Technische Universiteit Eindhoven
University of Technology

Quantum Lattice Cryptanalysis
Part 1: Sieving and Saturation

Thijs Laarhoven, Michele Mosca, Joop van de Pol

mail@thijs.com
http://www.thijs.com/

Schloss Dagstuhl, Wadern, Germany
(September 11, 2013)

Technische Universiteit
Eindhoven
University of Technology

Solving the Shortest Vector Problem in Lattices Faster Using Quantum Search

Thijs Laarhoven, Michele Mosca, Joop van de Pol

mail@thijs.com
http://www.thijs.com/

Schloss Dagstuhl, Wadern, Germany
(September 11, 2013)

Outline

Introduction
- Lattices
- Quantum Search
- Applications

SVP Algorithms
- Sieving
- Saturation

Overview

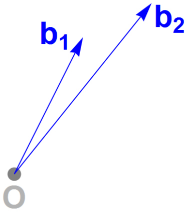
Conclusion

Lattices

What is a lattice?

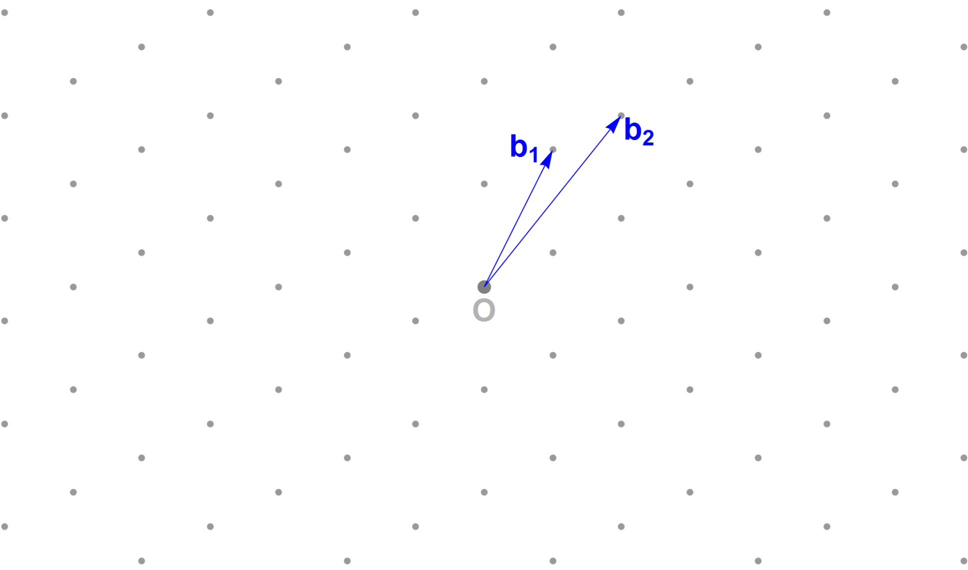
Lattices

What is a lattice?



Lattices

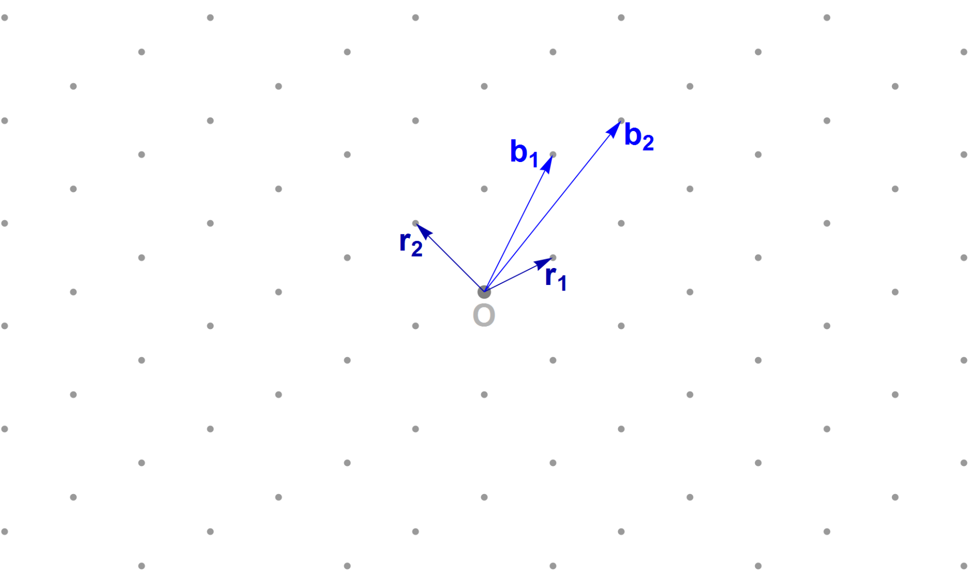
What is a lattice?



TU/e

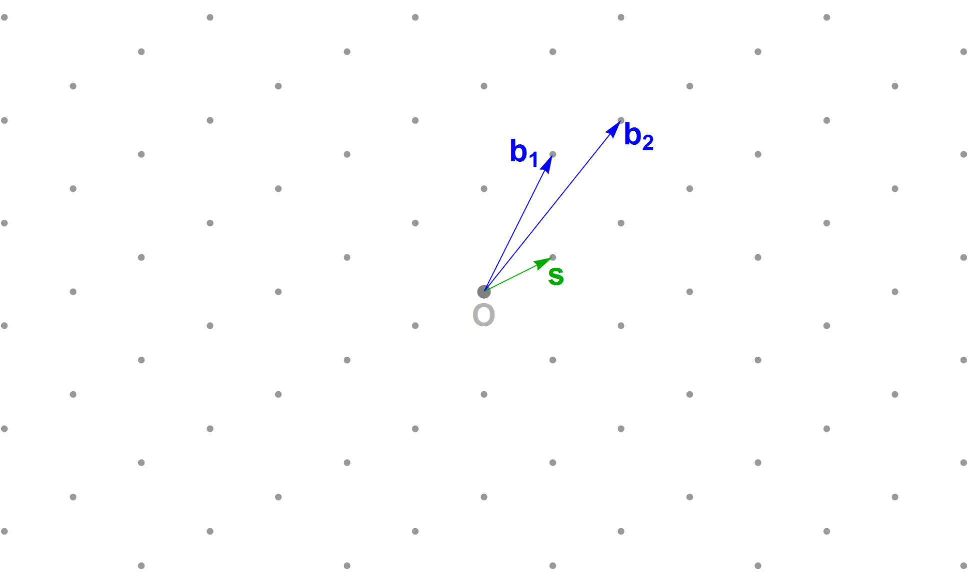
Lattices

Lattice Basis Reduction



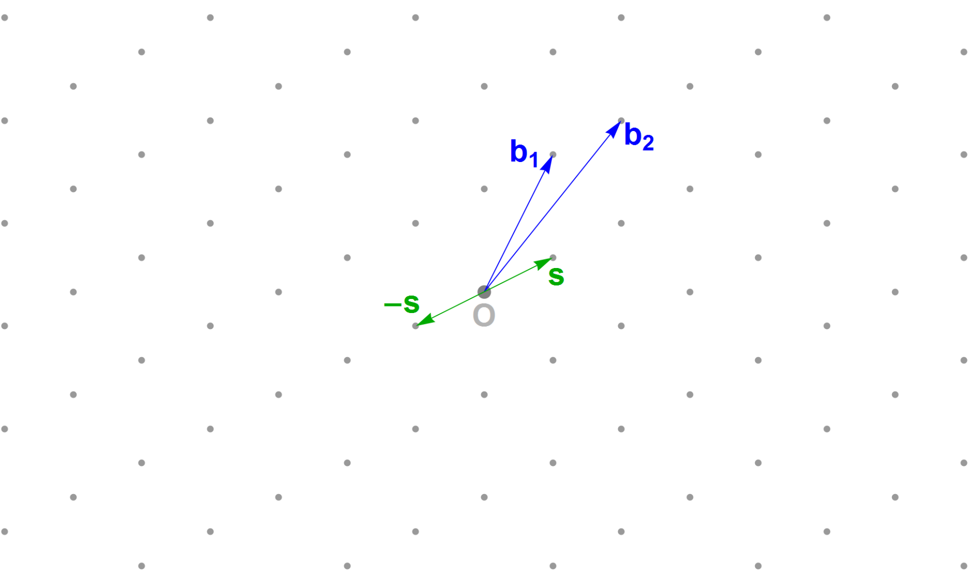
Lattices

Shortest Vector Problem (SVP)



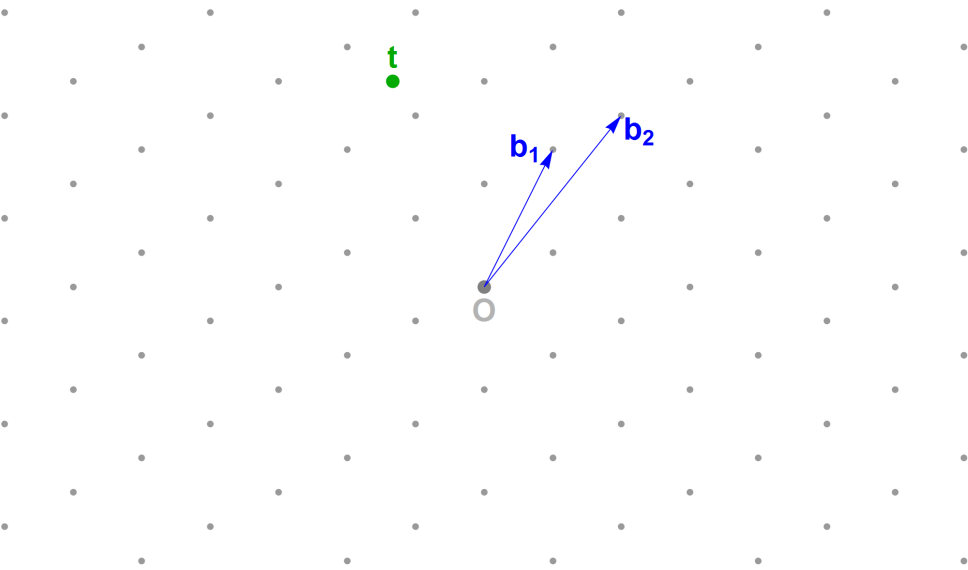
Lattices

Shortest Vector Problem (SVP)



Lattices

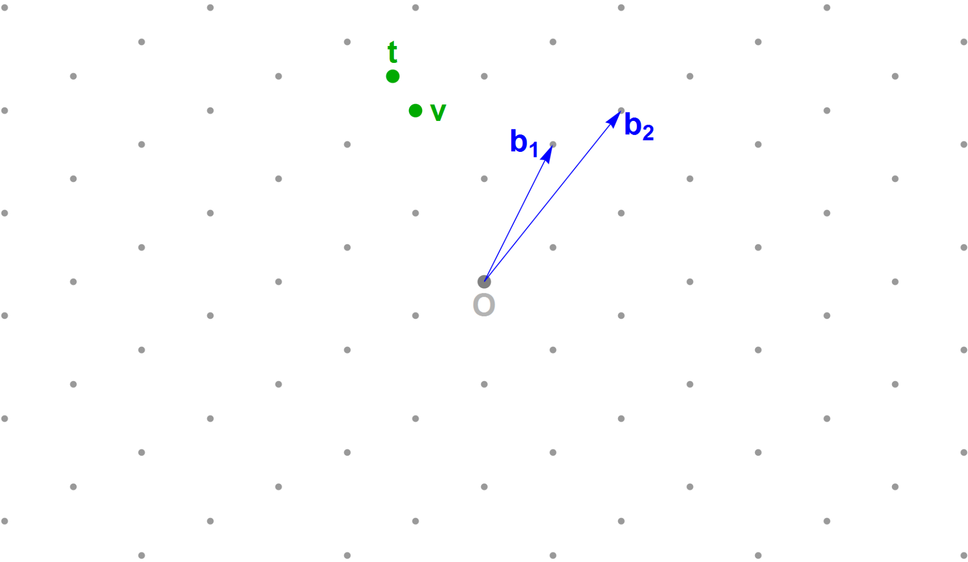
Closest Vector Problem (CVP)



TU/e

Lattices

Closest Vector Problem (CVP)



Quantum Search

Classical form

Problem: Given a list L of size N, and a function $f : L \to \{0, 1\}$ such that there is exactly one element $e \in L$ with $f(e) = 1$. Find this element e.

- Classical search: $\Theta(N)$ time
- Quantum search: $\Theta(\sqrt{N})$ time [Gro96]

Quantum Search

General form

Problem: Given a list L of size N, and a function $f : L \to \{0, 1\}$ such that there are $c = O(1)$ elements $e \in L$ with $f(e) = 1$. Find one such element e.

- Classical search: $\Theta(N/c)$ time
- Quantum search: $\Theta(\sqrt{N/c})$ time [Gro96]

Applications

(Why do we care?)

- “Constructive cryptography”: Lattice-based cryptosystems
  - Based on hard lattice problems (SVP, CVP)
  - NTRU cryptosystem [HPS98]
  - Fully Homomorphic Encryption [Gen09]
  - Candidate for post-quantum cryptography ("survivor")

Applications

(Why do we care?)

- "Constructive cryptography": Lattice-based cryptosystems
  - Based on hard lattice problems (SVP, CVP)
  - NTRU cryptosystem [HPS98]
  - Fully Homomorphic Encryption [Gen09]
  - Candidate for post-quantum cryptography ("survivor")
- "Destructive cryptography": Cryptanalysis
  - Attack knapsack-based cryptosystems [Sha82, LO85]
  - Attack RSA with Coppersmith's method [Cop97]
  - Attack DSA and ECDSA [NS02, NS03]
  - Attack lattice-based cryptosystems [Ngu99, JJ00]

Applications

(Why do we care?)

- "Constructive cryptography": Lattice-based cryptosystems
  - Based on hard lattice problems (SVP, CVP)
  - NTRU cryptosystem [HPS98]
  - Fully Homomorphic Encryption [Gen09]
  - Candidate for post-quantum cryptography ("survivor")
- "Destructive cryptography": Cryptanalysis
  - Attack knapsack-based cryptosystems [Sha82, LO85]
  - Attack RSA with Coppersmith's method [Cop97]
  - Attack DSA and ECDSA [NS02, NS03]
  - Attack lattice-based cryptosystems [Ngu99, JJ00]

How (quantum-)hard are hard lattice problems such as SVP?

Sieving

Studied since 2001 [AKS01, Reg04, NV08, . . ., ZPH13]

1. Generate a long list V of random lattice vectors

Sieving

Studied since 2001 [AKS01, Reg04, NV08, . . ., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R

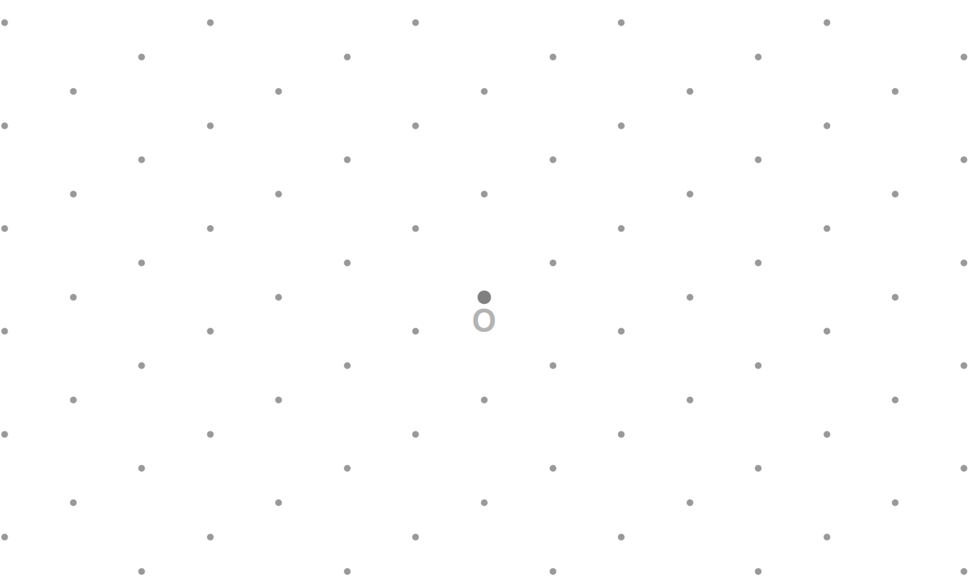
Sieving

Studied since 2001 [AKS01, Reg04, NV08, . . ., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R
3. Set $V = R$ and repeat until V contains a shortest vector

Sieving

1. Generate random lattice vectors



TU/e

Sieving

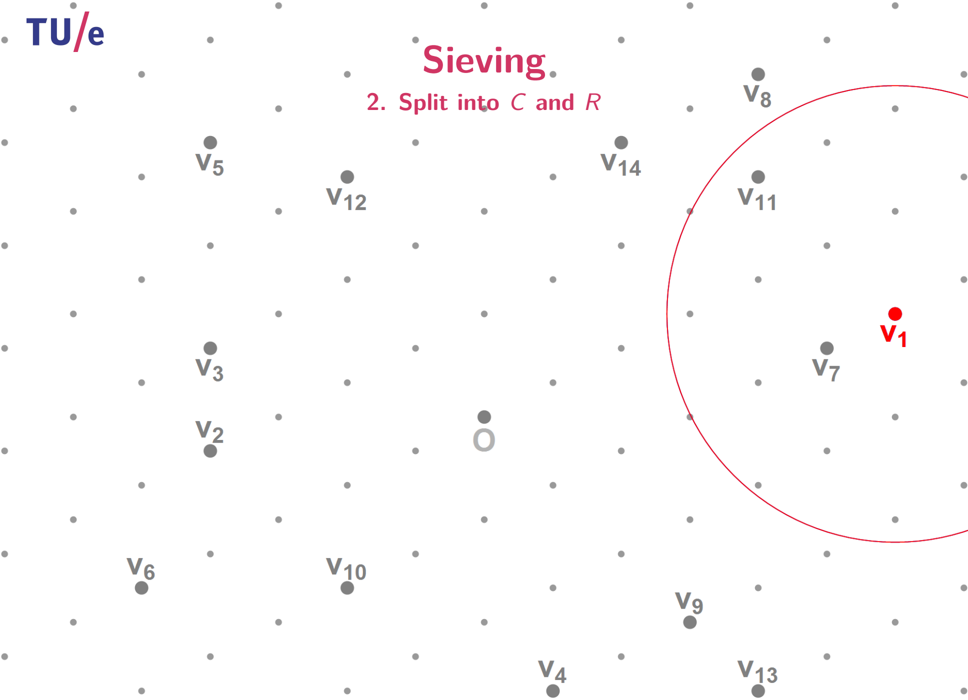
1. Generate random lattice vectors

Sieving

2. Split into C and R

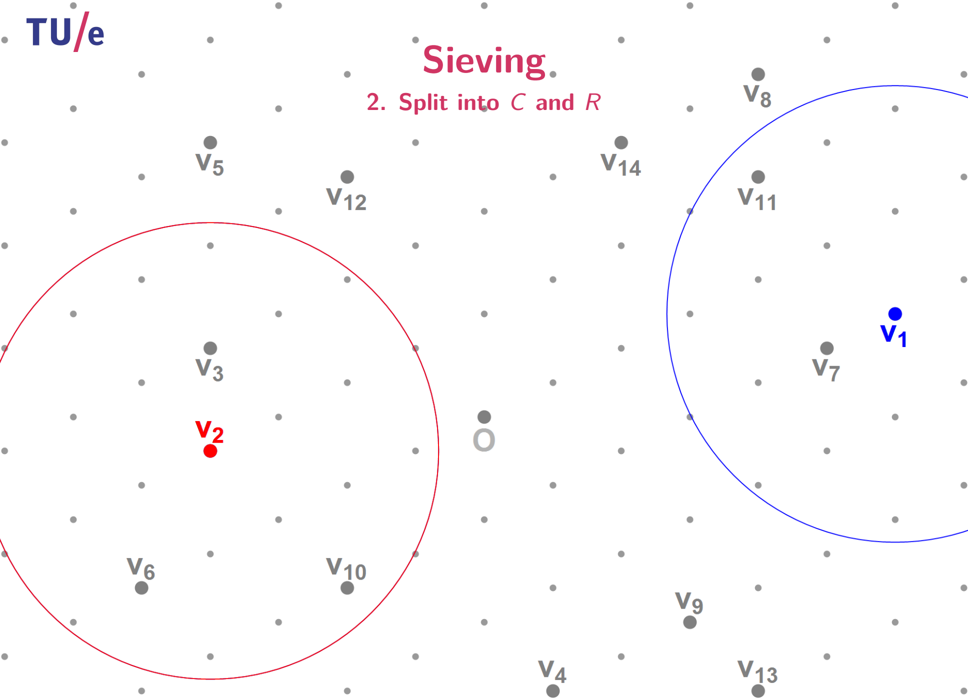
Sieving

2. Split into C and R



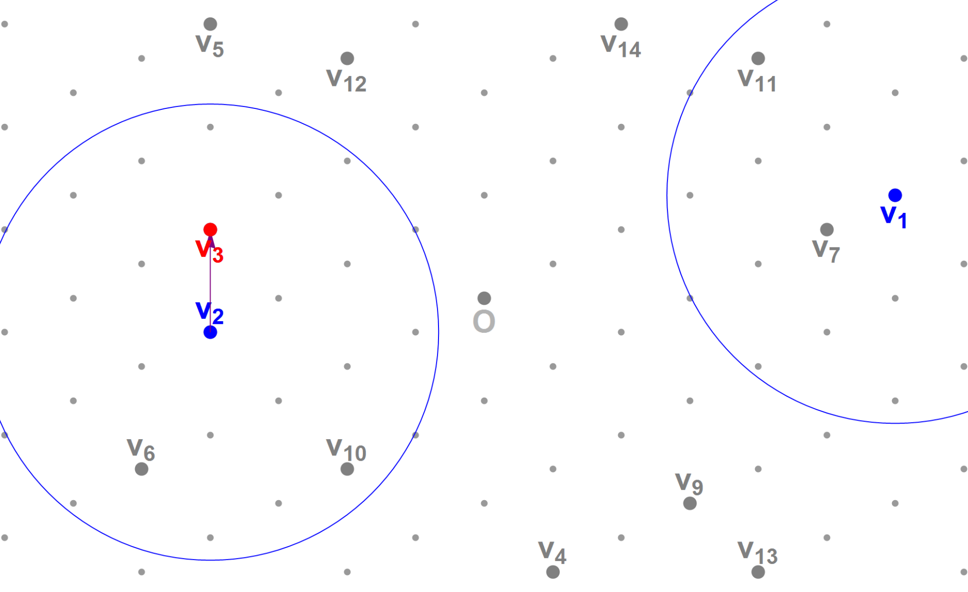
Sieving

2. Split into C and R



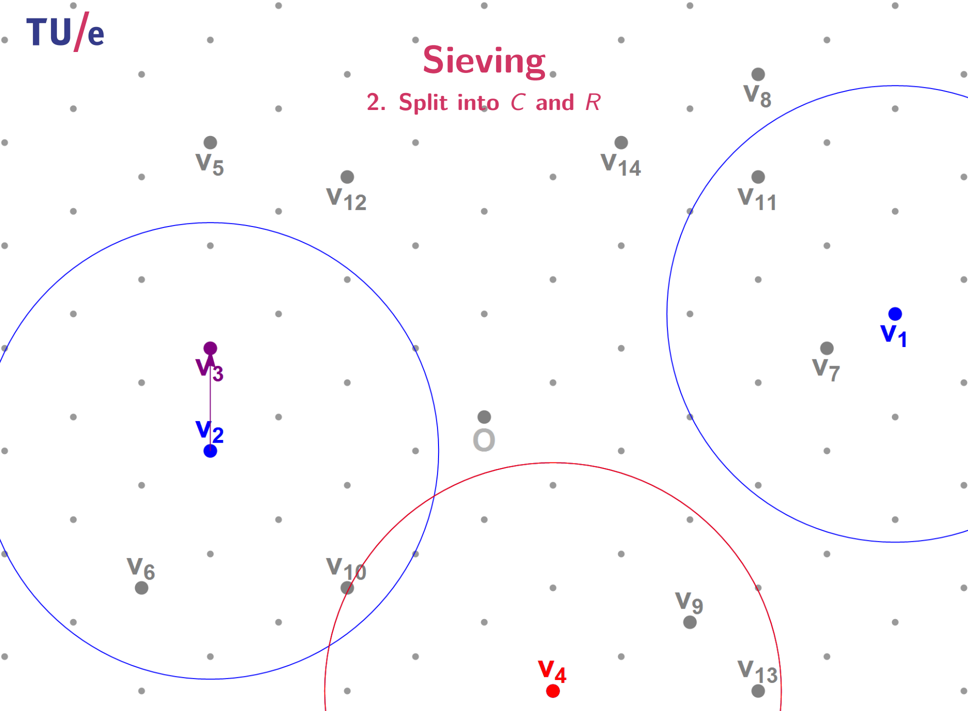
Sieving

2. Split into C and R



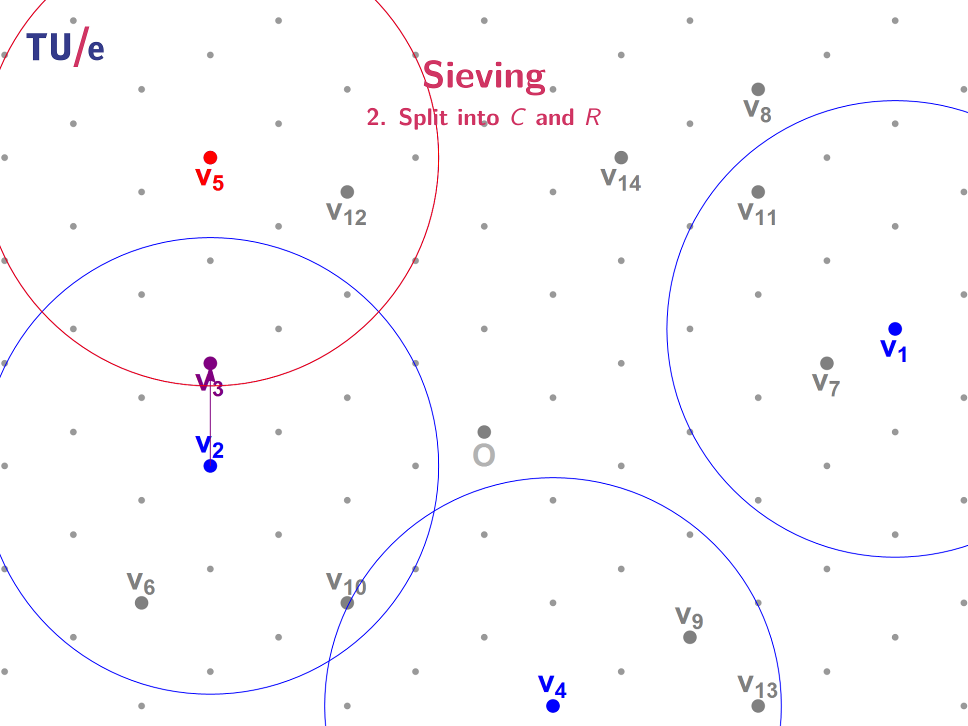
Sieving

2. Split into C and R



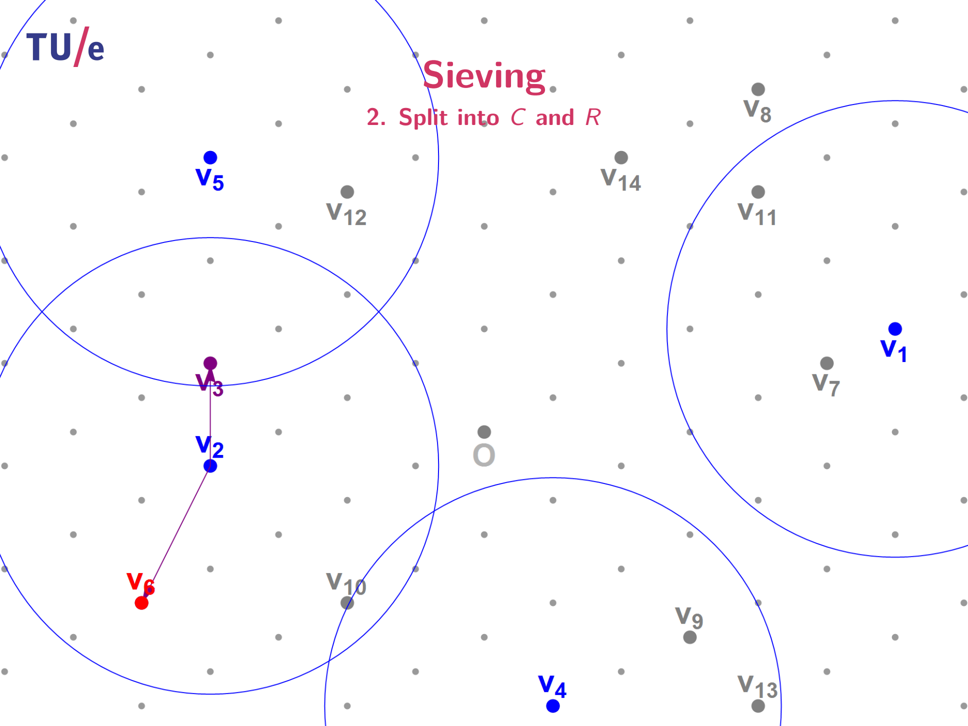
Sieving

2. Split into C and R



Sieving

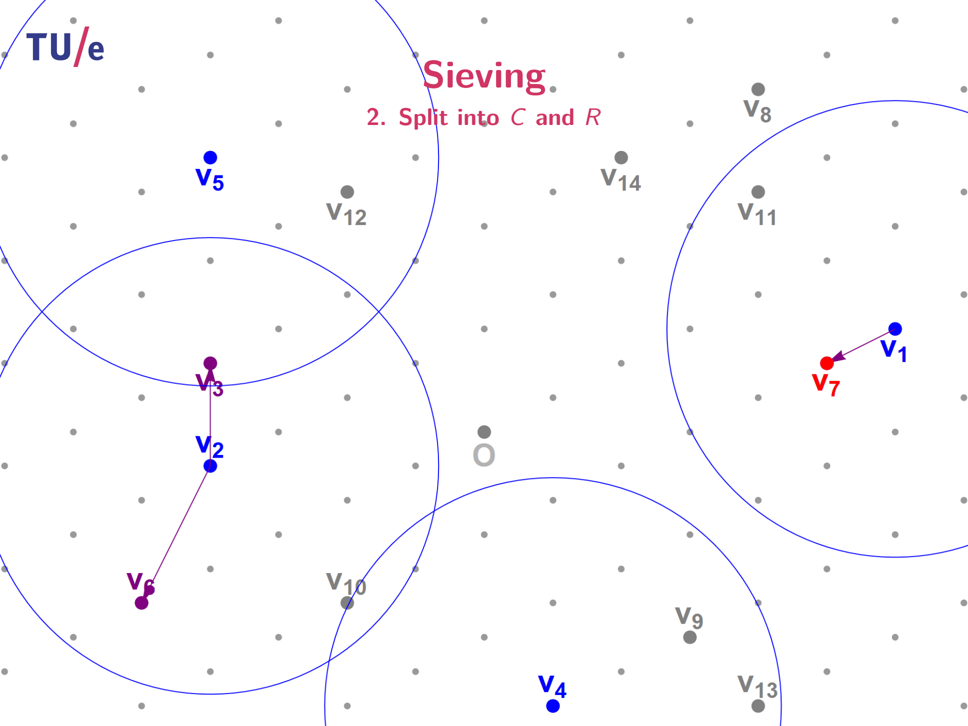
2. Split into C and R



TU/e

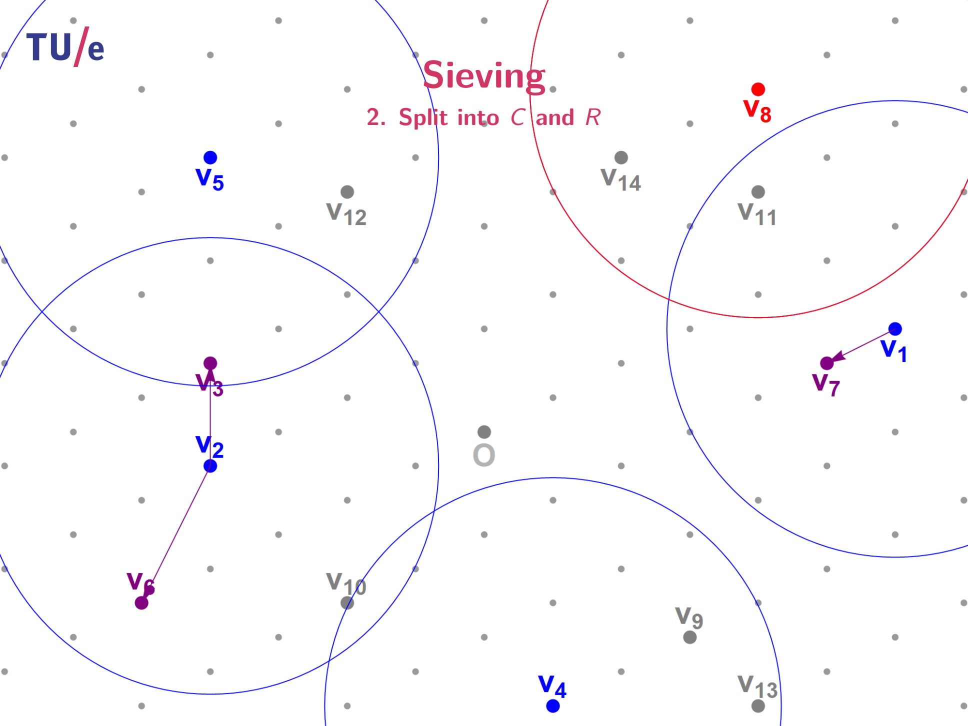
Sieving

2. Split into C and R



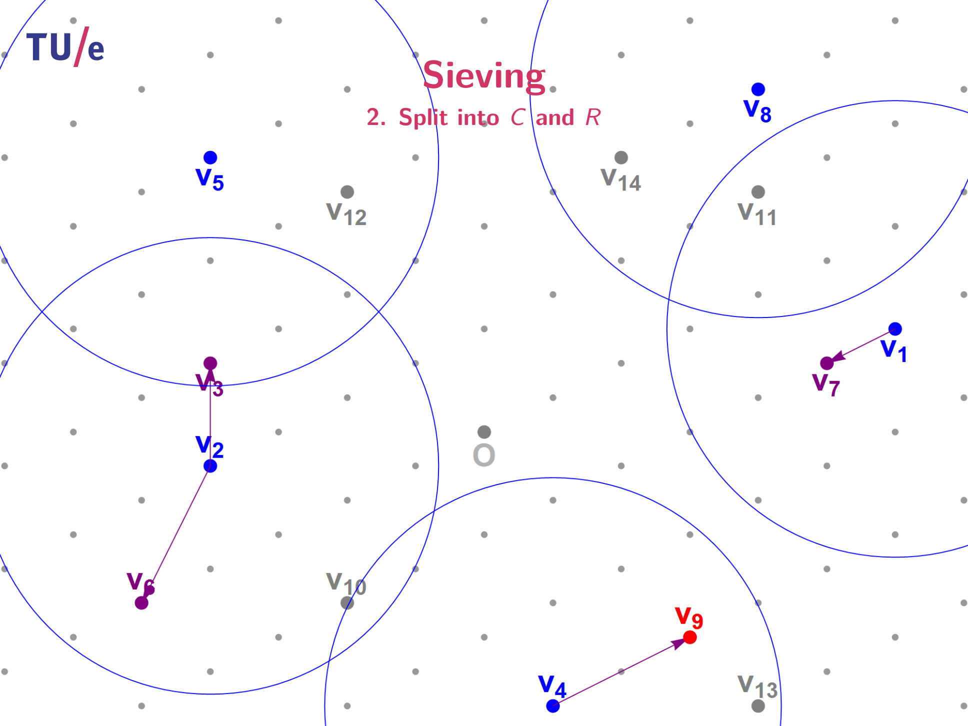
Sieving

2. Split into C and R



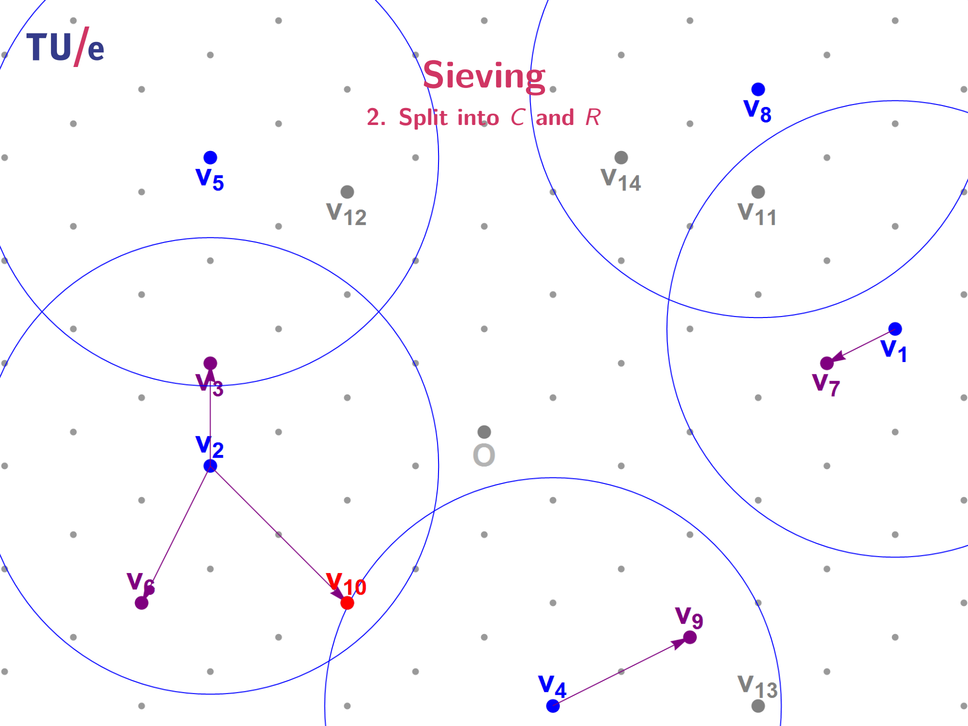
Sieving

2. Split into C and R



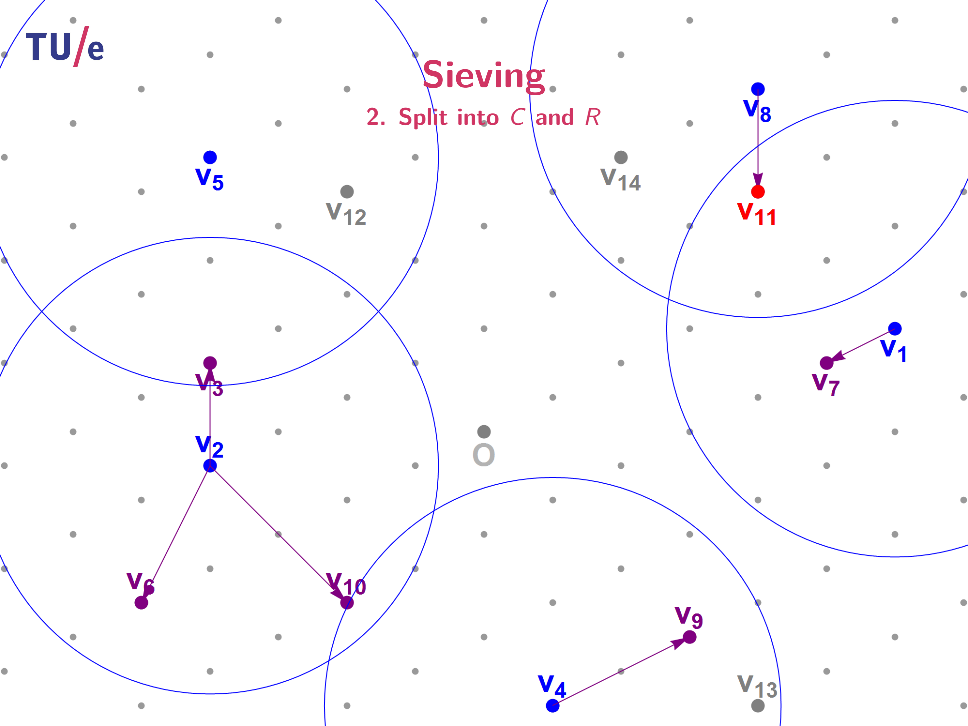
Sieving

2. Split into C and R



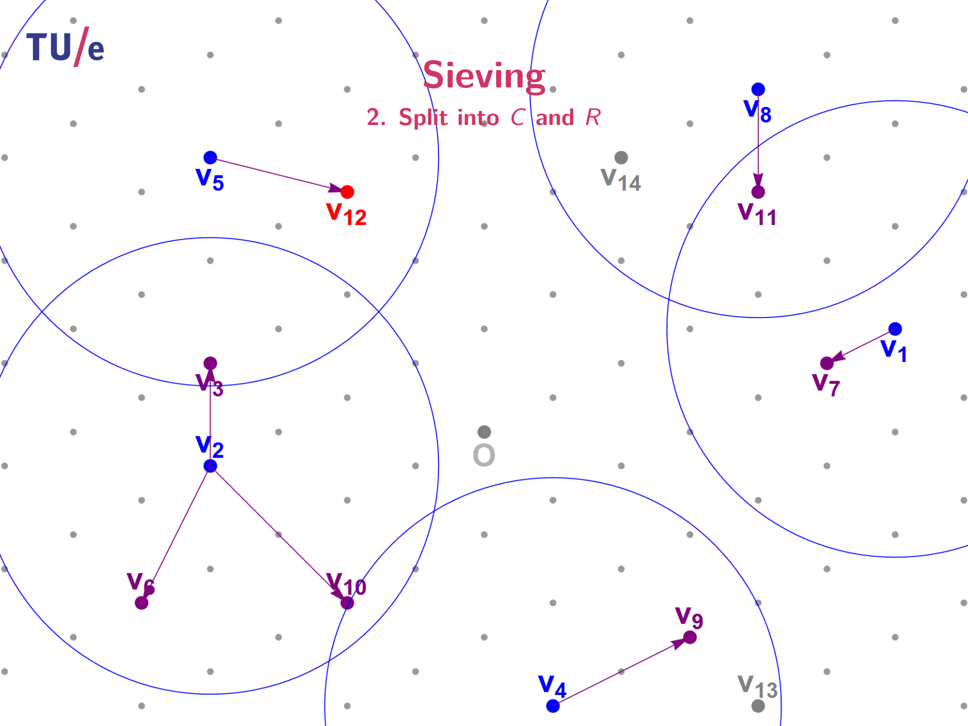
Sieving

2. Split into C and R



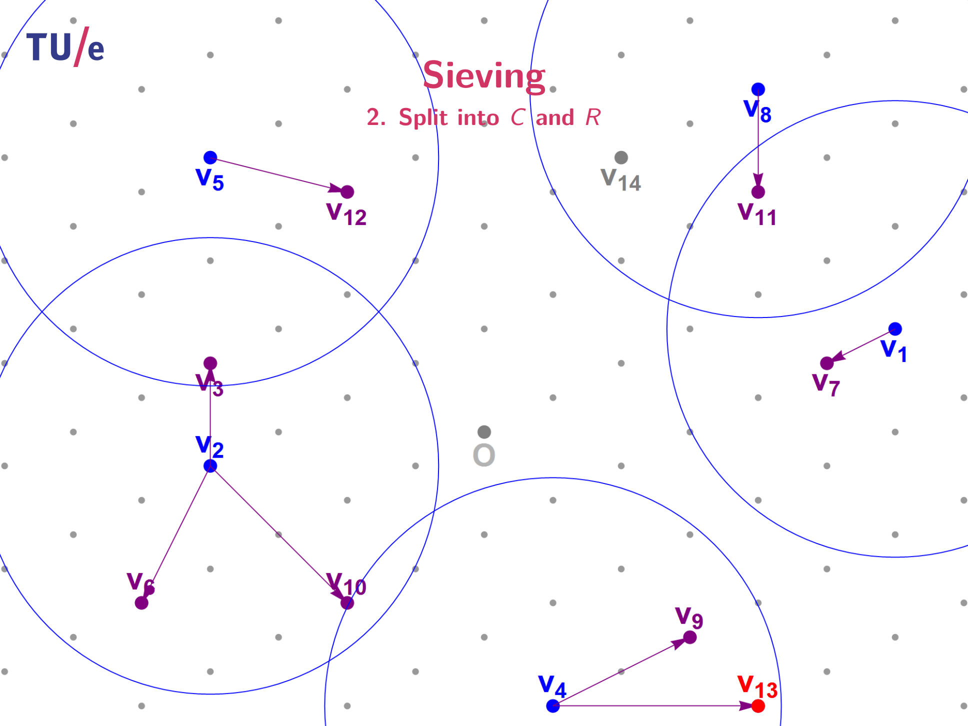
Sieving

2. Split into C and R



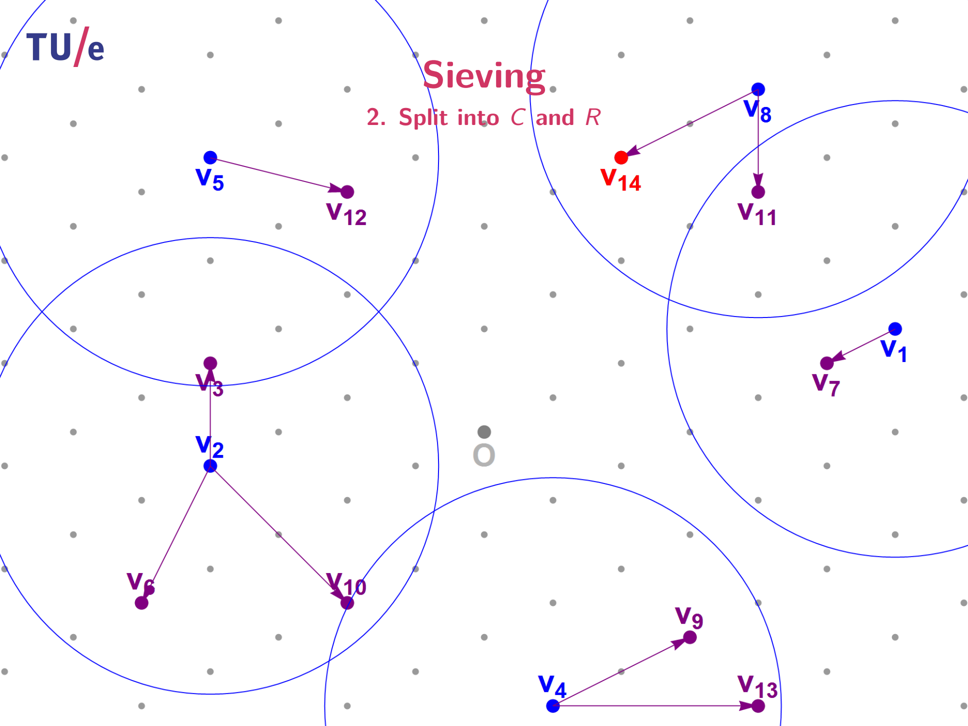
Sieving

2. Split into C and R



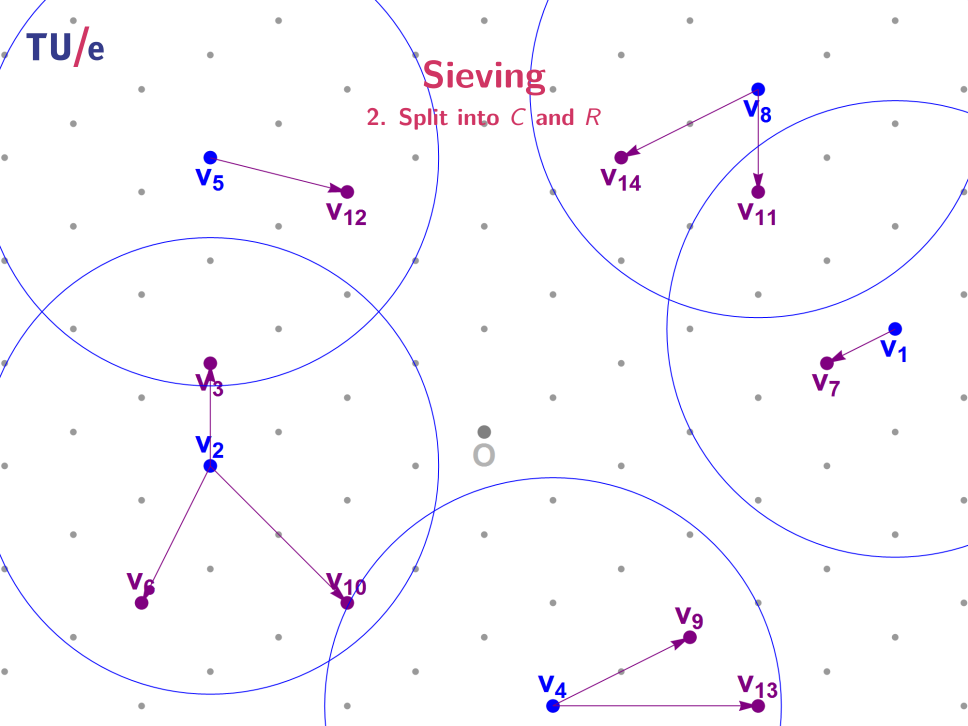
Sieving

2. Split into C and R



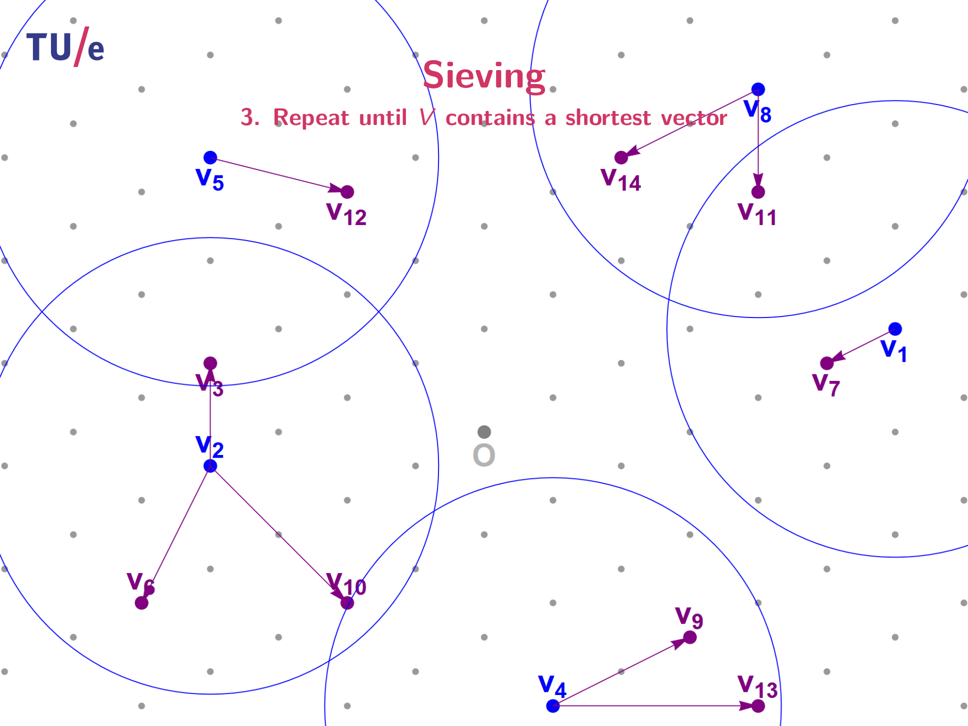
Sieving

2. Split into C and R



Sieving

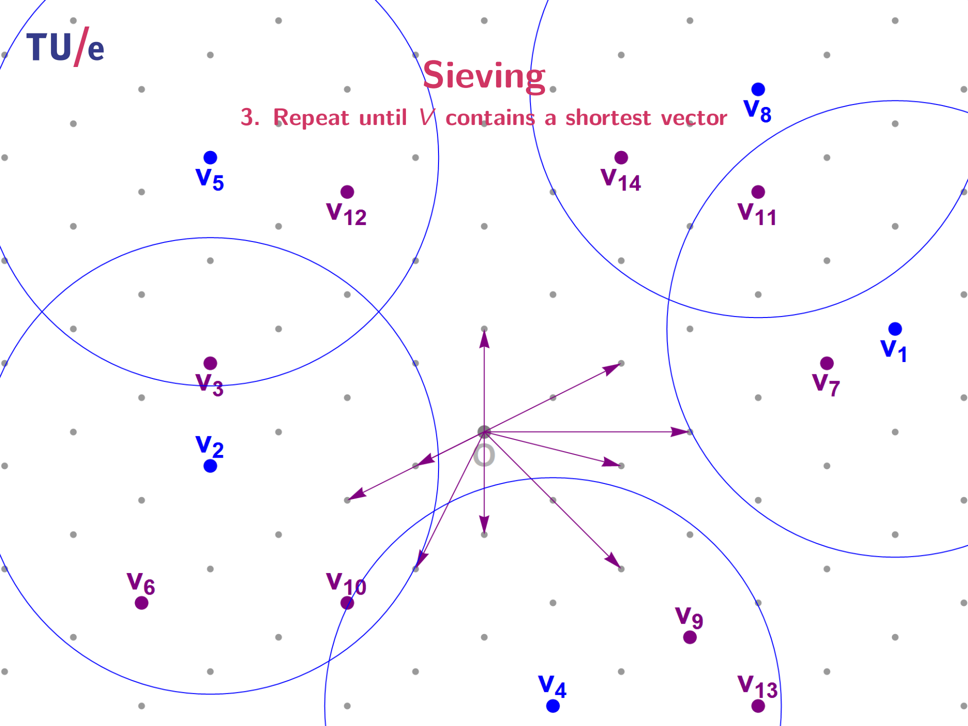
3. Repeat until V contains a shortest vector



TU/e

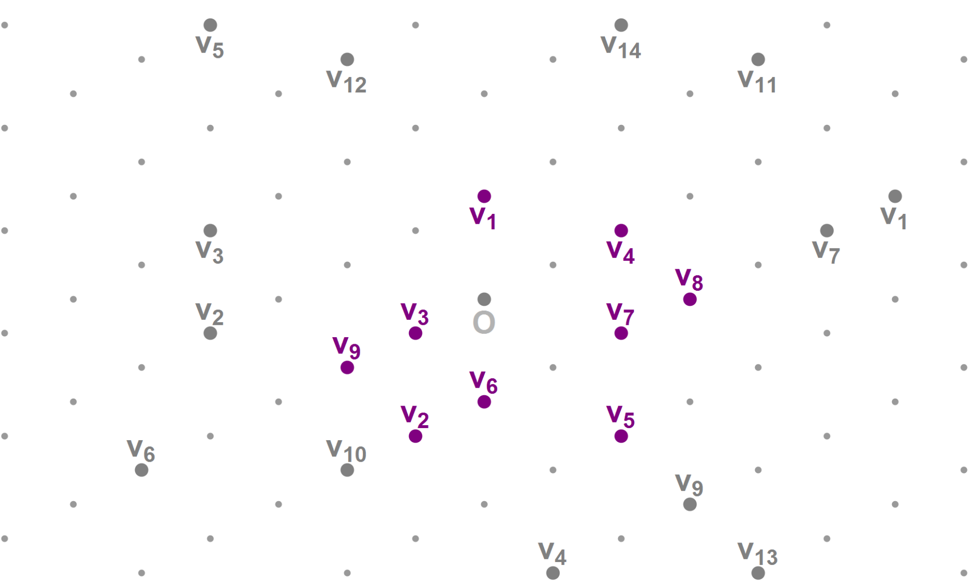
Sieving

3. Repeat until V contains a shortest vector



Sieving

3. Repeat until V contains a shortest vector



TU/e

Sieving

3. Repeat until V contains a shortest vector

Sieving

Studied since 2001 [AKS01, Reg04, NV08, . . ., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R
3. Set $V = R$ and repeat until V contains a shortest vector

Sieving

Studied since 2001 [AKS01, Reg04, NV08, ..., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R
3. Set $V = R$ and repeat until V contains a shortest vector

Complexity?

Sieving

Studied since 2001 [AKS01, Reg04, NV08, ..., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R
3. Set $V = R$ and repeat until V contains a shortest vector

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α

Sieving

Studied since 2001 [AKS01, Reg04, NV08, ..., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R
3. Set $V = R$ and repeat until V contains a shortest vector

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time:

Sieving

Studied since 2001 [AKS01, Reg04, NV08, ..., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R
3. Set $V = R$ and repeat until V contains a shortest vector

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time: $\approx 2^{\alpha n} \cdot 2^{\alpha n} = 2^{2\alpha n}$

Sieving

Studied since 2001 [AKS01, Reg04, NV08, . . ., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R
3. Set $V = R$ and repeat until V contains a shortest vector

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time: $\approx 2^{\alpha n} \cdot 2^{\alpha n} = 2^{2\alpha n}$
- Quantum Time:

TU/e

Sieving

Studied since 2001 [AKS01, Reg04, NV08, ..., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R
3. Set $V = R$ and repeat until V contains a shortest vector

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time: $\approx 2^{\alpha n} \cdot 2^{\alpha n} = 2^{2\alpha n}$
- Quantum Time: $\approx 2^{\alpha n} \cdot \sqrt{2^{\alpha n}} = 2^{\frac{3}{2}\alpha n}$

Sieving

Studied since 2001 [AKS01, Reg04, NV08, ..., ZPH13]

1. Generate a long list V of random lattice vectors
2. Split V into two sets C (centers, cover) and R (rest):
   - Set $C = \emptyset$ and $R = \emptyset$
   - For each $v \in V$, find the closest $c \in C$
     - If $\|v - c\|$ is "large", add v to C
     - If $\|v - c\|$ is "small", add $v - c$ to R
3. Set $V = R$ and repeat until V contains a shortest vector

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time: $\approx 2^{\alpha n} \cdot 2^{\alpha n} = 2^{2\alpha n}$
- Quantum Time: $\approx 2^{\alpha n} \cdot \sqrt{2^{\alpha n}} = 2^{\frac{3}{2}\alpha n}$
- Quantum speed-up: $\approx 25\%$ in the exponent

Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors

Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C

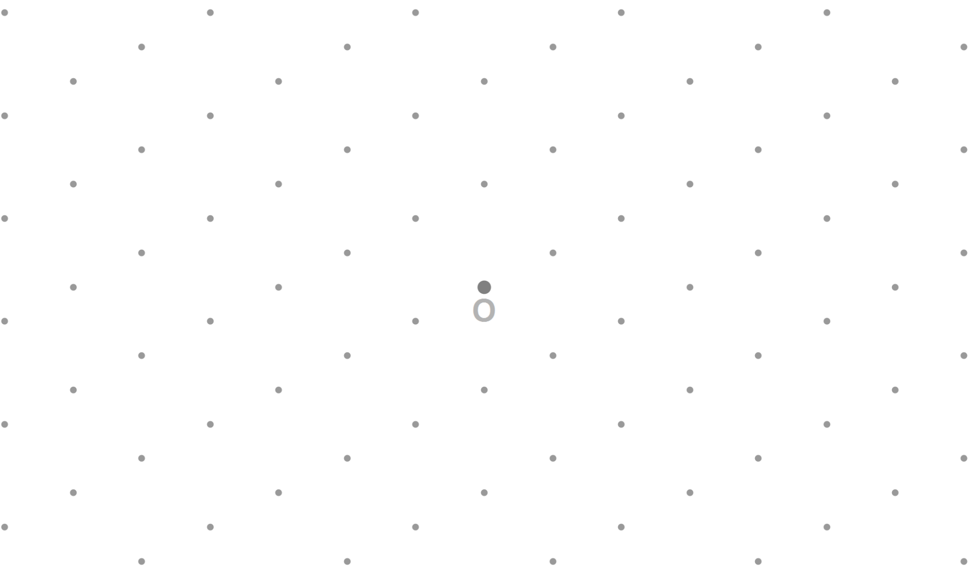
Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C
3. Search C for a shortest vector

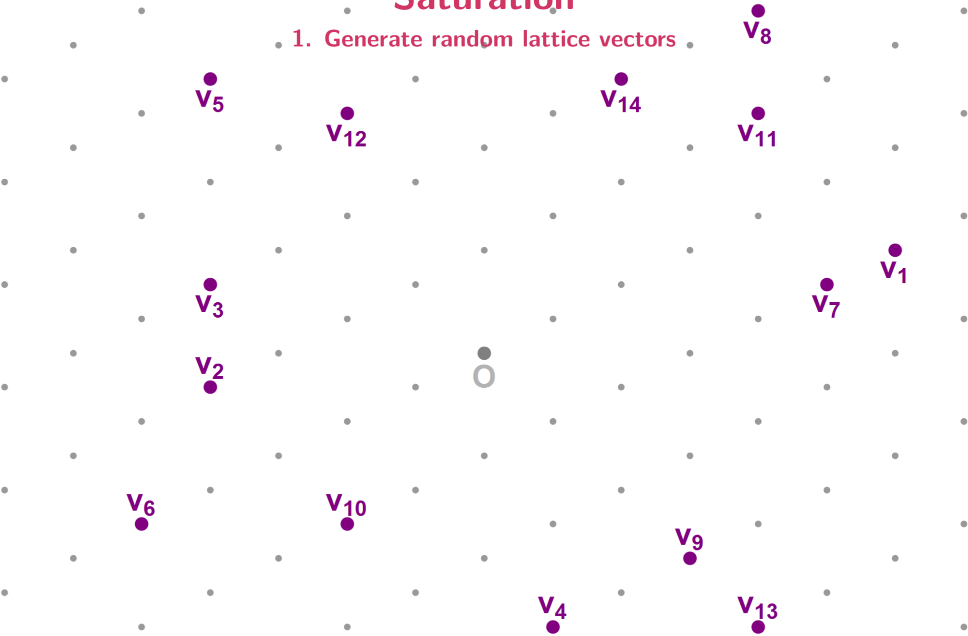
Saturation

1. Generate random lattice vectors



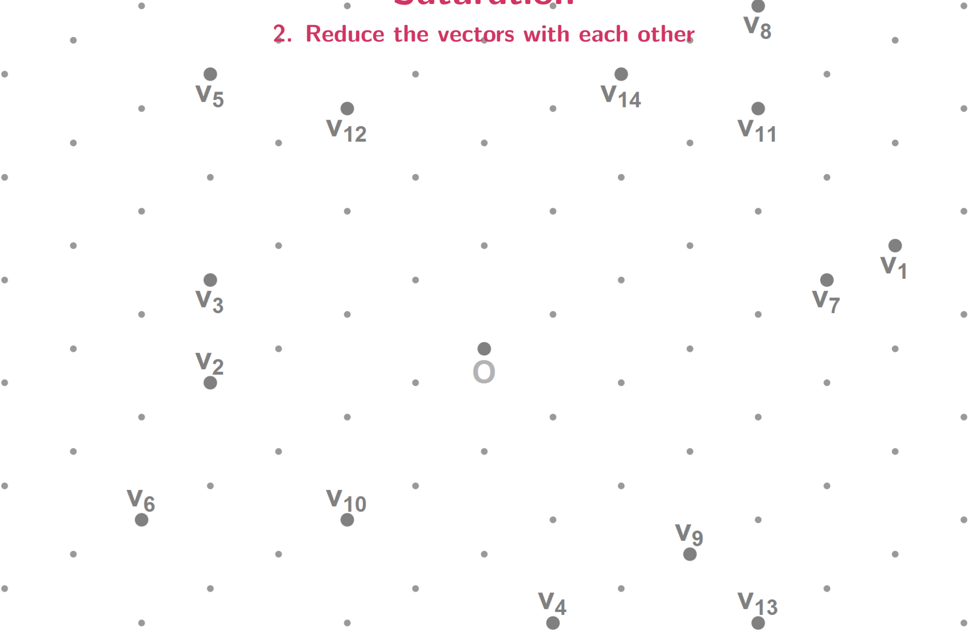
Saturation

1. Generate random lattice vectors



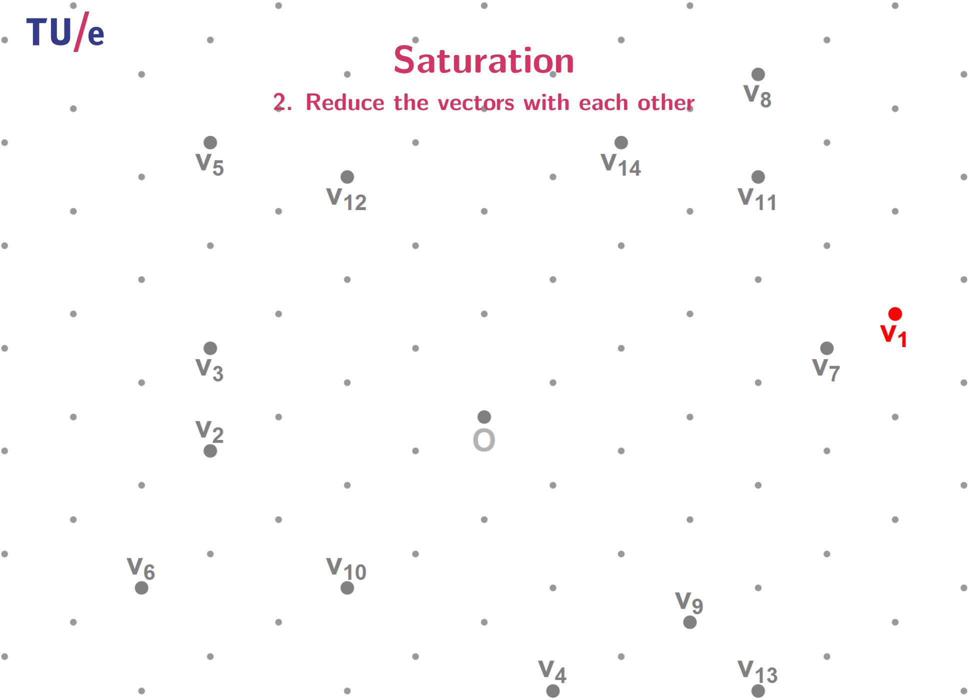
Saturation

2. Reduce the vectors with each other



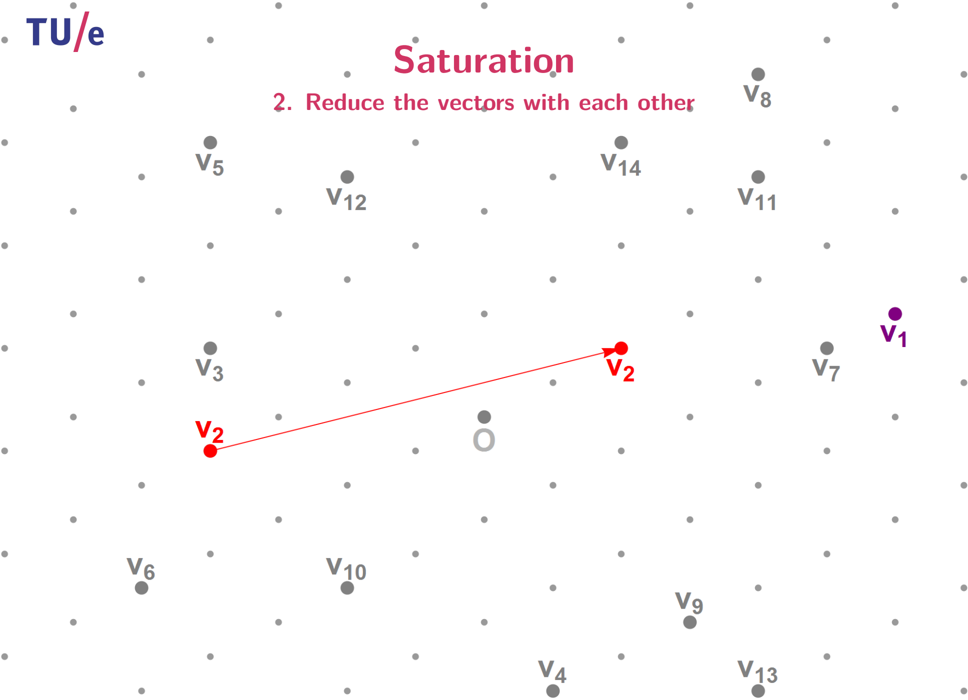
Saturation

2. Reduce the vectors with each other



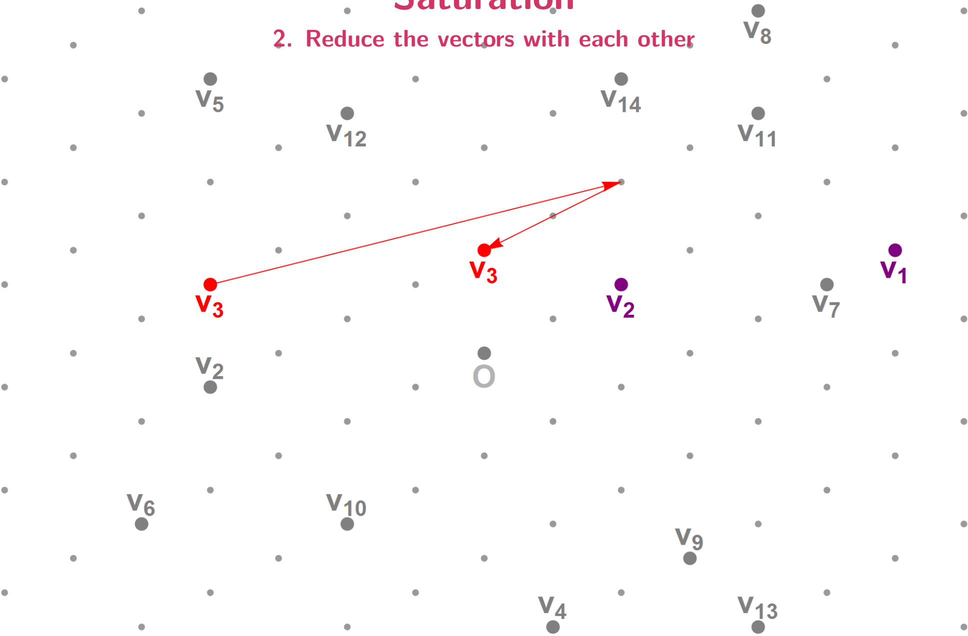
Saturation

2. Reduce the vectors with each other



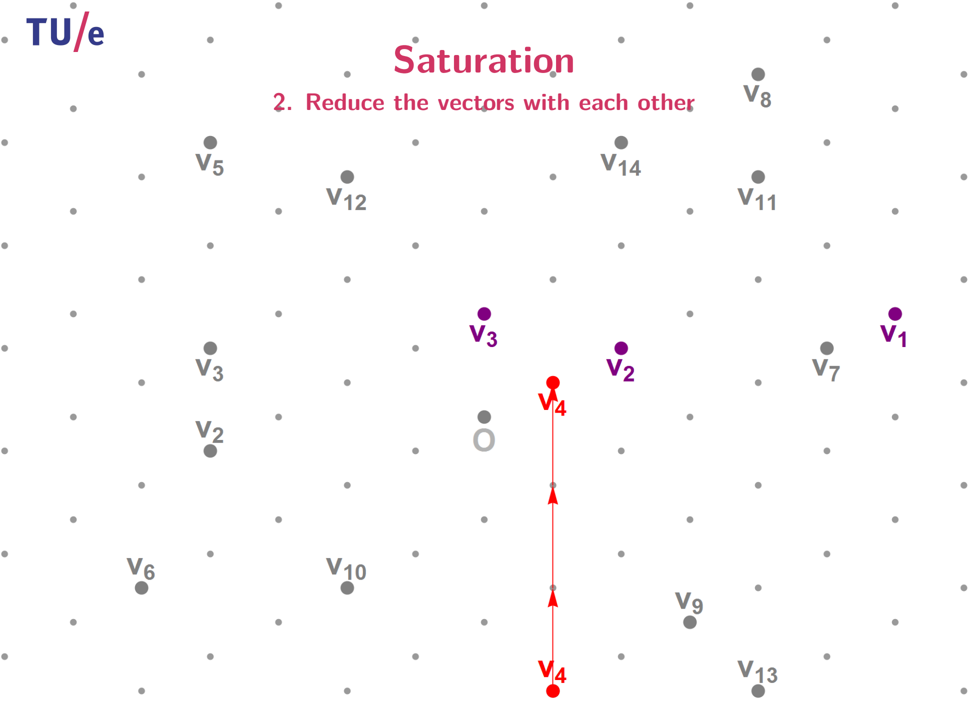
Saturation

2. Reduce the vectors with each other



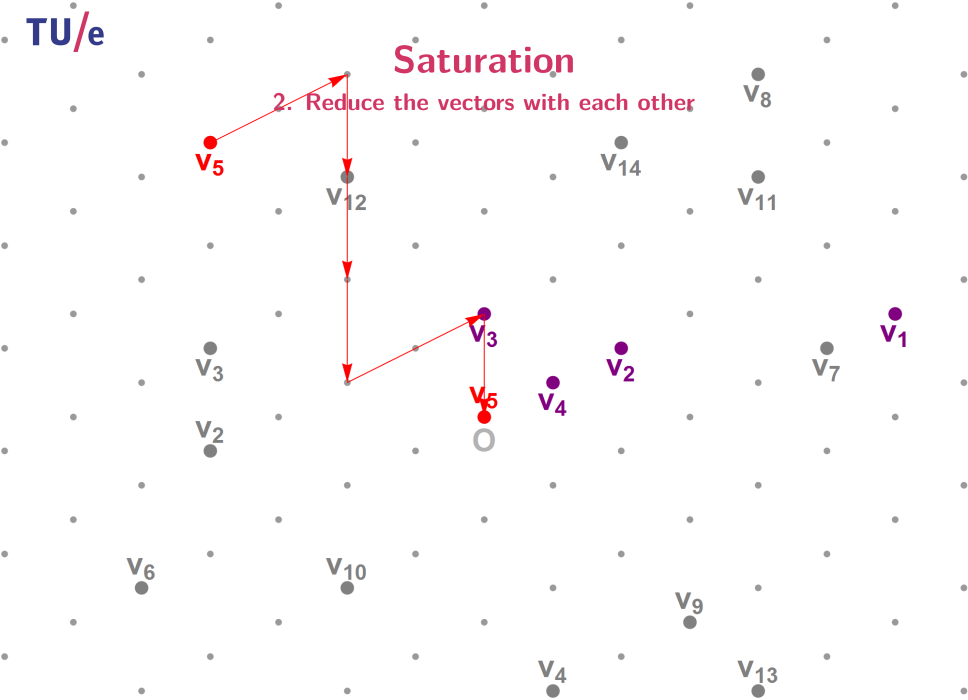
Saturation

2. Reduce the vectors with each other



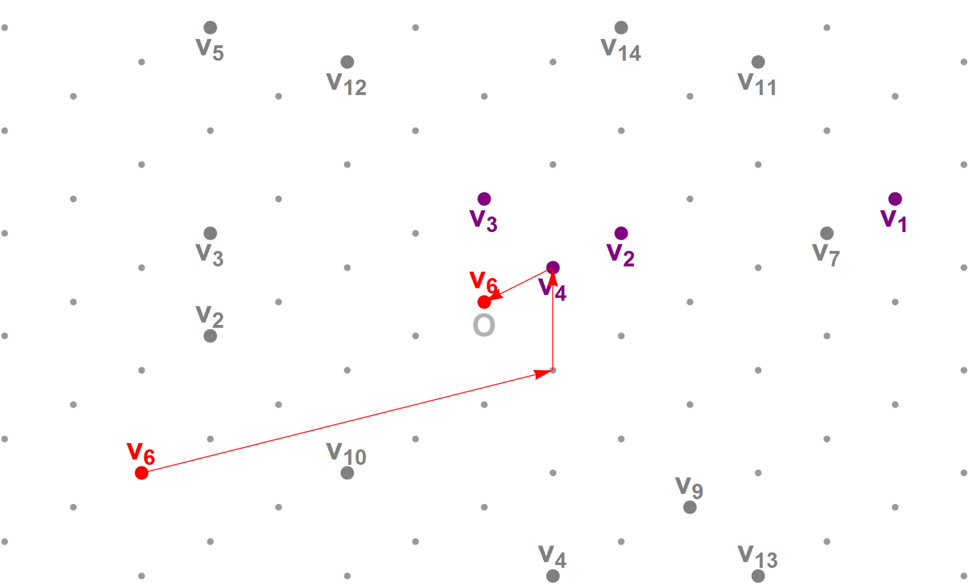
Saturation

2. Reduce the vectors with each other



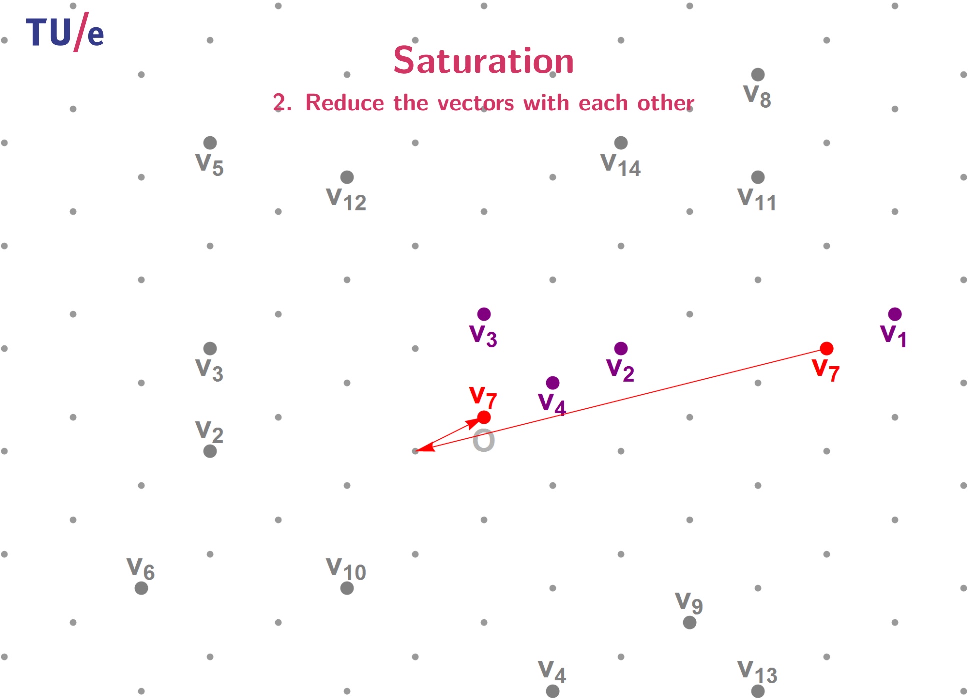
Saturation

2. Reduce the vectors with each other



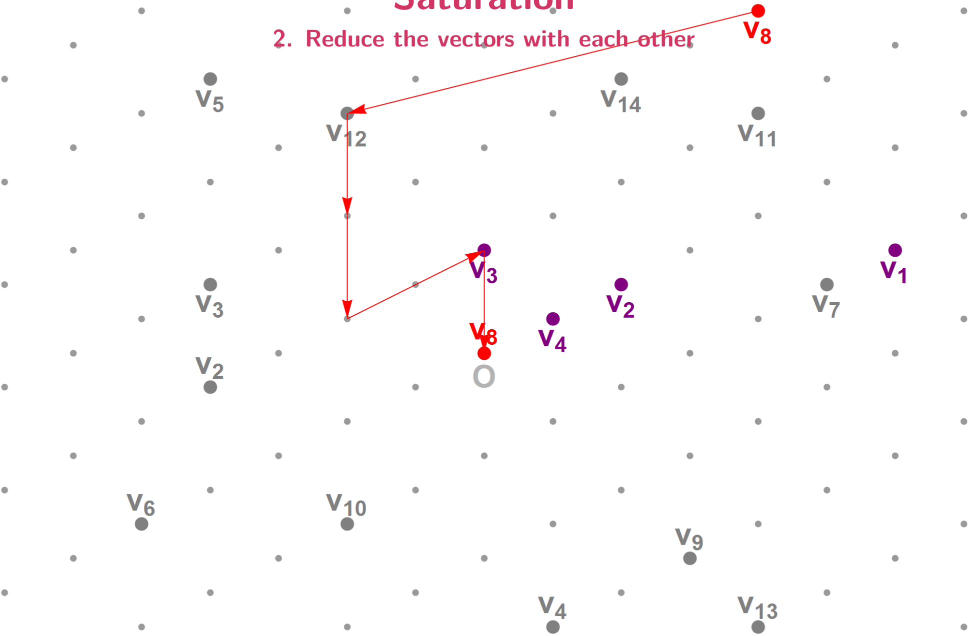
Saturation

2. Reduce the vectors with each other



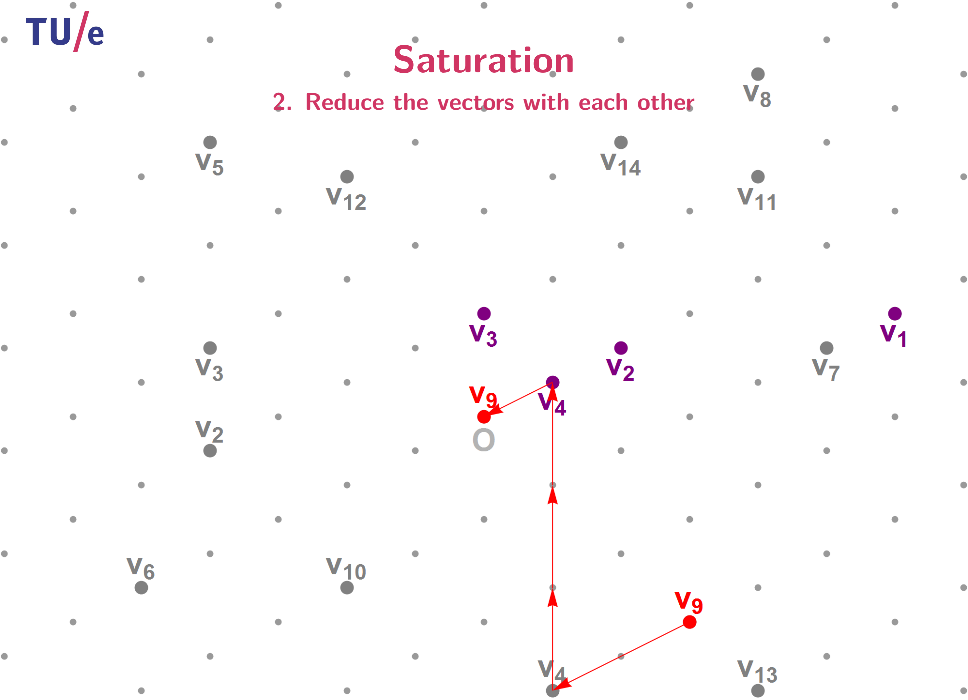
Saturation

2. Reduce the vectors with each other



Saturation

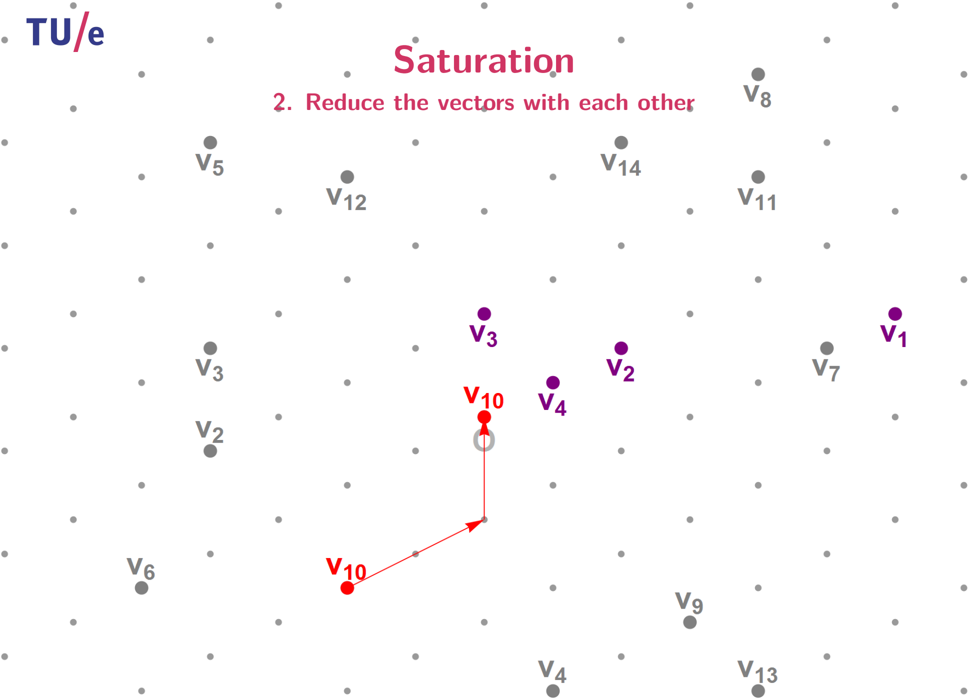
2. Reduce the vectors with each other



TU/e

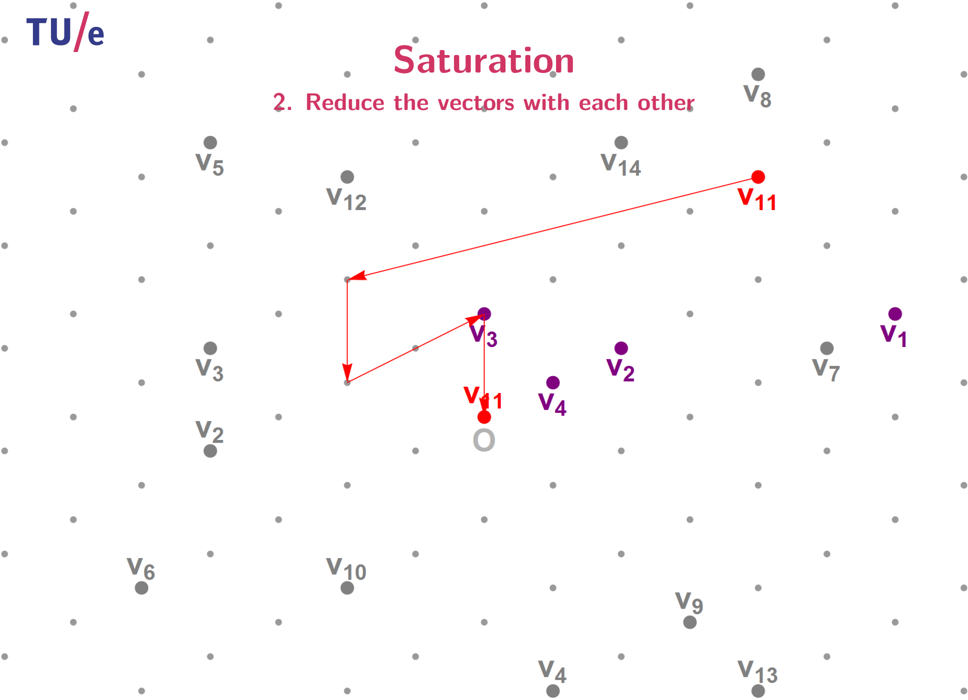
Saturation

2. Reduce the vectors with each other



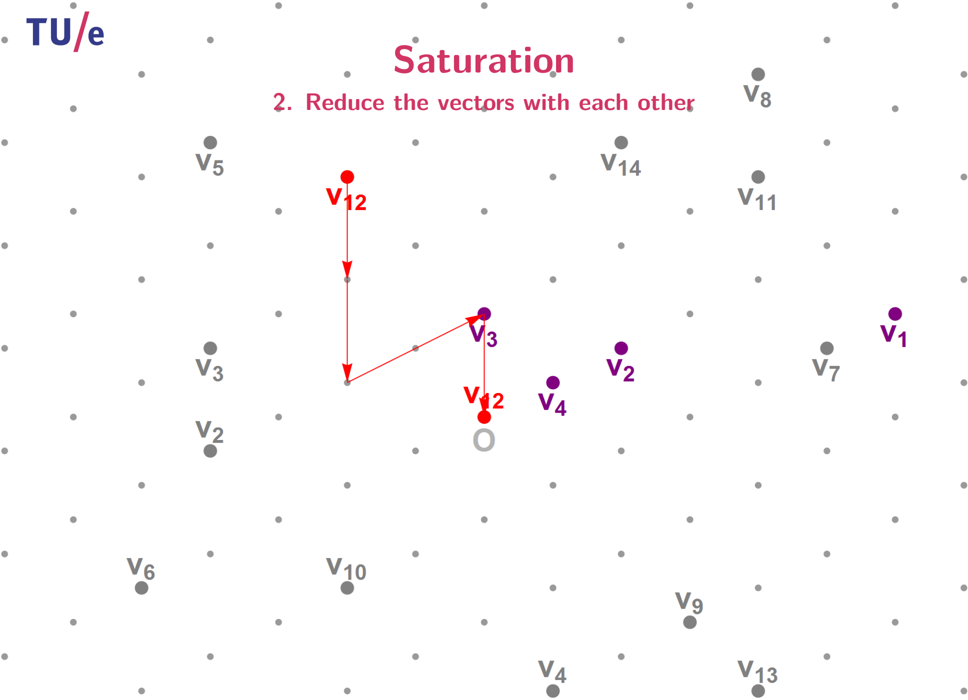
Saturation

2. Reduce the vectors with each other



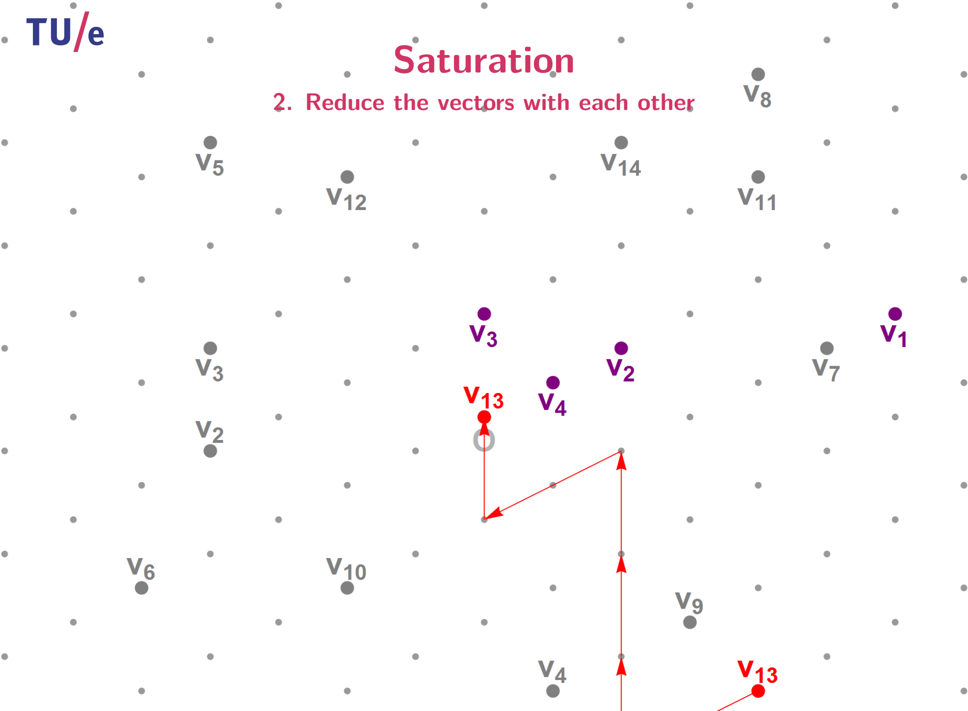
Saturation

2. Reduce the vectors with each other



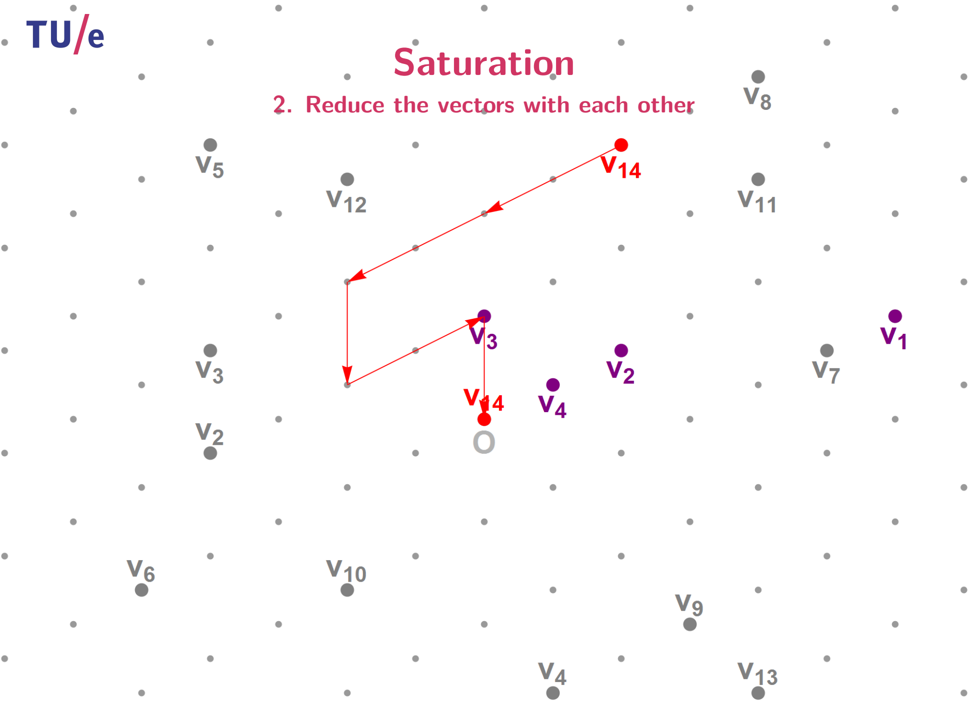
Saturation

2. Reduce the vectors with each other



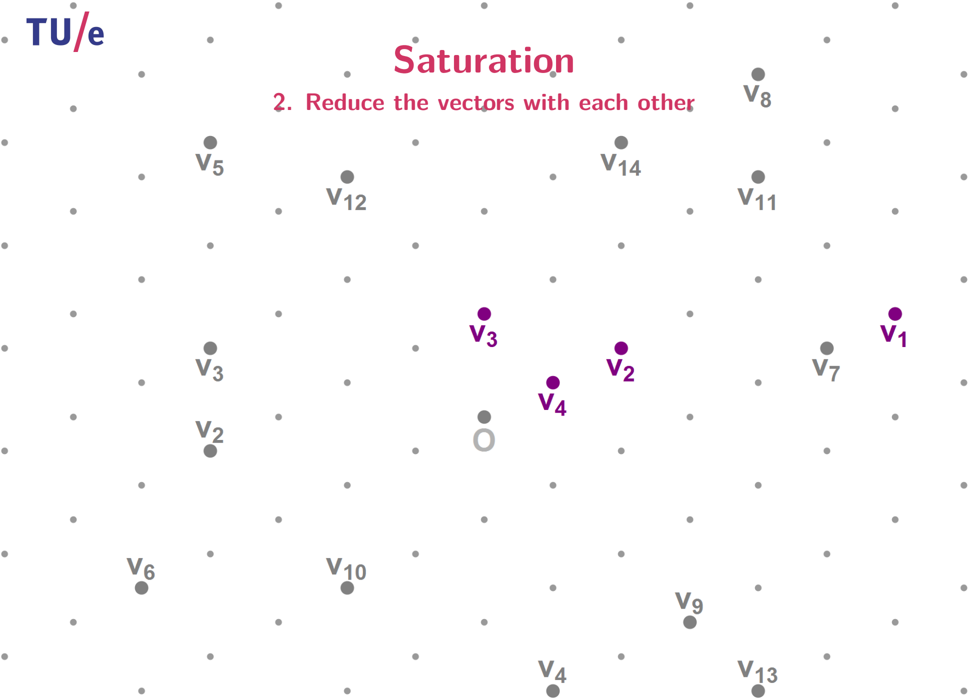
Saturation

2. Reduce the vectors with each other



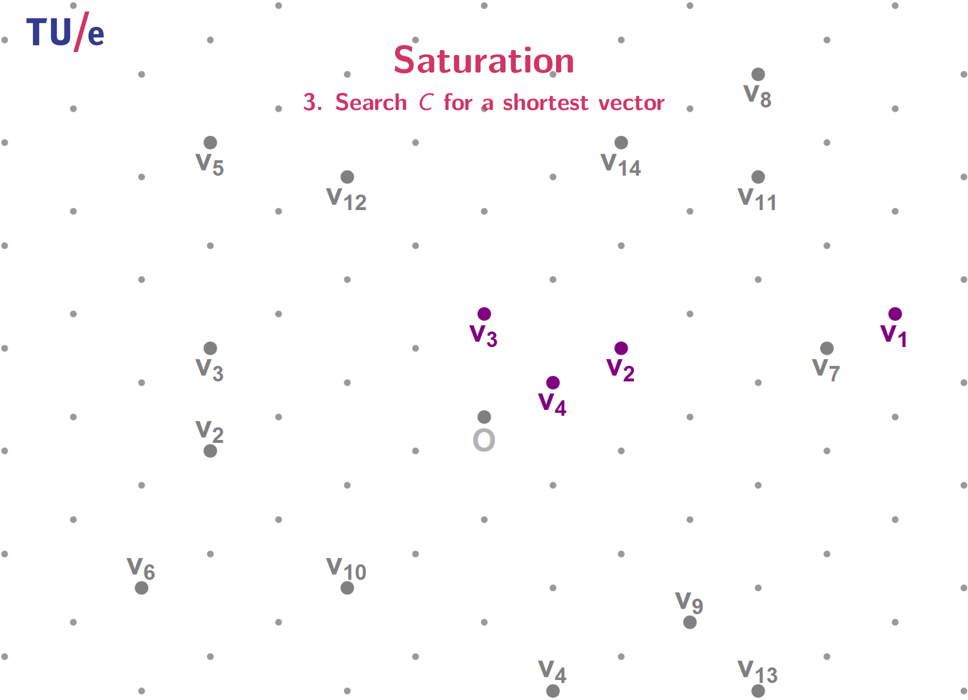
Saturation

2. Reduce the vectors with each other



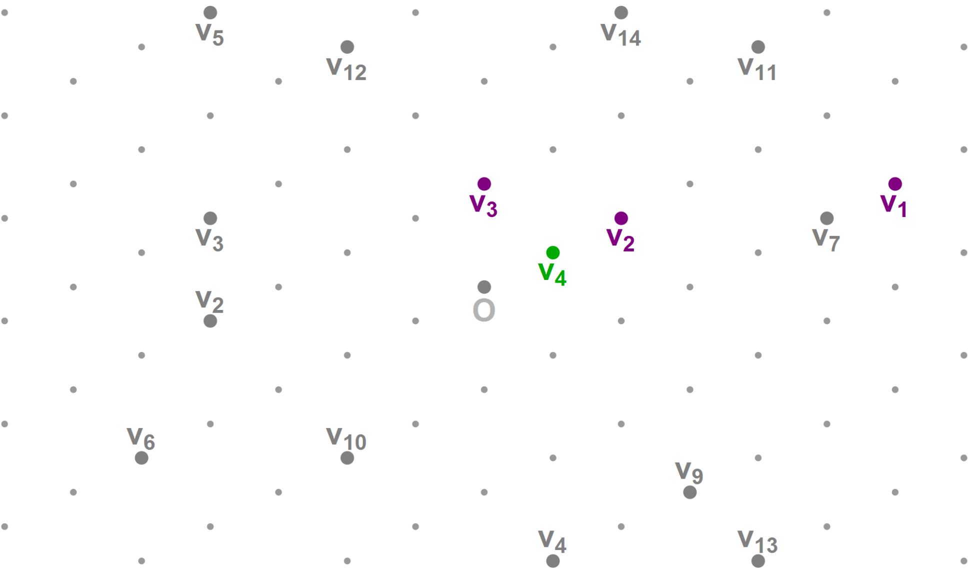
Saturation

3. Search C for a shortest vector



Saturation

3. Search C for a shortest vector



Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C
3. Find a shortest vector among the reduced vectors

Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C
3. Find a shortest vector among the reduced vectors

Complexity?

Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C
3. Find a shortest vector among the reduced vectors

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α

Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C
3. Find a shortest vector among the reduced vectors

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time:

Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C
3. Find a shortest vector among the reduced vectors

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time: $\approx 2^{\alpha n} \cdot 2^{\alpha n} = 2^{2\alpha n}$

Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C
3. Find a shortest vector among the reduced vectors

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time: $\approx 2^{\alpha n} \cdot 2^{\alpha n} = 2^{2\alpha n}$
- Quantum Time:

Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C
3. Find a shortest vector among the reduced vectors

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time: $\approx 2^{\alpha n} \cdot 2^{\alpha n} = 2^{2\alpha n}$
- Quantum Time: $\approx 2^{\alpha n} \cdot \sqrt{2^{\alpha n}} = 2^{\frac{3}{2}\alpha n}$

Saturation

Studied since 2009 [MV10, PS09, Sch11, IKMT13]

1. Generate a long list V of random lattice vectors
2. "Reduce the vectors with each other":
   - Set $C = \emptyset$
   - For each $v \in V$, find the closest vector $c \in C$
     - If $\|v - c\| < \|v\|$, set $v \leftarrow v - c$ and find new closest $c \in C$
     - If $\|v - c\| \geq \|v\|$, add v to C
3. Find a shortest vector among the reduced vectors

Complexity?

- Space: $|V|, |C|, |R| \leq 2^{\alpha n}$ for some α
- Classical Time: $\approx 2^{\alpha n} \cdot 2^{\alpha n} = 2^{2\alpha n}$
- Quantum Time: $\approx 2^{\alpha n} \cdot \sqrt{2^{\alpha n}} = 2^{\frac{3}{2}\alpha n}$
- Quantum speed-up: $\approx 25\%$ in the exponent

Overview

Provable results (large n asymptotics)

Table: Complexities of SVP algorithms in logarithmic leading order terms:

	Classical		Quantum	
Algorithm	Time	Space	Time	Space
Enum. [Kan83]	$O(n \log n)$	$O(\log n)$	$O(n \log n)$	$O(\log n)$
Sieving [PS09]	$2.65n$	$1.33n$	$2.65n$	$1.33n$
Saturation [PS09]	$2.47n$	$1.24n$	$2.47n$	$1.24n$
Voronoi cell [MV10]	$2.00n$	$1.00n$	$2.00n$	$1.00n$

TU/e

Overview

Provable results (large n asymptotics)

Table: Complexities of SVP algorithms in logarithmic leading order terms:

	Classical		Quantum	
Algorithm	Time	Space	Time	Space
Enum. [Kan83]	$O(n \log n)$	$O(\log n)$	$O(n \log n)$	$O(\log n)$
Sieving [PS09]	$2.65n$	$1.33n$	$2.65n$	$1.33n$
Saturation [LMP13]	$2.47n$	$1.24n$	**1.80n**	**1.29n**
Voronoi cell [MV10]	$2.00n$	$1.00n$	$2.00n$	$1.00n$

Overview

Heuristic/Experimental results ($n \approx 100$)

Table: Complexities of SVP algorithms in logarithmic leading order terms:

Algorithm	Classical Time	Classical Space	Quantum Time	Quantum Space
Enum. [GNR10]	$O(n \log n)$	$O(\log n)$	$O(n \log n)$	$O(\log n)$
Sieving [NV08]	$0.42n$	$0.21n$	$0.42n$	$0.21n$
Saturation [MV10]	$0.52n$	$0.21n$	$0.52n$	$0.21n$
Voronoi cell [MV10]	$2.00n$	$1.00n$	$2.00n$	$1.00n$

Overview

Heuristic/Experimental results ($n \approx 100$)

Table: Complexities of SVP algorithms in logarithmic leading order terms:

Algorithm	Classical Time	Classical Space	Quantum Time	Quantum Space
Enum. [GNR10]	$O(n \log n)$	$O(\log n)$	$O(n \log n)$	$O(\log n)$
Sieving [LMP13]	$0.42n$	$0.21n$	**0.32n**	**0.21n**
Saturation [LMP13]	$0.52n$	$0.21n$	**0.39n**	**0.21n**
Voronoi cell [MV10]	$2.00n$	$1.00n$	$2.00n$	$1.00n$

Conclusion

Using Grover search speeds up some SVP algorithms

- Faster sieving algorithms (exponent: $\approx -25\%$)
- Faster saturation algorithms (exponent: $\approx -25\%$)

Open quantum problems

- Other speed-ups for these algorithms?
- Speed-ups for other SVP algorithms? (see part 2)
- Speed-ups for lattice basis reduction?
- Speed-ups for lattices with more structure?

Questions