MR3376389 68Q25

Haviv, Ishay (IL-AC-SCS); **Regev, Oded** [**Regev, Oded**[1]] (1-NY-X)

★On the lattice isomorphism problem. (English summary)

Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, 391–404, *ACM*, *New York*, 2014.

An m-dimensional *lattice* of rank n is defined in the paper as the set of all integer combinations of n linearly independent vectors $b_1, \dots, b_n \in \mathbb{R}^m$; these vectors are called a *basis* of the lattice. The *Lattice Isomorphism Problem* (LIP) is defined in the abstract as the following problem: given two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ decide whether there exists an orthogonal linear transformation mapping $\mathcal{L}_1$ to $\mathcal{L}_2$.

The main result is an algorithm for this problem running in time $n^{O(n)}$ times a polynomial in the input size, where n is the rank of the input lattices. A crucial component in the proof of the correctness of this algorithm is a new generalized *isolation lemma*, which can isolate n linearly independent vectors in a given subset of $\mathbb{Z}^n$. It is also shown that LIP belongs to the complexity class SZK of problems whose solution can be validated by a statistical zero-knowledge proof.

Indeed, Theorem 1.1 of the paper states the existence of an algorithm that takes two bases of lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ of rank n as input, and outputs all the orthogonal linear transformations $O\colon \mathrm{span}(\mathcal{L}_1) \to \mathrm{span}(\mathcal{L}_2)$ for which we have $\mathcal{L}_2 = O(\mathcal{L}_1)$, in running time $n^{O(n)} \cdot s^{O(1)}$ and in polynomial space, where s denotes the input size. In addition, the number of these transformations is shown to be at most $n^{O(n)}$.

It is noted that the bound in Theorem 1.1 on the number of orthogonal linear transformations mapping one lattice to another is tight up to the constant in the exponent, and the running time of the algorithm is optimal, up to the constant in the exponent, given that it outputs all isomorphisms between the two input lattices. However, the challenge of finding a more efficient algorithm which only decides LIP is left open.

Saeed Salehi