L^3-reduced basis algorithm

1) Compute the GS-orthogonalization of $b_1, \dots, b_m$.
2) For $i = 2$ to m
   For $j = i-1$ to 1
   $$b_i \leftarrow b_i - \left\lfloor \frac{\langle b_i, b_j^* \rangle}{\|b_j^*\|^2} \right\rceil \cdot b_j \quad .$$
   ← rounding
3) If $\exists i$, $\|b_i^*\|^2 > \frac{4}{3} \cdot \|b_{i+1}^* + \mu_{i+1,i} b_i^*\|^2$
   then swap $\{b_i, b_{i+1}\}$ & GOTO (1).
4) Output $\{b_1, \dots, b_m\}$.

Analysis

Step 2: • Note that $b_2 \leftarrow b_2 - \left\lfloor \frac{\langle b_2, b_1 \rangle}{\|b_1\|^2} \right\rceil \cdot b_1$

$$\Rightarrow \frac{\langle b_2, b_1 \rangle}{\|b_1\|^2} \leftarrow \frac{\langle b_2, b_1 \rangle}{\|b_1\|^2} - \left\lfloor \frac{\langle b_2, b_1 \rangle}{\|b_1\|^2} \right\rceil \cdot \frac{\langle b_1, b_1 \rangle}{\|b_1\|^2}$$

$\Rightarrow |\mu_{2,1}|$ is being reduced to $\leq 1/2$.

• The same holds true for $|\mu_{i,i-1}|$, $i \in [m]$.
• Also, the transformation is unimodular, so the lattice remains unchanged.

Step 3: • To show that this step will not happen many times, we need a potential function:

$$D(b_1, \dots, b_m) := \prod_{i=1}^{m} \|b_i^*\|^{2(m-i)}.$$

• Step 2 has no effect on this.

While each Step 2 swap reduces D by a factor of $\frac{\|b_{i+1}^*\|^2}{\|b_i^*\|^2} < \left(\frac{3}{4} - \mu_{i+1,i}^2\right) < \frac{3}{4}$.

Lemma 3: $|D(b_1, \dots, b_m)|$ is a positive integer of value under $2^{\tilde{O}(n^5 l)}$.

Proof:

• Write D as $\prod_{j=1}^{m-1} D_j$, where $D_j := \prod_{i=1}^{j} \|b_i^*\|^2$.

• We now relate D_j with vol $(b_1, \dots, b_j)$:

• D_j is the det. of $(b_1^*, \dots, b_j^*)^T \cdot (b_1^*, \dots, b_j^*)$ which is the same as $((b_1, \dots, b_j) \cdot C)^T \cdot ((b_1, \dots, b_j) \cdot C)$, for a unimodular transformation C.

$\Rightarrow D_j = |(b_1, \dots, b_j)^T \cdot (b_1, \dots, b_j)| \in \mathbb{Z}_{>0}$.

• The bound follows from the size of b_j's:

$$D_j = 2^{\tilde{O}(n^3 \cdot \ell) \cdot j}$$

$$\Rightarrow D = 2^{\tilde{O}(n^5 \cdot \ell)}. \qquad \square$$

▷ Thus, Step 3 can repeat at most $\tilde{O}(n^5 \cdot \ell)$ times in the L^3-algorithm.

▷ A crude time estimate of poly. fact. algorithm is then $\tilde{O}(n^{11} \cdot \ell^2)$.

▷ Assuming $L := \max$ bit-size in b_i's, we get a crude estimate for the L^3-algo. (to approximate a shortest vector) of: $\tilde{O}(L \cdot m \cdot m^2)^2 = \tilde{O}(m^6 \cdot L^2)$.

for pre-processing ↑ ↑ for the potential-fn.

Application to simultaneous Diophantine approx.

- L^3-algorithm, & the idea of reduced basis, is used in many places.

- Eg. computational problems in algebraic number theory, faster arithmetic in number fields, knapsack problem, testing conjectures (Merten's conjecture, ABC-conjecture, ...).

- The main reason is the following property of L^3: [relation to the volume]

Theorem: If $b_1, \dots, b_n$ is a reduced basis for a lattice $L \triangleleft \mathbb{Z}^n$ & $b_1^*, \dots, b_n^*$ is its GSO, then:

(i) $\|b_j\| \leq 2^{\frac{i-1}{2}} \cdot \|b_i^*\|$, $\forall\ 1 \leq j \leq i \leq n$.

(ii) $d(L) \leq \prod_{i=1}^{n} \|b_i\| \leq 2^{n(n-1)/4} \cdot d(L)$.

(iii) $\|b_1\| \leq 2^{\frac{n-1}{4}} \cdot d(L)^{1/n}$.

[$d(L) := |(b_1, \dots, b_n)|$ is the determinant of L.]

Proof:

(i) We have $\|b_j\|^2 = \|b_j^*\|^2 + \sum_{k=1}^{j-1} \mu_{jk}^2 \cdot \|b_k^*\|^2$.

$\leq \|b_j^*\|^2 + \sum_{1 \leq k \leq j-1} \frac{1}{4} \cdot 2^{j-k} \cdot \|b_j^*\|^2$

$= \|b_j^*\|^2 \cdot \left(1 + \frac{2^{j-2}}{4}\right) \leq 2^{j-1} \cdot \|b_j^*\|^2$.

(ii) By unimodularity of GSO, $d(L) = |(b_1^*, \dots, b_n^*)|$.

$\Rightarrow \quad d(L) = \prod_{1 \le i \le n} \|b_i^*\|$

- Since $\|b_i^*\| \le \|b_i\|$, we get $d(L) \le \prod_{i=1}^{n} \|b_i\|$.

- From (i) we have, $\prod_{j=1}^{n} \|b_j\| \le \prod_{j=1}^{n} 2^{\frac{j-1}{2}} \cdot \|b_j^*\|$
$= 2^{n(n-1)/4} \cdot d(L)$.

(iii) Putting $j=1$ in (i) we have,

$$\prod_{1 \le i \le n} \|b_1\|^2 \le \prod_{1 \le i \le n} 2^{i-1} \cdot \|b_i^*\|^2$$

$$\Rightarrow \quad \|b_1\|^{2n} \le 2^{n(n-1)/2} \cdot d(L)^2$$

$$\Rightarrow \quad \|b_1\| \le 2^{\frac{n-1}{4}} \cdot d(L)^{1/n}.$$

□

— Suppose we are given <u>rationals</u> $\alpha_1, \dots, \alpha_n, \varepsilon$ & we want to find <u>integers</u> $p_1, \dots, p_n, q$ s.t. $\forall i, \ |p_i - q \cdot \alpha_i| \le \varepsilon$ & q is "small".

— L^3 provides a poly-time algorithm!

- Idea: Consider the lattice $\mathcal{L}$ generated by the columns of

$$B = \begin{pmatrix} 1 & 0 & \cdots & 0 & -\alpha_1 \\ 0 & 1 & \cdots & 0 & -\alpha_2 \\ \vdots & & & \vdots & \\ 0 & 0 & \cdots & 1 & -\alpha_n \\ 0 & 0 & \cdots & 0 & 2^{-n(n+1)/4} \cdot \varepsilon^{n+1} \end{pmatrix}$$

- It has elements like

$$(p_1 - q\alpha_1,\ p_2 - q\alpha_2,\ \ldots,\ p_n - q\alpha_n,\ q \cdot 2^{\frac{-n(n+1)}{4}} \cdot \varepsilon^{n+1}),$$

for integers $p_1, \ldots, p_n, q$. ------(a)

- By the previous theorem, L^3-algo. gives a vector b_1 in poly-time s.t.

$$\|b_1\| \leq 2^{n/4} \cdot d(\mathcal{L})^{1/n+1} = \varepsilon.$$

$\Rightarrow$ the p's & q corresponding to b_1 in eqn.(a) are not <u>too large</u>.

$\triangleright$ In particular, $q \leq 2^{n(n+1)/4} \cdot \varepsilon^{-n}$.