Parallel Gauss Sieve Algorithm : Solving the Ideal Lattice Challenge of 128 dimensions

Tsukasa Ishiguro[1] Shinsaku Kiyomoto[1]
Yutaka Miyake[1] Tsuyoshi Takagi[2]

KDDI R&D Laboratories Inc.[1]

Institute of Mathematics for Industry, Kyushu University[2]

2014/3/28

Background

- Some contests from TU Darmstadt
  - SVP Challenge, Ideal Lattice Challenge, Lattice Challenge

HALL OF FAME

Position	Dimension	Index	Seed	Euclidean norm	Contestant	Solution	Using Ideal Structure	Subm. Date
1	128	256	0	2959	Tsukasa Ishiguro, Shinsaku Kiyomoto, Yutaka Miyake, Tsuyoshi Takagi	vec	yes	2013-04-11
2	108	324	0	2669	Tsukasa Ishiguro, Shinsaku Kiyomoto, Yutaka Miyake, Tsuyoshi Takagi	vec	yes	2013-03-8
3	102	103	0	2670	Usatyuk Vasiliy	vec	no	2013-07-17
4	100	202	0	2660	Po-Chun Kuo, Po-Hsiang Hao	vec	no	2013-02-21
5	96	288	0	2493	Tsukasa Ishiguro, Shinsaku Kiyomoto, Yutaka Miyake, Tsuyoshi Takagi	vec	yes	2013-02-20
6	92	188	0	2534	Po-Chun Kuo, Po-Hsiang Hao	vec	no	2013-02-10
7	88	89	0	2482	Tsukasa Ishiguro, Shinsaku Kiyomoto, Yutaka Miyake, Tsuyoshi Takagi	vec	no	2013-02-8
8	82	83	0	2385	Usatyuk Vasiliy	vec	no	2013-02-8
9	80	220	0	2228	Tsukasa Ishiguro, Shinsaku Kiyomoto, Yutaka Miyake, Tsuyoshi Takagi	vec	no	2013-02-8
10	66	67	0	2191	T. Plantard and M. Schneider	vec	no	2012-[illegible]
11	64	160	0	2057	Bal Khadka, Spyros Magliveras	vec	yes	2013-

Background

- Some contests from TU Darmstadt
  - SVP Challenge, Ideal Lattice Challenge, Lattice Challenge

IDEAL LATTICE CHALLENGE

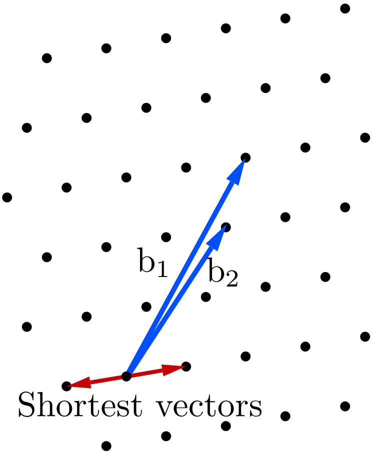
HALL OF FAME

Position	Dimension	Index	Seed	Euclidean norm	Contestant	Solution	Using Ideal Structure	Subm. Date
1	128	256	0	2959	Tsukasa Ishiguro, Shinsaku Kiyomoto, Yutaka Miyake, Tsuyoshi Takagi	vec	yes	2013-04-11
2	108	324	0	2669	Tsukasa Ishiguro, Shinsaku Kiyomoto, Yutaka Miyake, Tsuyoshi Takagi	vec	yes	2013-03-8
3	102	103	0	2670	Usatyuk Vasiliy	vec	no	2013-07-17
4	100	202	0	2660	Po-Chun Kuo, Po-Hsiang Hao	vec	no	2013-02-21

Our contributions

- A parallel version of an algorithm for solving SVP
- Improvements using ideal structures
- Solving the 128 dimensional SVP over ideal lattice

n dimensional lattice and SVP



- Lattice basis

$$\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{Z}^{n\times n}, \quad \mathbf{b}_i \in \mathbb{Z}^n$$

- Lattice

$$\mathcal{L}(\mathbf{B}) = \left\{ \sum_{1\le i\le n} \alpha_i \mathbf{b}_i, \alpha_i \in \mathbb{Z} \right\}$$

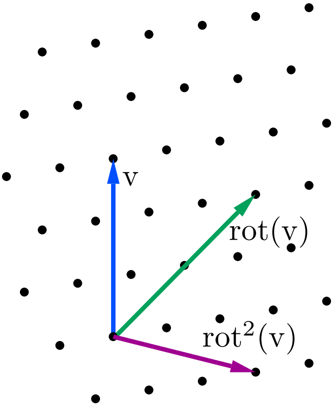
- (Euclidean) norm of $\mathbf{v} = (v_1, .., v_n)$

$$\|\mathbf{v}\| = \sqrt{\sum_{1\le i\le n} v_i^2}$$

Definition (Shortest Vector Problem(SVP))

Given a lattice $\mathcal{L}(\mathbf{B})$, find a shortest non-zero vector in $\mathcal{L}(\mathbf{B})$.

n dimensional ideal lattice



- Polynomial representation

$$\mathbf{v} = (v_1, \ldots, v_n) \in \mathcal{L}(\mathbf{B})$$
$$\Leftrightarrow \mathbf{v}(x) = \sum_{1 \le i \le n} v_i x^{i-1} \in \mathbb{Z}[x]$$

- Vector rotation

$$\mathbf{rot}(\mathbf{v}) = x\mathbf{v}(x) \bmod g(x)$$

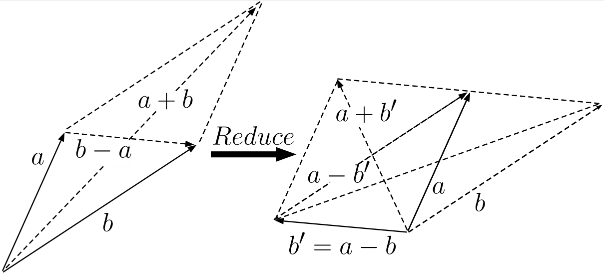
$g(x)$: monic, $\deg(g(x)) = n$

- If $\mathbf{rot}(\mathbf{v}) \in \mathcal{L}(\mathbf{B})$ for all $\mathbf{v} \in \mathcal{L}(\mathbf{B})$, then the $\mathcal{L}(\mathbf{B})$ is called *ideal lattice*

Gauss-reduced

Definition (Gauss-reduced)

If two different vectors $\mathbf{a}, \mathbf{b} \in \mathcal{L}(\mathbf{B})$ *satisfy* $||\mathbf{a} \pm \mathbf{b}|| \geq \max(||\mathbf{a}||, ||\mathbf{b}||)$, *then* $\mathbf{a}, \mathbf{b}$ *are called Gauss-reduced.*

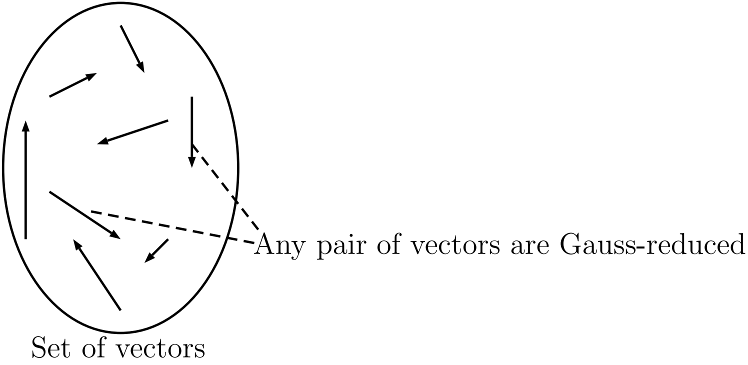


$\mathbf{a}, \mathbf{b}$ are not Gauss-reduced. $\mathbf{a}, \mathbf{b}'$ are Gauss-reduced.
We say that $\mathbf{b}$ (or $\mathbf{b}'$) was reduced by $\mathbf{a}$.

Pairwise-reduced

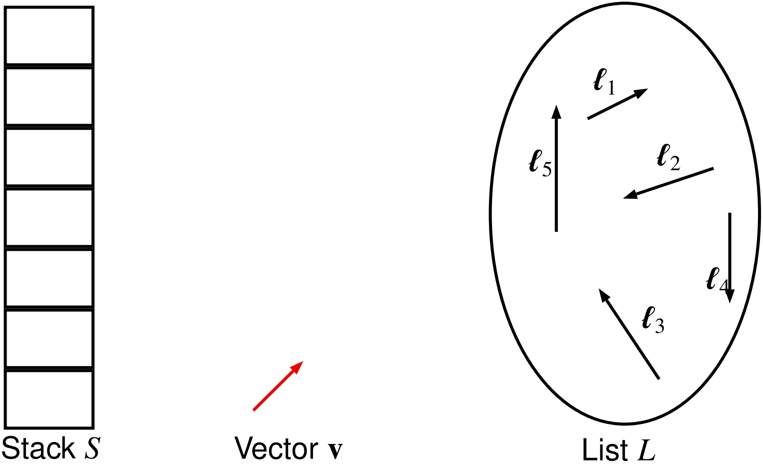
Definition (Pairwise-reduced)

Let A be a set of d vectors in $\mathcal{L}(\mathbf{B})$. If every pair of two vectors $(\mathbf{a}_i, \mathbf{a}_j)$ in A for $i, j = 1, \ldots, d, i \neq j$ is Gauss-reduced, then the A is called pairwise-reduced.



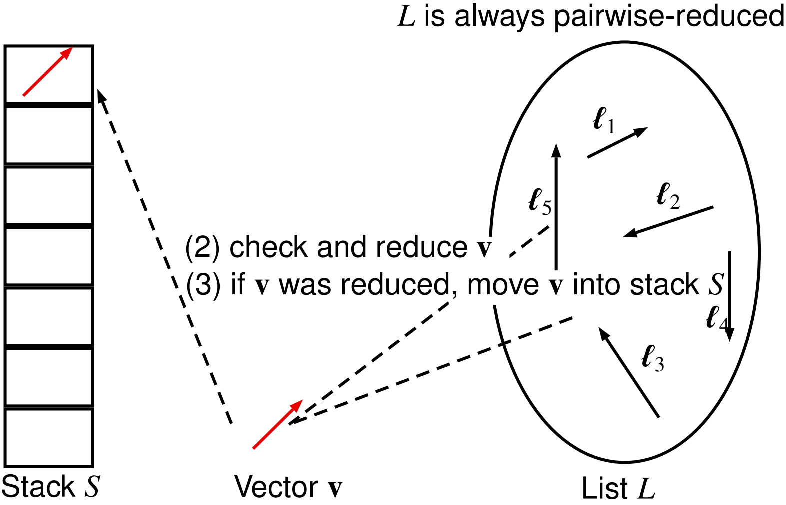
Gauss Sieve Algorithm[Micciancio, 2009]

L is always pairwise-reduced



(1) chosen at random or popped from stack S

Gauss Sieve Algorithm[Micciancio, 2009]

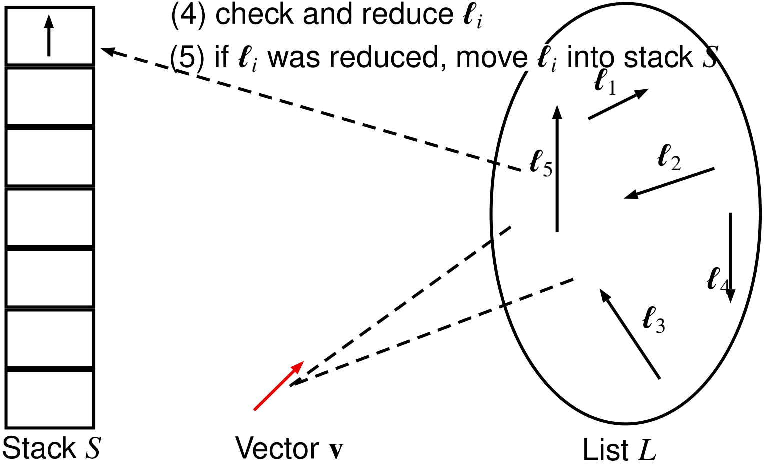


Gauss Sieve Algorithm[Micciancio, 2009]

L is always pairwise-reduced

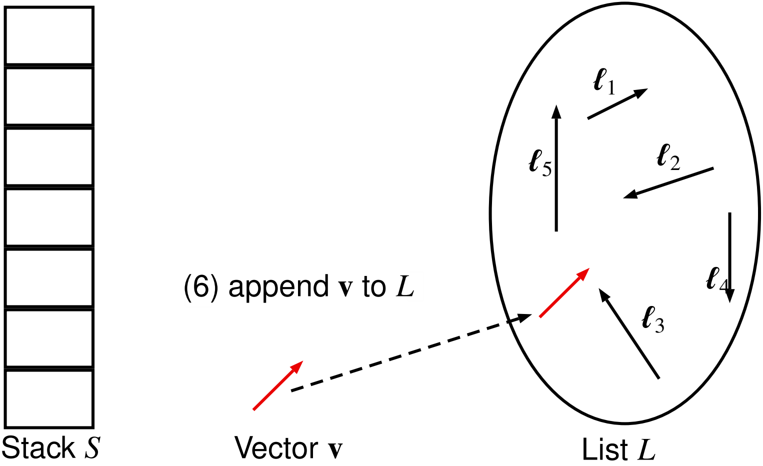
(4) check and reduce $\boldsymbol{\ell}_i$

(5) if $\boldsymbol{\ell}_i$ was reduced, move $\boldsymbol{\ell}_i$ into stack S



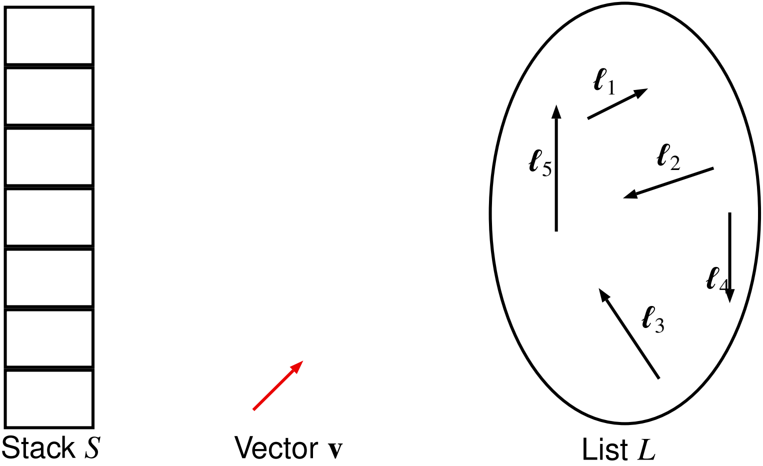
Gauss Sieve Algorithm[Micciancio, 2009]

L is always pairwise-reduced



Gauss Sieve Algorithm[Micciancio, 2009]

L is always pairwise-reduced



Gauss Sieve algorithm constructs a big list L of lattice vectors, which is always pairwise-reduced.
Finally, a shortest vector appeared in the list L.

Parallelization?

- The Gauss Sieve algorithm is not easy to be parallelized
- Milde and Schneider proposed a parallel implementation of the Gauss Sieve[Milde and Schneider, '10]
- Them algorithm does not keep the list L pairwise-reduced
- When they used 10 threads, the list L doubled size of original algorithm

Parallelization?

- The Gauss Sieve algorithm is not easy to be parallelized
- Milde and Schneider proposed a parallel implementation of the Gauss Sieve[Milde and Schneider, '10]
- Them algorithm does not keep the list L pairwise-reduced
- When they used 10 threads, the list L doubled size of original algorithm

Our goal

We propose a fully parallelized Gauss Sieve algorithm.

Parallelization?

- The Gauss Sieve algorithm is not easy to be parallelized
- Milde and Schneider proposed a parallel implementation of the Gauss Sieve[Milde and Schneider, '10]
- Them algorithm does not keep the list L pairwise-reduced
- When they used 10 threads, the list L doubled size of original algorithm

Our goal

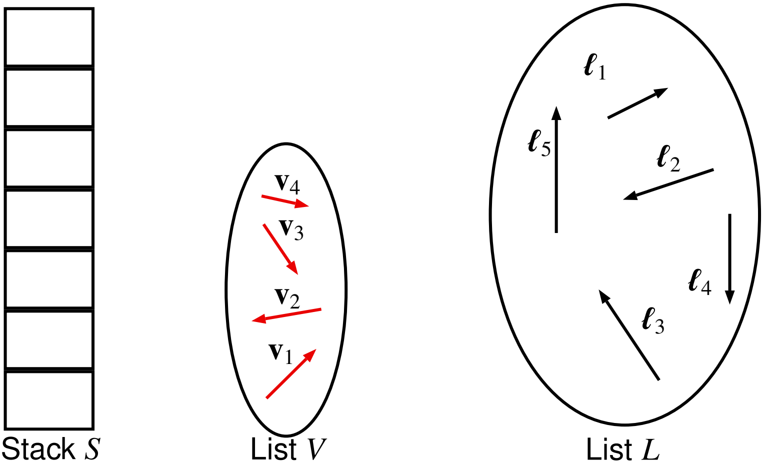
We propose a fully parallelized Gauss Sieve algorithm.

Our strategy

Our algorithm always keeps the list L pairwise-reduced without reference to the number of threads.

Parallel Gauss Sieve Algorithm

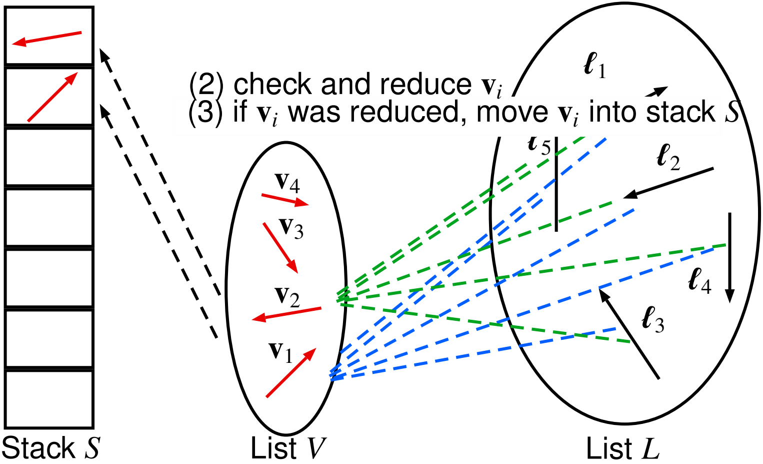
L is always pairwise-reduced



(1) choose at random or popped from stack S

Parallel Gauss Sieve Algorithm

L is always pairwise-reduced

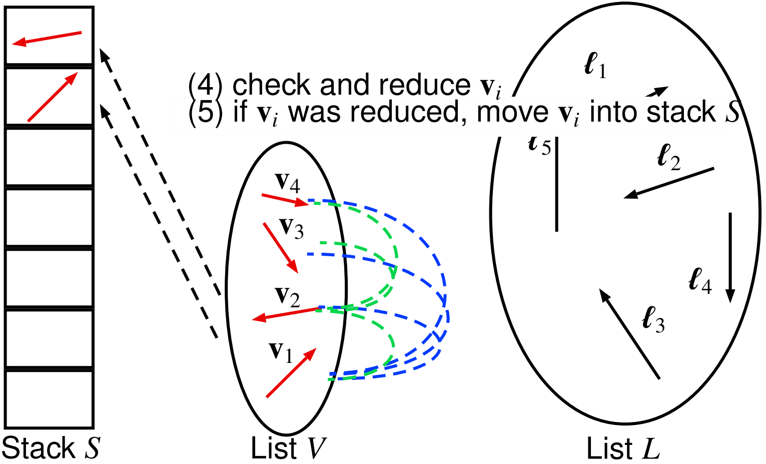


Parallel Gauss Sieve Algorithm

L is always pairwise-reduced

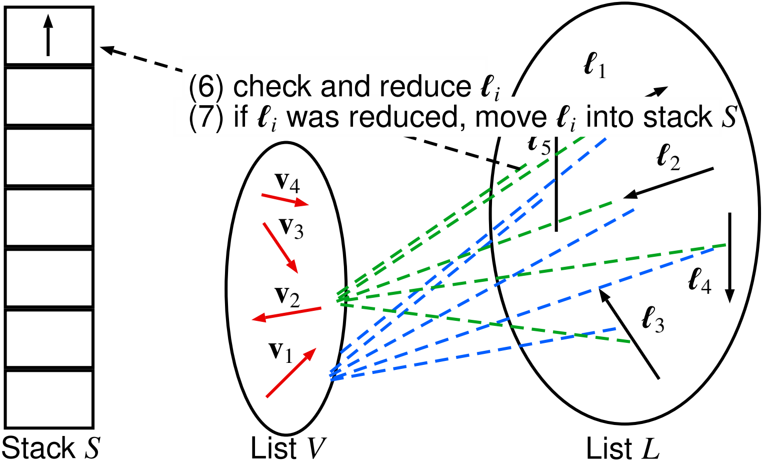
(4) check and reduce $\mathbf{v}_i$

(5) if $\mathbf{v}_i$ was reduced, move $\mathbf{v}_i$ into stack S



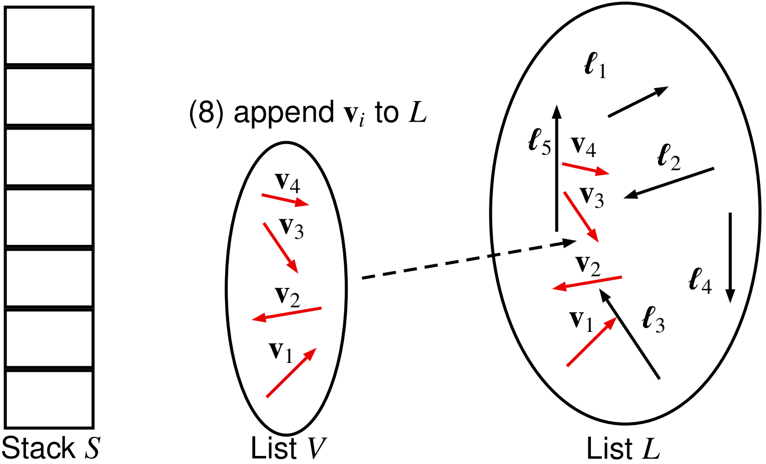
Parallel Gauss Sieve Algorithm

L is always pairwise-reduced

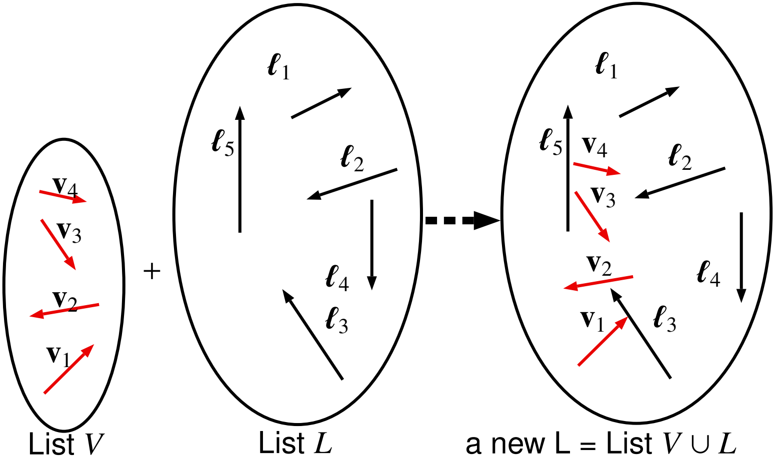


Parallel Gauss Sieve Algorithm

L is always pairwise-reduced



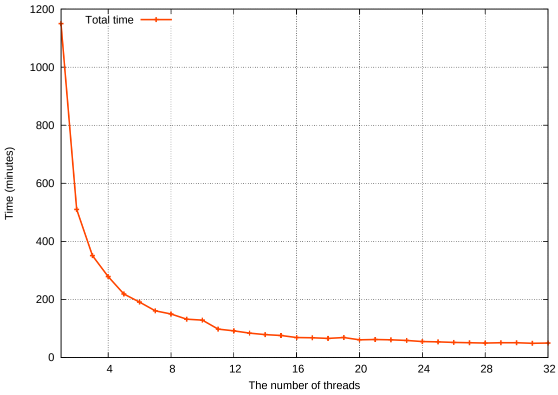
Is a new L pairwise-reduced?



- L and V are pairwise-reduced, respectivery
- All pairs $(\boldsymbol{\ell}_i, \mathbf{v}_j)$ are Gauss-reduced

$\rightarrow V \cup L$ is pairwise-reduced

Solving the 72 dimensional SVP



- This instance has 16 cores
- The running time dereases until 16 threads
- The sizes of the list L are most of the same

List of the improvements of Gauss Sieve

- Generic improvements
  - Sampling short vectors
    - Reduction of lengths of sampling vectors

    → about 5 times faster
  - Improvement of implementation
    - Using SIMD operations

    → $n = 80, 96, 128$

    → about 4 times fasters

- Specific improvements
  - Ideal Gauss Sieve for $n = 2^{\alpha}$ (Anti-cyclic lattice) [Schneider, '11]

    → $n = 128$
  - Trinomial lattice for $n = 2^s 3^t$
    - Inverse rotation $\mathbf{rot}^{-1}(\mathbf{v}) = x^{-1}\mathbf{v}(x) \bmod \boldsymbol{g}(x)$
    - Updating to short vectors

    → $n = 96$

  → more than 25 times faster

Experiment environment

- Amazon EC2 cc1.8xlarge instance
- OS: Ubuntu12.10
- Intel Xeon E5-2670(2.6Ghz), total 16 cores
- gsieve library [Voulgaris]
- compiler: g++4.1.2, OpenMP, OpenMPI

Improvement of implementation

- Our assumptions
  - All absolute values of norms of vectors are less than 2^{16}
  - Calculating time of inner product is most expensive
- We optimized inner product by using SIMD operations
  - 8-parallelization of 16-bit addition and multiplication (SSE4.2)

$\rightarrow$ about 4 times faster

Solving the Challenges

- SVP Challenge

dim n	CPU hours	#instances	#threads	type
80	0.9	1	32	*Random lattice*
96	200	4	128	*Random lattice*

- Ideal Lattice Challenge

dim n	CPU hours	#instances	#threads	type
80	0.9	1	32	*Ideal lattice*
96	8	1	32	*Trinomial lattice*
128	29,994	84	2,688	*Anti-cyclic lattice*

- Original gsieve library requires about 1 week for solving a 80 dimensional SVP
- Trinomial lattice : 25 times faster

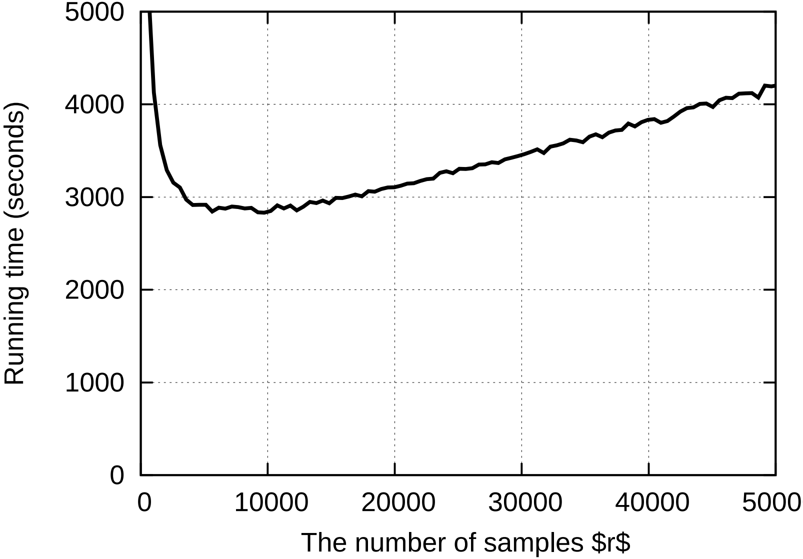
Conclusion

- We proposed a parallel version of the Gauss Sieve algorithm
- We found the new conditions to speed up the Gauss Sieve algorithm
- We solved a 128 dimensional SVP over ideal lattice, which had not been solved before
- The full-version is published in [ePrint 2013/388]

★ Open problems

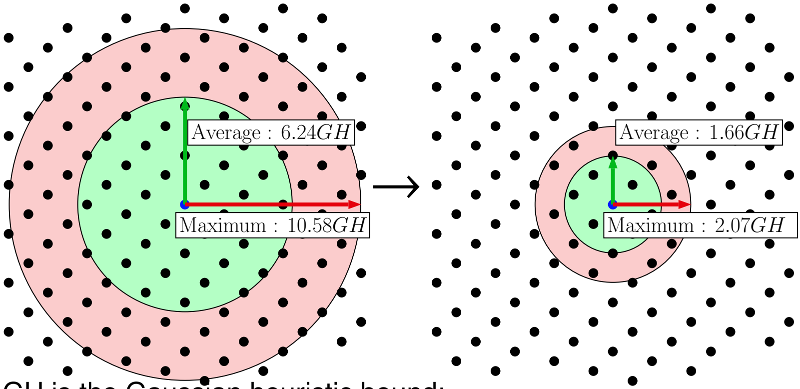
- How is the theoretical complexity of the Gauss Sieve, the Parallel Gauss Sieve, and the Ideal Gauss Sieve?
- Does there exist other conditions or techniques to speed up the Gauss Sieve algorithm?

Solving the 80 dimensional SVP



Sampling short vector

- Optimization of sampling algorithm, namely *SampleD* algorithm in Klein's randomized rounding algorithm.
- We try to adjust the parameter which determines the tradeoff between the length of the norm of sample vectors and the running time of our algorithm.



GH is the Gaussian heuristic bound:

$$GH = (1/\sqrt{\pi})\Gamma(\frac{n}{2}+1)^{\frac{1}{n}} \cdot \det(\mathcal{L}(\mathbf{B}))^{\frac{1}{n}}$$

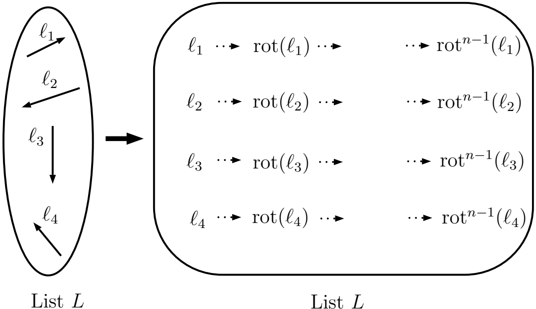
T.Ishiguro, S.Kiyomoto, Y.Miyake, T.Takagi

Applying Ideal Gauss Sieve [Schneider, ePrint 2011/458]

- Anti-cyclic lattice
  - $n = 2^\alpha, \alpha \in \mathbb{N}$
  - Cyclotomic polynomial: $g(x) = x^n + 1$
- Vector rotation

$$\begin{aligned} \mathbf{rot}(\mathbf{v}) &= (-v_n, v_1, \ldots, v_{n-1}) \\ \|\mathbf{rot}^i(\mathbf{v})\| &= \|\mathbf{v}\|, \quad (1 \le i \le n) \end{aligned}$$

- It is easy to generate $(n-1)$ independent vectors $\mathbf{rot}^i(\mathbf{v})$ of same length from one vector $\mathbf{v}$



T.Ishiguro, S.Kiyomoto, Y.Miyake, T.Takagi

Trinomial Lattice (1/2)

- Cyclotomic polynomial : $g(x) = x^n \pm x^{n/2} + 1$
  - (case 1) $n = 2 \cdot 3^m, m > 0$
  - (case 2) $n = 2^s 3^t, s > 1, t > 0$
- Vector rotation
  $$\mathbf{rot}(\mathbf{v}) = (-v_n, v_1, \ldots, v_{\frac{n}{2}-2}, v_{\frac{n}{2}-1} - v_{n-1}, v_{\frac{n}{2}}, \ldots, v_{n-1})$$
- Differential of norm
  $$\|\mathbf{rot}(\mathbf{v})\| - \|\mathbf{v}\| = (v_{n-1})^2 - 2v_{\frac{n}{2}-1}v_{n-1}$$

$\rightarrow$ If $(v_{n-1})^2 - 2v_{\frac{n}{2}-1}v_{n-1} < 0$, norm of a lattice vector decreases.

Trinomial Lattice (2/2)

- Improvement 3-1: Inverse rotation
  - $\mathbf{rot}^{-1}(\mathbf{v}) = x^{-1}\mathbf{v}(x) \bmod \boldsymbol{g}(x)$
    x^{-1}: inverse of x modulo $\boldsymbol{g}(x)$
- Improvement 3-2: Vector update
  - choosing the shortest vector in following vectors

$$\mathbf{rot}(\mathbf{v}), \mathbf{rot}^2(\mathbf{v}), \ldots, \mathbf{rot}^k(\mathbf{v})$$
$$\mathbf{rot}^{-1}(\mathbf{v}), \mathbf{rot}^{-2}(\mathbf{v}), \ldots, \mathbf{rot}^{-k}(\mathbf{v})$$

- Solving the 72 dimensional SVP