf{Bx} - \mathbf{t}\| \leq \gamma \cdot \|\mathbf{By} - \mathbf{t}\|$ for any $\mathbf{y} \in \mathbb{Z}^n$.*

Lattices have been investigated by mathematicians for centuries in the equivalent language of quadratic forms, and are the main object of study in the *geometry of numbers*, a field initiated by

Minkowski as a bridge between geometry and number theory. For a mathematical introduction to lattices see [3]. The reader is referred to [5, 11] for an introduction to lattices with an emphasis on computational and algorithmic issues.

2 KEY RESULTS

The problem of finding an efficient (polynomial time) solution to SVP_γ for lattices in arbitrary dimension was first solved by the celebrated *lattice reduction* algorithm of Lenstra, Lenstra and Lovász [10], commonly known as the *LLL* algorithm.

Theorem 1. *There is a polynomial time algorithm to solve* SVP_γ *for* $\gamma = (2/\sqrt{3})^n$*, where* n *is the rank of the input lattice.*

The LLL algorithm achieves more than just finding a relatively short lattice vector: it finds a so-called *reduced basis* for the input lattice, i.e., an entire basis of relatively short lattice vectors. Shortly after the discovery of the LLL algorithm, Babai [2] showed that reduced bases can be used to efficiently solve CVP_γ as well within similar approximation factors.

Corollary 1. *There is a polynomial time algorithm to solve* CVP_γ *for* $\gamma = O(2/\sqrt{3})^n$*, where* n *is the rank of the input lattice.*

The reader is referred to the original papers [2, 10] and [11, Chapter 2] for details. Introductory presentations of the LLL algorithm can also be found in many other texts, e.g., [15, Chapter 16] and [14, Chapter 27]. It is interesting to note that CVP is at least as hard as SVP (see [11, Chapter 2]) in the sense that any algorithm that solves CVP_γ can be efficiently adapted to solve SVP_γ within the same approximation factor.

Both SVP_γ and CVP_γ are known to be NP-hard in their exact ($\gamma = 1$) or even approximate versions for small values of γ, e.g., constant γ independent of the dimension. (See [11, Chapters 3 and 4] and [4, 9] for the most recent results.) So, no efficient algorithm is likely to exist to solve the problems exactly in arbitrary dimension. For any fixed dimension n, both SVP and CVP can be solved exactly in polynomial time using an algorithm of Kannan [8]. However, the dependency of the running time on the lattice dimension is $n^{O(n)}$. Using randomization, exact SVP can be solved probabilistically in $2^{O(n)}$ time and space using the *sieving* algorithm of Ajtai, Kumar and Sivakumar [1].

As for approximate solutions, the LLL lattice reduction algorithm has been improved both in terms of running time and approximation guarantee. (See [13] and references therein.) Currently, the best (randomized) polynomial time approximation algorithm achieves approximation factor $\gamma = 2^{O(n \log\log n / \log n)}$.

3 APPLICATIONS

Despite the large (exponential in n) approximation factor, the LLL algorithm has found numerous applications and lead to the solution of many algorithmic problems in computer science. The number and variety of applications is too large to give a comprehensive list. Some of the most representative applications in different areas of computer science are mentioned below.

The first motivating applications of lattice basis reduction were the solution of integer programs with a fixed number of variables and the factorization of polynomials with rationals coefficients. (See

[10], [7] and [15, Chapter 16].) Other classic applications are the solution of random instances of low-density subset-sum problems, breaking (truncated) linear congruential pseudorandom generators, simultaneous Diophantine approximation, and the disproof of Mertens' conjecture. (See [7] and [15, Chapter 17].)

More recently, lattice basis reduction has been extensively used to solve many problems in cryptanalysis and coding theory, including breaking several variants of the RSA cryptosystem and the DSA digital signature algorithm, finding small solutions to modular equations, and list decoding of CRT (Chinese Reminder Theorem) codes. The reader is referred to [6, 12] for a survey of recent applications, mostly in the area of cryptanalysis.

One last class of applications of lattice problems is the design of cryptographic functions (e.g., collision resistant hash functions, public key encryption schemes, etc.) based on the apparent intractability of solving SVP_γ within small approximation factors. The reader is referred to [11, Chapter 8] and [12] for a survey of such applications, and further pointers to relevant literature. One distinguishing feature of many such lattice based cryptographic functions is that they can be proved to be hard to break *on the average*, based on a *worst-case* intractability assumption about the underlying lattice problem.

4 OPEN PROBLEMS

The main open problems in the computational study of lattices is to determine the complexity of approximate SVP_γ and CVP_γ for approximation factors $\gamma = n^c$ polynomial in the rank of the lattice. Specifically,

- Are there polynomial time algorithm that solve SVP_γ or CVP_γ for polynomial factors $\gamma = n^c$? (Finding such algorithms even for very large exponent c would be a major breakthrough in computer science.)
- Is there an $\epsilon > 0$ such that approximating SVP_γ or CVP_γ to within $\gamma = n^\epsilon$ is NP-hard? (The strongest known inapproximability results [4] are for factors of the form $n^{O(1/\log n)}$ which grow faster than any poly-logarithmic function, but slower than any polynomial.)

There is theoretical evidence that for large polynomials factors $\gamma = n^c$, SVP_γ and CVP_γ are not NP-hard. Specifically, both problems belong to complexity class coAM for approximation factor $\gamma = O(\sqrt{n/\log n})$. (See [11, Chapter 9].) So, the problems cannot be NP-hard within such factors unless the polynomial hierarchy PH collapses.

5 URL to CODE

The LLL lattice reduction algorithm is implemented in most library and packages for computational algebra, e.g.,

- GAP (`http://www.gap-system.org`)
- LiDIA (`http://www.cdc.informatik.tu-darmstadt.de/TI/LiDIA/`)
- Magma (`http://magma.maths.usyd.edu.au/magma/`)
- Maple (`http://www.maplesoft.com/`)

- Mathematica (`http://www.wolfram.com/products/mathematica/index.html`)
- NTL (`http://shoup.net/ntl/`).

NTL also includes an implementation of Block Korkine-Zolotarev reduction that has been extensively used for cryptanalysis applications.

6 CROSS REFERENCES

Knapsack. Sphere packing problem. Cryptographic hardness of learning. Discrete logarithm. Factoring. Learning heavy Fourier coefficients over Z_N.

7 RECOMMENDED READING

References

[1] M. Ajtai, R. Kumar, and D. Sivakumar, *A sieve algorithm for the shortest lattice vector problem*, in Proceedings of the thirty-third annual ACM symposium on theory of computing - STOC 2001, Heraklion, Crete, Greece, July 2001, ACM, pp. 266–275.

[2] L. Babai, *On Lovasz' lattice reduction and the nearest lattice point problem*, Combinatorica, 6 (1986), pp. 1–13. Preliminary version in STACS 1985.

[3] J. W. S. Cassels, *An introduction to the geometry of numbers*, Springer-Verlag, New York, 1971.

[4] I. Dinur, G. Kindler, R. Raz, and S. Safra, *Approximating CVP to within almost-polynomial factors is NP-hard*, Combinatorica, 23 (2003), pp. 205–243. Preliminary version in FOCS 1998.

[5] M. Grotschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, vol. 2 of Algorithms and Combinatorics, Springer-Verlag, second ed., 1993.

[6] A. Joux and J. Stern, *Lattice reduction: A toolbox for the cryptanalyst*, Journal of Cryptology, 11 (1998), pp. 161–185.

[7] R. Kannan, *Annual reviews of computer science*, vol. 2, Annual Review Inc., Palo Alto, California, 1987, ch. "Algorithmic geometry of numbers", pp. 231–267.

[8] ——, *Minkowski's convex body theorem and integer programming*, Mathematics of operation research, 12 (1987), pp. 415–440.

[9] S. Khot, *Hardness of Approximating the Shortest Vector Problem in Lattices*, Journal of the ACM, 52 (2005), pp. 789–808. Preliminary version in FOCS 2004.

[10] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen, 261 (1982), pp. 513–534.

[11] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*, vol. 671 of The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Boston, Massachusetts, Mar. 2002.

[12] P. Nguyen and J. Stern, *The two faces of lattices in cryptology*, in Cryptography and lattices conference – CaLC 2001, J. Silverman, ed., vol. 2146 of Lecture Notes in Computer Science, Providence, RI, USA, Mar. 2001, Springer-Verlag, pp. 146–180.

[13] C. P. Schnorr, *Fast LLL-type lattice reduction*, Information and Computation, 204 (2006), pp. 1–25.

[14] V. V. Vazirani, *Approximation Algorithms*, Springer, 2001.

[15] J. von zur Gathen and J. Gerhard, *Modern Comptuer Algebra*, Cambridge, second ed., 2003.