Lattice basis reduction with GP-pari

G. Hanrot (INRIA Lorraine & LORIA)

e-mail : hanrot@loria.fr

September 17, 2004

Plan of the talk

I- Lattice basis reduction. What is it, how does it work ?

II- A few applications of lattice basis reduction

  a- "Black box" applications : polynomial factorization, [...]

  b- other applications :

   – simultaneous rational approximation

   – exponential diophantine equations

   – small roots of modular equations

III- Implementation and perspectives

  a- the pari implementation

  b- comparison with NTL, magma

  c- other notions of reduction and applications

Lattice basis reduction

Def. A lattice is a discrete subgroup of $\mathbb{R}^n$. Basis $= (e_i)_{1\le i\le k}$.

- $(e_i)_{1\le i\le k}$ basis $\Rightarrow M(e_i)_{1\le i\le k}$ basis for $M \in SL_k(\mathbb{Z})$.
- Bases are not all of equivalent interest.
- Which is the best basis ? Is it easy to compute ?
- Tradeoff ?
- Can also be formulated in terms of quadratic forms on $\mathbb{Z}^k$.

Notations

Gram-Schmidt orthogonalization :

$$b_i^* = b_i - \sum_{j<i} m_{ij} b_j^*, \quad m_{ij} = (b_i | b_j^*)/\|b_j^*\|^2.$$

$$\det(L) = \prod \|b_i^*\|.$$

Notions of reduction

- Size-reduced bases : $m_{ij} \leq 1/2$. Polynomial.
- Optimality : Minkowski reduction (shortest possible vectors). NP-hard.
- Korkine-Zolotarev : proper + $\|b_i^*\|$ minimal in $\text{proj}^{\perp}_{<b_1,\ldots,b_{i-1}>} < b_1, \ldots, b_i >$. NP-hard.
- s-Siegel reduction : proper + $\|b_{i+1}^*\|/\|b_i^*\| \geq s$. Polynomial (LLL).
- t-Lovász reduction : proper + $\|b_{i+1}^*\|/\|b_i^*\| \geq \sqrt{1/t^2 - m_{i+1,i}^2}$. (hence $(1/t^2 - 1/4)^{1/2}$-Siegel reduced). Polynomial (LLL).

Properties of LLL-reduced bases

– $|m_{ij}| \leq 1/2$.
– (almost) shortest vectors : s-Siegel reduced bases have $\|b_i\| \geq s^{1-n}\lambda_i(L)$

– almost orthogonal : $\prod \|b_i^*\| \geq s^{p(1-p)/2} \prod \|b_i\|$.

The LLL algorithm

From NTL documentation (V. Shoup) :

> (...) I think it is safe to say that nobody really understands how the LLL algorithm works. The theoretical analyses are a long way from describing what "really" happens in practice. (...)

The LLL algorithm

- Due to Lenstra, Lenstra, Lovász (82), initially developed for polynomial factorization over $\mathbb{Q}$.
- Gives a Lovász-reduced basis in polynomial time for $t < 1$.
- Gives short vectors of the lattice (and transformation matrix).
- Gives provably almost shortest vectors.

The LLL algorithm, principles

- Start with $i = 2$.
- Size-reduce b_i wrt. b_{i-1}.
- If Lovász condition is verified, $i \leftarrow i + 1$ + size-reduce b_i wrt b_{i-2}, ..., b_1 ; else swap b_i and b_{i-1} and $i \leftarrow \max(2, i - 1)$.

Note : need to update Gram-Schmidt coefficients m_{ij} after each size-reduction / swap.

The LLL algorithm, complexity

– $\Delta = \prod_{i=1}^{n} \prod_{j=1}^{i} \|b_i^*\| \leq M^{n^2}$
– SR does not change Δ
– swap decreases Δ by a factor $\geq t$.

$\Rightarrow$ at most $O(n^2 \log_{1/t} M)$ steps.

The LLL algorithm, Gram-Schmidt coefficients

- Exact coefficients : rational numbers $\Rightarrow$ tons of gcds or huge height.
- Integral variant (de Weger) : precompute a common denominator for all m_{ij}.
- Floating-point variants : compute m_{ij} as floating-point numbers.

For the execution time, $3 > 2 > 1$. But SR $\rightarrow m_{ij} \leftarrow m_{ij} - \lfloor m_{ij} \rceil \Rightarrow$ huge precision loss...

A few heuristics on reduced bases

Warning : valid only for "random" lattices, with no real short vectors.

– almost orthogonal, hence $\|b_i\| \approx \|b_i^*\|$.
– all $\|b_i^*\|$ roughly of the same size
– hence $\|b_i\| \approx (\det(L))^{1/dim(L)}$.
– usually all coordinates of b_i of the same order of magnitude.

Applications of LLL

Part I - blackbox applications

- linear/polynomial relations ;
- factorization over $\mathbb{Q}$;
- polynomial reduction ;
- ideal reduction.

Knapsack and linear relations

see also PSLQ. Let $\alpha_1, \ldots, \alpha_n \in \mathbb{R}^n$. A short vector of the lattice generated by the columns of

$$\begin{pmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \\ [C\alpha_1] & [C\alpha_2] & \ldots & [C\alpha_{n-1}] & [C\alpha_n] \end{pmatrix}$$

is a n-uple $(a_1, \ldots, a_n)$ such that $a_1^2 + \ldots + a_{n-1}^2 + (\sum_{i=1}^n a_i[C\alpha_i])^2$ is (almost) short(est). Heuristics $\Rightarrow$ if no relation, shortest $\approx C^{1/n}$, with $a_i \approx C^{1/n}$. *See* `lindep, algdep`.

Knapsack and linear relations (2)

Further facts :

– Can be used for simultaneous approximation (eg. when several conjugates are known) ;

– If no relation, can get a Thm : for any $|a_i| \leq B$, $|\sum a_i \alpha_i| \geq \lambda$.

– Can force different orders of magnitude for various b_i by modifying the 1 on diagonal (recall that all coordinates of short vectors have the same order).

– Applications to cryptoanalysis of knapsack systems.

Example : `lindep`, `algdep`.

Knapsack and linear relations (3)

CVP : given a lattice $L \subset \mathbb{R}^n$ and a vector $v \in \mathbb{R}^n$, find $x \in L$ so that $|x - v|$ minimal.

Naive idea : let B be a matrix generating L, take $x = \lfloor B^{-1}v \rceil$.

Good approximation when the basis is close to orthogonal, may be arbitrarily bad in general $\Rightarrow$ first LLL-reduce B.

Other way : LLL-reduce $L \times \{0\} \oplus (x_1, \ldots, x_n, 1)\mathbb{Z} \subset \mathbb{R}^{n+1}$.

Knapsack and linear relations (4)

Application : simultaneous rational approximation. Let $\alpha_1, \ldots, \alpha_n$ be real numbers, find $p_1, \ldots, p_n, q$ such that $|q\alpha_i - p_i|$ all small.
Pigeonhole principle $\Rightarrow$ there is a solution with $|q\alpha_i - p_i| \leq q^{-1/n}$.

Take

$$M = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ [C\alpha_1] & -C & 0 & \ldots & 0 \\ [C\alpha_2] & 0 & -C & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ [C\alpha_n] & 0 & \ldots & 0 & -C \end{pmatrix}$$

Determinant $\approx C^n$, dimension $n+1$ hence shortest vector should be $C^{n/(n+1)}$.

Exponential diophantine equations

well, some of them.

Example (de Weger, 1987) : search for ABC examples.

Try to solve $|p_1^{t_1} \ldots p_k^{t_k} - q_1^{s_1} \ldots q_l^{s_l}| \leq D$, $t_i \in \mathbb{Z}$. ie.

$$|\sum t_i \log p_i - \sum s_l \log q_l| \leq D q_1^{-s_1} \ldots q_l^{-s_l} \ll D \exp(-O(\max|s_i|)). (1)$$

Baker's bound $\Rightarrow$ lhs $\gg \exp(-C \log \max|s_i|)$, hence $|s_i|, |t_i| \leq E$, E usually huge.

But (1) + thm on lower bounds for relations yields $|s_i|, |t_i| \leq E'$, and so on... until $E^{(j)}$ quite small and hopefully feasible.

Similar : Thue equations.

Polynomial reduction

Let $P \in \mathbb{Z}[X]$ be an irreducible polynomial, find Q such that $\mathbb{Q}[X]/Q(X) \approx \mathbb{Q}[X]/P(X)$ and Q "small(est)".

Idea : search for a root α of Q as an element the ring of integers of $\mathbb{Q}[X]/P(X)$ with small conjugates.

$\Rightarrow$ Reduce the quadratic form with coefficients

$$m_{i,j} = \sum_{k=1}^{n} \sigma_k(\omega_i)\overline{\sigma_k}(\omega_j),$$

and a short vector $(b_1, \ldots, b_n)$ yields a small value of $\sum_{k=1}^{n} |\sigma_k(\sum b_i\omega_i)|^2$ + compute the minimal polynomial of $\alpha = \sum b_i\omega_i$.

Use Fincke-Pohst for the smallest. *see* `polred`, `polredabs`.

Polynomial factorization over $\mathbb{Q}$

- historical application (LLL'82), polynomial. Unpractical (dimension of the lattice = degree of the polynomial).
- van Hoeij (2002), practical.

Start by factoring over $\mathbb{Z}/p\mathbb{Z}$ for some suitable p, and lifting over $\mathbb{Z}/p^t\mathbb{Z}$, t large enough.

$$P(x) = \prod_{i=1}^{\ell} P_i(x), P_i \in \mathbb{Z}/p^t\mathbb{Z}[x].$$

Find recombinations, i.e. $e_i \in \{0,1\}^\ell$ with $\prod_{i=1}^{\ell} P_i^{e_i} \in \mathbb{Z}[x]$.

Polynomial factorization over $\mathbb{Q}$

Find recombinations, i.e. $e_i \in \{0,1\}^\ell$ with $\prod_{i=1} \ell P_i^{e_i} \in \mathbb{Z}[x]$.

Idea 1 : use $S_k(P_i) = \sum \alpha_i^k$, where α_i goes through the roots of P_i, which depend linearly on e_i :

$$S_k(\prod P_i^{e_i}) = \sum e_i S_k(P_i) \bmod p^t.$$

Idea 2 : if t large and $\prod_{i=1} \ell P_i^{e_i} \in \mathbb{Z}[x]$, then $S_k(\prod_{i=1} \ell P_i^{e_i})$ "small", say $< p^\alpha$.

$\Rightarrow$ find $(e_i), \lambda_k \in \mathbb{Z}$ such that $\sum e_i \lfloor S_k(P_i)/p^\alpha \rfloor - \lambda_k p^{t-\alpha}$ small...

$\Rightarrow$ knapsack.

An iterative step

Goal : find the lattice of (e_i) such that $\prod_{i=1} \ell P_i^{e_i} \in \mathbb{Q}(x)$.

Start with $L = \mathbb{Z}^\ell$, at each step build $L' \subset L$ until L' has a 0/1 basis giving $\mathbb{Z}[x]$ factors of P.

Lemma. Let (b_i) a basis of a lattice. Si $\|b_j^*\| > B$ for $j \geq j_0$, then if v is such that $\|v\| \leq B$, $v \in< b_1, \ldots, b_{j_0-1} >$.

Way better if (b_i) LLL-reduced... Since any valid (e_i) yields a small linear combination, $(e_i) \in$ sublattice of L.

Generalization to number fields by K. Belabas.

Example : `factor`.

Ideal reduction

Data : an algebraic number field $\mathbb{K} = \mathbb{Q}(\alpha)$, an ideal I in $\mathbb{K}$, in HNF. Is I principal ?

- Hard in general.
- can be approximated using LLL to find a candidate generator (in a better world an element of smallest norm).
- Find a small element in the ideal, like in `polred`

Take the quadratic form $\sum_{k=1}^{n} \sigma_k(\theta_i)\overline{\sigma}_k(\theta_j) \Rightarrow$ find α "small" in I.

Possibly reduce in several "directions" by adding weights. Replace I by I/α, where α is "small".

Ideal reduction (2)

A funny example : find a generator for $Cl(\mathbb{Q}(\zeta_{83}))$ (for which $h(\mathbb{K}) = 3$).

- $(3) = \mathfrak{p}_3\overline{\mathfrak{p}}_3$
- `idealred`$(\mathfrak{p}_3)$ is nontrivial ;
- `idealred`$(\mathfrak{p}_3^3)$ yields a generator ξ ;

but if $\mathfrak{p}_3$ were principal, ξ would be a cube (up to a unit), hence $\xi/\overline{\xi}$ would be a cube $\Rightarrow$ find some p modulo which this is not the case...

Small roots of modular equations

- Method due to Hastad and independently to Girault, Vallée, Toffin, generalized by Coppersmith.
- Given $P(X) \bmod N$, find all $z \leq N^{\alpha}$ such that $P(z) = 0 \bmod N$...
- ... without factoring N.
- ... and generalization to several variables.

Small roots of modular equations (II)

- Idea : find Q such that $\mathrm{Zeros}(P \bmod N) \subset \mathrm{Zeros}(Q \bmod N)$
- and Q has small $\|.\|_2 = \left(\sum_{i=0}^{d} q_i^2\right)^{1/2}$
- But $|z| \leq X \Rightarrow |Q(z)| \leq \|Q\|_2 \sqrt{d} X^d$
- rhs $< N \Rightarrow Q(z) = 0$ in $\mathbb{Z}$.

Small roots of modular equations (III)

Finding Q

- Idea : build a lattice of polynomials Q st for any $Q \in L$, $P(z) = 0 \bmod N \Rightarrow Q(z) = 0 \bmod N$.
- Use N, Nx^k, $P(x)$, $x^k P(x)$ for various values of k.
- Find a short(est) vector in L

Small roots of modular equations (IV)

Finding Q

A few improvements :

– replace $P(x)$ by $P(xX)$ (all coefficients of same size, sharper ineq.)

– build polynomials that are 0 mod N^ℓ, using $N^i P(xX)^{\ell-i} x^j$.

– choose bases that generate lattices with smallest possible "root-discriminant".

– several variables generalization is not difficult, but cannot get a theorem.

Small roots of modular equations (IV)

Theorem One can find all $|x| \leq N^{1/d-\epsilon}$ such that $P(x) = 0 \bmod N$ in polynomial time, where d is the degree of P.

Straightforward application : RSA, $e = 3$, if the high $2/3$ of the cleartext are known, the last third can be recovered.

Application 1 : Factoring with high bits known

[Coppersmith] Assume $N = p \cdot q$ with P given so that $|p - P| \leq N^\tau$. Put $f(x) = P + x$, then we want to solve $f(x) = 0 \bmod p$ with $x < X = N^\tau$.

Choose the families $g_i(x) = N^{m-i} f^i(x), i = 0..m$ and $h_j(x) = x^j f^m(x), j = 0..m \Rightarrow (2m+1)$-dimensional lattice with det $= N^{m(m+1)/2} X^{m(2m+1)}$.

Need $\det(L)^{1/(2m+1)} \approx \|f(xX)\| < p^m/\sqrt{n}$, which yields asymptotically $N^{1/4}X < p < N^{1/2}$ hence $X < N^{1/4}$.

[Boneh, Durfee, Howgrave-Graham] Similar approach yields polynomial factorization of $N = p^r q$ asymptotically as soon as $|P - p| < p^{1-c/(r+c)}$, with $q < p^c$.

Application 2. List decoding of CRT codes

[Boneh, 2002]

Problem : N an integer, p_i a set of primes with $P = \prod p_i \gg N$. Let $\phi : x \leq N \mapsto (x \bmod p_i)$ be an error correcting code. *How many errors can be list-decoded*, ie. if R is received and $M(m) = \gcd(P, m - R)$, what is the smallest M for which we can find all possible $x \leq N$?

Put

$$g_i(x) = P^{a-i}(x - R)^i, i = 0..a - 1$$

$$h_i(x) = (x - R)^a x^i, i = 0..a' - 1$$

Application 2. List decoding of CRT codes

We have $g_i(m) = 0 \bmod M^a$ and $h_i(m) = 0 \bmod M^a$ for all i.

Then $\det(L) = P^{a(a+1)/2} B^{(a+a')(a+a'-1)/2}$ and dimension is $a + a'$.

We want $\det(L)^{1/(a+a')} \approx M^a$, hence we need $M > P^{(a+1)/2(a+a')} N^{(a+a'-1)/2a}$, and we can fix $a + a'$ and optimize a.

Cryptanalysis of low-exponent RSA

[Boneh, Durfee] Assume d is small, $(p-1, q-1) = 2$. Since $de = 1 \bmod \varphi(N)/2$, have $de - 1 = k((N+1)/2 - (p+q)/2)$, ie. $k(A+s) = 0 \bmod e$, where k and s are unknown and small.

Thm. If $d < N^{1-1/sqrt2} \approx N^{0.292}$, bivariate Coppersmith method can yield p, q in polynomial time.

Proof is more technical. $N^{0.284}$ easy : use $x^i f^k(x,y) e^{m-k}$ and $y^j f^k(x,y) e^{m-k}$ + careful optimization.

Application 4(?) : LD on EC

[Semaev 2003]

Let E be an elliptic curve over a field $\mathbb{K}$ of characteristic > 3. Define $f_n(X_1, \ldots, X_n)$ such that for $(x_1, \ldots, x_n) \in \overline{K}$, $f_n(x_1, \ldots, x_n) = 0$ iff. $\exists (y_i) \in \overline{K}$ with $(x_i, y_i) \in E$ and the n points sum to 0.

f_n is of degree 2^{n-2} in each variable.

$$\begin{aligned} f_n(x_1, \ldots, x_n) = \quad & \mathrm{Res}_X \quad (f_{n-k}(X_1, \ldots, X_{n-k-1}, X), \\ & \qquad\qquad f_{k+2}(X_{n-k}, \ldots, X_n, X)), n \geq 4. \end{aligned}$$

Application 4(?) : LD on EC

Take $\mathbb{K} = \mathbb{Z}/p\mathbb{Z}$ and $FB = \{R \in E(K), |x(R)| \leq p^{\tau}\}$. Put $(x, y) = l_1 P + l_2 Q$. If we can find small solutions to $f_{n+1}(x_1, \ldots, x_n, x) = 0 \bmod p$, we can find relations $l_1 P + l_2 Q = \sum_{R \in FB} a_i R_i$.

$\Rightarrow$ with enough relations, solve DL.

Is Coppersmith applicable ?

Implementation and perspectives

Goals :

- evaluate pari efficiency wrt. other systems (NTL, magma)
- how to improve efficiency in the floating-point case ?
- recent (or less recent) technology that could be included ?

pari/NTL/magma comparison

Quick and dirty bench on P4 3 GHz, pari-cvs June 11, NTL-5.3.2, magma-2.10 Knapsack-type lattices.

Integer versions, small coefficients (100 bits)

	pari	NTL/GMP	Magma
5x5	0.7ms	0.7ms	2ms
10x10	4.6ms	4.9ms	8.5ms
20x20	12ms	36ms	58ms
50x50	92ms	338ms	440ms
100x100	0.57s	0.77s	0.91s
200x200	3.3s	2.9s	3.9s
500x500	35s	29s	477s

pari/NTL/magma comparison

Quick and dirty bench on P4 3 GHz, pari-cvs June 11, NTL-5.3.2, magma-2.10 Knapsack-type lattices.

Integer versions, medium coefficients (1000 bits)

	pari	NTL/GMP	Magma
5x5	0.1s	0.05ms	0.27s
10x10	0.72s	0.37s	1.7s
20x20	5.7s	2.8s	11.8s
50x50	35s	30s	117s
100x100	120s	161s	468s
200x200	180s	160s	20min

pari/NTL/magma comparison

Quick and dirty bench on P4 3 GHz, pari-cvs June 11, NTL-5.3.2, magma-2.10 Knapsack-type lattices.

Integer versions, huge coefficients (10000 bits)

	pari	NTL/GMP	Magma
5x5	57s	12s	121s
10x10	420s	140s	800s
20x20	52min	17min	>60min

pari/NTL/magma comparison

Quick and dirty bench on P4 3 GHz, pari-cvs June 11, NTL-5.3.2, magma-2.10 Knapsack-type lattices.

FP versions, small coefficients (100 bits)

	pari	NTL/GMP	Magma
5x5	1ms	0.3ms(F)	0.6ms
10x10	4.6ms	1.5ms(F)	2.7ms
20x20	0.1s	9ms(F)	15ms
50x50	5s	80ms(F)	180ms
100x100	81s	0.23s(F)	0.46s
200x200	>2min	1.2s(F)	2.3s
500x500	>20min	12s	27s

pari/NTL/magma comparison

Quick and dirty bench on P4 3 GHz, pari-cvs June 11, NTL-5.3.2, magma-2.10 Knapsack-type lattices.

FP versions, medium coefficients (1000 bits)

	pari	NTL/GMP	Magma
5x5	19ms	5ms(F,G)	9ms
10x10	84ms	29ms(F,G)	43ms
20x20	0.7s	0.2s(F,G)	0.3s
50x50	20s	2.3s(F,G)	4.8s
100x100	>20min	13s(F,G)	31s
200x200	>20min	63s(F,G)	263

pari/NTL/magma comparison

Quick and dirty bench on P4 3 GHz, pari-cvs June 11, NTL-5.3.2, magma-2.10 Knapsack-type lattices.

FP versions, huge coefficients (10000 bits)

	pari	NTL/GMP	Magma
5x5	12s	0.2s(XD)	0.46s
10x10	29s	0.9s(XD)	1.7s
20x20	3min	6s(XD)	9.8s

Plans (?) for further development

A few (very personal) suggestions.

- Improve floating-point implementation (Schnorr-Euchner) and test unimplemented technologies (Koy-Schnorr ?)
- A generic unimodular Coppersmith implementation ;
- Exact (Minkowski) reduction in dimension ≤ 4 [Nguyen-Stehle 2004] ;
- Sharper reductions : exact SVP, CVP ;
- BKZ ?

Floating-point variants

- The results seem to show that pari floating-point code should be checked thoroughly.
- Wild guess (no time to check it) : awkward management of precision.
- Comparison with NTL code.
- continue experiments with Koy-Schnorr ideas & more stable orthogonalization techniques (`#if 0 ... #endif` currently).

SVP, CVP

– SVP = find shortest vector in a lattice. NP-Hard. Algorithm known in $n^{n/2+o(n)}$, randomized in $2^{O(n)}$, the latter unpractical.

– CVP = find the vector in a lattice which is closest to a given vector. Still harder. Algorithm in $n^{3n/2+o(n)}$, randomized for $(1+\epsilon)$-CVP in $2^{O(n(1+1/\epsilon))}$.

Can be done up to, say, dimension 20. Basic block for BKZ.

Sharper (BKZ) reduction, SVP

Korkine-Zolotarev : size-reduced + $\|b_i^*\|$ minimal in $\mathrm{proj}^{\perp}_{<b_1,\ldots,b_{i-1}>} < b_1, \ldots, b_i >$.

LLL : size-reduced + (b_i, b_{i+1}) (almost)-Gauss reduced

k-BKZ reduction : size-reduced + $(b_i, \ldots, b_{i+k-1})$ (almost)-KZ-reduced for all $i < n - k + 1$.

Need to solve a k-dimensional SVP.

Main interest : improve the quality of the basis obtained.
Implemented in NTL, can be made efficient via some fine tuning.
Mostly useful for crypto-ish applications, where one really wants *the* shortest vector ?