Lattice Basis Reduction
Part II: Algorithms

Sanzheng Qiao

Department of Computing and Software
McMaster University, Canada
qiao@mcmaster.ca
www.cas.mcmaster.ca/~qiao

November 8, 2011, revised February 2012

Joint work with W. Zhang and Y. Wei, Fudan University
Outline

1. Hermite Reduction
2. LLL Reduction
3. HKZ Reduction
4. Minkowski Reduction
5. A Measurement
Outline

1. Hermite Reduction
2. LLL Reduction
3. HKZ Reduction
4. Minkowski Reduction
5. A Measurement
Hermite-reduced
A lattice basis \( \{b_1, b_2, \ldots, b_n\} \) is called size-reduced if its QR decomposition satisfies

\[ |r_{i,i}| \geq 2|r_{i,j}|, \quad \text{for all} \quad 1 \leq i < j \leq n, \]
Hermite reduction (size reduction)

Hermite-reduced

A lattice basis \( \{ b_1, b_2, \ldots, b_n \} \) is called size-reduced if its QR decomposition satisfies

\[
|r_{i,i}| \geq 2|r_{i,j}|, \quad \text{for all} \quad 1 \leq i < j \leq n,
\]

Procedure \textbf{Reduce}(i, j)

\[
\begin{bmatrix}
 r_{i,i} & r_{i,j} \\
 r_{j,i} & r_{j,j}
\end{bmatrix} \begin{bmatrix}
 1 & \left\lfloor \frac{r_{i,j}}{r_{i,i}} \right\rfloor \\
 0 & 1
\end{bmatrix} = \begin{bmatrix}
 r_{i,i} & r_{i,j} - r_{i,i} \left\lfloor \frac{r_{i,j}}{r_{i,i}} \right\rfloor \\
 r_{j,i} & r_{j,j}
\end{bmatrix}
\]

\[
|r_{i,i}| \geq 2 \left| r_{i,j} - r_{i,i} \left\lfloor \frac{r_{i,j}}{r_{i,i}} \right\rfloor \right|
\]
Gauss reduction

A unimodular transformation

\[
\begin{bmatrix}
1 & -\mu \\
0 & 1
\end{bmatrix}
\]
or
\[
\begin{bmatrix}
1 & 0 \\
-\mu & 1
\end{bmatrix}
\]

Also called

Integer Gauss transformation
Integer elementary matrix
Outline

1. Hermite Reduction
2. LLL Reduction
3. HKZ Reduction
4. Minkowski Reduction
5. A Measurement
LLL-reduced

A lattice basis \( \{b_1, b_2, \ldots, b_n\} \) is called LLL-reduced if it is size-reduced and \( R \) in the QR decomposition satisfies

\[
r_{i+1,i+1}^2 + r_{i,i+1}^2 \geq \omega \ r_{i,i}^2
\]
LLL reduction

LLL-reduced

A lattice basis \( \{b_1, b_2, \ldots, b_n\} \) is called LLL-reduced if it is size-reduced and \( R \) in the QR decomposition satisfies

\[
    r_{i+1,i+1}^2 + r_{i,i+1}^2 \geq \omega r_{i,i}^2
\]

Procedure \texttt{SwapRestore}(i)

Find a Givens plane rotation \( G \):

\[
    G \begin{bmatrix}
        r_{i-1,i-1} & r_{i-1,i} \\
        0 & r_{i,i}
    \end{bmatrix}
    \begin{bmatrix}
        0 & 1 \\
        1 & 0
    \end{bmatrix}
    =
    \begin{bmatrix}
        \hat{r}_{i-1,i-1} & \hat{r}_{i-1,i} \\
        0 & \hat{r}_{i,i}
    \end{bmatrix}.
\]

Unimodular transformation: Permutation
LLL algorithm

```plaintext
k = 2;
while k <= n {
    if \(|r(k-1,k) / r(k-1,k-1)| > 1/2

        if r(k,k)^2 + r(k-1,k)^2 < w*r(k-1,k-1)^2 {

            }
    
} else {

    }
```
LLL algorithm

\[ k = 2; \]
\[ \text{while } k \leq n \{ \]
\[ \quad \text{if } \left| \frac{r(k-1,k)}{r(k-1,k-1)} \right| > \frac{1}{2} \]
\[ \quad \quad \text{Reduce}(k-1,k); \]
\[ \quad \text{if } r(k,k)^2 + r(k-1,k)^2 < w \cdot r(k-1,k-1)^2 \{ \]
\[ \quad \quad \} \text{ else } \{ \]
\[ \quad \} \]
\[ \} \]
LLL algorithm

\[ k = 2; \]
\[ \text{while } k \leq n \{ \]
\[ \quad \text{if } \left| \frac{r(k-1,k)}{r(k-1,k-1)} \right| > 1/2 \]
\[ \quad \quad \text{Reduce}(k-1,k); \]
\[ \quad \text{if } r(k,k)^2 + r(k-1,k)^2 < w \cdot r(k-1,k-1)^2 \{ \]
\[ \quad \quad \text{SwapRestore}(k); \]
\[ \quad \quad k = \text{max}(k-1, 2); \]
\[ \} \quad \text{else} \{ \]
\[ \} \]
\[ \} \]
k = 2;
while k <= n {
    if |r(k-1,k) / r(k-1,k-1)| > 1/2
        Reduce(k-1,k);
    if r(k,k)^2 + r(k-1,k)^2 < w*r(k-1,k-1)^2 {
        SwapRestore(k);
        k = max(k-1, 2);
    } else {
        for i = k-2 downto 1
            if |r(i,k) / r(i,i)| > 1/2
                Reduce(i,k);
        k = k+1;
    }
}
LLL algorithm

k = 2;
while k <= n {
    if |r(k-1,k) / r(k-1,k-1)| > 1/2
        Reduce(k-1,k);
    if r(k,k)^2 + r(k-1,k)^2 < w * r(k-1,k-1)^2 {
        SwapRestore(k);
        k = max(k-1, 2);
    } else {
        for i = k-2 downto 1
            if |r(i,k) / r(i,i)| > 1/2
                Reduce(i,k);
        k = k+1;
    }
}

Redundant size reductions.
k = 2;
while k <= n
    g = round(r(k-1,k) / r(k-1,k-1));
    if r(k,k)^2 + (r(k-1,k) - g*r(k-1,k-1))^2 < w*r(k-1,k-1)^2
        ReduceSwapRestore(k);
        k = max(k-1, 2);
    else
        k = k + 1;

for k = 2 to n
    for i = k-1 downto 1
        if |r(i,k) / r(i,i)| > 1/2
            Reduce(i,k);
An improvement: Delayed size reduction

\[
\begin{align*}
&k = 2; \\
&\text{while } k \leq n \\
&\quad \quad g = \text{round}(r(k-1,k) / r(k-1,k-1)); \\
&\quad \quad \text{if } r(k,k)^2 + (r(k-1,k) - g \cdot r(k-1,k-1))^2 < w \cdot r(k-1,k-1)^2 \\
&\quad \quad \quad \text{ReduceSwapRestore}(k); \\
&\quad \quad \quad k = \max(k-1, 2); \\
&\quad \text{else} \\
&\quad \quad \quad k = k + 1; \\
&\text{for } k = 2 \text{ to } n \\
&\quad \text{for } i = k-1 \text{ downto } 1 \\
&\quad \quad \quad \text{if } |r(i,k) / r(i,i)| > 1/2 \\
&\quad \quad \quad \quad \text{Reduce}(i,k); \\
\end{align*}
\]

Produces identical results at 50% cost.
Outline

1. Hermite Reduction
2. LLL Reduction
3. HKZ Reduction
4. Minkowski Reduction
5. A Measurement
HKZ reduction

A lattice basis \( \{b_1, b_2, \ldots, b_n\} \) is called HKZ-reduced if it is size-reduced and for each trailing \((n - i + 1) \times (n - i + 1)\), \(1 \leq i < n\), submatrix of \( R \) in the QR decomposition, its first column is a shortest nonzero vector in the lattice generated by the submatrix.
HKZ reduction

A lattice basis \( \{b_1, b_2, \ldots, b_n\} \) is called HKZ-reduced if it is size-reduced and for each trailing \((n - i + 1) \times (n - i + 1)\), \(1 \leq i < n\), submatrix of \( R \) in the QR decomposition, its first column is a shortest nonzero vector in the lattice generated by the submatrix.

Two problems

- Shortest vector problem (SVP)
- Expansion to a basis
SVP

\[
\min_z \|Bz\|_2^2
\]
SVP

\[
\min_{z} \| Bz \|_2^2
\]

Sphere decoding
Determine a search sphere

\[
\| Bz \|_2^2 \leq \rho^2
\]
SVP

\[
\min_z \|Bz\|_2^2
\]

Sphere decoding
Determine a search sphere

\[
\|Bz\|_2^2 \leq \rho^2
\]

A simple choice of \(\rho\): the length of the first (or shortest) column of \(B\).
Example

\[ Rz = \begin{bmatrix} 4 & 1 & 5 \\ 0 & 4 & 4 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \]

\[ \rho = 4 \]
Example

\[ Rz = \begin{bmatrix} 4 & 1 & 5 \\ 0 & 4 & 4 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \]

\[ \rho = 4 \]

A necessary condition for \( z_3 \): \(|3z_3| \leq 4\).

Possible values of \( z_3 \): 0, –1, 1
Example

For each possible values of $z_3$, say $z_3 = 0$,

\[
Rz = \begin{bmatrix}
4 & 1 & 5 \\
0 & 4 & 4 \\
0 & 0 & 3
\end{bmatrix}
\begin{bmatrix}
z_1 \\
z_2 \\
z_3
\end{bmatrix}
= \begin{bmatrix}
4 & 1 \\
0 & 4 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
z_1 \\
z_2
\end{bmatrix}
+ 0 \begin{bmatrix}
5 \\
4 \\
3
\end{bmatrix}
\]

The problem size is reduced.
Example

For each possible values of $z_3$, say $z_3 = 0$, 

$$Rz = \begin{bmatrix} 4 & 1 & 5 \\ 0 & 4 & 4 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 0 & 4 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + 0 \begin{bmatrix} 5 \\ 4 \\ 3 \end{bmatrix}$$

The problem size is reduced.

The necessary condition for $z_2$: $|4z_2| \leq 4$

Possible values of $z_2$: 0, −1, 1
Example

The search tree

```
          0
         / \  
       -1   1
      /    / \
0     0  \ 1
       /     /
-1    -1  1
```

The solution

\[
Rz = \begin{bmatrix} 4 & 1 & 5 \\ 0 & 4 & 4 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -3 \end{bmatrix}
\]
Expanding to a basis

Problem:
Transform the basis matrix

\[ A = \begin{bmatrix} 4 & 1 & 5 \\ 0 & 4 & 4 \\ 0 & 0 & 3 \end{bmatrix} \]

into a new basis matrix whose first column is the shortest vector

\[ Az = \begin{bmatrix} 0 \\ 0 \\ -3 \end{bmatrix} \]
Expanding to a basis

Problem:
Transform the basis matrix 

\[
A = \begin{bmatrix}
4 & 1 & 5 \\
0 & 4 & 4 \\
0 & 0 & 3 \\
\end{bmatrix}
\]

into a new basis matrix whose first column is the shortest vector 

\[
Az = \begin{bmatrix}
0 \\
0 \\
-3 \\
\end{bmatrix}
\]

That is, find a unimodular matrix \( Z \): 

\[
Az = AZe_1 \quad \text{or} \quad z = Ze_1, \quad Z^{-1}z = e_1
\]

Unimodular transformation that introduces zeros into an integer vector.
A plane unimodular transformation (Luk, Zhang, and Q, 2010).

\[ \gcd(p, q) = \pm d, \ ap + bq = \pm d. \]

Form the unimodular matrix

\[
\begin{bmatrix}
    a & b \\
    -q/d & p/d \\
\end{bmatrix}
\begin{bmatrix}
    p \\
    q \\
\end{bmatrix} =
\begin{bmatrix}
    d \\
    0 \\
\end{bmatrix}
\]
A unimodular transformation (Luk, Zhang, and Q, 2010).

\[ \gcd(p, q) = \pm d, \; ap + bq = \pm d. \]

Form the unimodular matrix

\[
\begin{bmatrix}
  a & b \\
  -q/d & p/d
\end{bmatrix}
\begin{bmatrix}
  p \\
  q
\end{bmatrix} =
\begin{bmatrix}
  d \\
  0
\end{bmatrix}
\]

Its inverse

\[
\begin{bmatrix}
  p/d & -b \\
  q/d & a
\end{bmatrix}
\]
Example

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 1 \\
\end{bmatrix}
\begin{bmatrix}
1 \\
1 \\
-1 \\
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
4 & 1 & 5 \\
0 & 4 & 4 \\
0 & 0 & 3 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & -1 & 1 \\
\end{bmatrix}
= 
\begin{bmatrix}
4 & -4 & 5 \\
0 & 0 & 4 \\
0 & -3 & 3 \\
\end{bmatrix}
\]
Example

\[
\begin{pmatrix}
1 & 0 & 0 \\
-1 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
1 \\
1 \\
0
\end{pmatrix}
=
\begin{pmatrix}
1 \\
0 \\
0
\end{pmatrix}
\]

\[
\begin{pmatrix}
4 & -4 & 5 \\
0 & 0 & 4 \\
0 & -3 & 3
\end{pmatrix}
\begin{pmatrix}
1 & 0 & 0 \\
1 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
=
\begin{pmatrix}
0 & -4 & 5 \\
0 & 0 & 4 \\
-3 & -3 & 3
\end{pmatrix}
\]
Improving Kannan’s algorithm

Kannan, 1987

Expansion method

In the $k$th, $k = 1, \ldots, n$, recursion, solve a $k$-dim system ($O(k^3)$).
Total $O(n^4)$
Improving Kannan’s algorithm

Kannan, 1987
Expansion method

In the $k$th, $k = 1, ..., n$, recursion, solve a $k$-dim system ($O(k^3)$).
Total $O(n^4)$

Determine whether a set of vectors are linearly dependent.
Improving Kannan’s algorithm

Kannan, 1987
Expansion method
In the $k$th, $k = 1, \ldots, n$, recursion, solve a $k$-dim system ($O(k^3)$).
Total $O(n^4)$
Determine whether a set of vectors are linearly dependent.

Our method
Efficient, $O(n^2)$
Accurate, unimodular (integer) transformations.
Properties

- Efficient.
- Exact, integer arithmetic.
- Include permutation and identity as special cases.
- Can triangularize an integer matrix.
- Any unimodular can be decomposed into a product of this plan unimodular and integer Gauss transformations.
Application

Cryptography

Find a large vector

\[ \mathbf{v} = \begin{bmatrix} 997 \\ 1234 \\ 56789 \end{bmatrix}, \quad \gcd(v_i) = 1 \]
Application

Cryptography

Find a large vector

\[
v = \begin{bmatrix}
997 \\
1234 \\
56789
\end{bmatrix}, \quad \gcd(v_i) = 1
\]

Determine a unimodular matrix

\[
Z^{-1}v = \begin{bmatrix}
997 & -1 & 0 \\
1234 & 0 & -543 \\
6789 & 0 & -24989
\end{bmatrix}^{-1} v = \begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix}
\]

\[
\text{cond}(Z) = 1.55 \times 10^{12}
\]
Application

Cryptography

Find a large vector

\[
v = \begin{bmatrix} 997 \\ 1234 \\ 56789 \end{bmatrix}, \quad \gcd(v_i) = 1
\]

Determine a unimodular matrix

\[
Z^{-1}v = \begin{bmatrix} 997 & -1 & 0 \\ 1234 & 0 & -543 \\ 6789 & 0 & -24989 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}
\]

\[
\operatorname{cond}(Z) = 1.55 \times 10^{12}
\]

Choose a diagonal \(A\) as a private key

\[
B = AZ \quad \text{(ill-conditioned)}
\]

as public key
Outline

1. Hermite Reduction
2. LLL Reduction
3. HKZ Reduction
4. Minkowski Reduction
5. A Measurement
Minkowski reduction

A lattice basis \( \{b_1, b_2, \ldots, b_n\} \) is called Minkowski-reduced if for each \( b_i, i = 1, 2, \ldots, n \), its length
\[
\|b_i\|_2 = \min(\|\hat{b}_i\|_2, \|\hat{b}_{i+1}\|_2, \ldots, \|\hat{b}_n\|_2)
\]
over all sets \( \{\hat{b}_i, \hat{b}_{i+1}, \ldots, \hat{b}_n\} \) of lattice points such that \( \{b_1, b_2, \ldots, b_{i-1}, \hat{b}_i, \hat{b}_{i+1}, \ldots, b_n\} \) form a basis for the lattice.
Existing Minkowski reduction algorithms

- Lagrange, 1773, dimension two
- Semaev, 2001, dimension three
- Nguyen and Stehleé, 2009, dimension four
- Afflerbach and Grothe, 1985, up to dimension seven
- Helfrish, 1985, theoretical value, very expensive
Existing Minkowski reduction algorithms

- Lagrange, 1773, dimension two
- Semaev, 2001, dimension three
- Nguyen and Stehleé, 2009, dimension four
- Afflerbach and Grothe, 1985, up to dimension seven
- Helfrish, 1985, theoretical value, very expensive

Zhang, Q, Wei, 2011
Problem

For $\rho = 1, 2, ..., n$, find $b_\rho$: a shortest vector such that $\{b_1, ..., b_\rho\}$ can be extended to a basis for the lattice.
For $p = 1, 2, \ldots, n$, find $b_p$: a shortest vector such that 
\{b_1, \ldots, b_p\} can be extended to a basis for the lattice.

Algorithm:

\begin{verbatim}
for $p = 1 \ldots n$
    find a shortest $v = Bz$ such that
        \{b_1, \ldots, b_{p-1}, v\} is expandable to a basis;
    set $b_p = v$ and expand \{b_1, \ldots, b_p\} to a basis;
end
\end{verbatim}
A proposition:
Let $B = [b_1, \ldots, b_n]$ be a generator matrix for a lattice $L$ and a lattice vector $v = Bz$, then $\{b_1, \ldots, b_{p-1}, v\}$ is expandable to a basis for $L$ if and only if $\gcd(z_p, \ldots, z_n) = \pm 1$. 
Minkowski reduction algorithm

A proposition:
Let $B = [b_1, \ldots, b_n]$ be a generator matrix for a lattice $L$ and a lattice vector $v = Bz$, then \{$b_1, \ldots, b_{p-1}, v$\} is expandable to a basis for $L$ if and only if $\gcd(z_p, \ldots, z_n) = \pm 1$.

Constrained minimization problem:

$$\min_z \|Bz\|_2 \quad \text{subject to} \quad \gcd(z_p, \ldots, z_n) = \pm 1$$
A proposition:
Let $B = [\mathbf{b}_1, ..., \mathbf{b}_n]$ be a generator matrix for a lattice $L$ and a lattice vector $\mathbf{v} = Bz$, then $\{\mathbf{b}_1, ..., \mathbf{b}_{p-1}, \mathbf{v}\}$ is expandable to a basis for $L$ if and only if $\gcd(z_p, ..., z_n) = \pm 1$.

Constrained minimization problem:

$$\min_{z} \|Bz\|_2 \quad \text{subject to} \quad \gcd(z_p, ..., z_n) = \pm 1$$

Modified sphere decoding:
While searching for short lattice vectors, enforce the condition $\gcd(z_p, ..., z_n) = \pm 1$. 
Measuring orthogonality

Lattice reduction is to transform a lattice basis into another that becomes “more orthogonal”.

Measuring orthogonality

Lattice reduction is to transform a lattice basis into another that becomes “more orthogonal”.

How do we measure the degree of orthogonality of a basis?
Measuring orthogonality

Lattice reduction is to transform a lattice basis into another that becomes “more orthogonal”.

How do we measure the degree of orthogonality of a basis?

Usual choice: condition number of matrix.
Lattice reduction is to transform a lattice basis into another that becomes “more orthogonal”.

How do we measure the degree of orthogonality of a basis?

Usual choice: *condition number* of matrix.

Consider the matrix
\[
\begin{bmatrix}
1 & 0 \\
0 & 10^k
\end{bmatrix}
\]
Lattice reduction is to transform a lattice basis into another that becomes “more orthogonal”.

How do we measure the degree of orthogonality of a basis?

Usual choice: \textit{condition number} of matrix.

Consider the matrix \[
\begin{bmatrix}
1 & 0 \\
0 & 10^k
\end{bmatrix}.
\]

Its condition number is $10^k$, but the columns are orthogonal.
Lattice reduction is to transform a lattice basis into another that becomes “more orthogonal”.

How do we measure the degree of orthogonality of a basis?

Usual choice: *condition number* of matrix.

Consider the matrix
\[
\begin{bmatrix}
1 & 0 \\
0 & 10^k
\end{bmatrix}.
\]

Its condition number is $10^k$, but the columns are orthogonal.

*Condition #* ignores intermediate singular values of $n \times n$ matrix.
An interpretation

\[
\begin{bmatrix}
  r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} \\
  0 & r_{2,2} & r_{2,3} & r_{2,4} \\
  0 & 0 & r_{3,3} & r_{3,4} \\
  0 & 0 & 0 & r_{4,4}
\end{bmatrix}
\]

\[
\sin \theta_i = \frac{|r_{i,i}|}{\|r_{:.i}\|_2}
\]
In particular, the geometric mean $\sigma$:

$$\sigma^n = \prod_{i=1}^{n} \sin \theta_i = \prod_{i=1}^{n} \frac{|r_{i,i}|}{\|r_{i,i}\|_2} = \frac{d(L)}{\prod_{i=1}^{n} \|b_i\|_2}$$

Hadamard’s inequality

$$\det(B) \leq \prod_{i=1}^{n} \|b_i\|_2$$

The equality holds if and only if $b_i$ are orthogonal.
Also called Hadamard ratio or orthogonality defect.
Note that $0 \leq \sigma \leq 1$, $\sigma = 1$ for any diagonal matrix, and $\sigma = 0$ for any singular matrix.
Note that $0 \leq \sigma \leq 1$, $\sigma = 1$ for any diagonal matrix, and $\sigma = 0$ for any singular matrix.

Since $V_n = \prod_{i=1}^{n} |r_{i,i}| = d(L)$ is a constant for a given $L$, we can improve $\sigma$ by reducing $\|b_i\|_2$. 
Note that $0 \leq \sigma \leq 1$, $\sigma = 1$ for any diagonal matrix, and $\sigma = 0$ for any singular matrix.

Since $V_n = \prod_{i=1}^{n} |r_{i,i}| = d(L)$ is a constant for a given $L$, we can improve $\sigma$ by reducing $\|b_i\|_2$.

Possible measurements other than the geometric mean?
Thank you!
Thank you!

Questions?