

OVERVIEW

What:

- develop strategies to improve the running time of lattice basis reduction algorithms.

Why:

- lattice basis reduction is an important tool in cryptography, cryptanalysis, and algorithmic number theory.
- possible impact on quantum computing.

How:

- generate unimodular lattice bases using different construction methods.
- sort the basis vectors in ascending or descending order and observe the effects on our reduction algorithms.
- find strategy to decide which algorithm and sorting to use.

LATTICE BASES

We are using unimodular lattice bases L with $\det L = \pm 1$ to concentrate on the running time of lattice basis reduction rather than the quality of the reduced basis.

We developed three different construction methods for generating unimodular lattice bases:

- M_{1x} : multiply lower and upper triangular matrices with $\det = \pm 1$.
- M_{2y} : start with identity matrix and perform unimodular transformations (swaps and translations).
- M_{3z} : hybrid approach: multiply matrices generated by M_{1x} and M_{2y} .

We derived 25 different unimodular basis types based on these construction methods.

We created 1000 bases for each type with entries up to 500 bit.

EXPERIMENTS

We tested variants V_1 and V_2 of the LLL algorithm in combination with sorting the bases in ascending or descending orders.

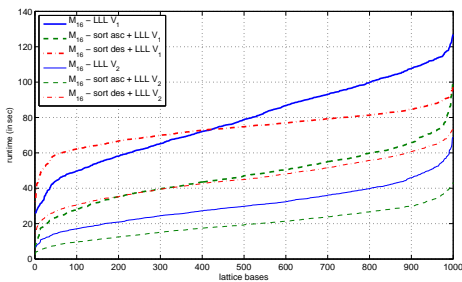


FIGURE 1: For M_{16} bases sorting in ascending or descending order has a **major effect** on the running time of the tested reduction algorithms.

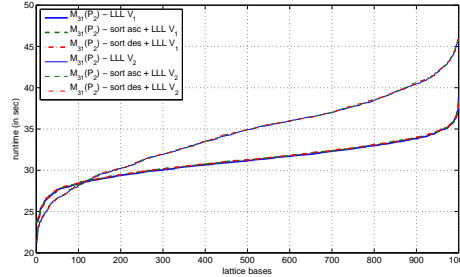


FIGURE 2: For M_{31} with parameter set P_2 bases sorting in ascending or descending order has **no effect** on the running time of the reduction algorithms.

OBSERVATIONS

Dependent on the basis type, we observe a big difference in the performance of LLL variants V_1 and V_2 .

Dependent on the basis type, sorting will decrease or in some cases increase the running time.

LLL variant V_2 seems to perform better for lattice bases with higher reduction times.

Sorting in ascending order decreases the running time of the reduction algorithm for the majority of the lattice bases.

STRATEGY

Develop decision algorithm that determines:

- which variant of the LLL to choose (V_1 or V_2)
- what kind of sorting (ascending or descending)

The goal is to find a decision algorithm that based on the type of the unimodular basis finds the optimal combination with a probability of 90%.

RESULTS

A strategy based on the decision algorithm reduces the running time of the LLL for all 25 types of unimodular lattice basis.

