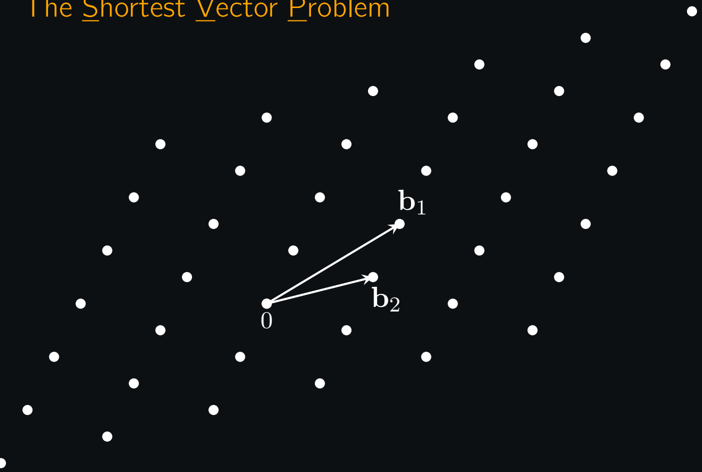
Quantum time-memory trade-offs for lattice sieving algorithms

Elena Kirshanova

based on joint work with Erik Mårtensson, Eamonn W. Postlethwaite, Subhayan Roy Moulik

Dagstuhl, Germany
October 15, 2019

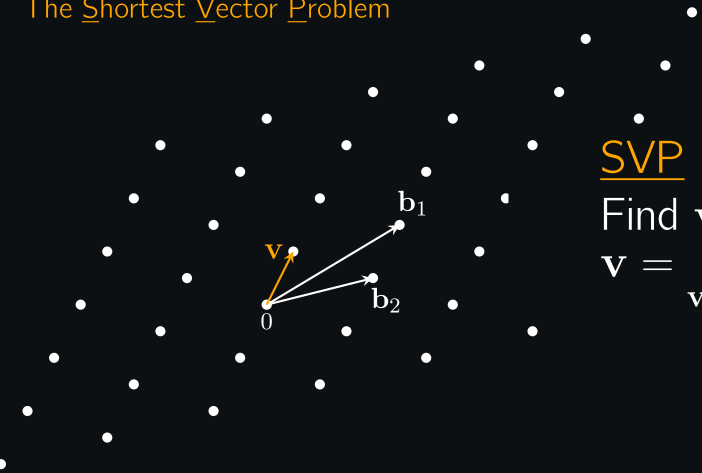
The Shortest Vector Problem



A lattice is a set $\mathcal{L} = \{\sum_{i \le n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z}\}$ for some linearly independent $\mathbf{b}_i \in \mathbb{R}^n$

$\{\mathbf{b}_i\}_i$– a basis of $\mathcal{L}$

The Shortest Vector Problem



SVP

Find $\mathbf{v} \in \mathcal{L}$:

$$\mathbf{v} = \min_{\mathbf{v} \neq 0 \in \mathcal{L}} \|\mathbf{v}\|$$

A lattice is a set $\mathcal{L} = \{\sum_{i \leq n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z}\}$ for some linearly independent $\mathbf{b}_i \in \mathbb{R}^n$

$\{\mathbf{b}_i\}_i -$ a basis of $\mathcal{L}$

Asymptotical ($+o()$ everywhere) Hardness of SVP. $n := \dim \mathcal{L}$

Classical	**Quantum**

Asymptotical ($+o()$ everywhere) Hardness of SVP. $n := \dim \mathcal{L}$

Classical		**Quantum**
	Enumeration	
$\log \text{Time} = \frac{1}{2e} n \log n$		$\log \text{Time} = \frac{1}{4e} n \log n$
$\text{Mem} = \text{poly}(n)$		

Asymptotical ($+o()$ everywhere) Hardness of SVP. $n := \dim \mathcal{L}$

Classical	**Quantum**

Enumeration

Classical	Quantum
$\log \text{Time} = \frac{1}{2e} n \log n$	$\log \text{Time} = \frac{1}{4e} n \log n$
$\text{Mem} = \text{poly}(n)$	

Sieving (provable)

Classical	Quantum
$\log \text{Time} = 2.465n$ or $1.0n$	$\log \text{Time} = 2.25n$ or $1.0n$

Sieving (heuristical)

Classical	Quantum
$\log \text{Time} = 0.396n$	$\log \text{Time} = 0.299n$
$\log \text{Mem} = 0.189n$	$\log \text{Mem} = 0.139n$
Time/Memory trade-offs exist	Time/Memory trade-offs exist

Asymptotical ($+o()$ everywhere) Hardness of SVP. $n := \dim \mathcal{L}$

	Classical	**Quantum**
Enumeration	$\log \text{Time} = \frac{1}{2e} n \log n$ $\text{Mem} = \text{poly}(n)$	$\log \text{Time} = \frac{1}{4e} n \log n$
Sieving (provable)	$\log \text{Time} = 2.465n$ or $1.0n$	$\log \text{Time} = 2.25n$ or $1.0n$
Sieving (heuristical)	$\log \text{Time} = 0.396n$ $\log \text{Mem} = 0.189n$ Time/Memory trade-offs exist	$\log \text{Time} = 0.299n$ $\log \text{Mem} = 0.139n$ Time/Memory trade-offs exist
Sieving + filtering techniques	$\log \text{Time} = 0.292n$ $\log \text{Mem} = 0.208n$	$\log \text{Time} = 0.265n$ $\log \text{Mem} = 0.265n$

Asymptotical ($+o()$ everywhere) Hardness of SVP. $n := \dim \mathcal{L}$

Classical	**Quantum**

Enumeration

Classical	Quantum
$\log \text{Time} = \frac{1}{2e} n \log n$	$\log \text{Time} = \frac{1}{4e} n \log n$
$\text{Mem} = \text{poly}(n)$	

Sieving (provable)

Classical	Quantum
$\log \text{Time} = 2.465n$ or $1.0n$	$\log \text{Time} = 2.25n$ or $1.0n$

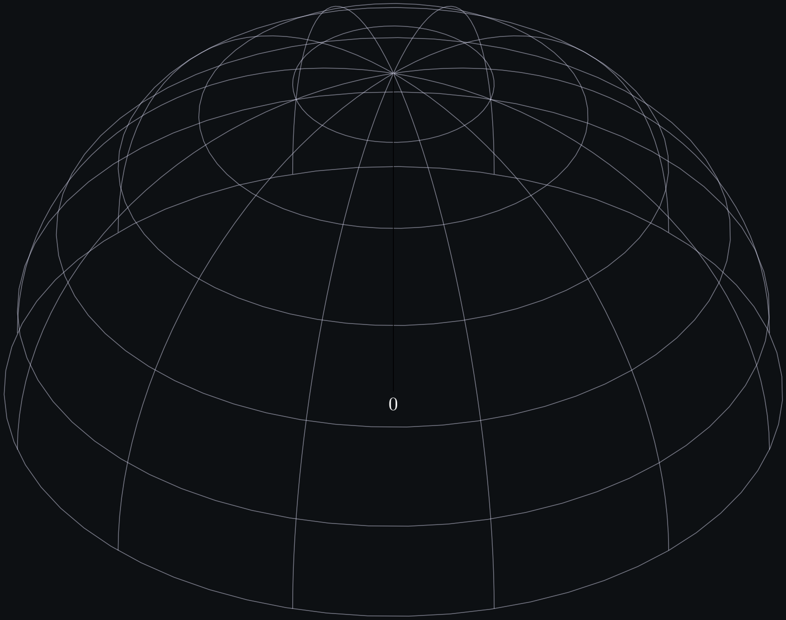
Sieving (heuristical)

Classical	Quantum
$\log \text{Time} = 0.396n$	$\boxed{\log \text{Time} = 0.299n}$
$\log \text{Mem} = 0.189n$	$\boxed{\log \text{Mem} = 0.139n}$
Time/Memory trade-offs exist	Time/Memory trade-offs exist

Sieving + filtering techniques

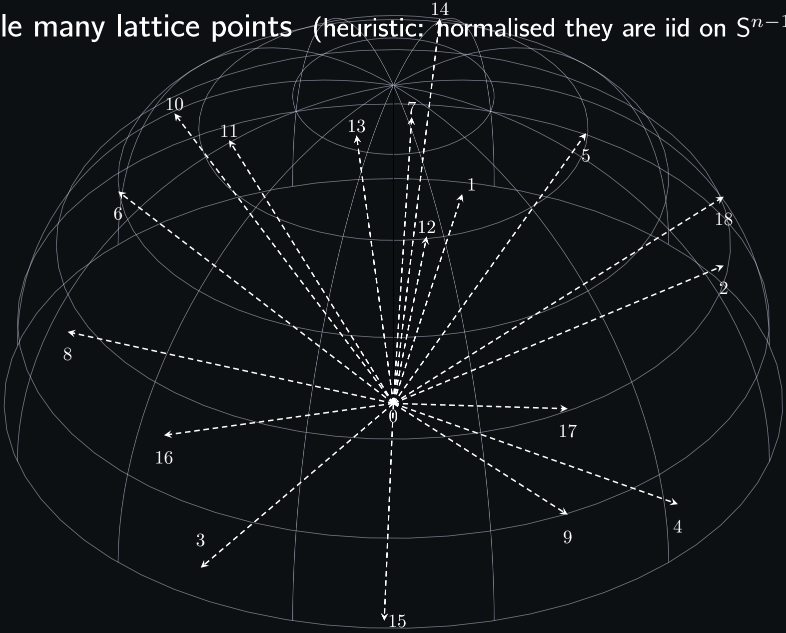
Classical	Quantum
$\log \text{Time} = 0.292n$	$\log \text{Time} = 0.265n$
$\log \text{Mem} = 0.208n$	$\log \text{Mem} = 0.265n$

3-Sieve as 3-List problem for ℓ_2 norm



3-Sieve as 3-List problem for ℓ_2 norm

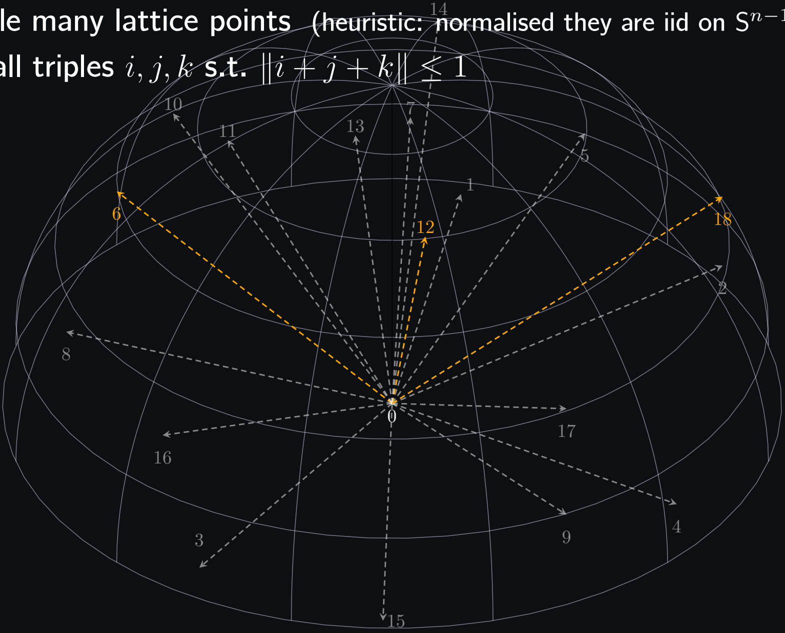
Sample many lattice points (heuristic: normalised they are iid on S^{n-1})



3-Sieve as 3-List problem for ℓ_2 norm

Sample many lattice points (heuristic: normalised they are iid on S^{n-1})

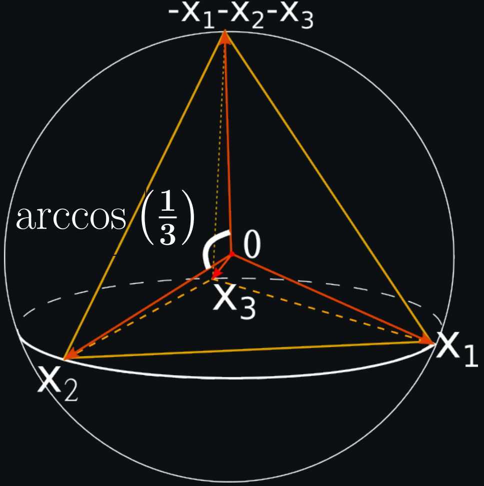
Find all triples i, j, k **s.t.** $\|i + j + k\| \leq 1$



3-Sieve as 3-List problem for ℓ_2 norm

$$\|12 + 6 + 18\| \leq 1$$

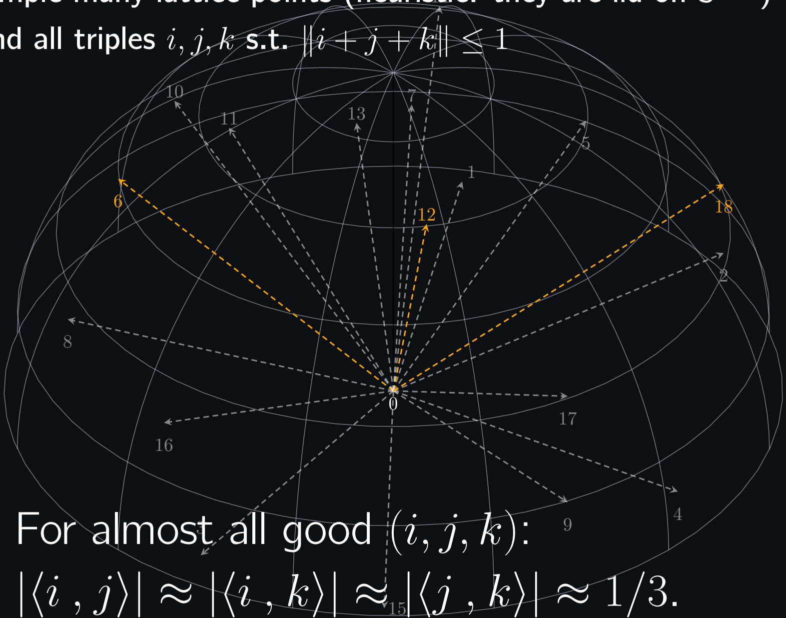
Configuration of good triples



All good triples are concentrated in the shape of 3-simplex

3-Sieve as 3-List problem for ℓ_2 norm

1. Sample many lattice points (heuristic: they are iid on S^{n-1})
2. Find all triples i, j, k s.t. $\|i + j + k\| \leq 1$



For almost all good (i, j, k):

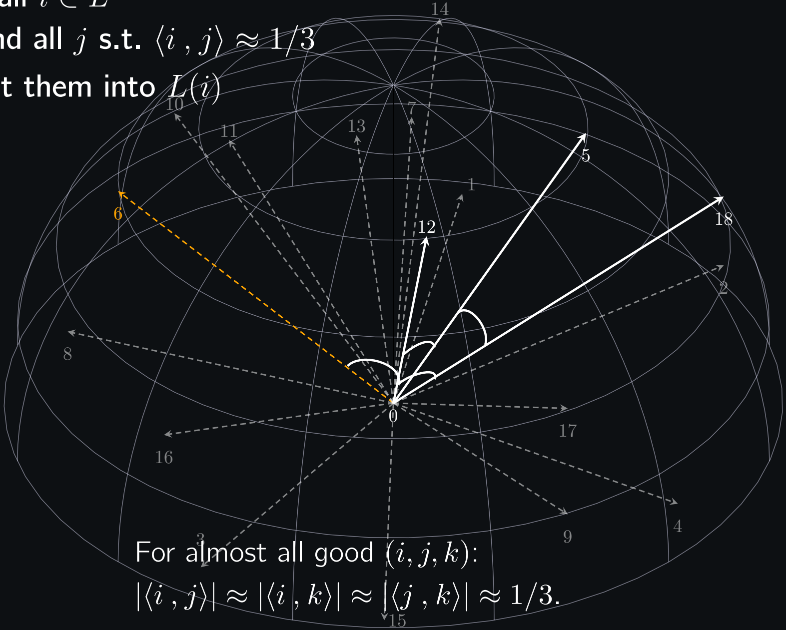
$$|\langle i\ , j\rangle| \approx |\langle i\ , k\rangle| \approx |\langle j\ , k\rangle| \approx 1/3.$$

3-Sieve as 3-List problem for ℓ_2 norm

For all $i \in L$

  Find all j s.t. $\langle i\,,\,j\rangle \approx 1/3$

  Put them into $L(i)$



For almost all good (i, j, k):

$|\langle i\,,\,j\rangle| \approx |\langle i\,,\,k\rangle| \approx |\langle j\,,\,k\rangle| \approx 1/3.$

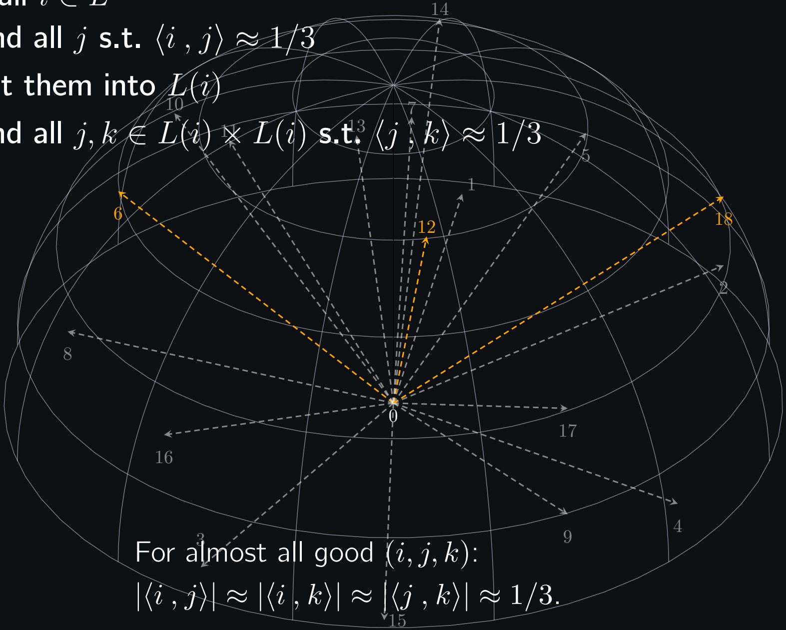
3-Sieve as 3-List problem for ℓ_2 norm

For all $i \in L$

Find all j s.t. $\langle i\,,\,j\rangle \approx 1/3$

Put them into $L(i)$

Find all $j, k \in L(i) \times L(i)$ s.t. $\langle j\,,\,k\rangle \approx 1/3$



For almost all good (i, j, k):

$|\langle i\,,\,j\rangle| \approx |\langle i\,,\,k\rangle| \approx |\langle j\,,\,k\rangle| \approx 1/3.$

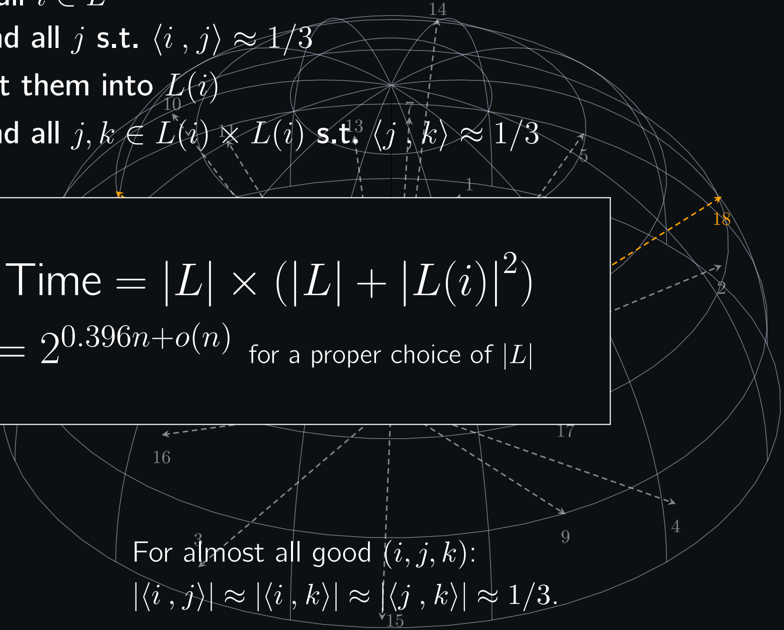
3-Sieve as 3-List problem for ℓ_2 norm

For all $i \in L$

Find all j s.t. $\langle i\,, j\rangle \approx 1/3$

Put them into $L(i)$

Find all $j, k \in L(i) \times L(i)$ s.t. $\langle j\,, k\rangle \approx 1/3$



$$\text{Time} = |L| \times (|L| + |L(i)|^2)$$

$= 2^{0.396n+o(n)}$ for a proper choice of $|L|$

For almost all good (i, j, k):

$|\langle i\,, j\rangle| \approx |\langle i\,, k\rangle| \approx |\langle j\,, k\rangle| \approx 1/3.$

Classical 3-Sieve

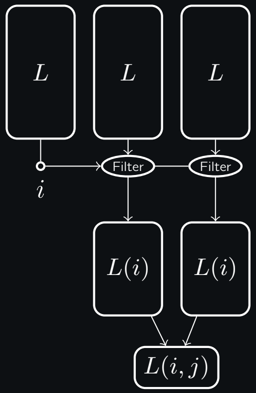
For all $i \in L$:

$\quad L(i) \leftarrow \{j \;:\; \langle i\,,j\rangle \approx 1/3\}$

$\quad$For all $j \in L(i)$

$\quad\quad L(i,j) \leftarrow \{k \in L(i) \;:\; \langle j\,,k\rangle \approx 1/3\}$

$\quad\quad$Output all $(i, j, L(i,j))$



$$\text{Time} = |L| \times (|L| + |L(i)|^2)$$

Classical 3-Sieve

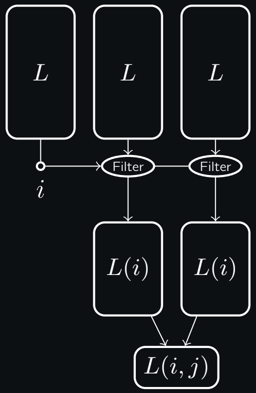
For all $i \in L$:

$L(i) \leftarrow \{j \ : \ \langle i \ , j \rangle \approx 1/3\}$

For all $j \in L(i)$

$L(i,j) \leftarrow \{k \in L(i) \ : \langle j \ , k \rangle \approx 1/3\}$

Output all $(i, j, L(i,j))$



$$\text{Time} = |L| \times (|L| + |L(i)|^2)$$

Set $|L|$ s.t. we output $\approx |L|$ triples

For all $i \in L$:

1. $\frac{1}{|L|} \sum_{j,k \in L \times L} |j\rangle |k\rangle$

For all $i \in L$:

1. $\frac{1}{|L|} \sum_{j,k \in L \times L} |j\rangle\,|k\rangle$

$$\xrightarrow[f(i,j)=1 \Leftrightarrow |\langle i,j \rangle| \approx 1/3]{\text{Grover}}, \xrightarrow[f(i,k)=1 \Leftrightarrow |\langle i,k \rangle| \approx 1/3]{\text{Grover}}$$

2. $\frac{1}{|L(i)|} \sum_{j,k \in L(i) \times L(i)} |j\rangle\,|k\rangle$

For all $i \in L$:

1. $\frac{1}{|L|} \sum_{j,k \in L \times L} |j\rangle |k\rangle$

$$\xrightarrow[f(i,j)=1 \Leftrightarrow |\langle i,j \rangle| \approx 1/3]{\text{Grover}}, \xrightarrow[f(i,k)=1 \Leftrightarrow |\langle i,k \rangle| \approx 1/3]{\text{Grover}}$$

2. $\frac{1}{|L(i)|} \sum_{j,k \in L(i) \times L(i)} |j\rangle |k\rangle \xrightarrow[f(j,k)=1 \Leftrightarrow |\langle j,k \rangle| \approx 1/3]{\text{Grover}}$

3. $\sum_{j,k \text{ are good}} |j\rangle |k\rangle$

For all $i \in L$:

1. $\frac{1}{|L|} \sum_{j,k \in L \times L} |j\rangle |k\rangle$

$$\xrightarrow[f(i,j)=1 \Leftrightarrow |\langle i,j\rangle| \approx 1/3]{\text{Grover}}, \xrightarrow[f(i,k)=1 \Leftrightarrow |\langle i,k\rangle| \approx 1/3]{\text{Grover}}$$

2. $\frac{1}{|L(i)|} \sum_{j,k \in L(i) \times L(i)} |j\rangle |k\rangle \xrightarrow[f(j,k)=1 \Leftrightarrow |\langle j,k\rangle| \approx 1/3]{\text{Grover}}$

3. $\sum_{j,k \text{ are good}} |j\rangle |k\rangle$ Expect $\mathcal{O}(1)$ good (j,k)

4. Measure, output (i,j,k)

Quantum 3-Sieve

For all $i \in L$:

1. $\frac{1}{|L|} \sum_{j,k \in L \times L} |j\rangle |k\rangle$

$$\xrightarrow[f(i,j)=1 \Leftrightarrow |\langle i,j \rangle| \approx 1/3]{\text{Grover}}, \xrightarrow[f(i,k)=1 \Leftrightarrow |\langle i,k \rangle| \approx 1/3]{\text{Grover}}$$

2. $\frac{1}{|L(i)|} \sum_{j,k \in L(i) \times L(i)} |j\rangle |k\rangle \xrightarrow[f(j,k)=1 \Leftrightarrow |\langle j,k \rangle| \approx 1/3]{\text{Grover}}$
3. $\sum_{j,k \text{ are good}} |j\rangle |k\rangle$ Expect $\mathcal{O}(1)$ good (j, k)
4. Measure, output (i, j, k)

Quantum 3-Sieve

For all $i \in L$:

1. $\frac{1}{|L|} \sum_{j,k \in L \times L} |j\rangle |k\rangle$

$$\xrightarrow[f(i,j)=1 \Leftrightarrow |\langle i,j \rangle| \approx 1/3]{\text{Grover}}, \xrightarrow[f(i,k)=1 \Leftrightarrow |\langle i,k \rangle| \approx 1/3]{\text{Grover}}$$

2. $\frac{1}{|L(i)|} \sum_{j,k \in L(i) \times L(i)} |j\rangle |k\rangle \xrightarrow[f(j,k)=1 \Leftrightarrow |\langle j,k \rangle| \approx 1/3]{\text{Grover}}$
3. $\sum_{j,k \text{ are good}} |j\rangle |k\rangle$ Expect $\mathcal{O}(1)$ good (j,k)
4. Measure, output (i,j,k)

$$\text{TimeQ} = |L| \times \left(2\sqrt{\frac{|L|}{|L(i)|}} \cdot |L(i)| \right) = 2^{0.335n}$$

$$\text{TimeC} = |L| \times (|L| + |L(i)|^2) = 2^{0.396n}$$

Quantum k-Sieve

The algorithm generalises to larger $k = \Theta(1)$ and time-optimal inner product leading to

Time = $2^{0.299n+o(n)}$ Memory = $2^{0.139n+o(n)}$

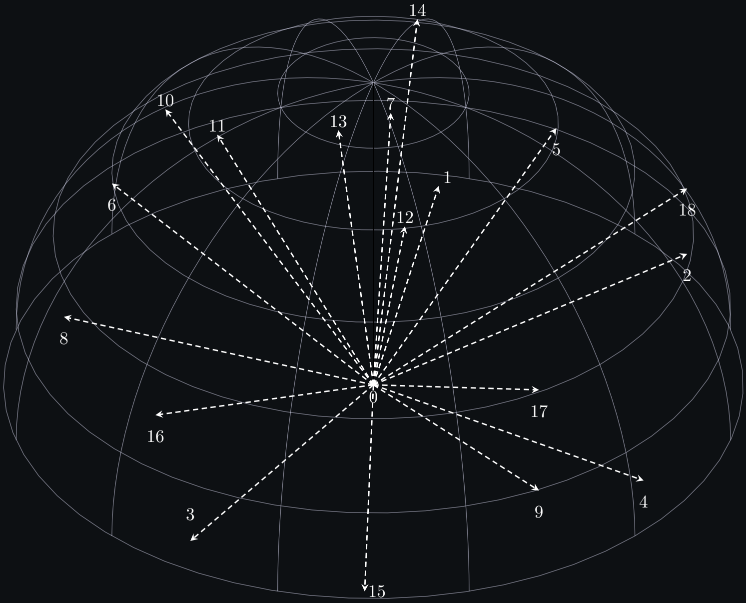
Asymptotically time-optimal algorithm uses hashing techniques for $k = 2$ and achieves (T.Laarhoven)

Time = $2^{0.265n+o(n)}$ Memory = $2^{0.265n+o(n)}$

3-Sieve via triangle finding

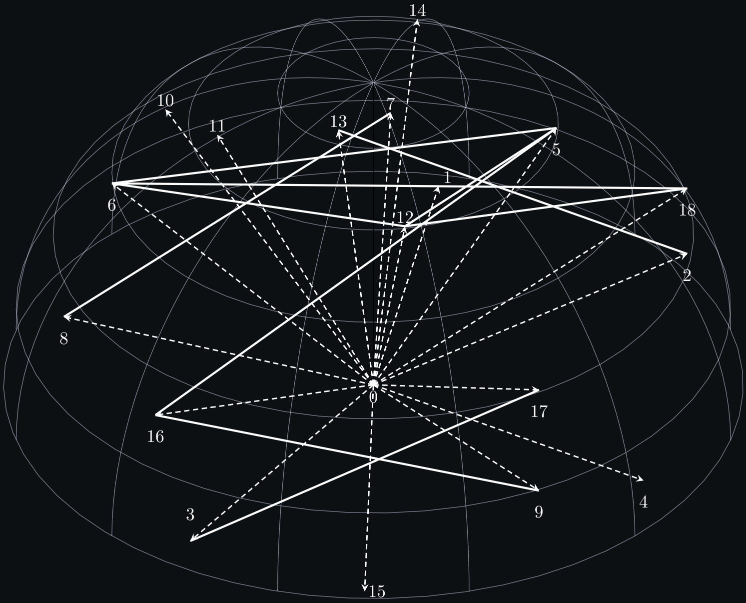
3-Sieve via triangle finding

Connect two points $(i, j) \Leftrightarrow |\langle i\ , j \rangle| \approx 1/3$



3-Sieve via triangle finding

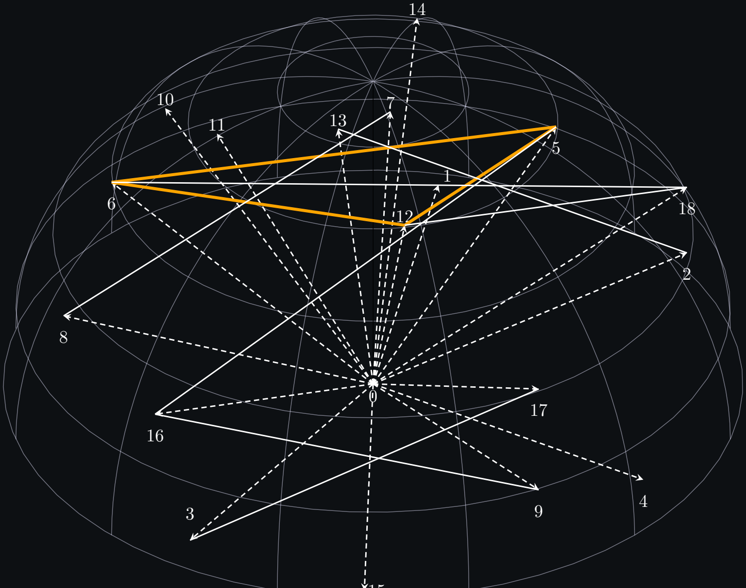
Connect two points $(i, j) \Leftrightarrow |\langle i\ , j\rangle| \approx 1/3$



3-Sieve via triangle finding

Connect two points $(i, j) \Leftrightarrow |\langle i\,,\, j\rangle| \approx 1/3$

Good triples $(i, j, k) \Leftrightarrow$ triangles



Apply quantum triangle (k-clique) finding

$G = \{V, E\}$, V – lattice vectors, $e(v_i, v_j) \in E \Leftrightarrow |\langle v_i\,, v_j\rangle| \approx 1/3$

Run triangle listing on G (it's a sparse graph!)

Apply quantum triangle (k-clique) finding

$G = \{V, E\}$, V – lattice vectors, $e(v_i, v_j) \in E \Leftrightarrow |\langle v_i\,, v_j\rangle| \approx 1/3$

Run triangle listing on G (it's a sparse graph!)

Vast literature on quantum triangle finding but in the **query** model

Apply quantum triangle (k-clique) finding

$G = \{V, E\}$, V – lattice vectors, $e(v_i, v_j) \in E \Leftrightarrow |\langle v_i\,, v_j\rangle| \approx 1/3$

Run triangle listing on G (it's a sparse graph!)

Vast literature on quantum triangle finding but in the **query** model

Adapt the triangle **finding** algorithm of [Buhrman–de Wolf-Dürr–Heiligman–Høyer–Magniez–Santha]:

$$\text{Time (find } \triangle) = \sqrt{|E|} \implies \text{Time (list all } \triangle's) = |V|\sqrt{|E|}$$

Gives the same runtime complexity as the previous algorithm for any $k = \Theta(1)$.

Large quantum memory sieving for $k = 2$

Task: Given a large list L of vectors find all (i, j) s.t.

$$|\langle L[i]\, , L[j] \rangle| \approx 1/2$$

Large quantum memory sieving for $k = 2$

Task: Given a large list L of vectors find all (i, j) s.t.

$$|\langle L[i]\,, L[j] \rangle| \approx 1/2$$

We have $|L|$ quantum processors and a shared quantum memory of size $|L|$

[Beals–Brierley–Gray–Harrow–Kutin–Linden–Shepherd–Stather]:

For a list L and $|L|$ functions

$$f_i(j) = \begin{cases} 1, & |\langle L[i]\,, L[j] \rangle| \approx 1/2 \\ 0, & \text{else} \end{cases}$$

define a solution $\mathsf{s} \in \{1 \ldots |L|\}^{|L|} :\ f_i(\mathsf{s}_i) = 1$ for all i.

Large quantum memory sieving for $k = 2$

Task: Given a large list L of vectors find all (i, j) s.t.

$$|\langle L[i]\,, L[j]\rangle| \approx 1/2$$

We have $|L|$ quantum processors and a shared quantum memory of size $|L|$

[Beals–Brierley–Gray–Harrow–Kutin–Linden–Shepherd–Stather]:

For a list L and $|L|$ functions

$$f_i(j) = \begin{cases} 1, & |\langle L[i]\,, L[j]\rangle| \approx 1/2 \\ 0, & \text{else} \end{cases}$$

define a solution $\mathbf{s} \in \{1 \ldots |L|\}^{|L|} : \; f_i(\mathbf{s}_i) = 1$ for all i.

There exists a quantum algorithm that returns $\mathbf{s}_i$ for each i. It can be implemented using a quantum circuit of width $\widetilde{\mathcal{O}}(L)$ and depth $\widetilde{\mathcal{O}}(\sqrt{L})$.

Large quantum memory sieving for $k = 2$

[Beals–Brierley–Gray–Harrow–Kutin–Linden–Shepherd–Stather]:

For a list L and $|L|$ functions

$$f_i(j) = \begin{cases} 1, & |\langle L[i]\,, L[j]\rangle| \approx 1/2 \\ 0, & \text{else} \end{cases}$$

define a solution $\mathsf{s} \in \{1 \dots |L|\}^{|L|} :\ f_i(\mathsf{s}_i) = 1$ for all i.

Large quantum memory sieving for $k = 2$

[Beals–Brierley–Gray–Harrow–Kutin–Linden–Shepherd–Stather]:

For a list L and $|L|$ functions

$$f_i(j) = \begin{cases} 1, & |\langle L[i]\,, L[j]\rangle| \approx 1/2 \\ 0, & \text{else} \end{cases}$$

define a solution $\mathbf{s} \in \{1 \ldots |L|\}^{|L|} : \ f_i(\mathbf{s}_i) = 1$ for all i.

There exists a quantum algorithm that returns $\mathbf{s}_i$ for each i. It can be implemented using a quantum circuit of width $\widetilde{\mathcal{O}}(L)$ and depth $\widetilde{\mathcal{O}}(\sqrt{L})$.

There exists a quantum circuit that implements 2-Sieve of width $2^{0.2075n+o(n)}$ and depth $2^{0.1037n+o(n)}$.