Lattice algorithms for the shortest vector problem

Thijs Laarhoven

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

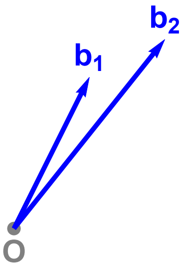
Aussois Winter School
(March 21st, 2019)

Lattices

What is a lattice?

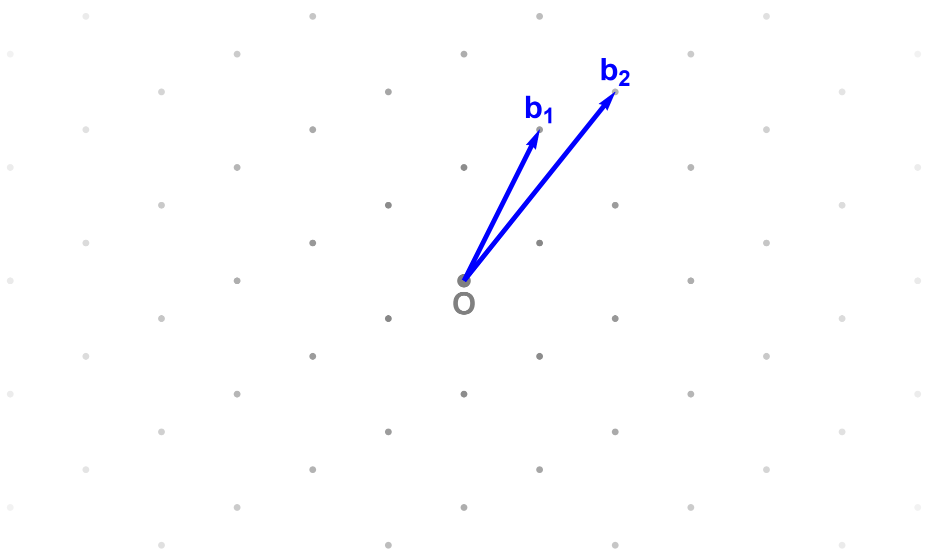
Lattices

What is a lattice?



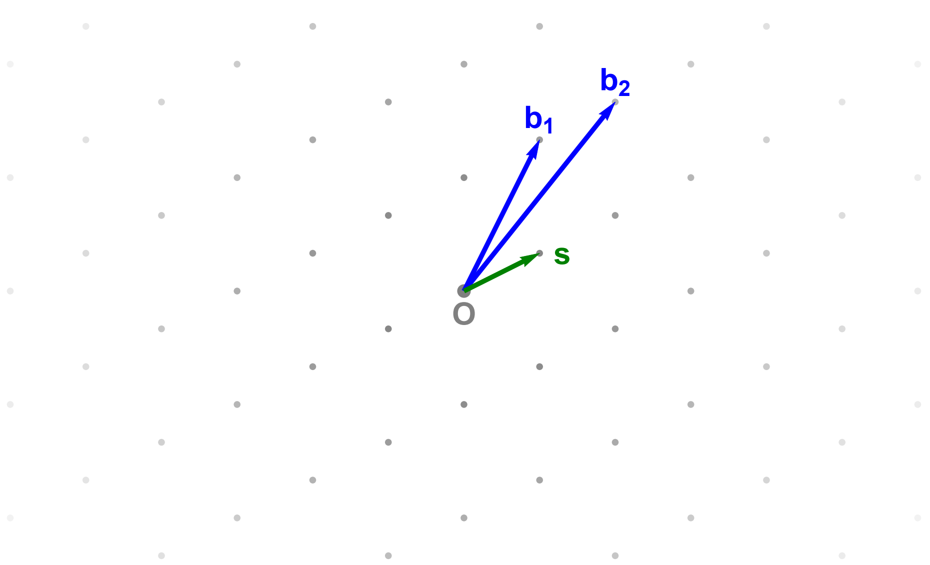
Lattices

What is a lattice?



Lattices

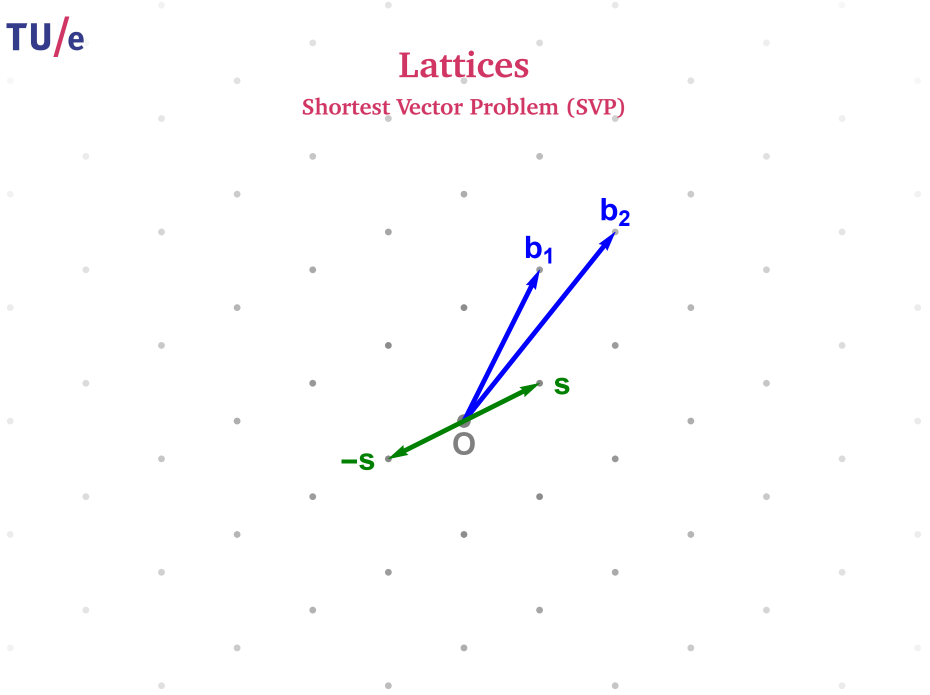
Shortest Vector Problem (SVP)



TU/e

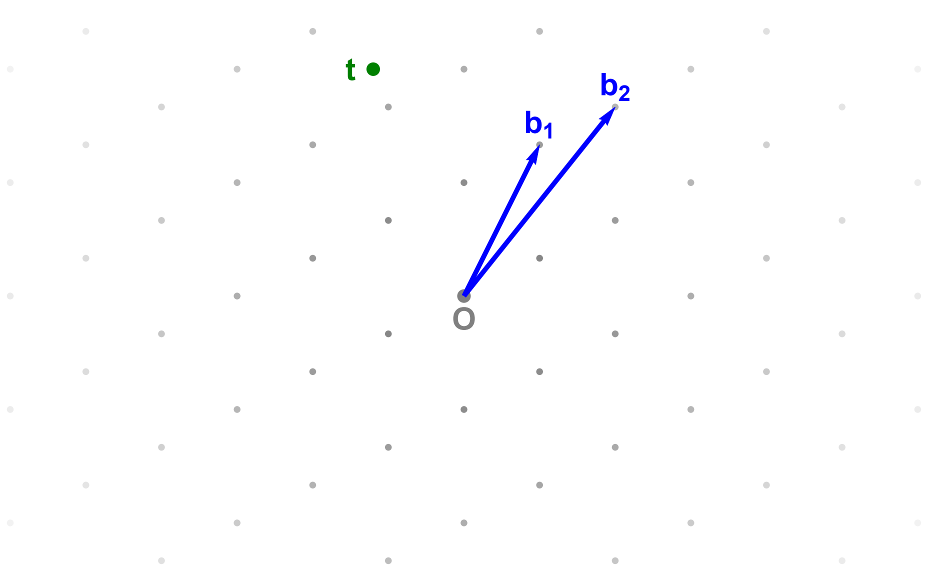
Lattices

Shortest Vector Problem (SVP)



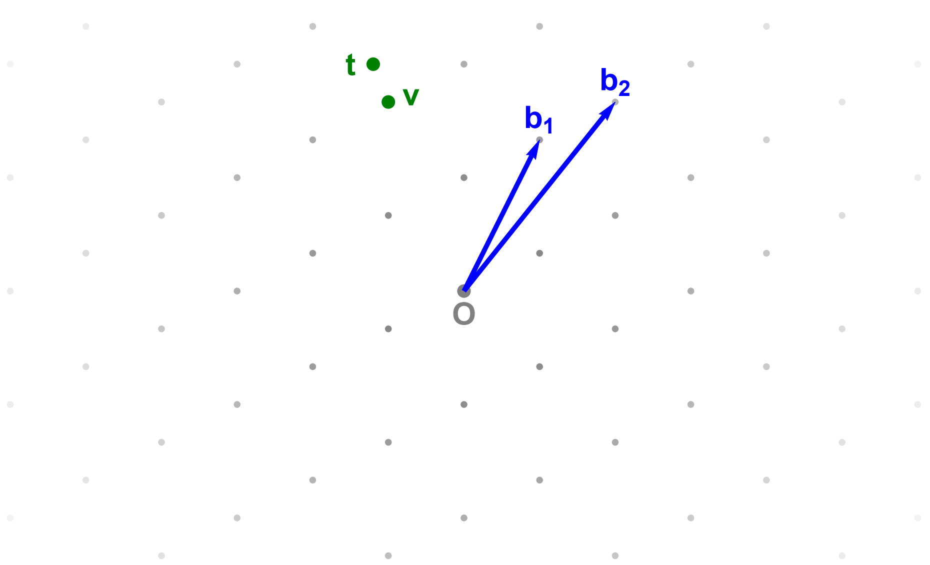
Lattices

Closest Vector Problem (CVP)



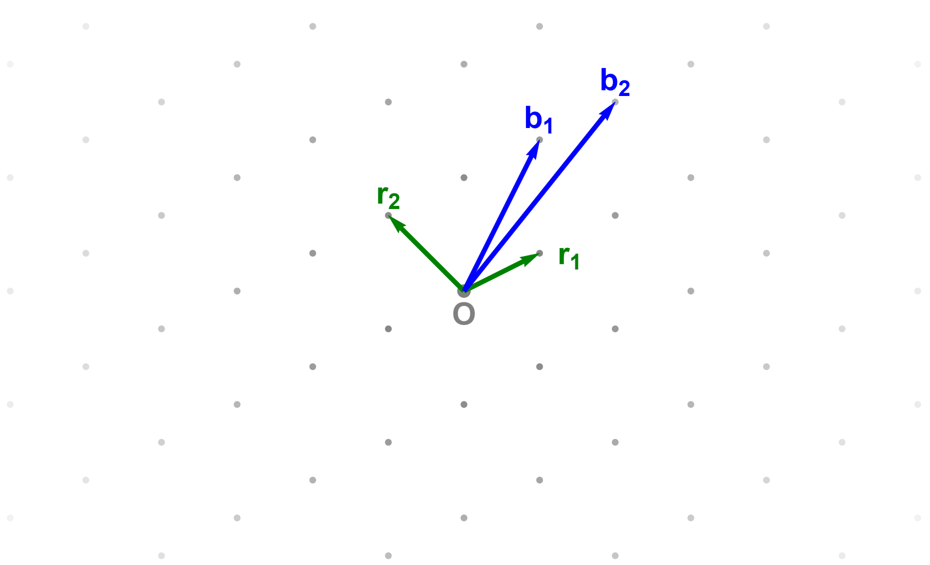
Lattices

Closest Vector Problem (CVP)



Lattices

Lattice basis reduction



TU/e

Lattices

Lattice-based cryptanalysis

Problem: Security of lattice-based cryptographic primitives

- Most lattice problems solvable via (approximate) SVP
- State-of-the-art: BKZ basis reduction [Sch87, SE94, ...]
  - Leo's algorithmic ant and the sandpile
  - BKZ uses exact SVP algorithm as subroutine
  - Complexity of BKZ dominated by *exact* SVP calls

Problem: How hard is SVP in high dimensions?

Outline

Lattices

SVP algorithms

- Enumeration
- Sieving

SVP hardness

- Theory
- Practice

Conclusion

Outline

Lattices

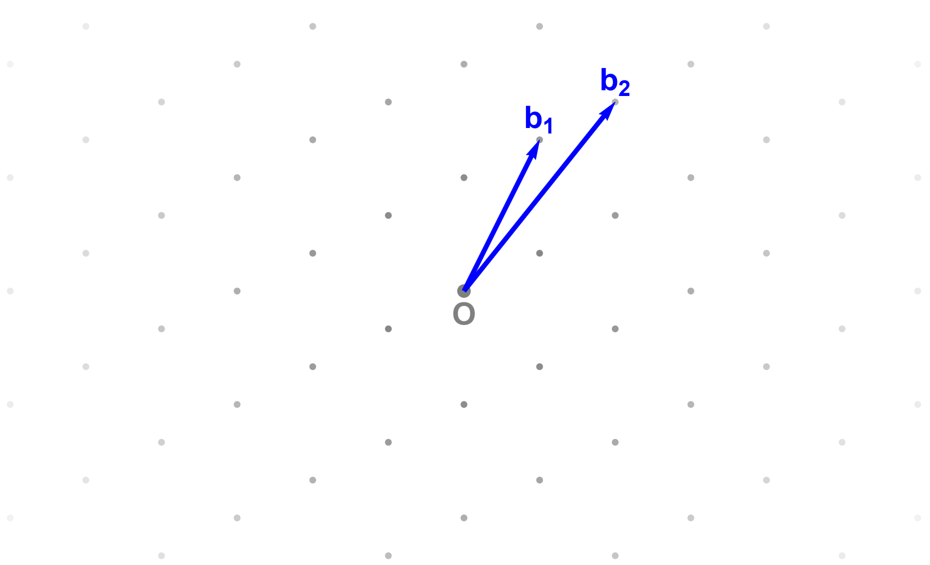
SVP algorithms
- Enumeration
- Sieving

SVP hardness
- Theory
- Practice

Conclusion

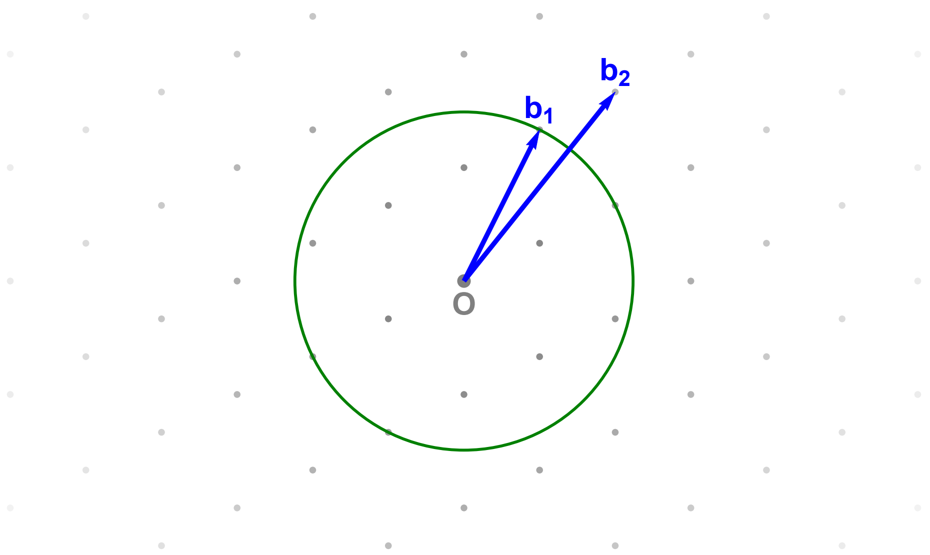
Enumeration

Determine possible coefficients of b_2



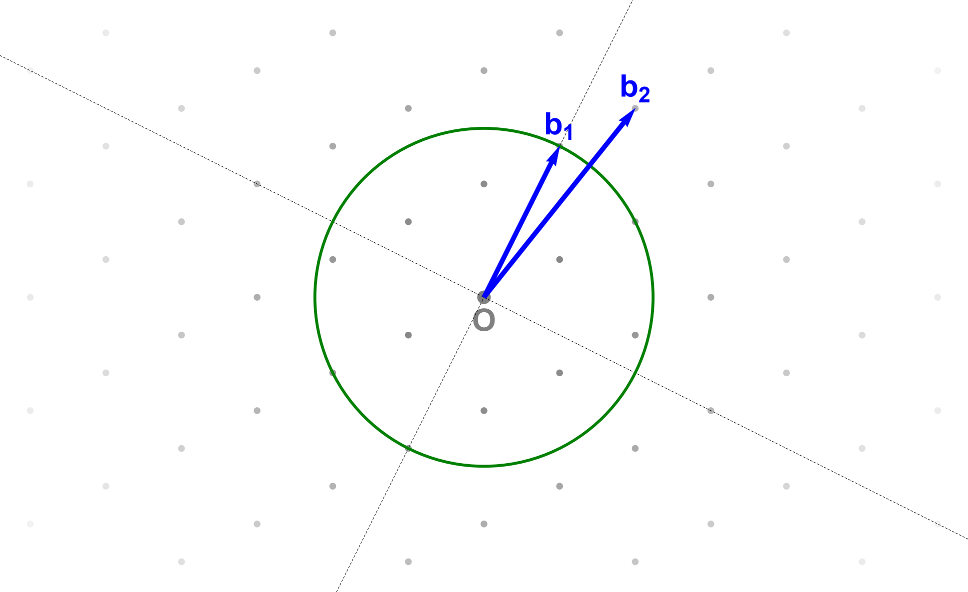
Enumeration

Determine possible coefficients of b_2



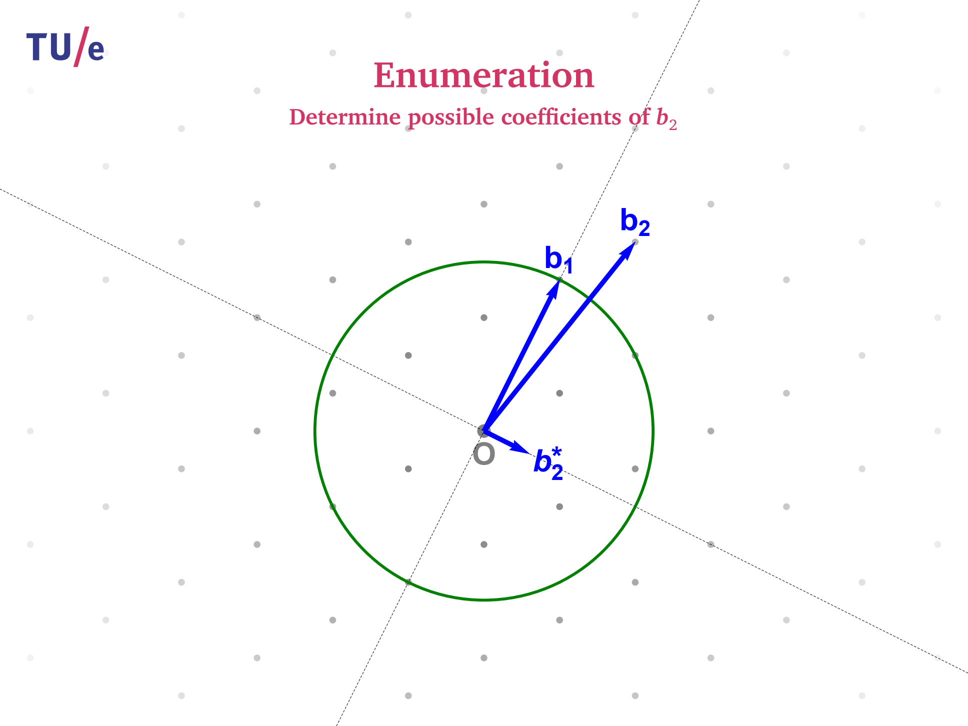
Enumeration

Determine possible coefficients of b_2



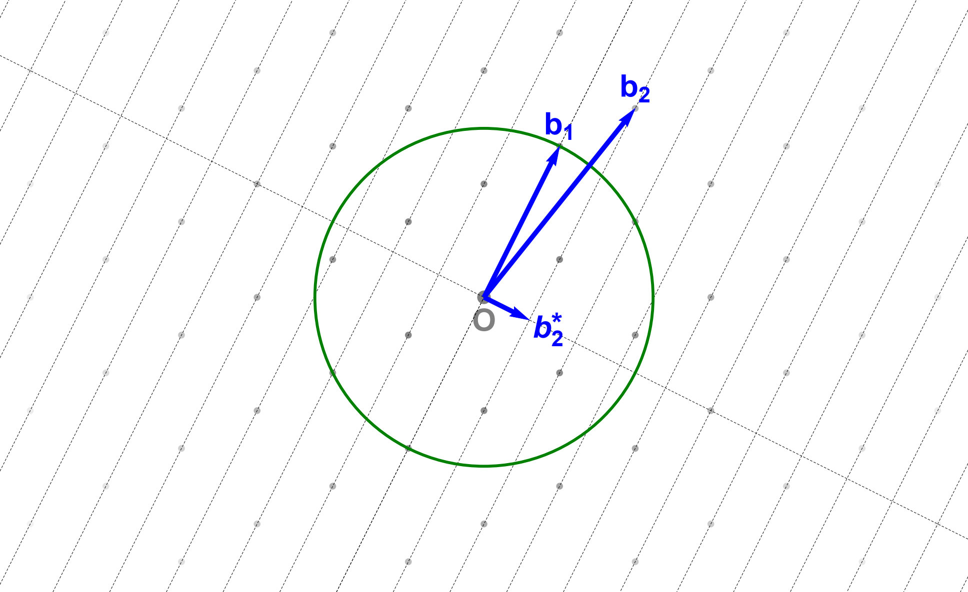
Enumeration

Determine possible coefficients of b_2



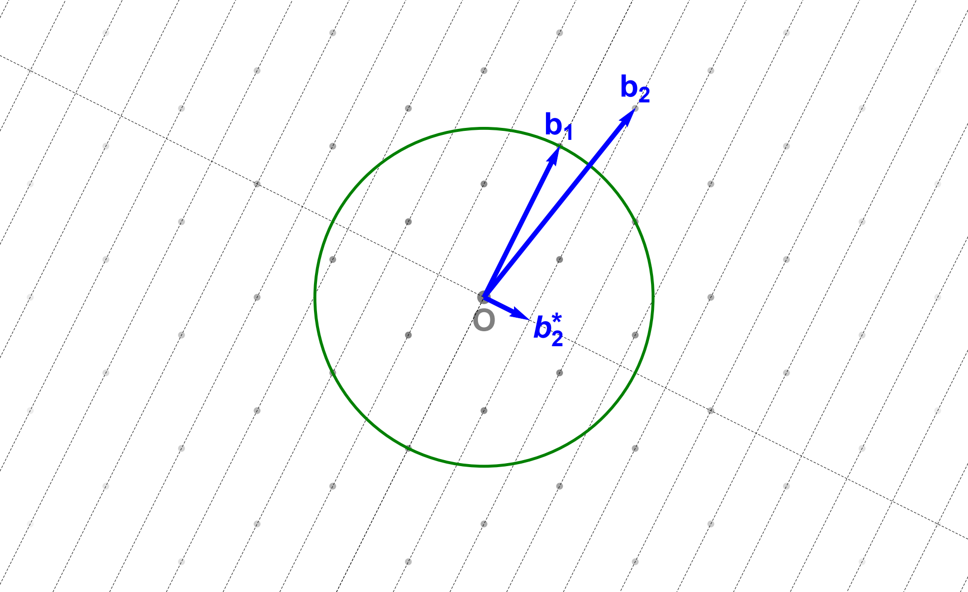
Enumeration

Determine possible coefficients of b_2



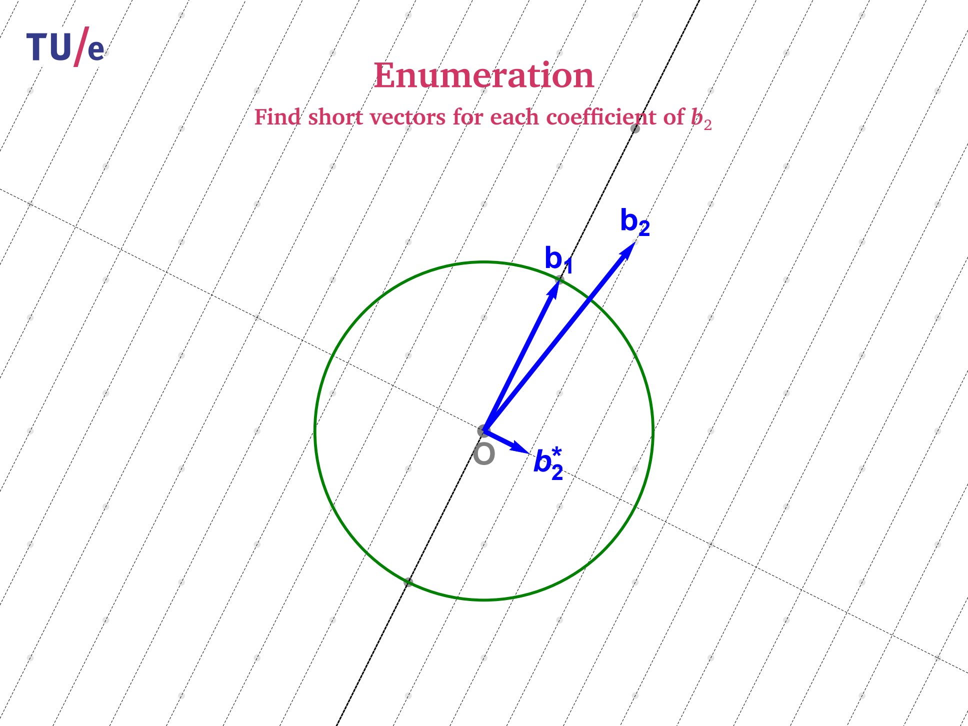
Enumeration

Find short vectors for each coefficient of b_2



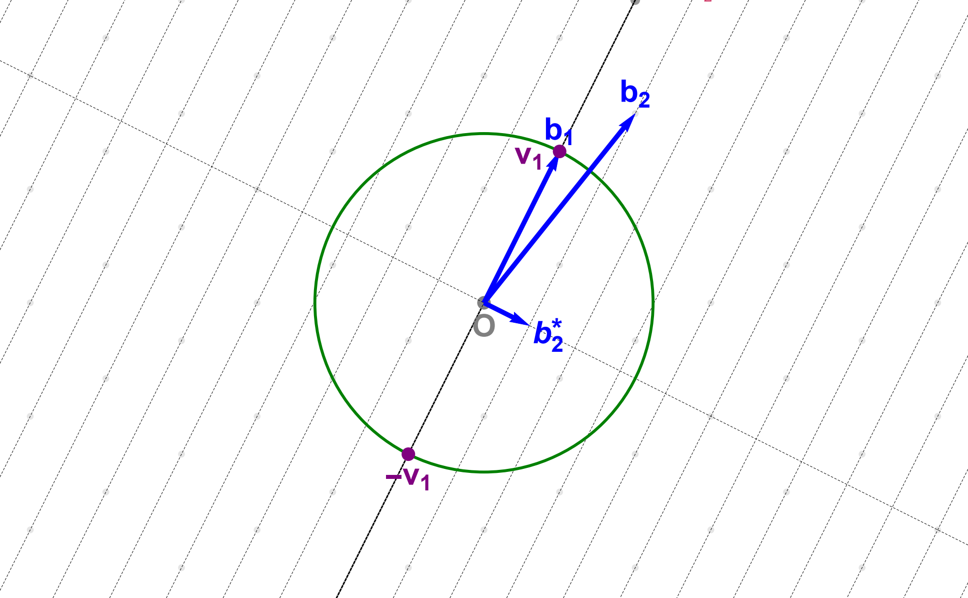
Enumeration

Find short vectors for each coefficient of b_2



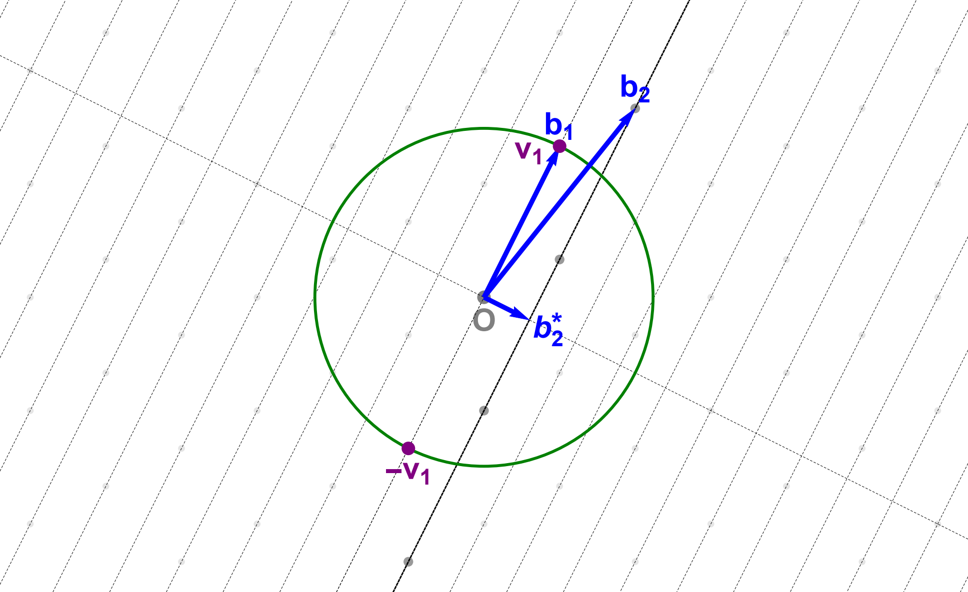
Enumeration

Find short vectors for each coefficient of b_2



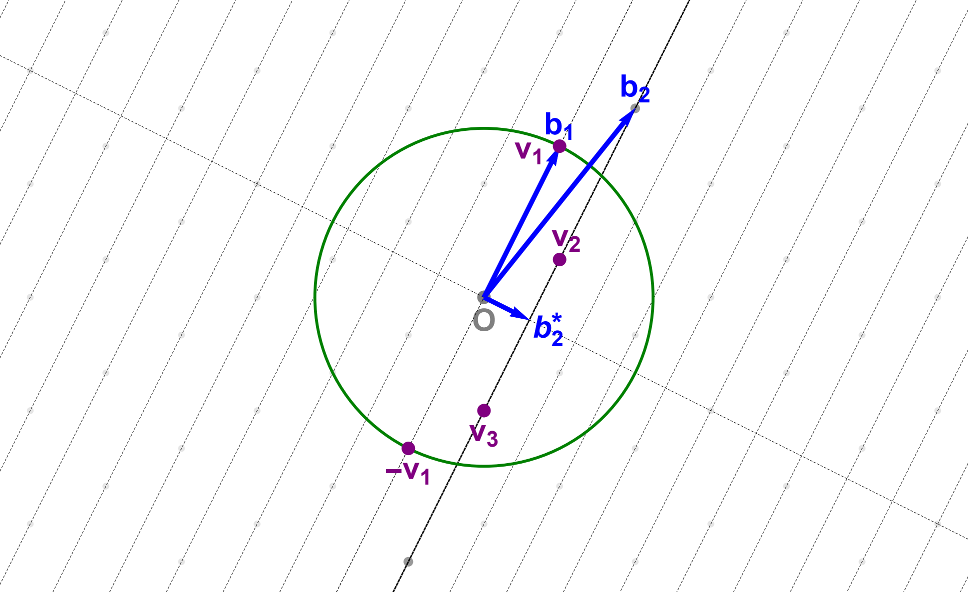
Enumeration

Find short vectors for each coefficient of b_2



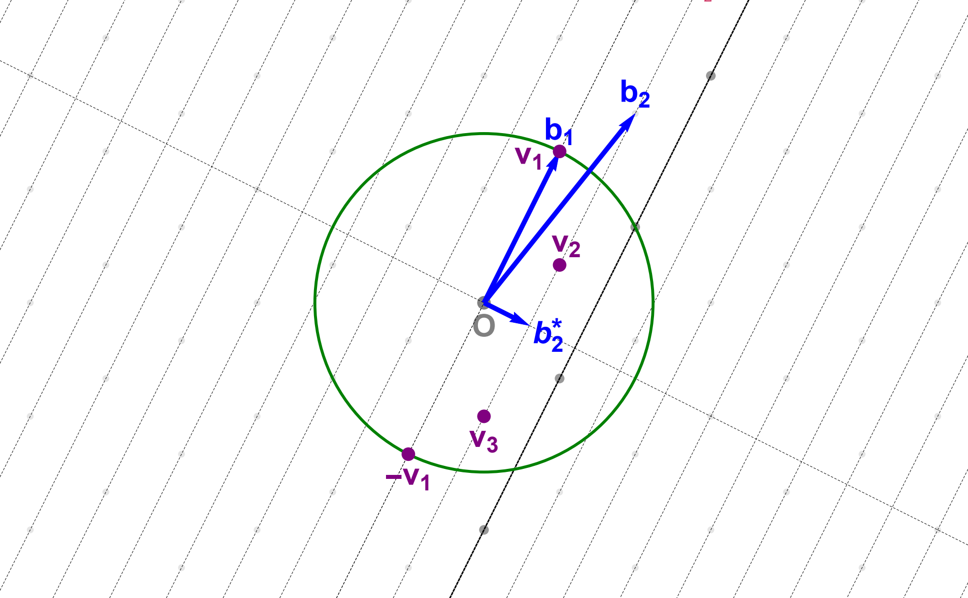
Enumeration

Find short vectors for each coefficient of b_2



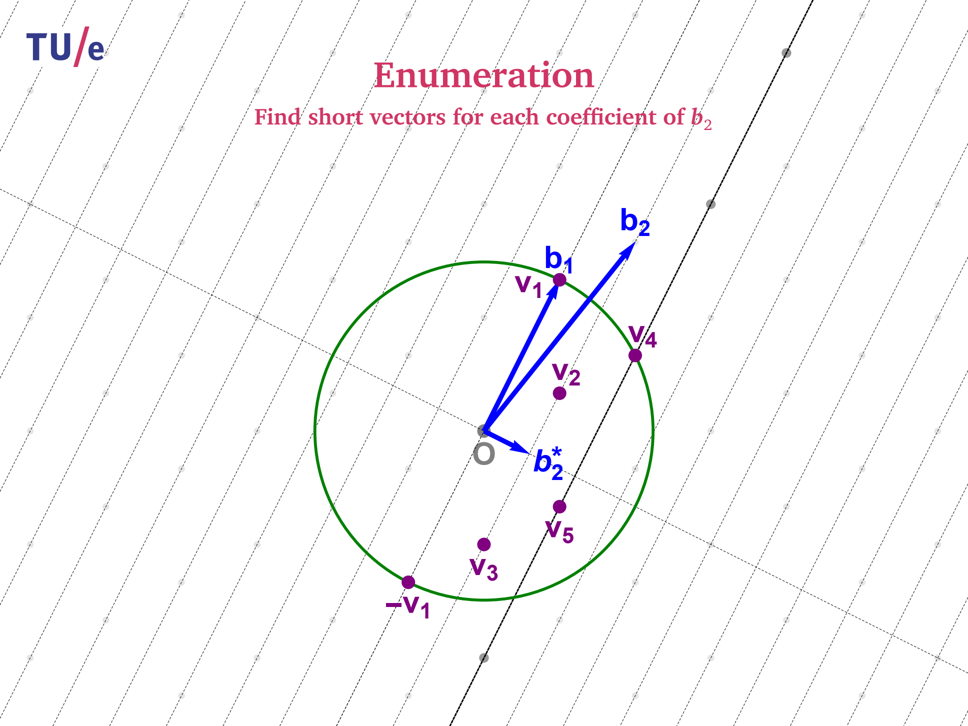
Enumeration

Find short vectors for each coefficient of b_2



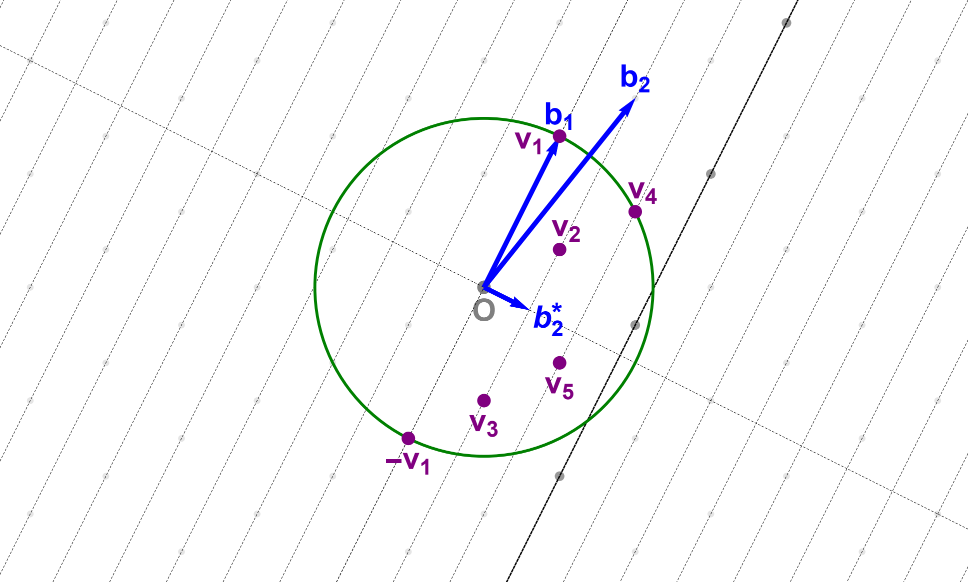
Enumeration

Find short vectors for each coefficient of b_2



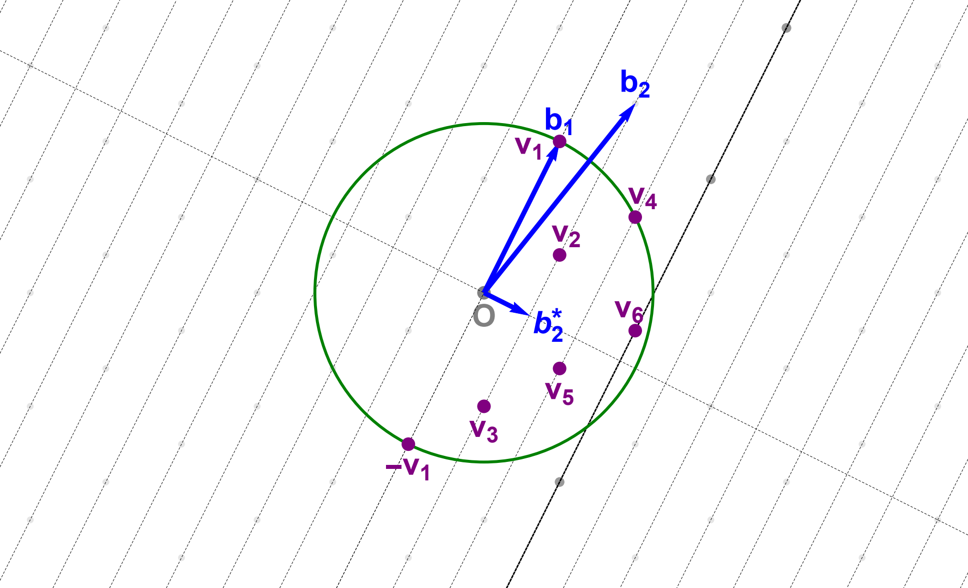
Enumeration

Find short vectors for each coefficient of b_2



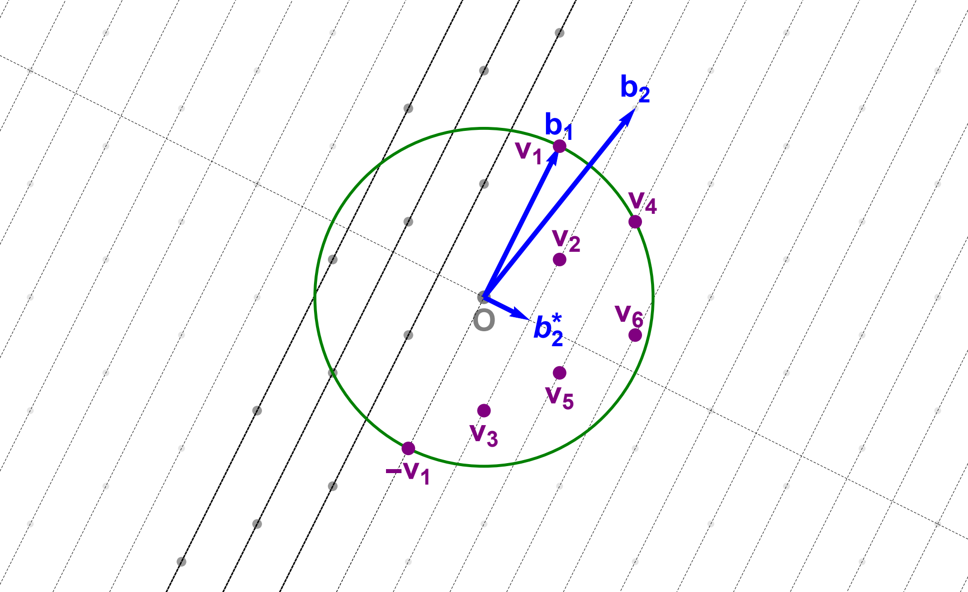
Enumeration

Find short vectors for each coefficient of b_2



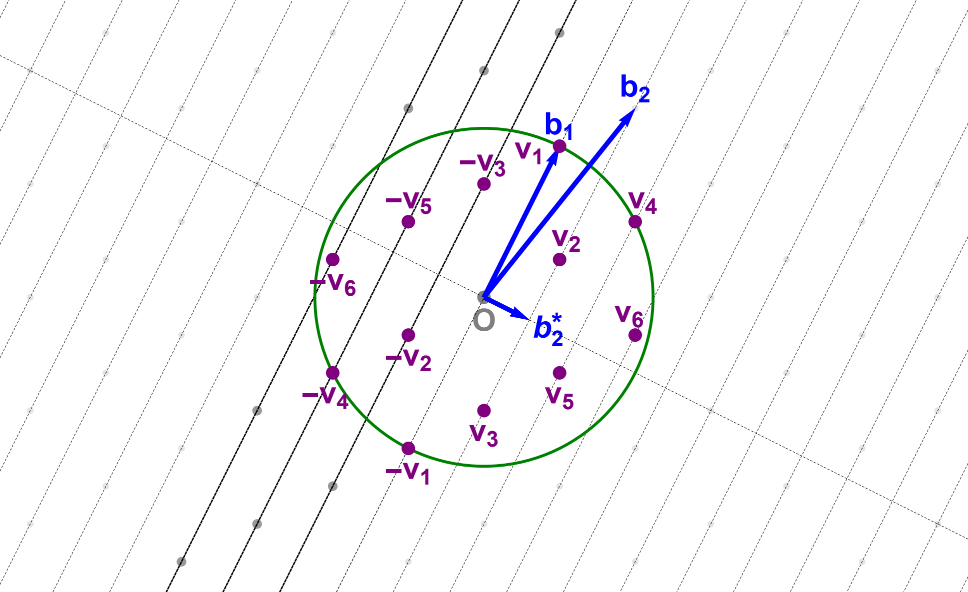
Enumeration

Find short vectors for each coefficient of b_2



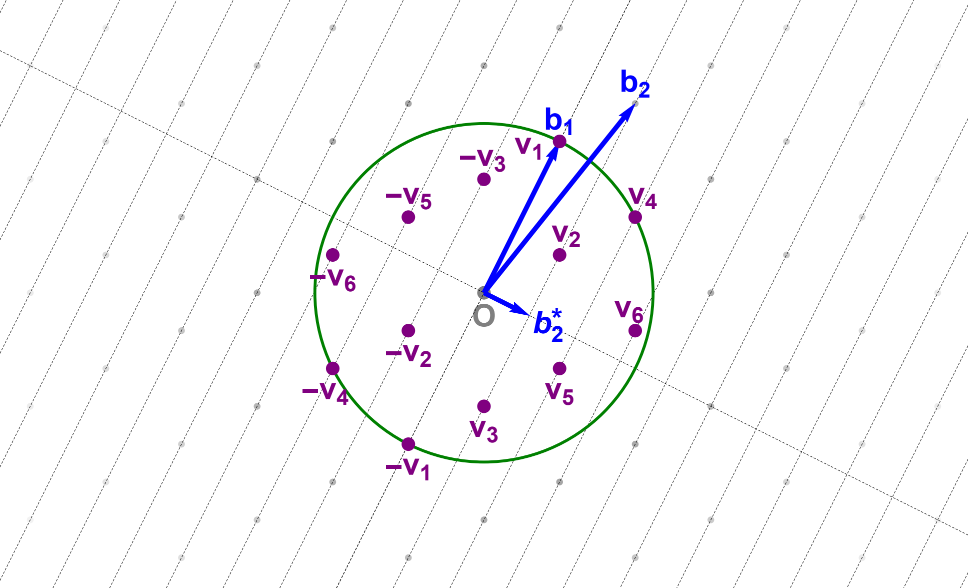
Enumeration

Find short vectors for each coefficient of b_2



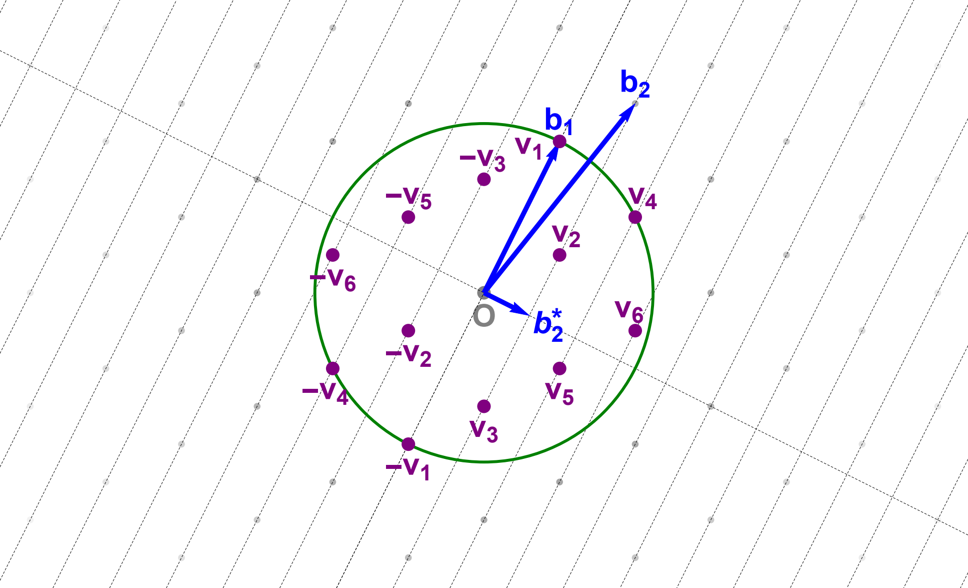
Enumeration

Find short vectors for each coefficient of b_2



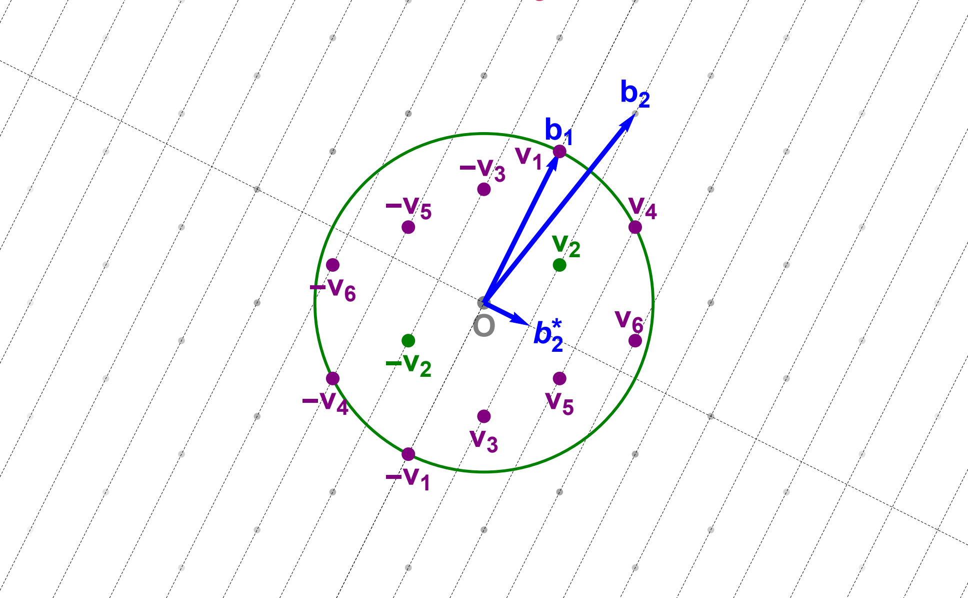
Enumeration

Find a shortest vector among all found vectors



Enumeration

Find a shortest vector among all found vectors



Enumeration

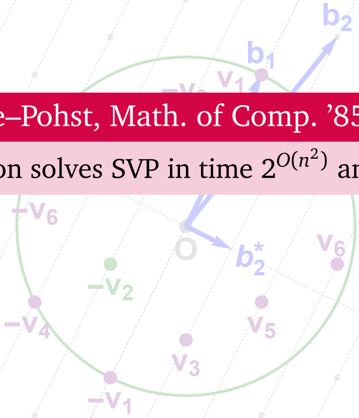
Overview

Enumeration

Overview

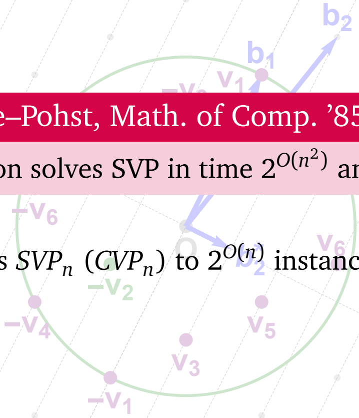
Theorem (Fincke–Pohst, Math. of Comp. '85)

Lattice enumeration solves SVP in time $2^{O(n^2)}$ and space poly(n).



Enumeration

Overview



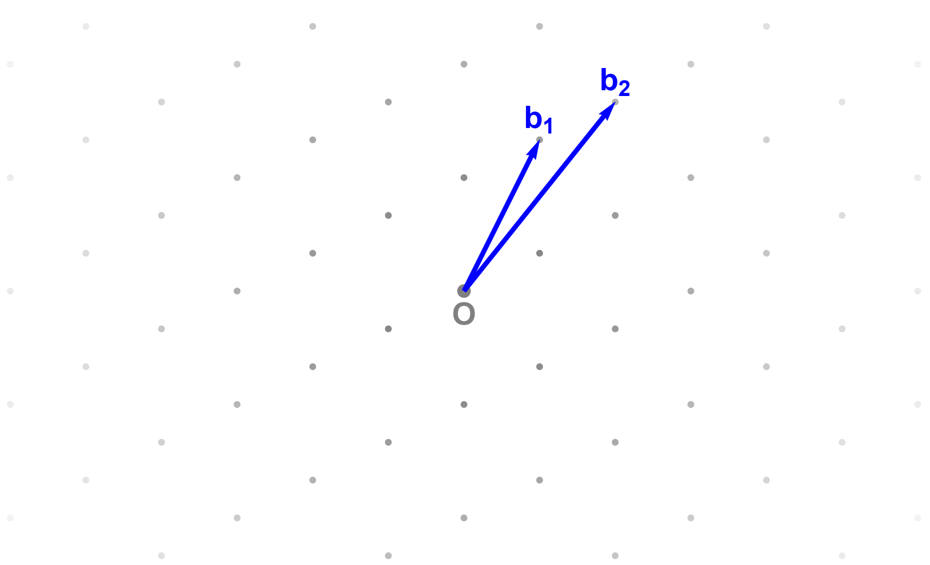
Theorem (Fincke–Pohst, Math. of Comp. ’85)

Lattice enumeration solves SVP in time $2^{O(n^2)}$ and space poly(n).

Essentially reduces SVP_n (CVP_n) to $2^{O(n)}$ instances of CVP_{n-1}.

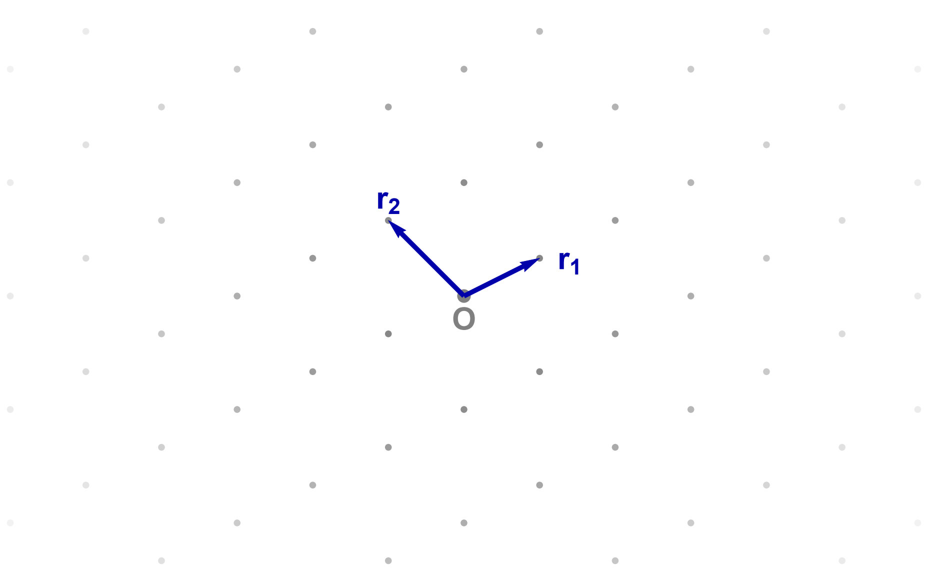
Enumeration

Better bases



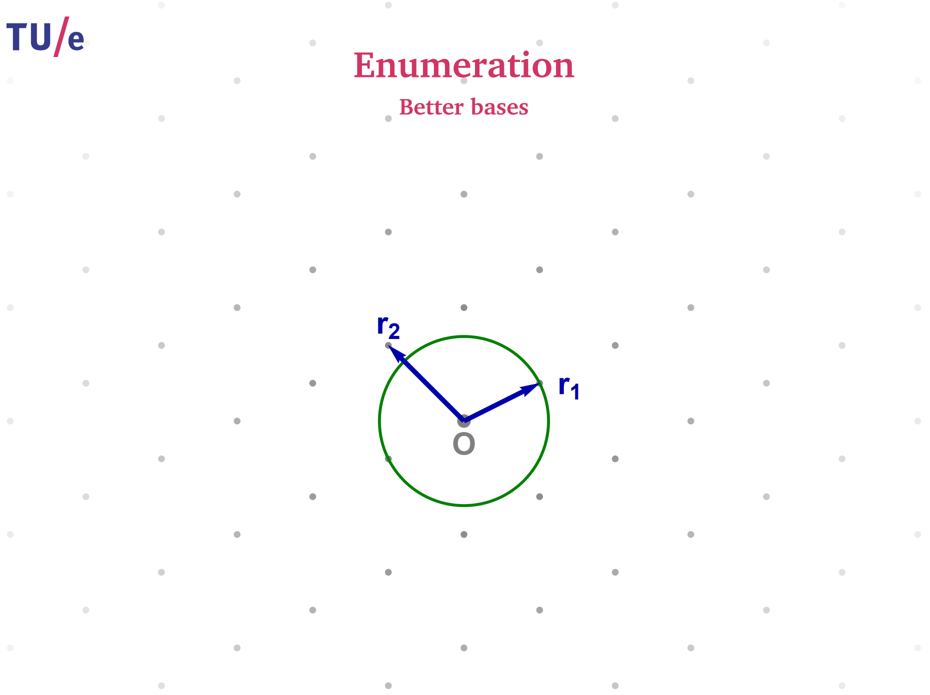
Enumeration

Better bases



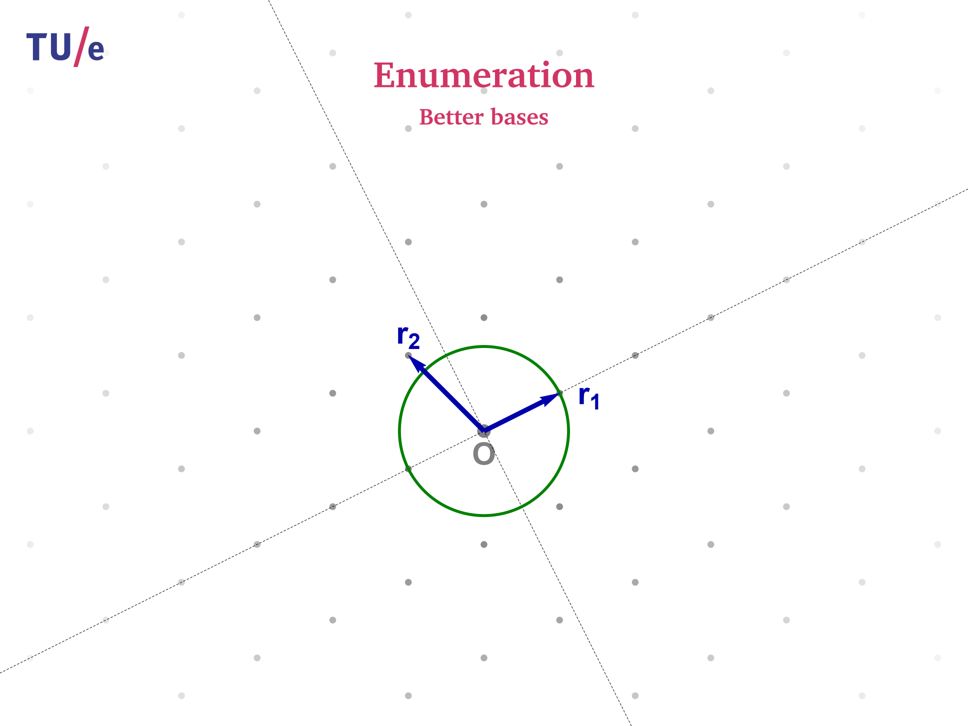
Enumeration

Better bases



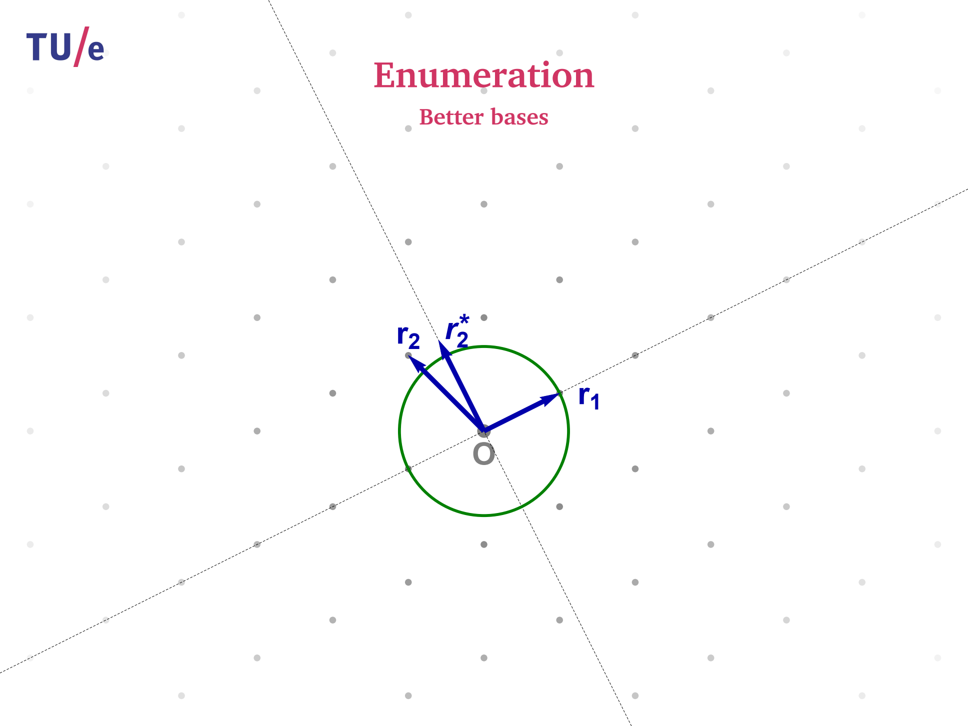
Enumeration

Better bases



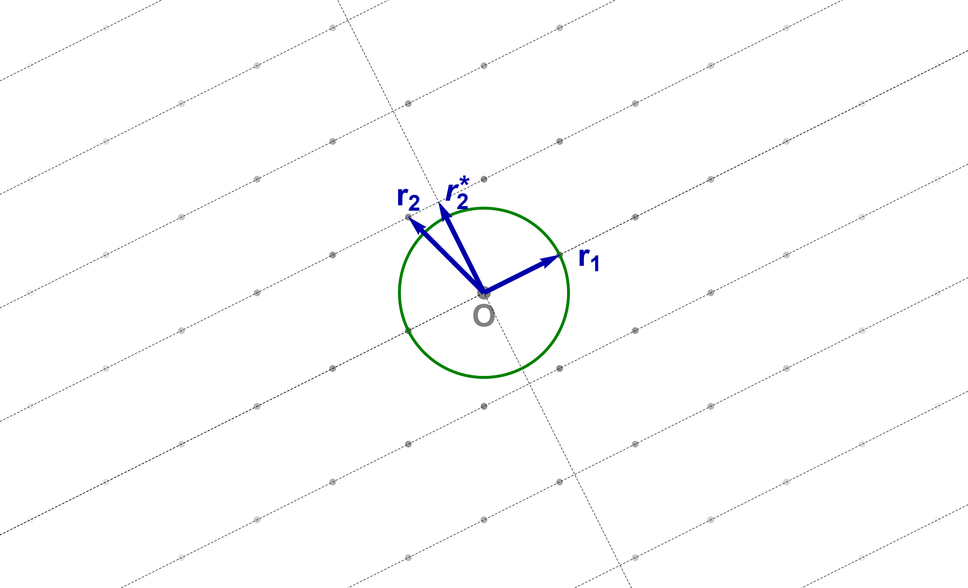
Enumeration

Better bases



Enumeration

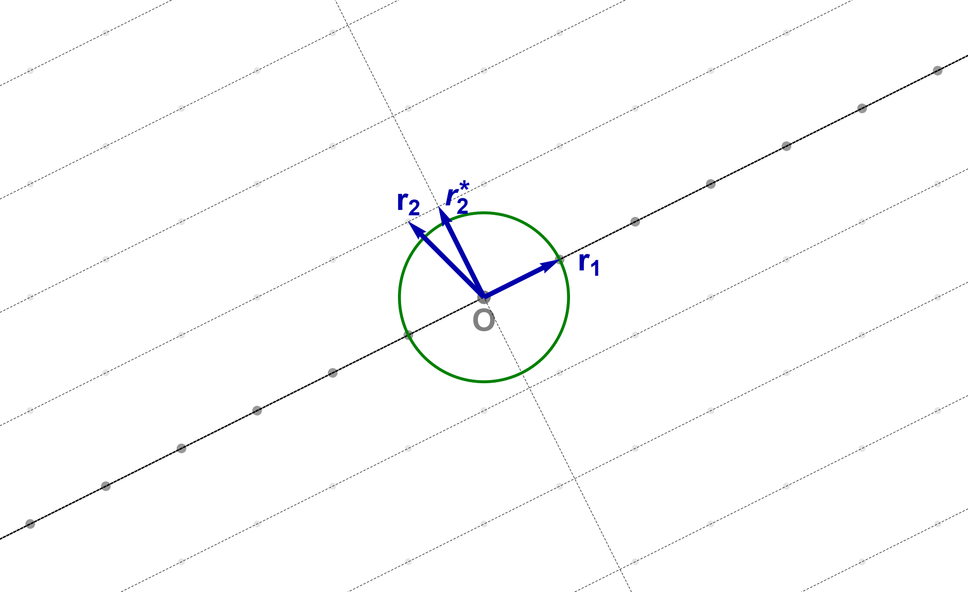
Better bases



TU/e

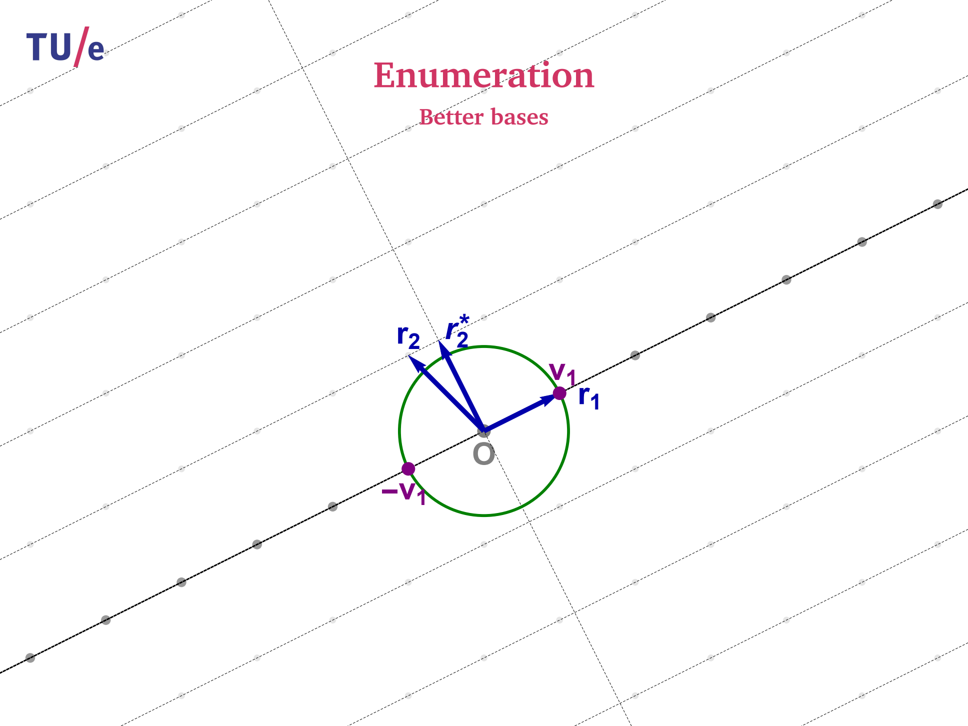
Enumeration

Better bases



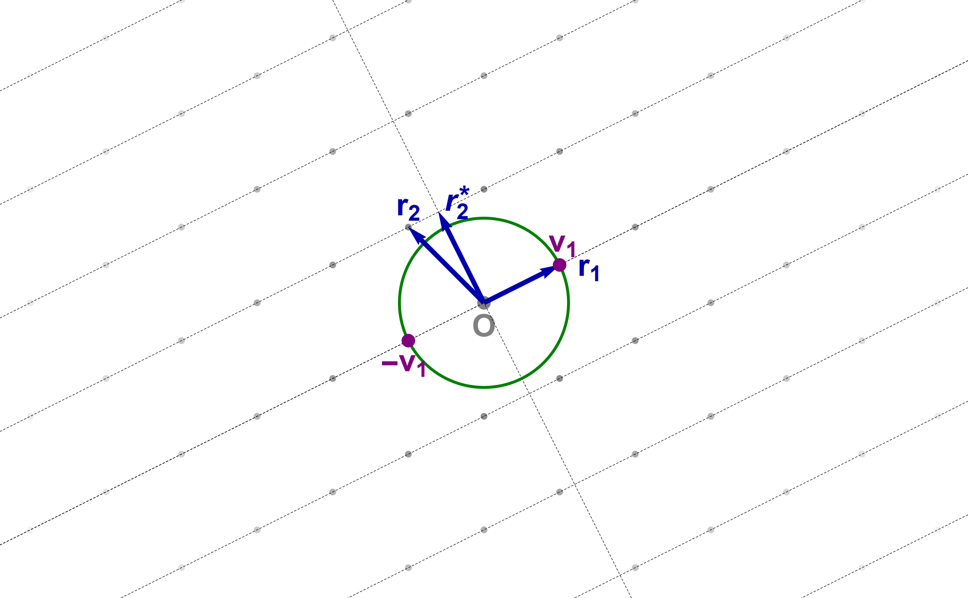
Enumeration

Better bases



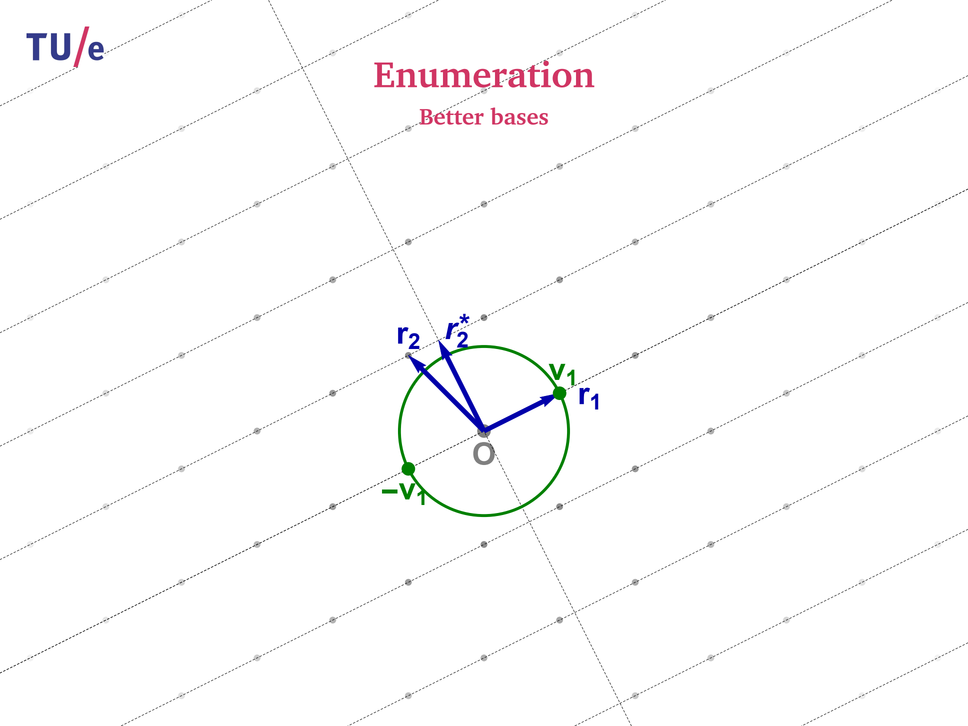
Enumeration

Better bases



Enumeration

Better bases



Enumeration

Better bases

Enumeration

Better bases

Theorem (Kannan, STOC'83)

Combining enumeration with stronger basis reduction, one can solve SVP in time $2^{O(n \log n)}$ and space $\mathrm{poly}(n)$.

Enumeration

Better bases

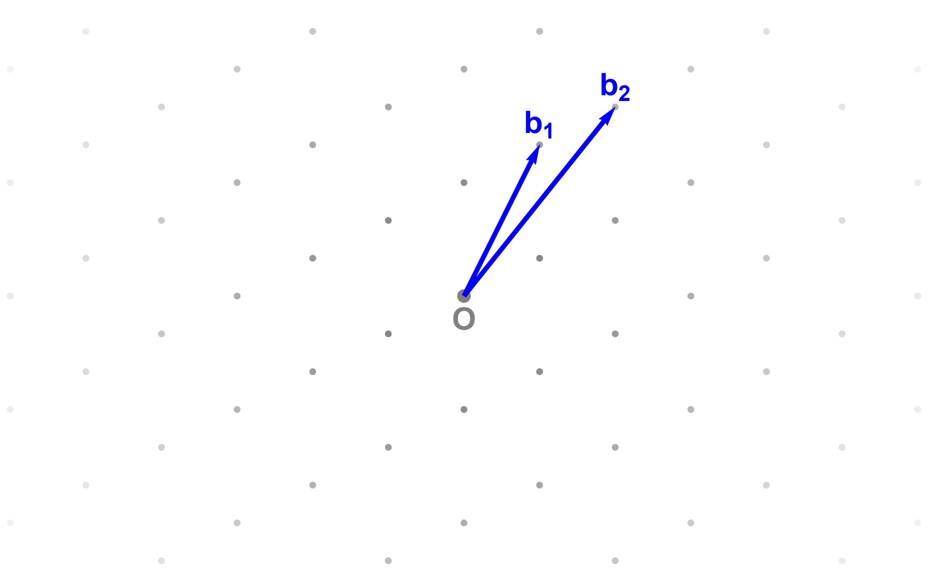
Theorem (Kannan, STOC'83)

Combining enumeration with stronger basis reduction, one can solve SVP in time $2^{O(n \log n)}$ and space $\mathrm{poly}(n)$.

> *"Our algorithm reduces an n-dimensional problem to polynomially many (instead of $2^{O(n)}$) $(n-1)$-dimensional problems. [...] The algorithm we propose, first finds a more orthogonal basis for a lattice in time $2^{O(n \log n)}$."*
>
> – Kannan, *STOC'83*

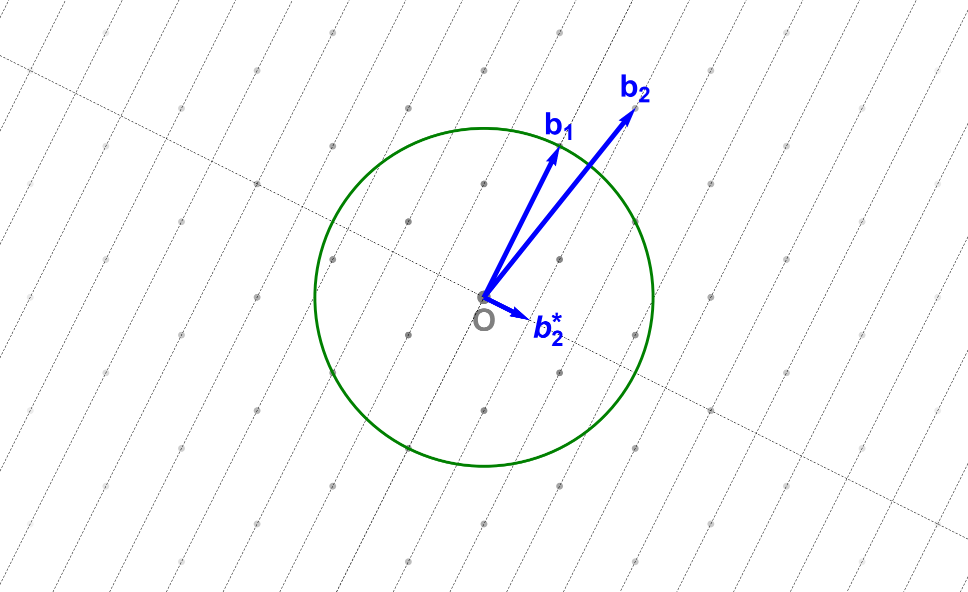
Enumeration

Pruning the enumeration tree



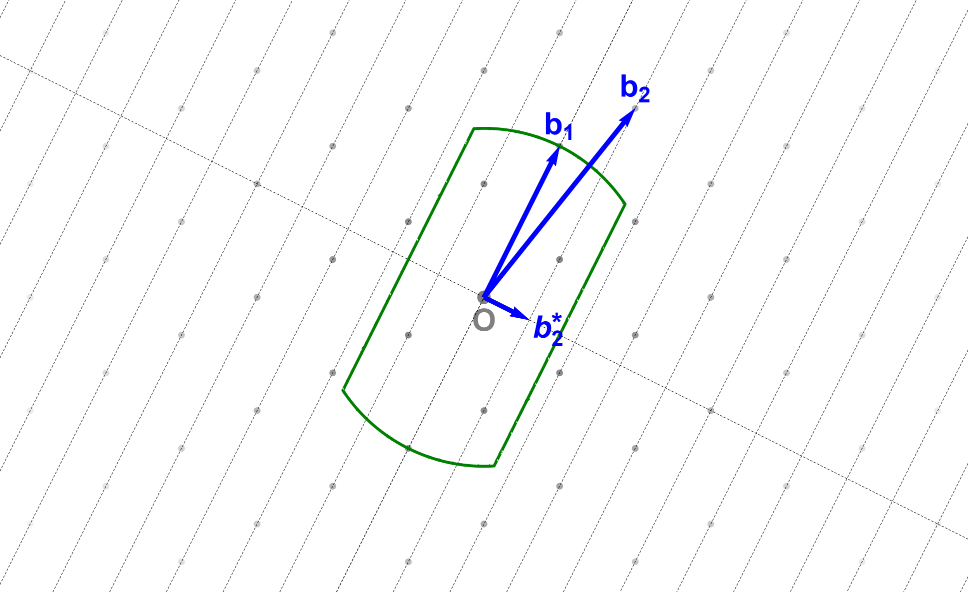
Enumeration

Pruning the enumeration tree



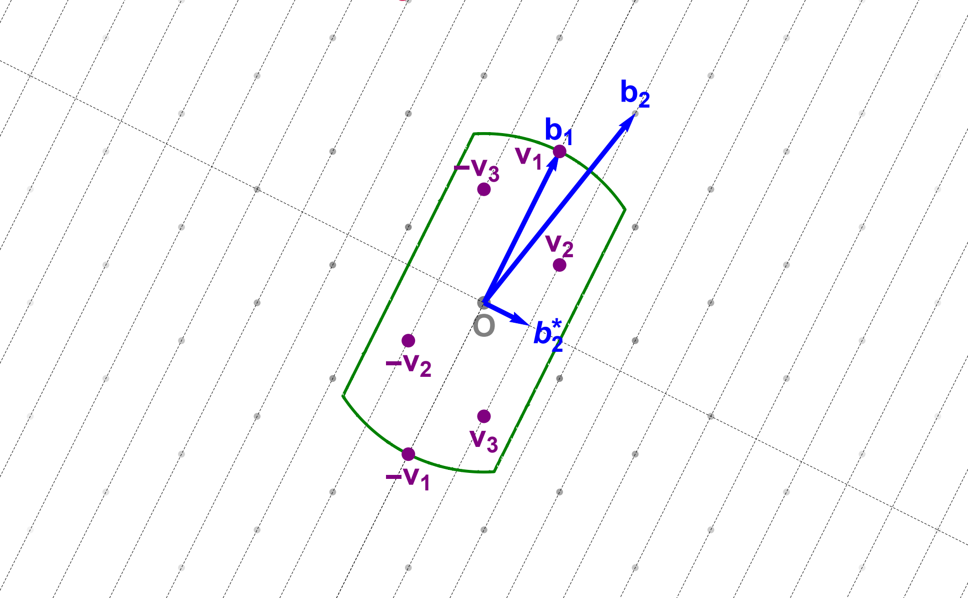
Enumeration

Pruning the enumeration tree



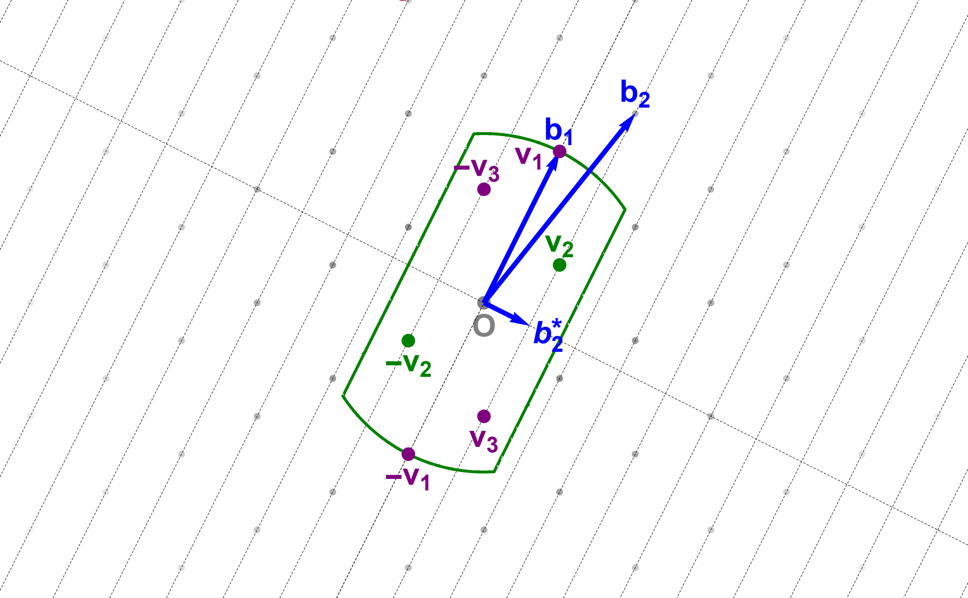
Enumeration

Pruning the enumeration tree



Enumeration

Pruning the enumeration tree



Outline

Lattices

SVP algorithms
- Enumeration
- Sieving

SVP hardness
- Theory
- Practice

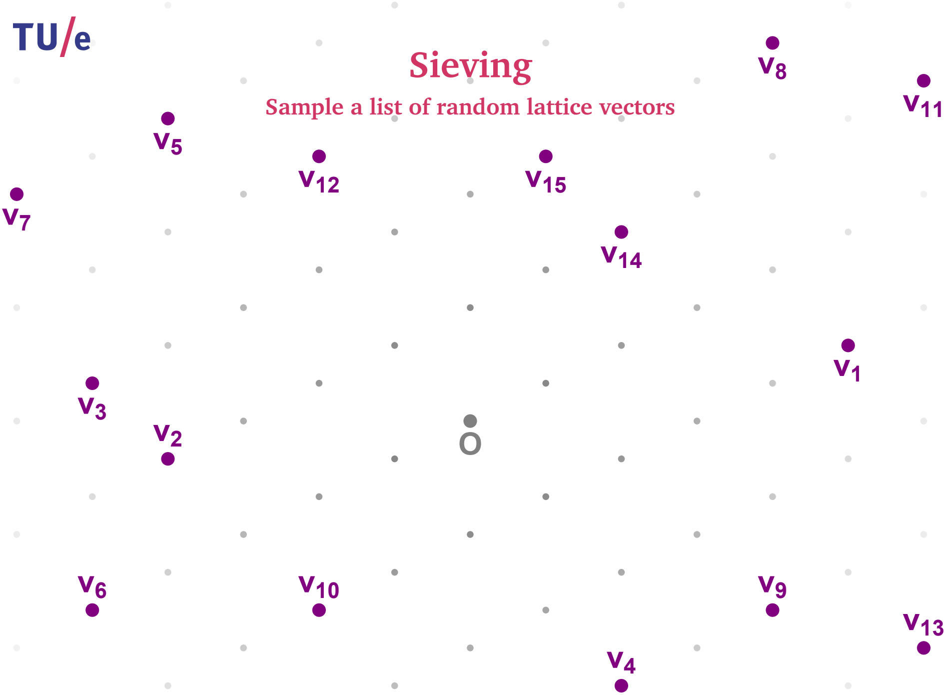
Conclusion

Sieving

Sample a list of random lattice vectors

Sieving

Sample a list of random lattice vectors

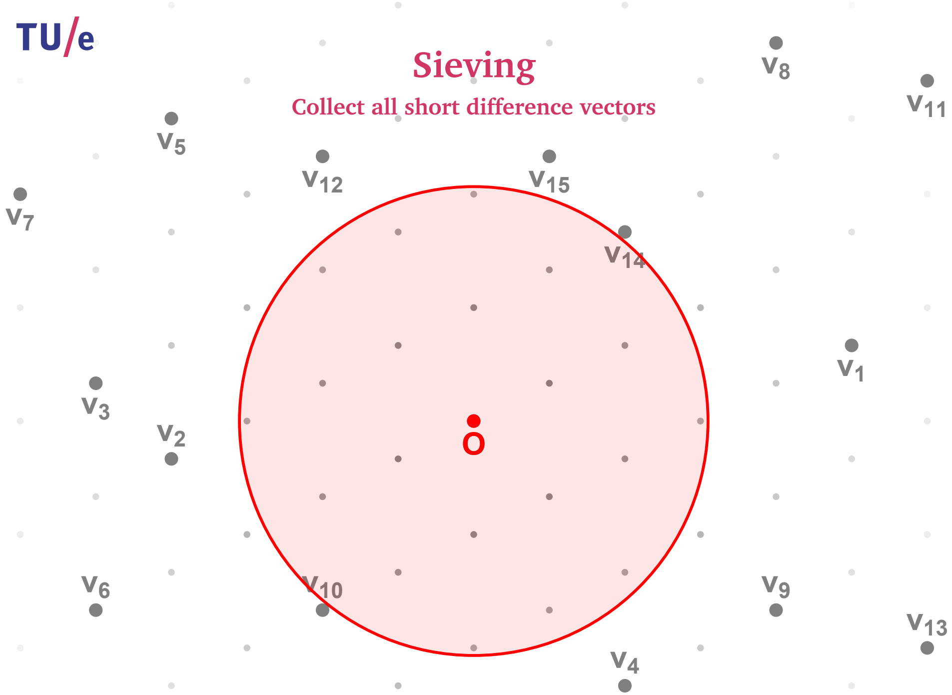


Sieving

Collect all short difference vectors

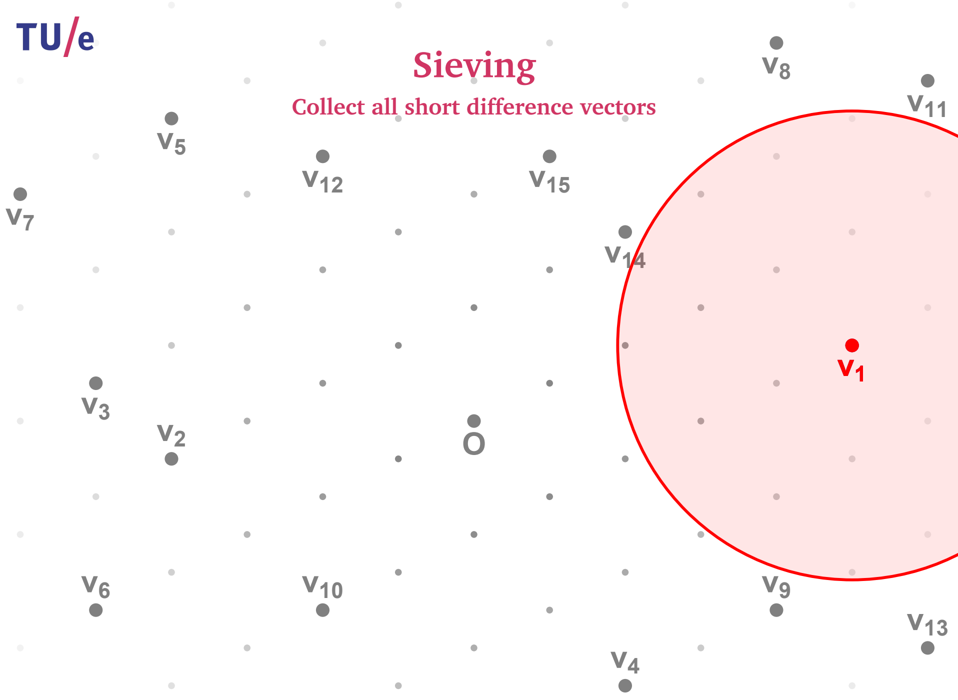
Sieving

Collect all short difference vectors



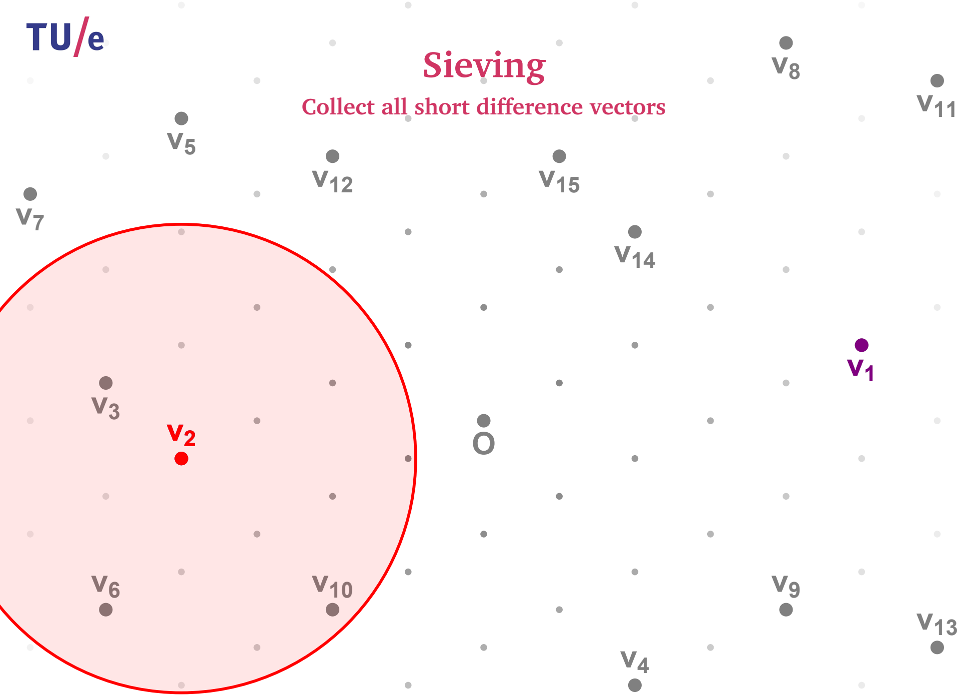
Sieving

Collect all short difference vectors



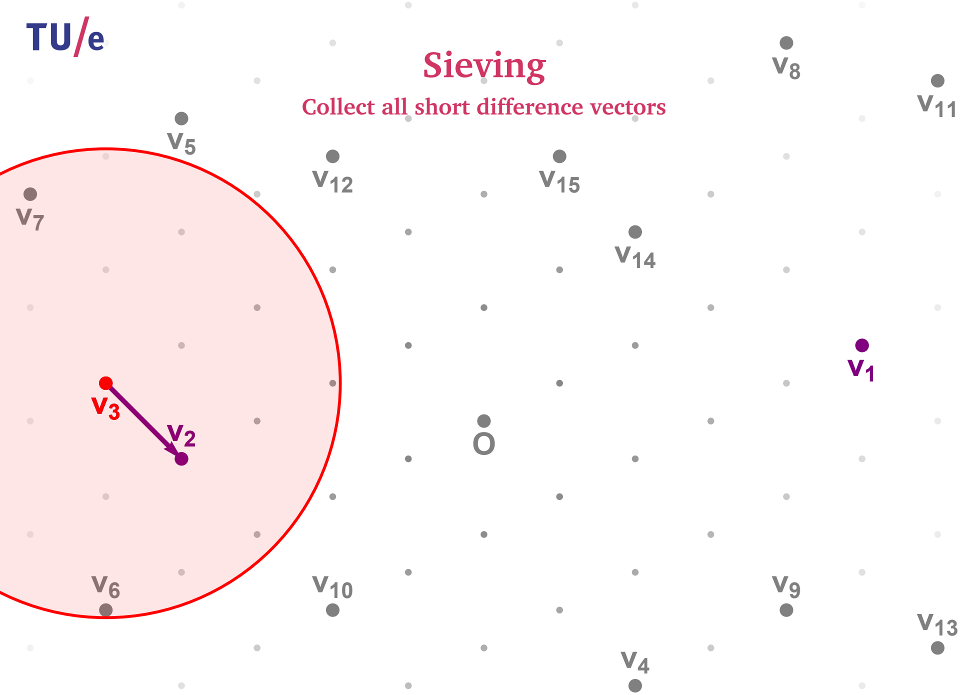
Sieving

Collect all short difference vectors



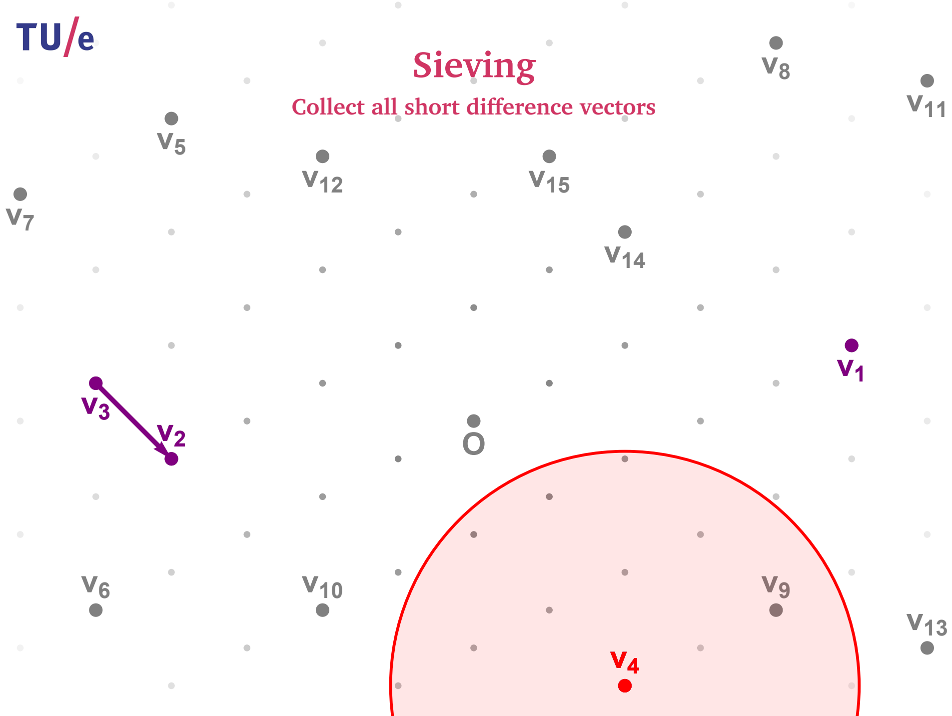
Sieving

Collect all short difference vectors



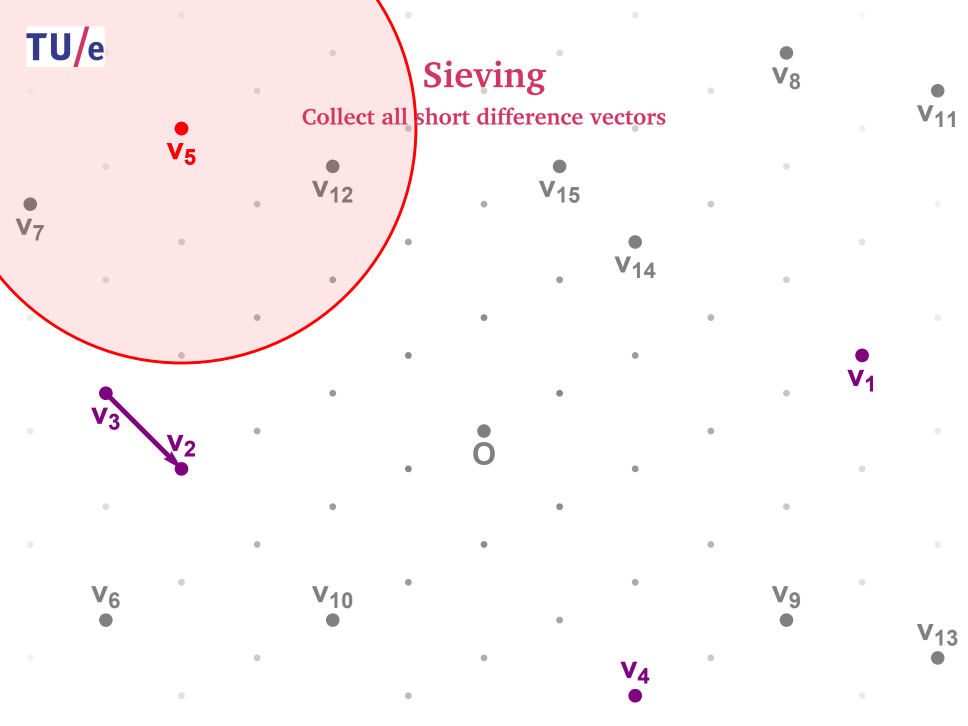
Sieving

Collect all short difference vectors



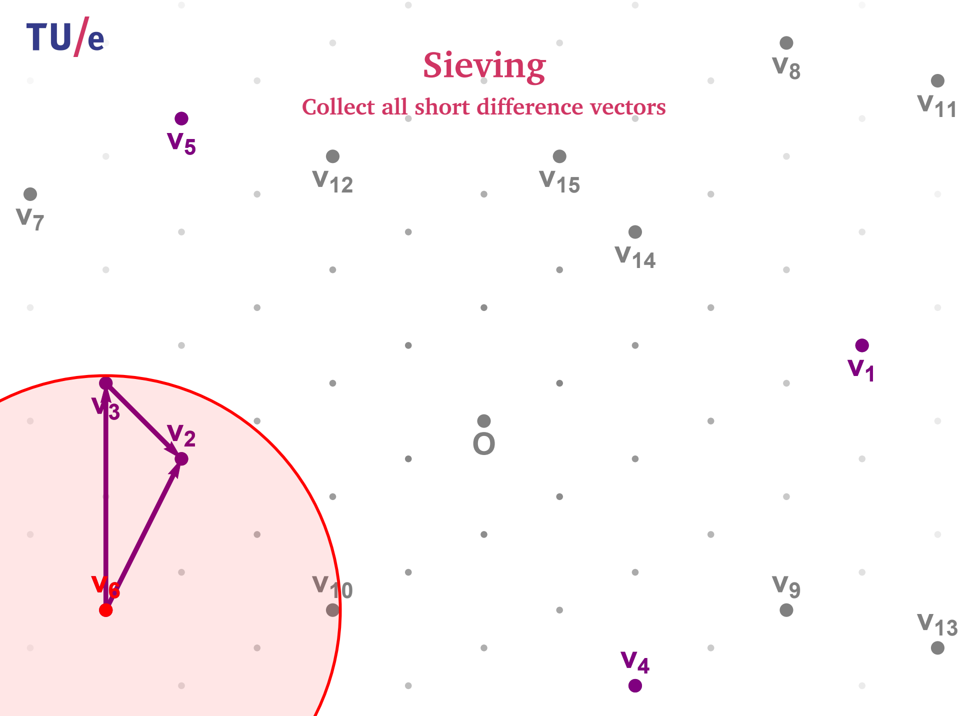
Sieving

Collect all short difference vectors



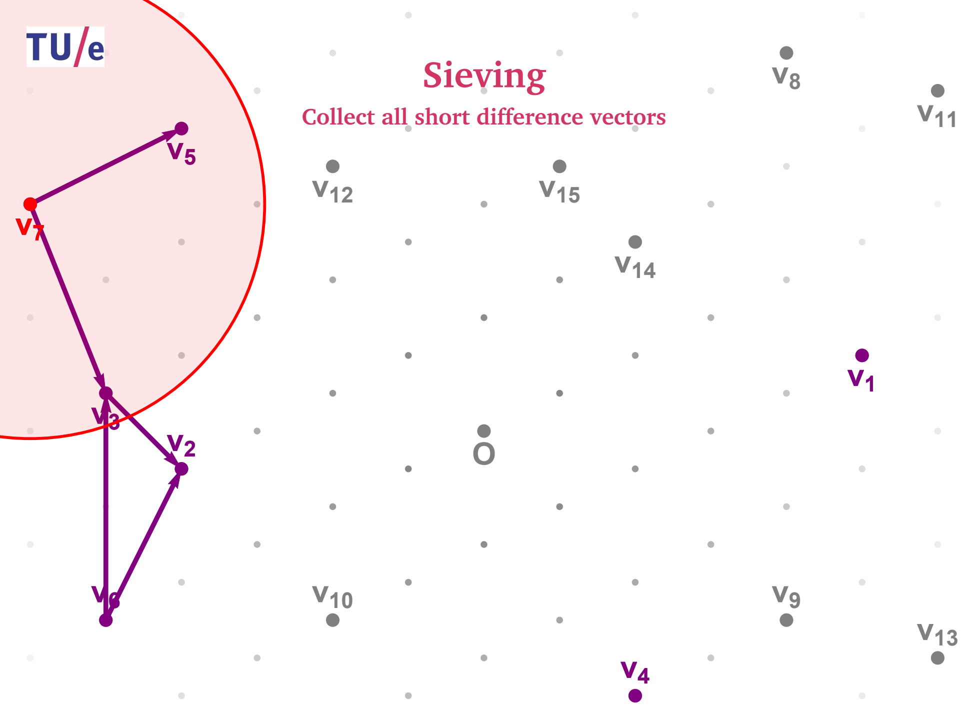
Sieving

Collect all short difference vectors



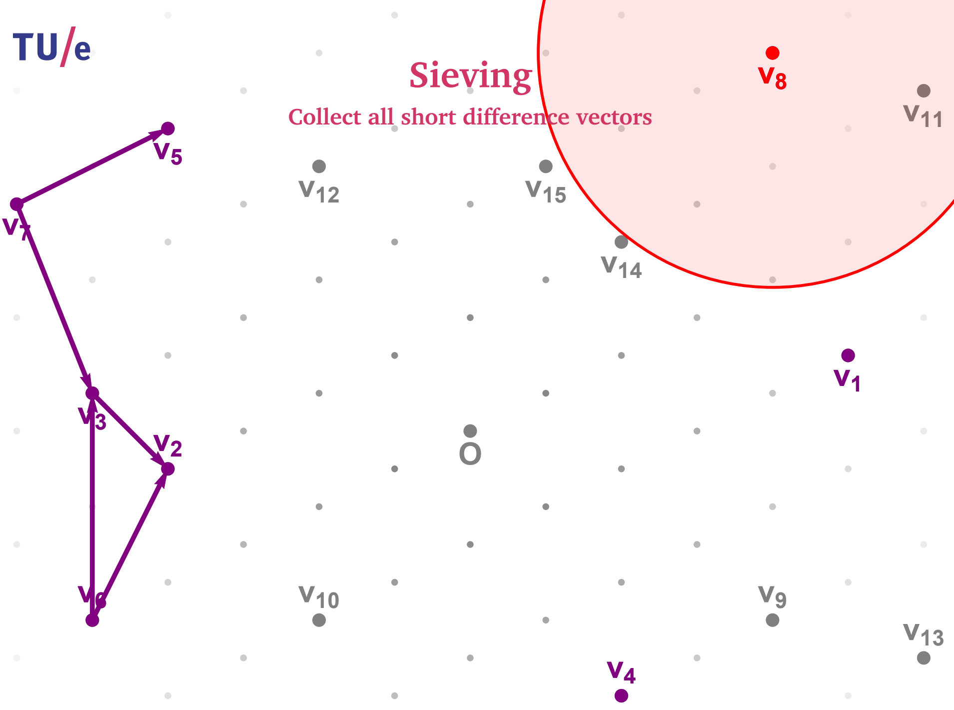
Sieving

Collect all short difference vectors



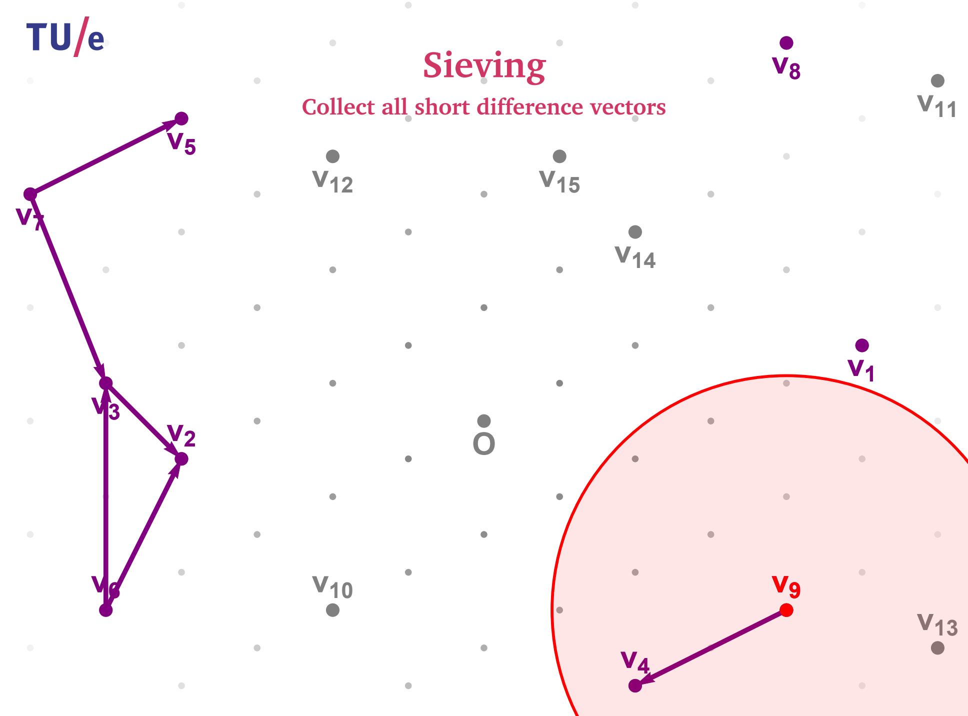
Sieving

Collect all short difference vectors



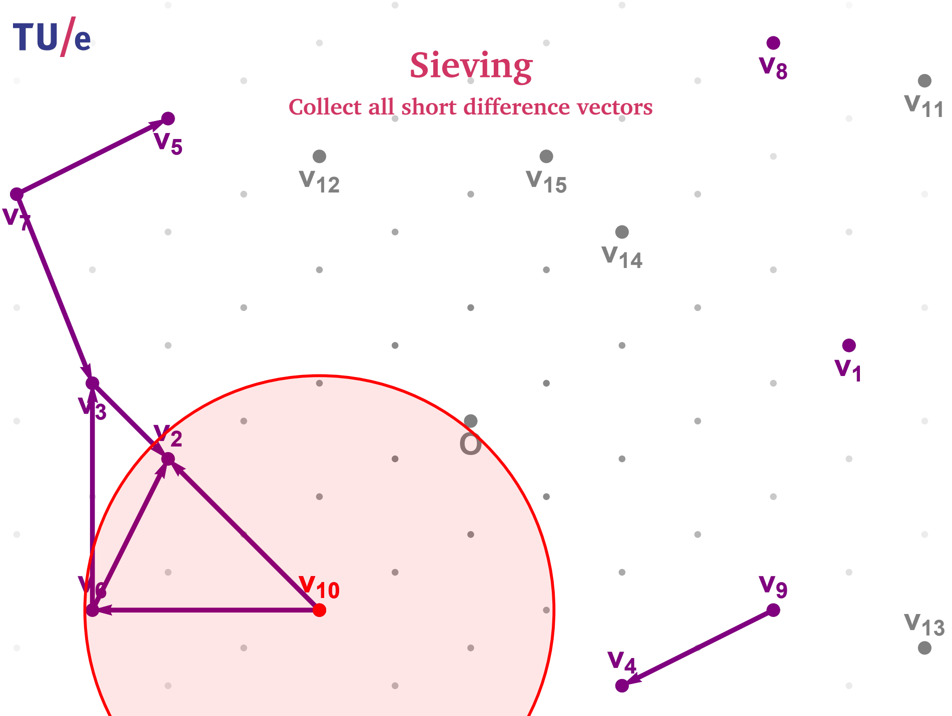
Sieving

Collect all short difference vectors



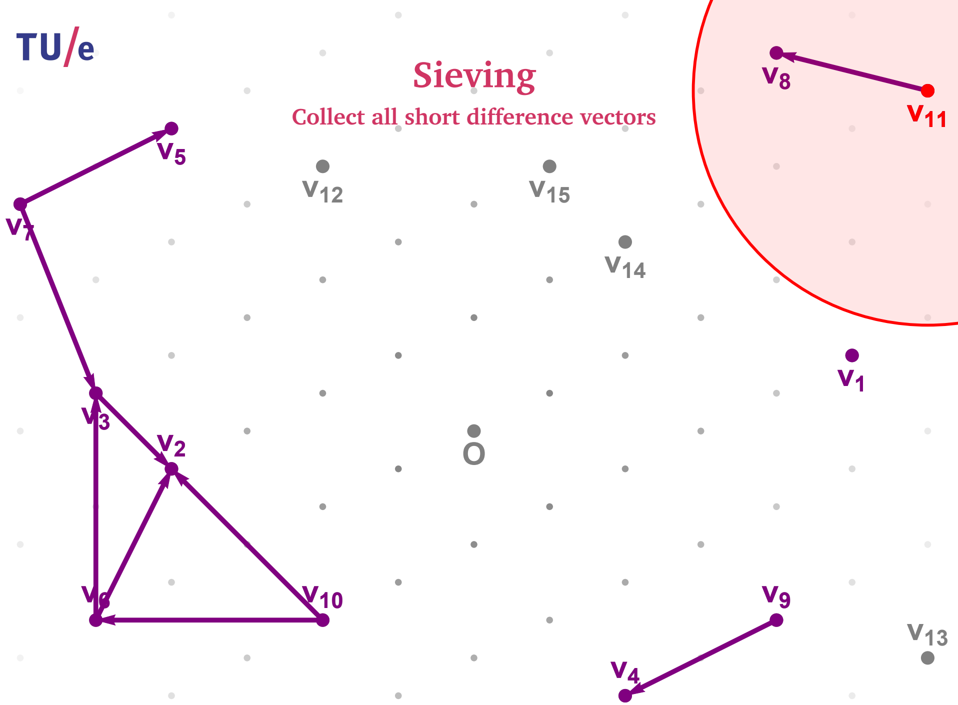
Sieving

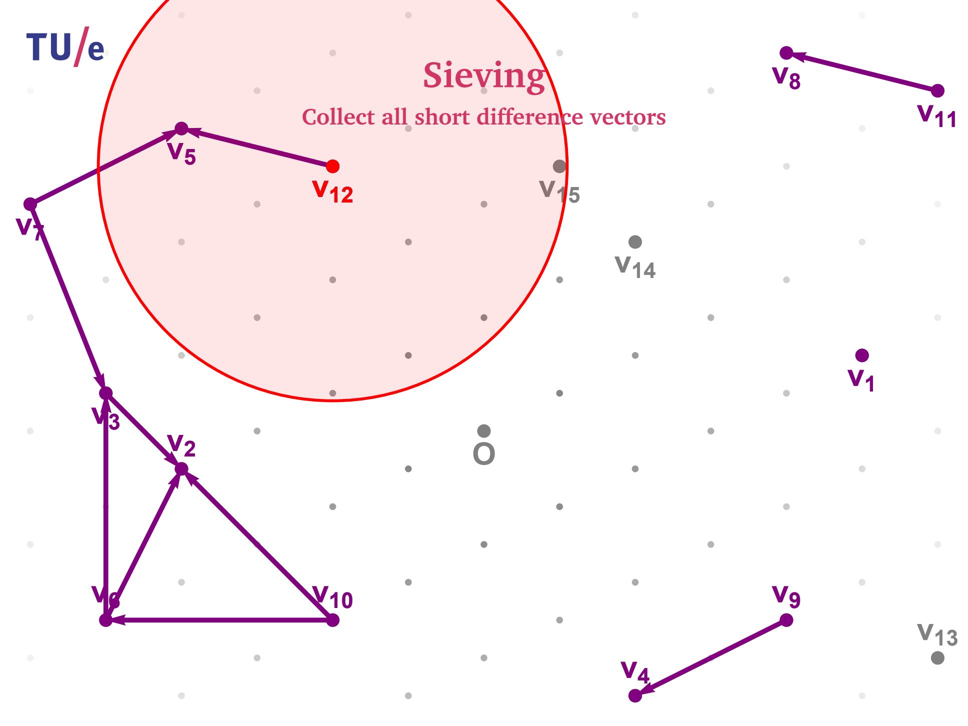
Collect all short difference vectors



Sieving

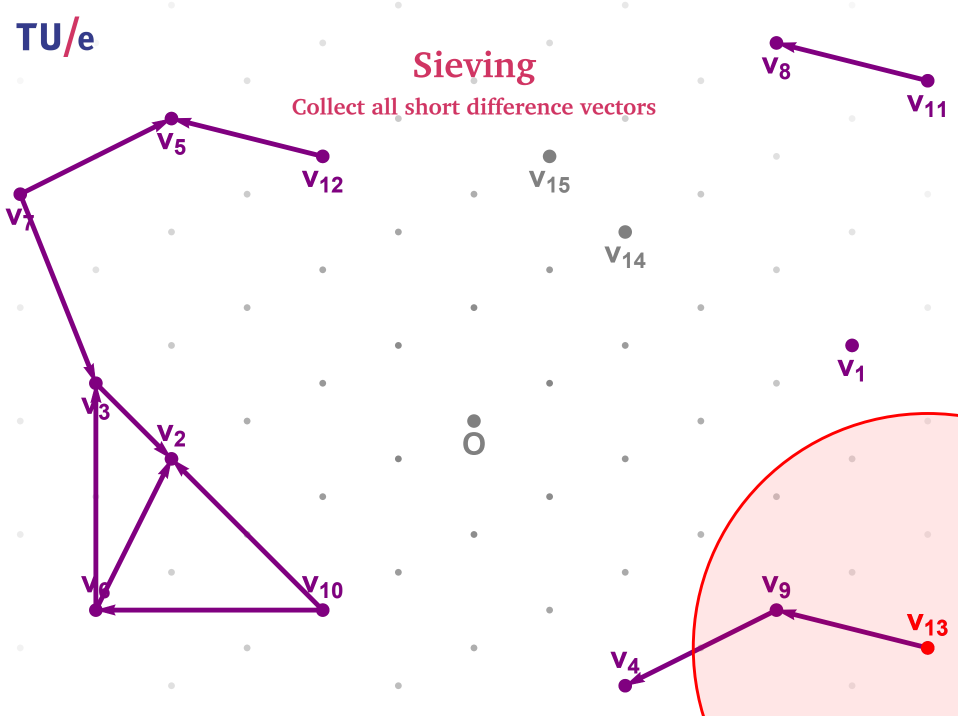
Collect all short difference vectors




TU/e
Sieving
Collect all short difference vectors
v8
v11
v5
v12
v15
v7
v14
v1
v3
v2
O
v6
v10
v9
v13
v4

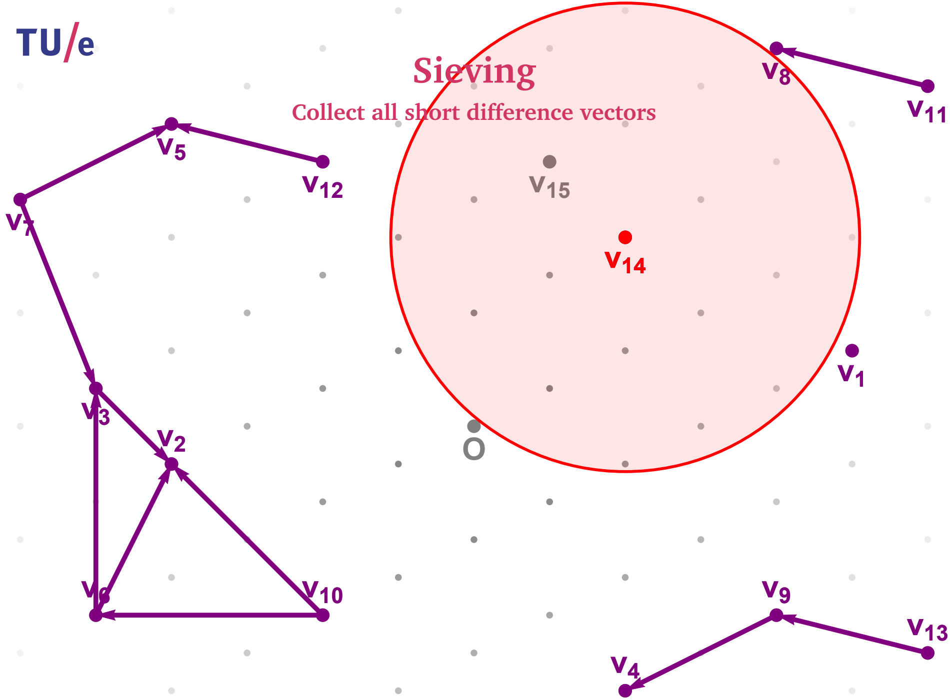
Sieving

Collect all short difference vectors



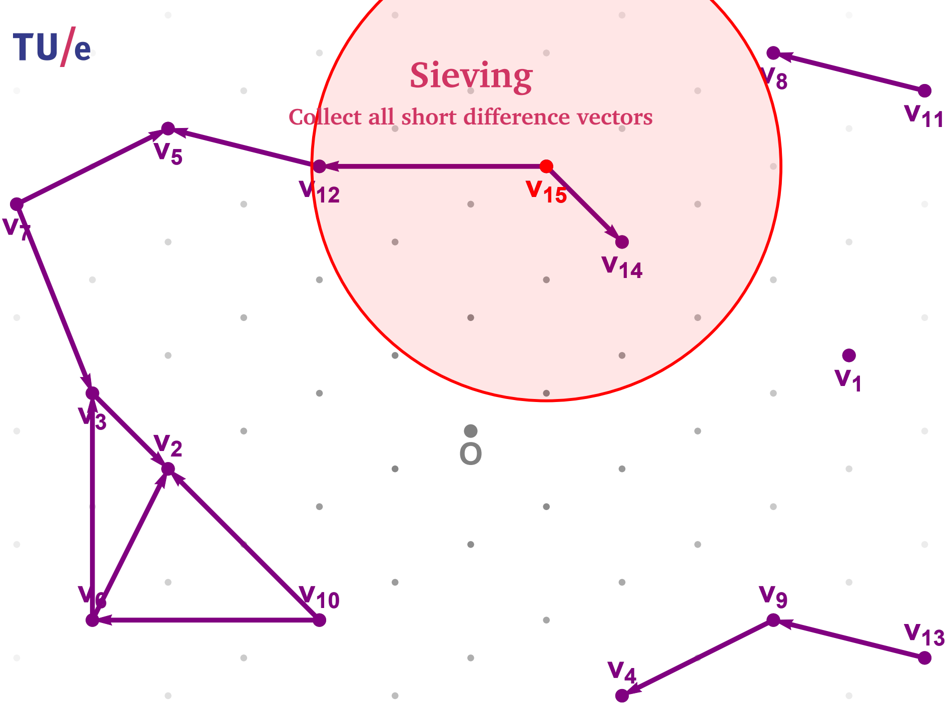
Sieving

Collect all short difference vectors



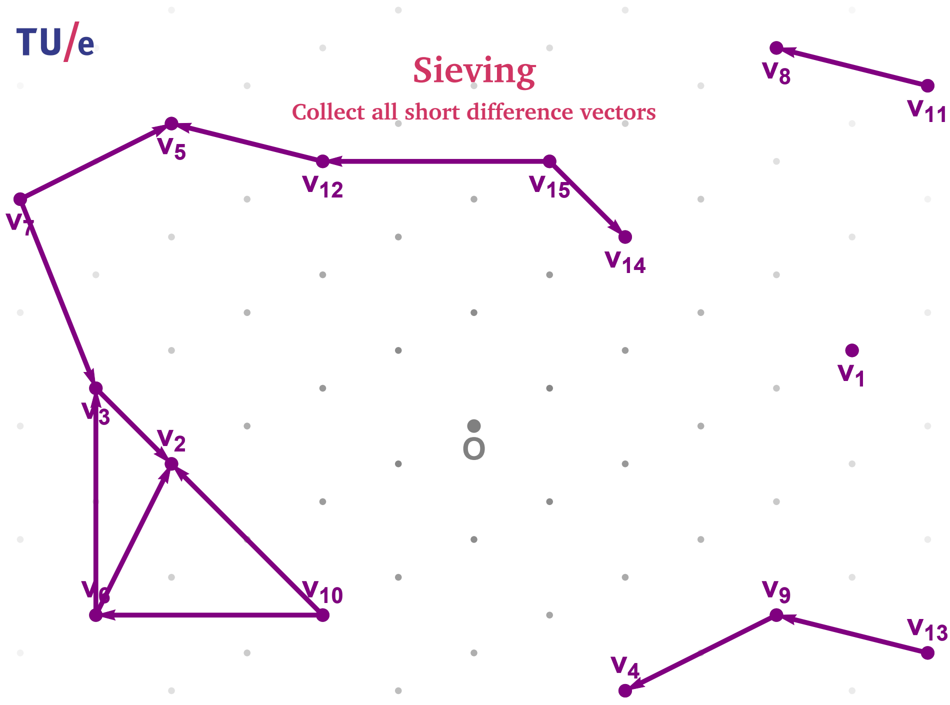
Sieving

Collect all short difference vectors



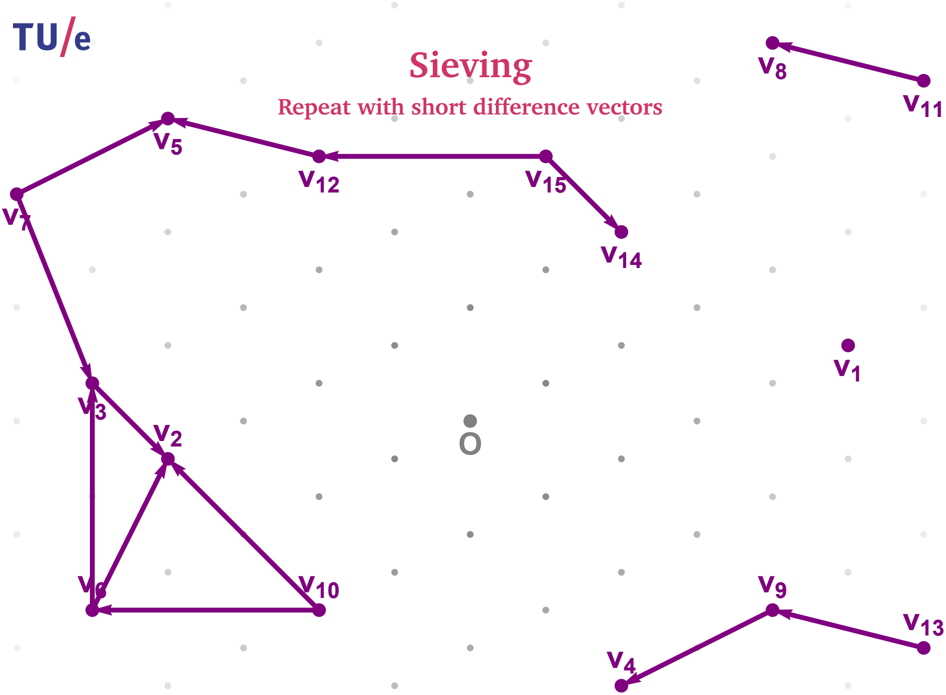
Sieving

Collect all short difference vectors



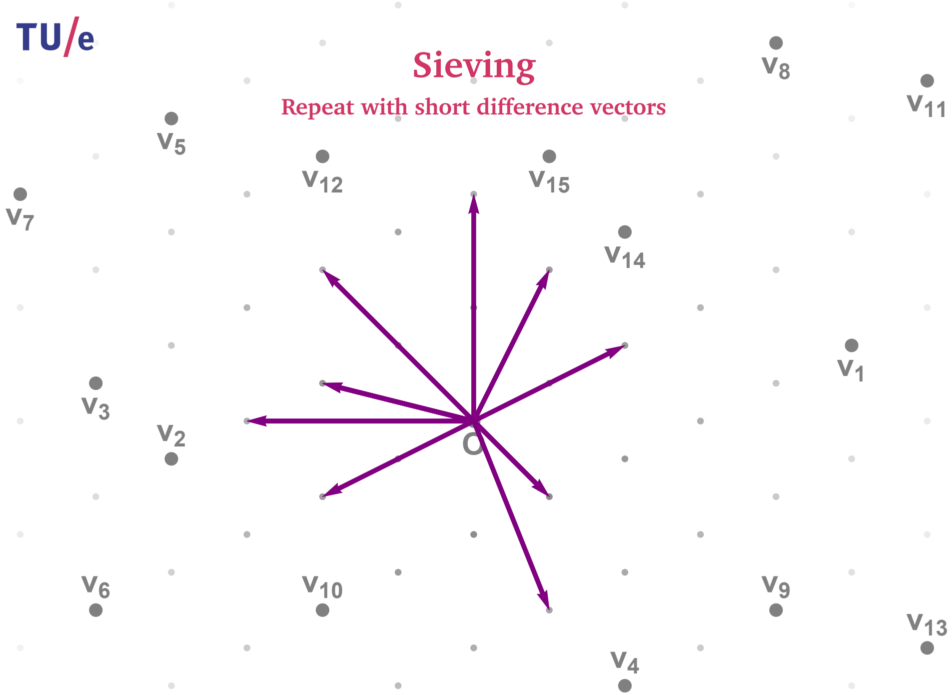
Sieving

Repeat with short difference vectors



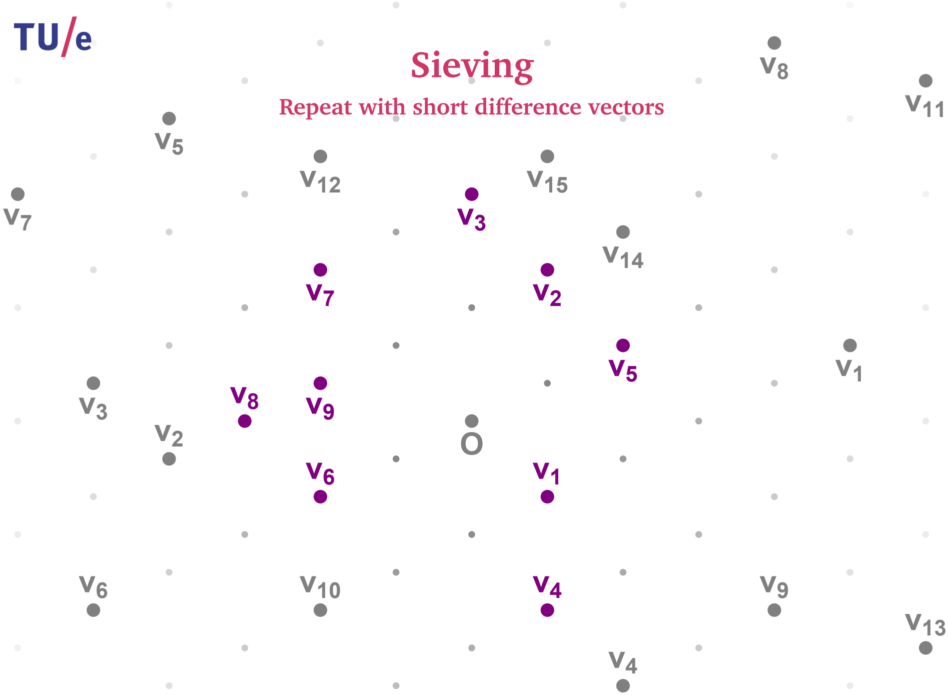
Sieving

Repeat with short difference vectors



Sieving

Repeat with short difference vectors



Sieving

Overview

Sieving

Overview

Heuristic (Nguyen–Vidick, J. Math. Crypt. '08)

Sieving solves SVP in time $(4/3)^{n+o(n)}$ and space $(4/3)^{n/2+o(n)}$.

Sieving

Overview

Heuristic (Nguyen–Vidick, J. Math. Crypt. ’08)

Sieving solves SVP in time $(4/3)^{n+o(n)}$ and space $(4/3)^{n/2+o(n)}$.

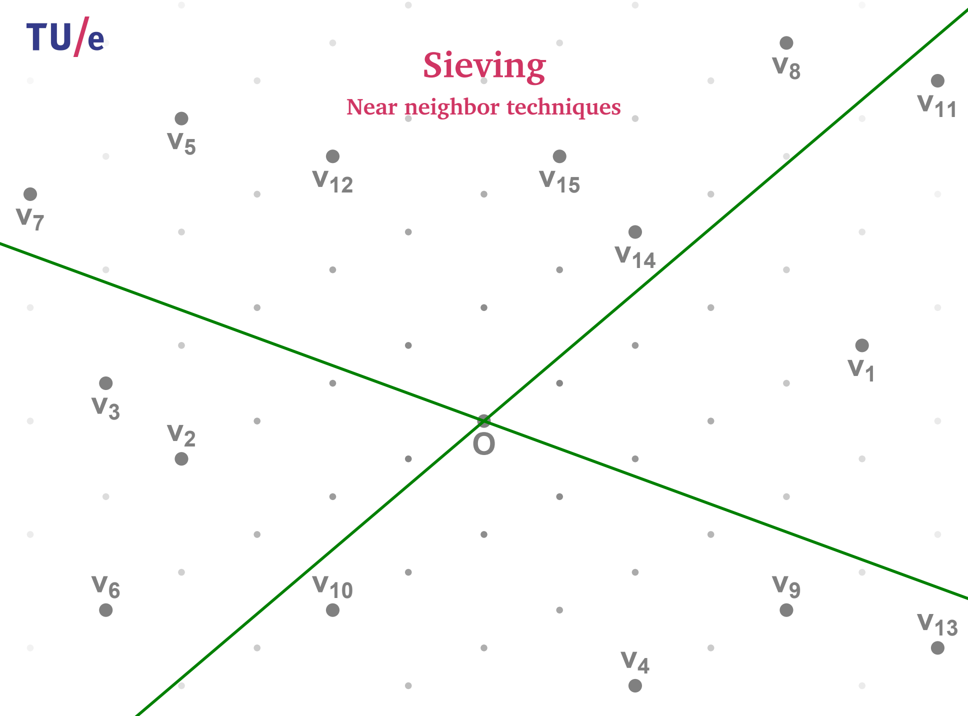
The list size comes from heuristic packing/saturation arguments, the time complexity is quadratic in the list size.

Sieving

Near neighbor techniques

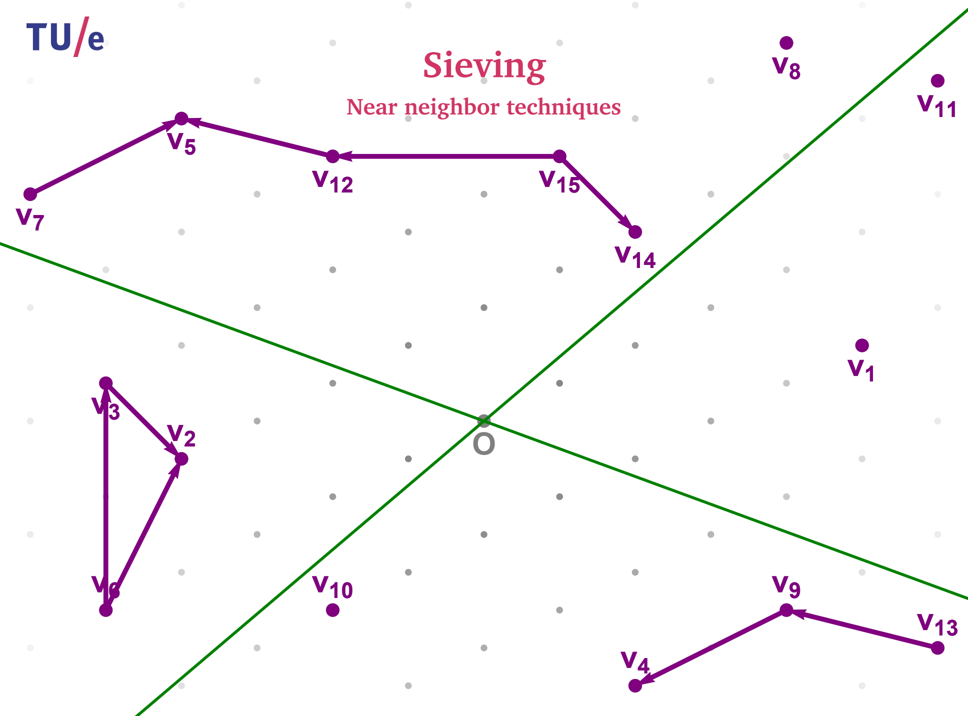
Sieving

Near neighbor techniques



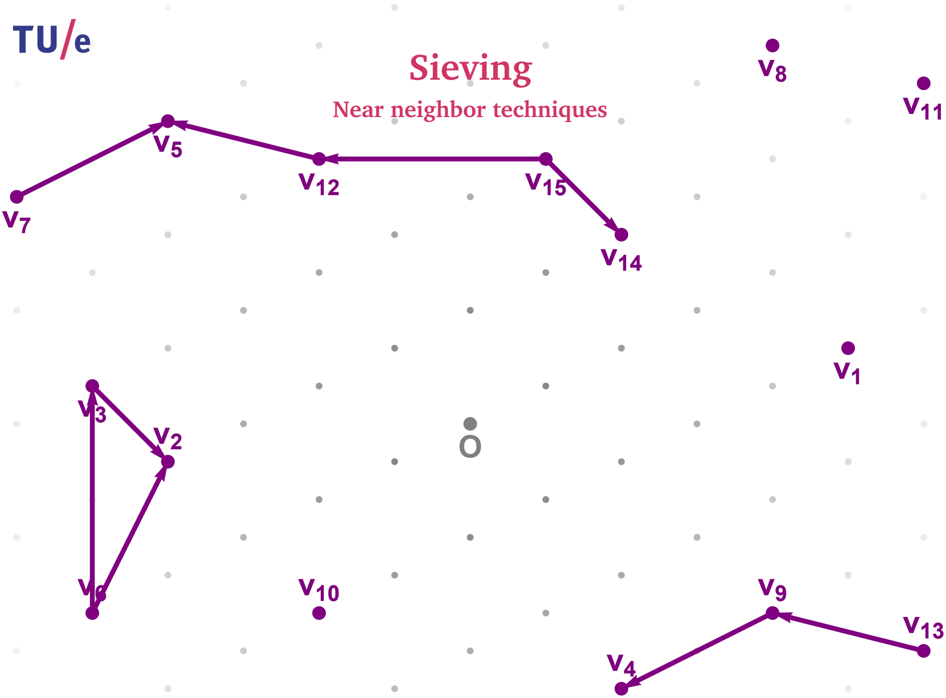
Sieving

Near neighbor techniques



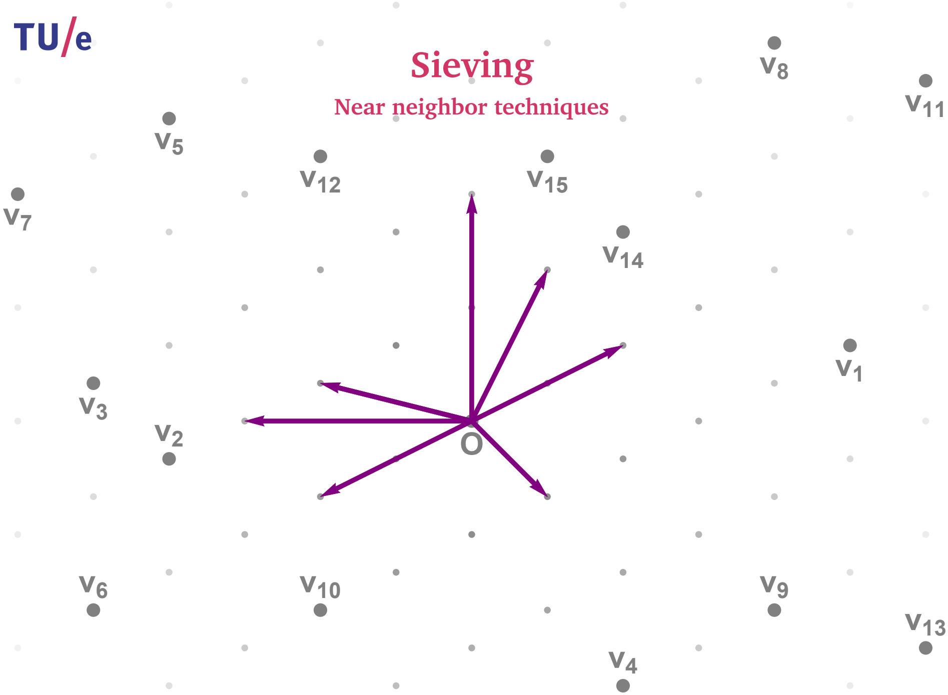
Sieving

Near neighbor techniques



Sieving

Near neighbor techniques



Sieving

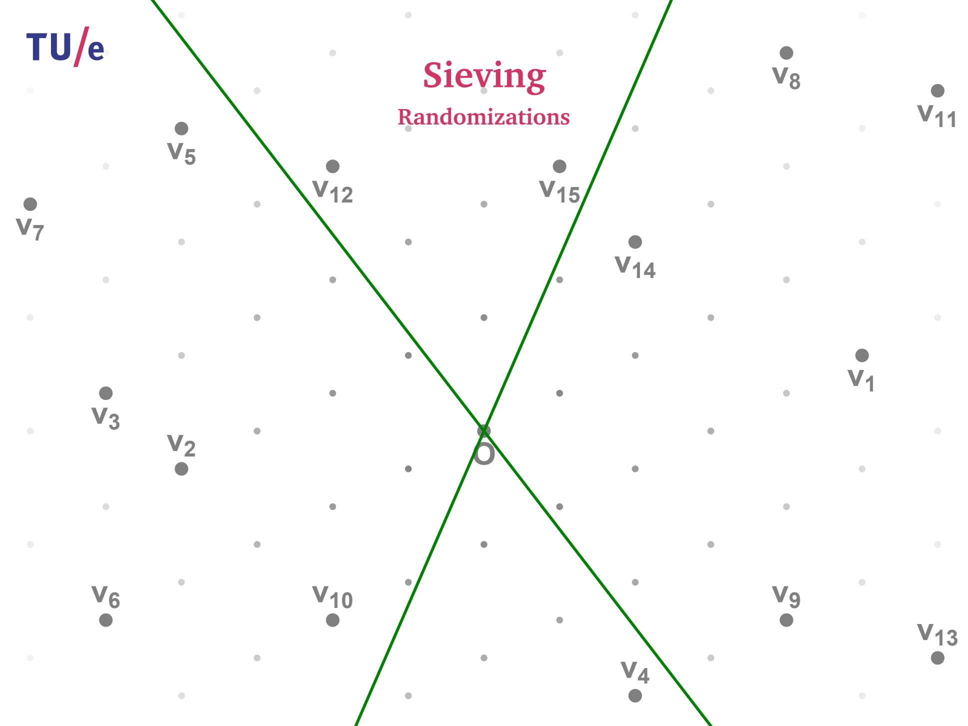
Near neighbor techniques

Sieving

Randomizations

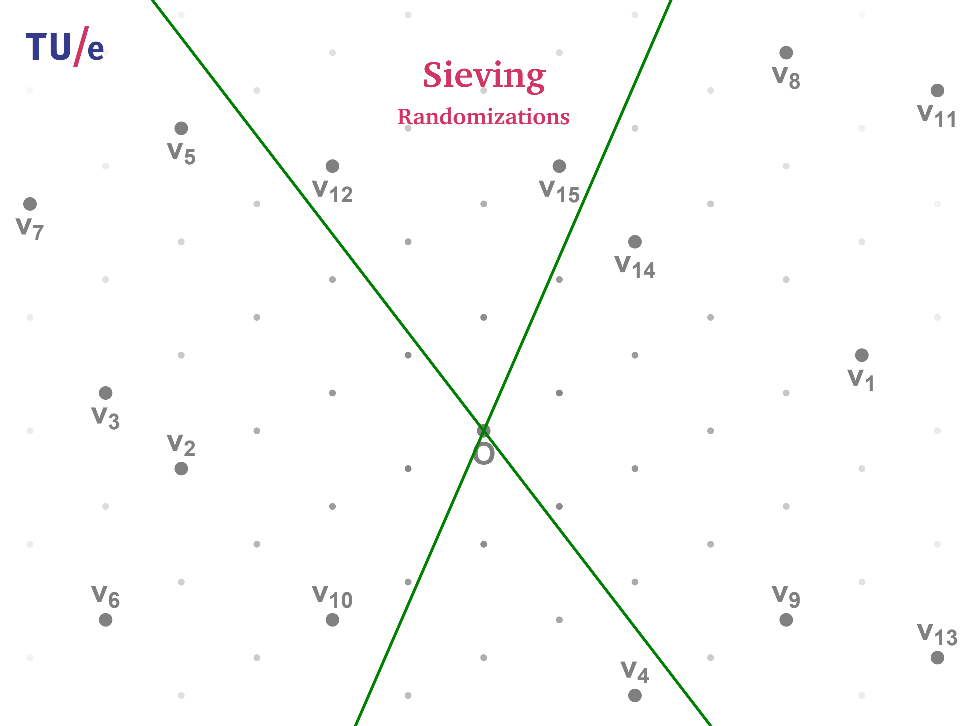
Sieving

Randomizations



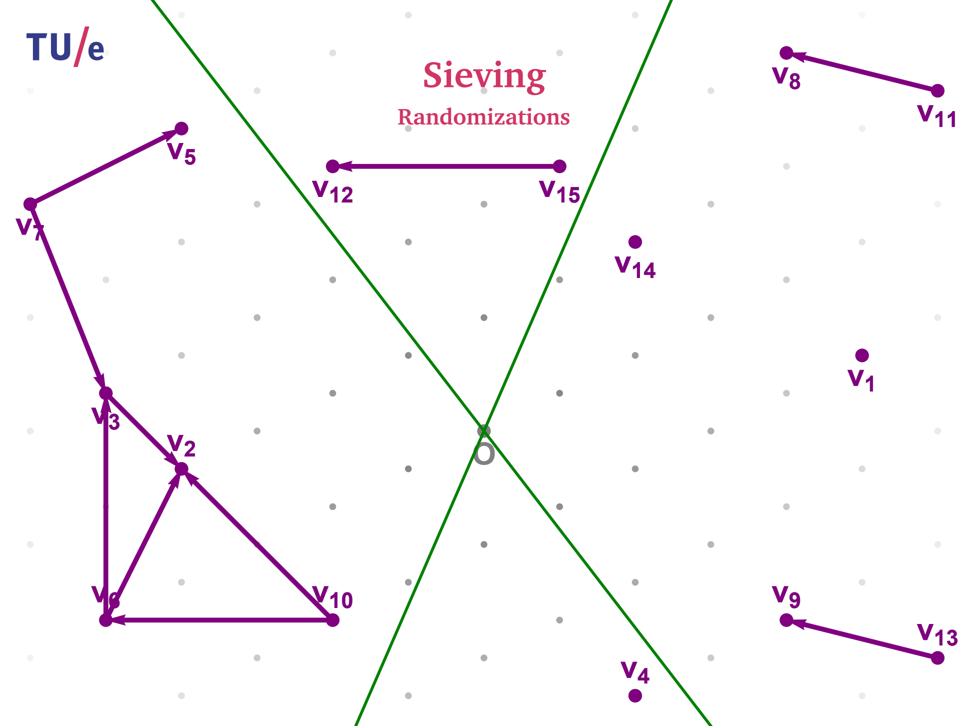
Sieving

Randomizations



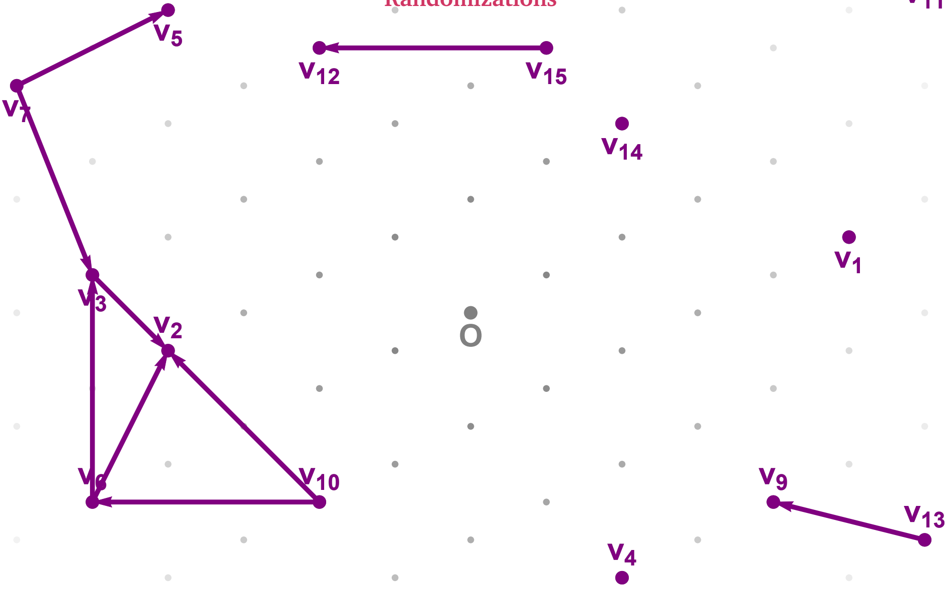
Sieving

Randomizations



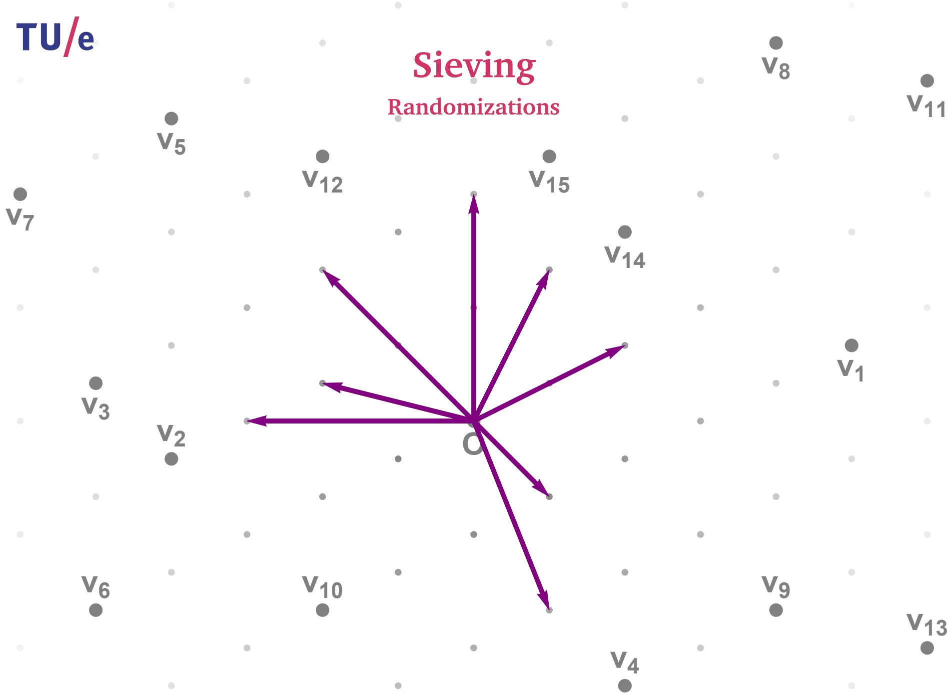
Sieving

Randomizations



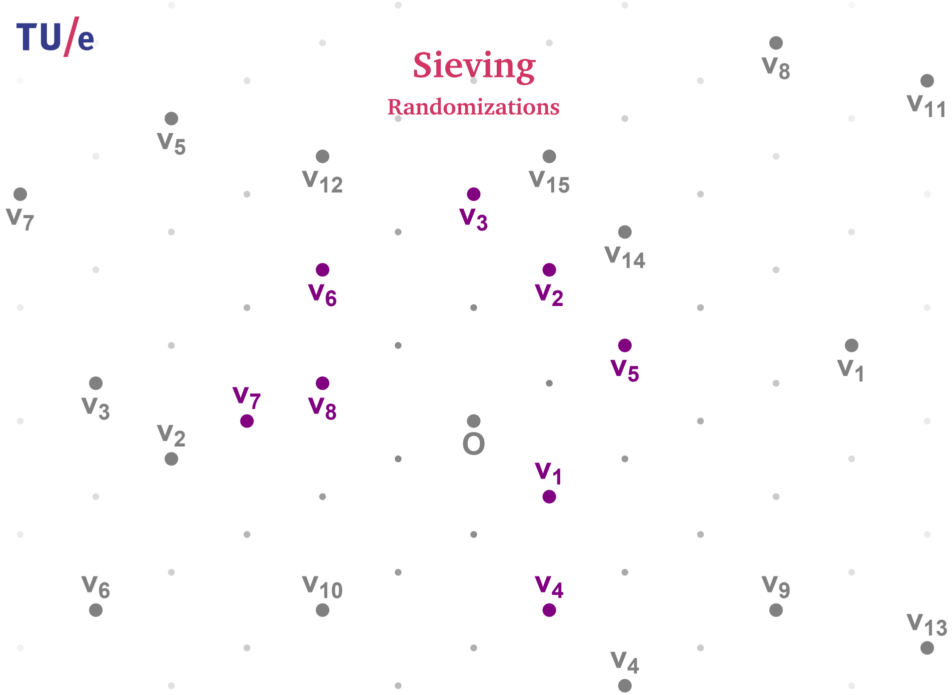
Sieving

Randomizations



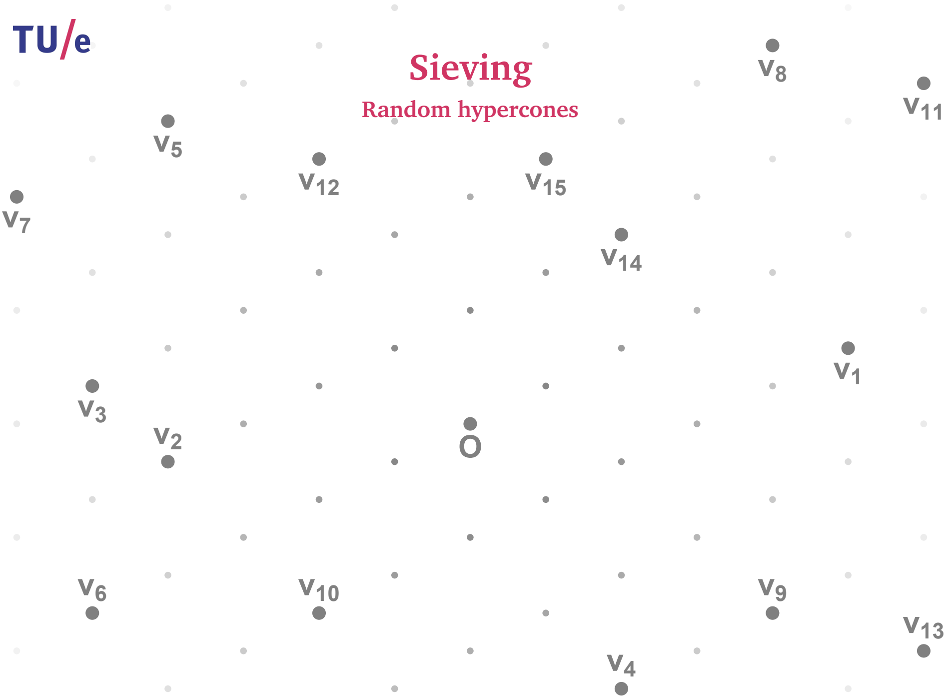
Sieving

Randomizations



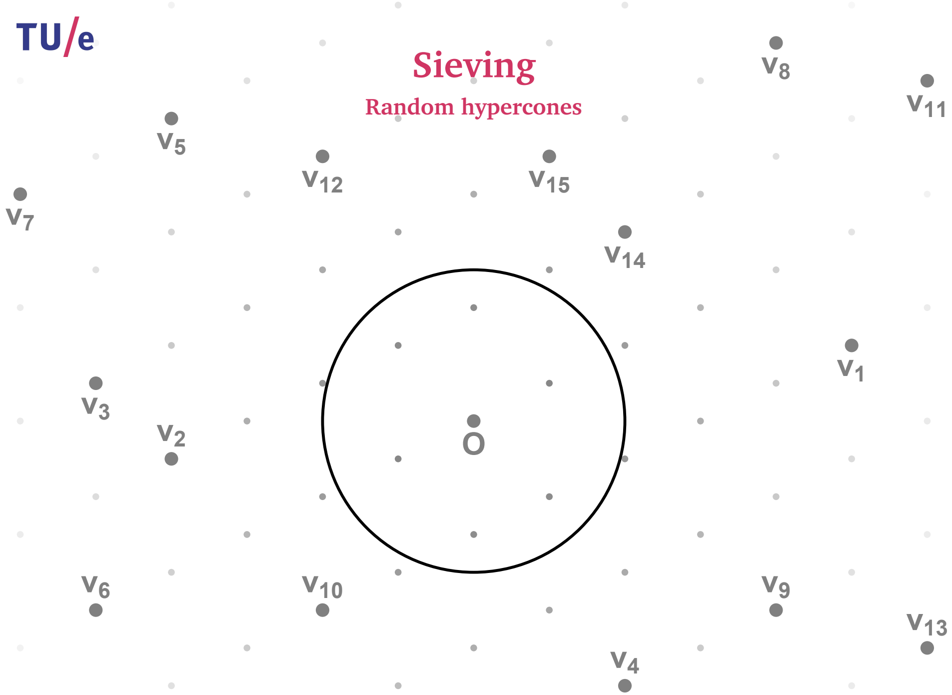
Sieving

Random hypercones



Sieving

Random hypercones

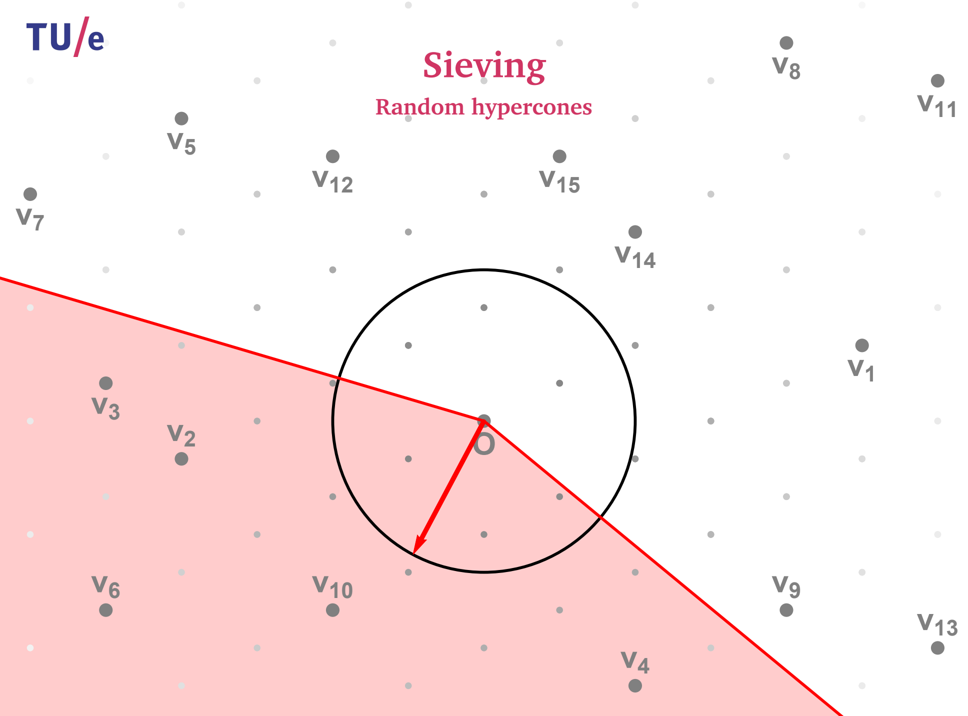


Sieving

Random hypercones

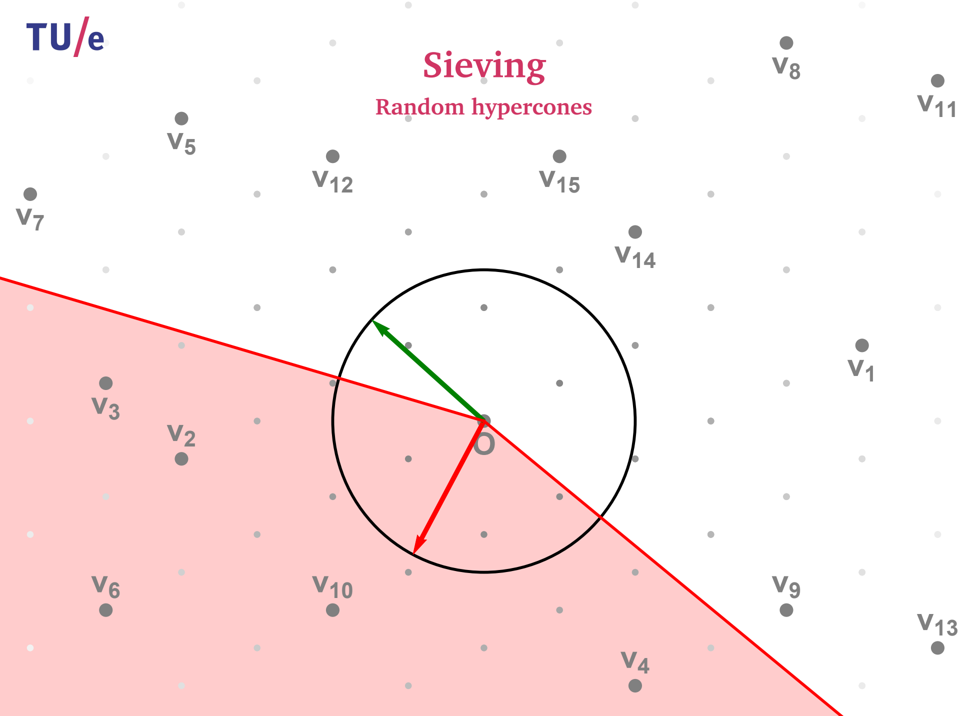
Sieving

Random hypercones



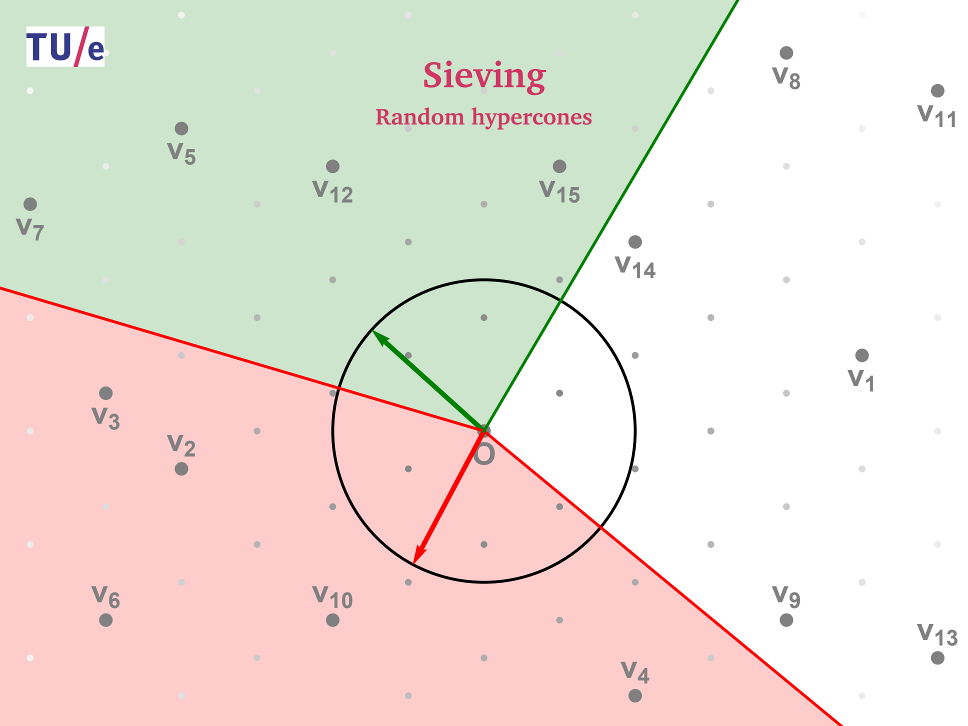
Sieving

Random hypercones



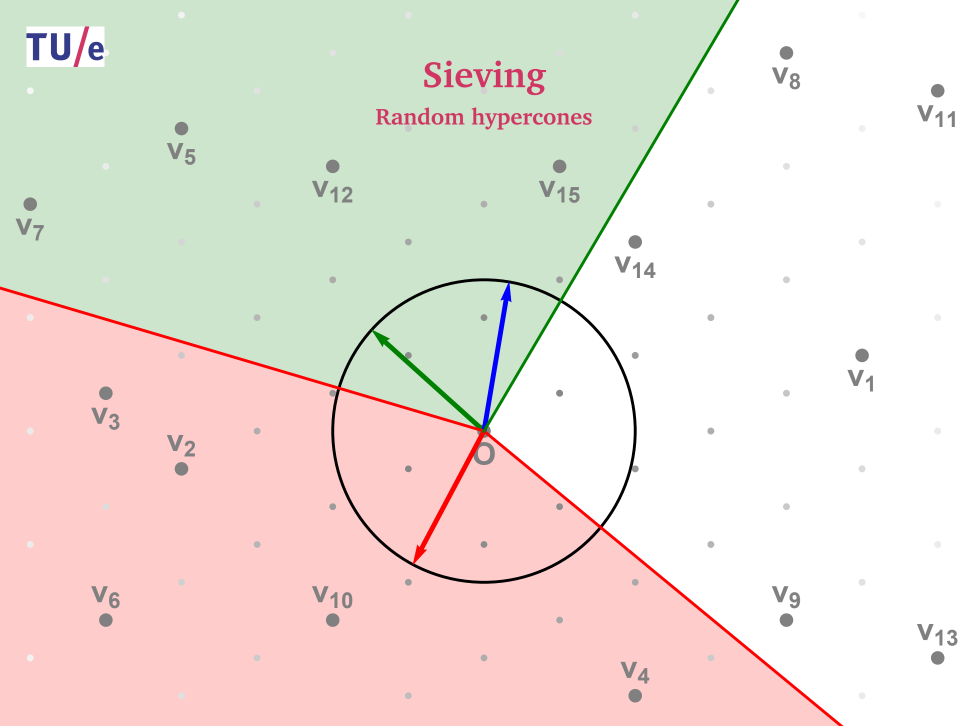
Sieving

Random hypercones



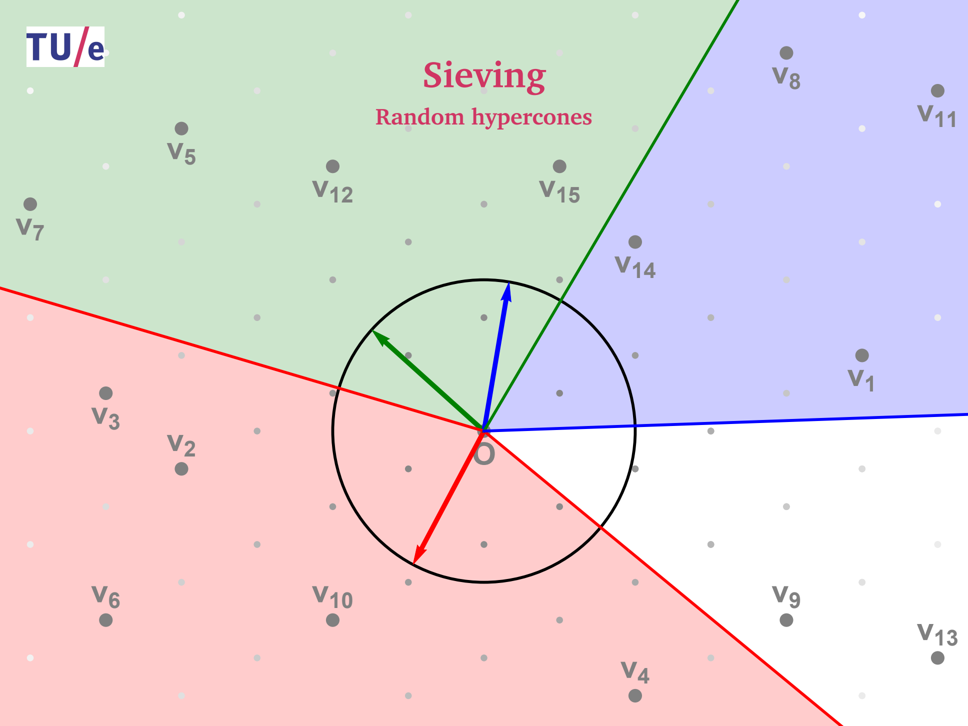
Sieving

Random hypercones



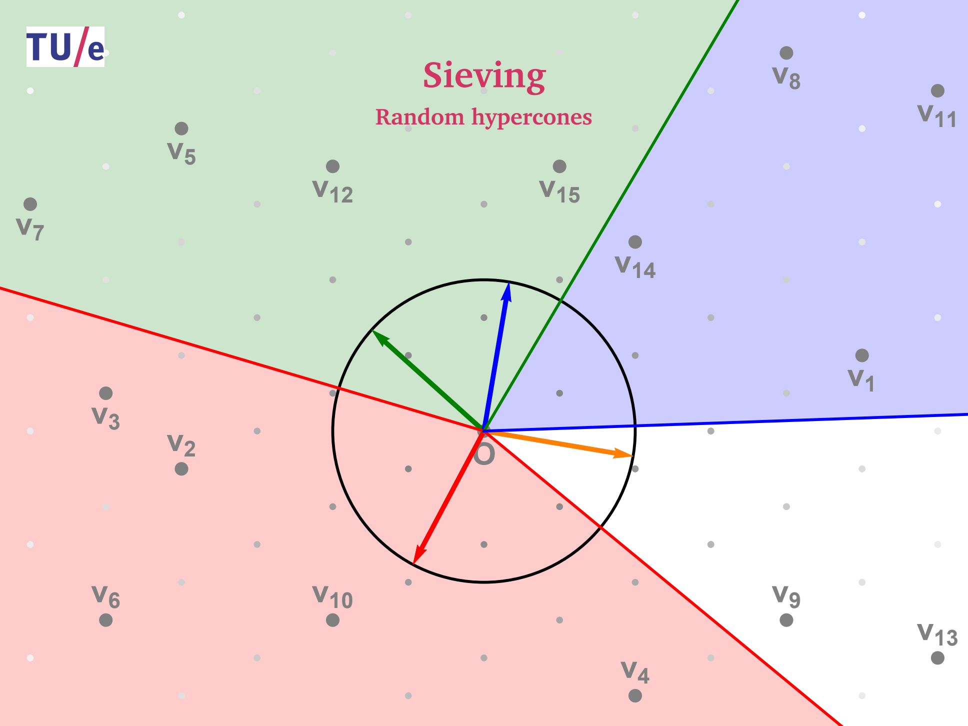
Sieving

Random hypercones



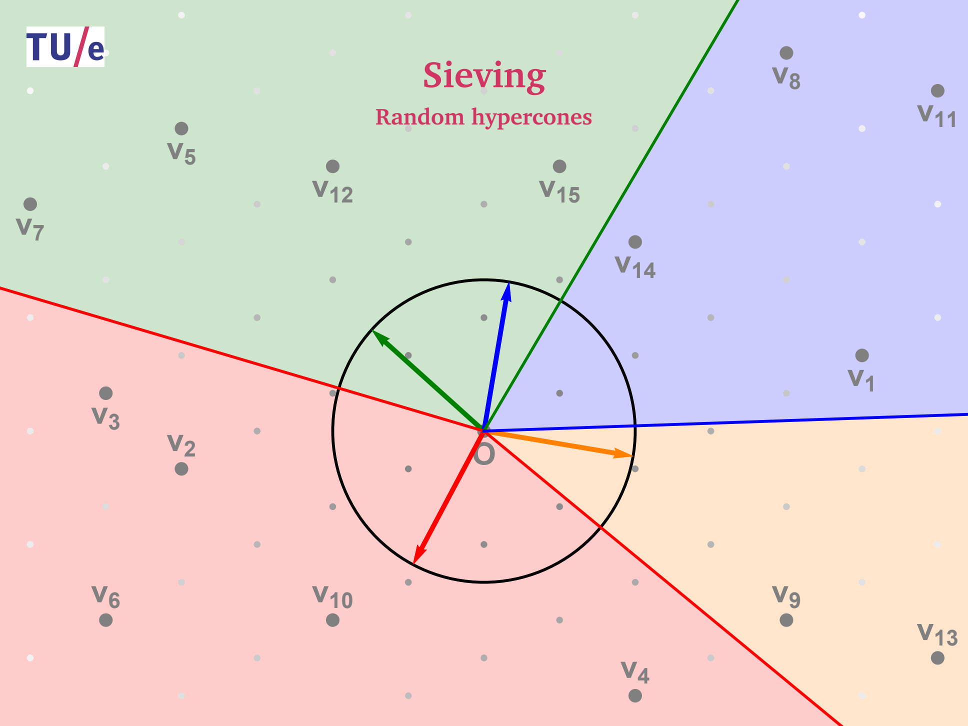
Sieving

Random hypercones



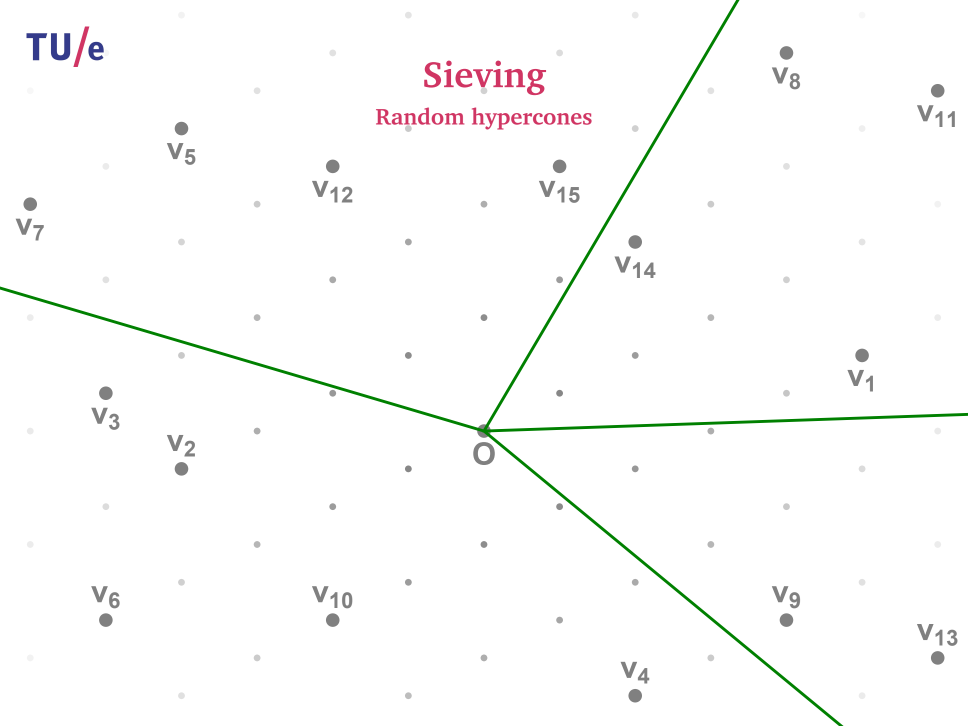
Sieving

Random hypercones



Sieving

Random hypercones

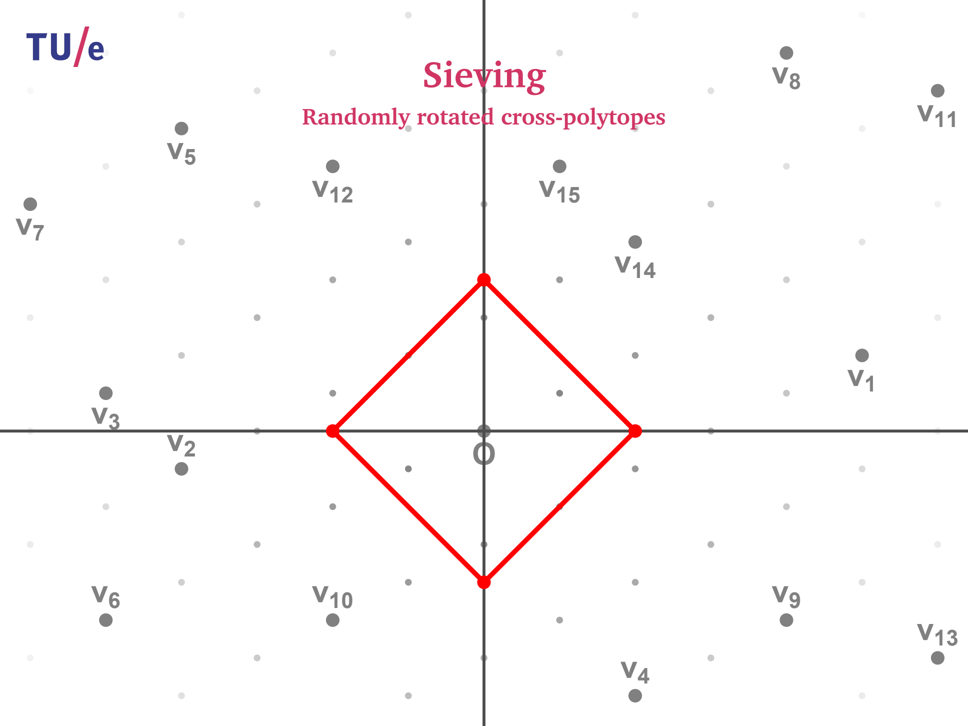


Sieving

Randomly rotated cross-polytopes

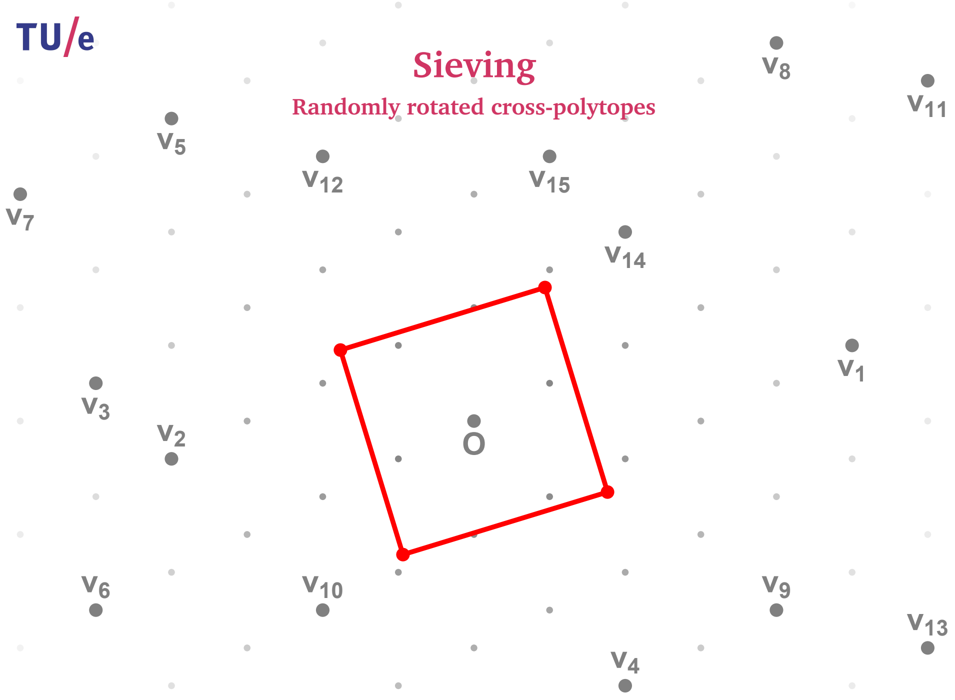
Sieving

Randomly rotated cross-polytopes



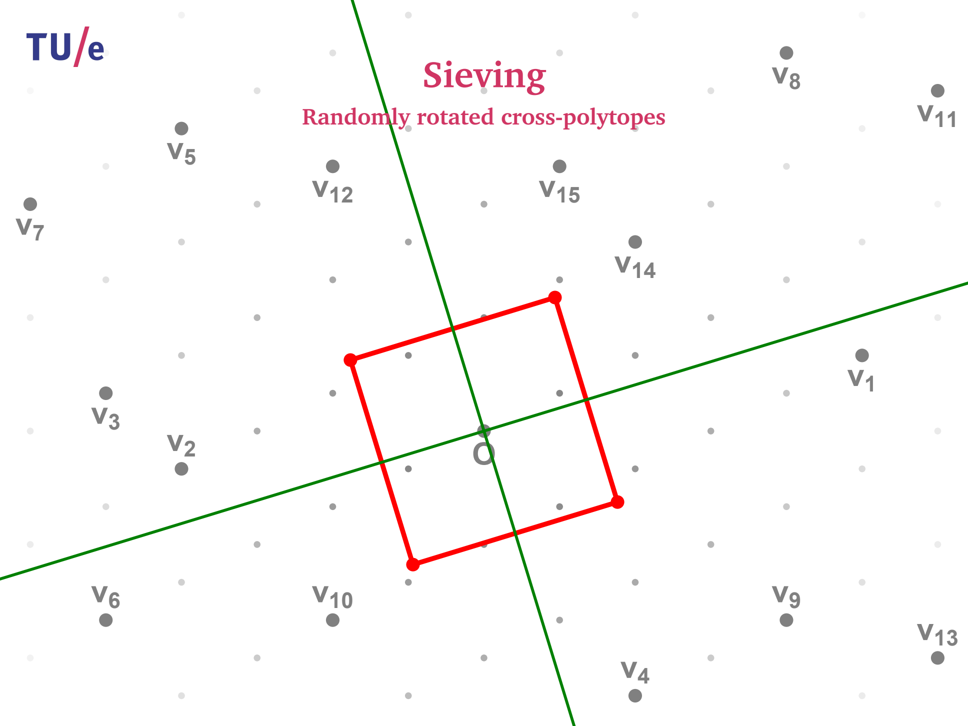
Sieving

Randomly rotated cross-polytopes



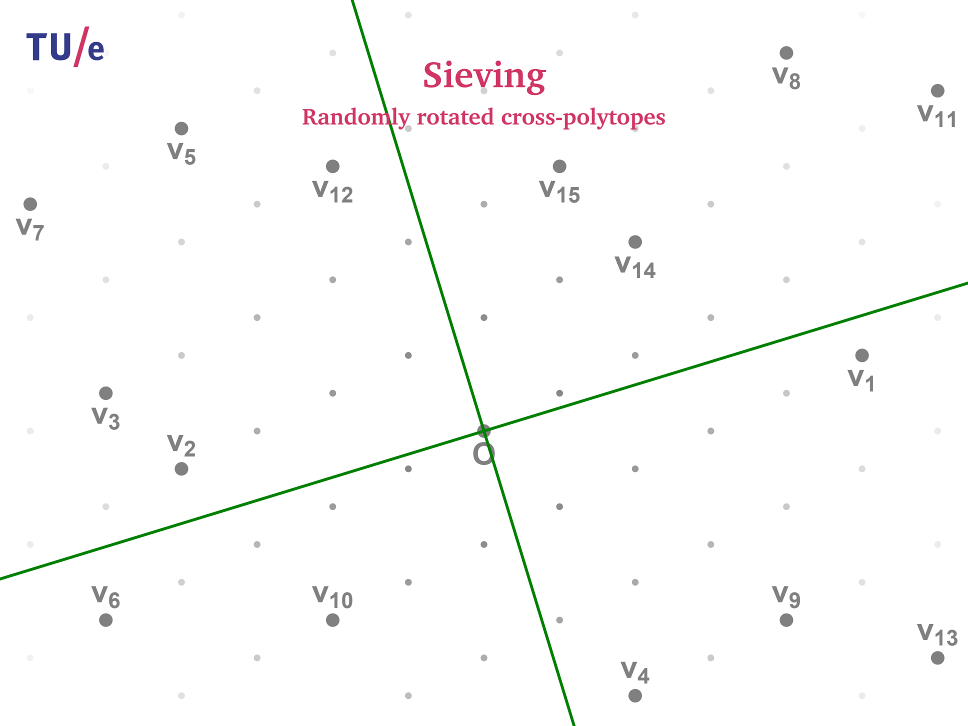
Sieving

Randomly rotated cross-polytopes



Sieving

Randomly rotated cross-polytopes

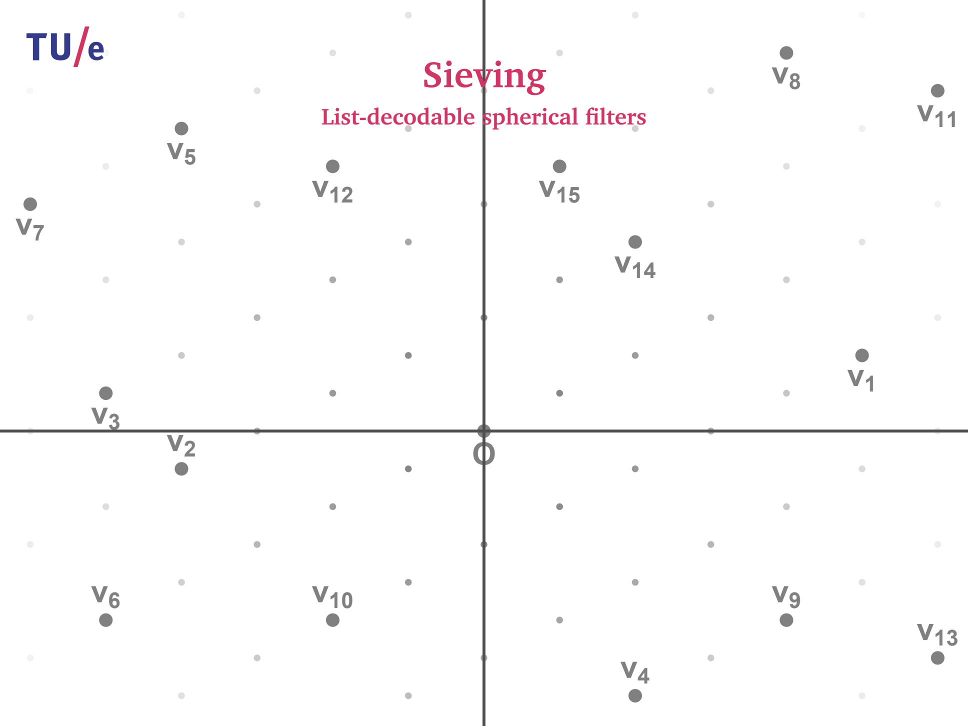


Sieving

List-decodable spherical filters

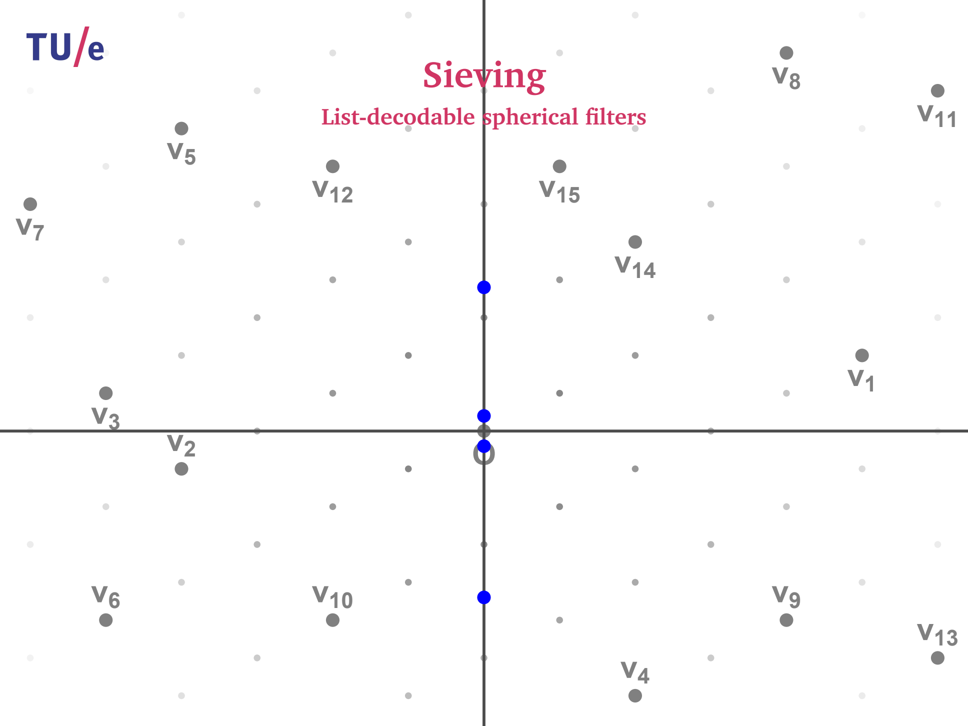
Sieving

List-decodable spherical filters



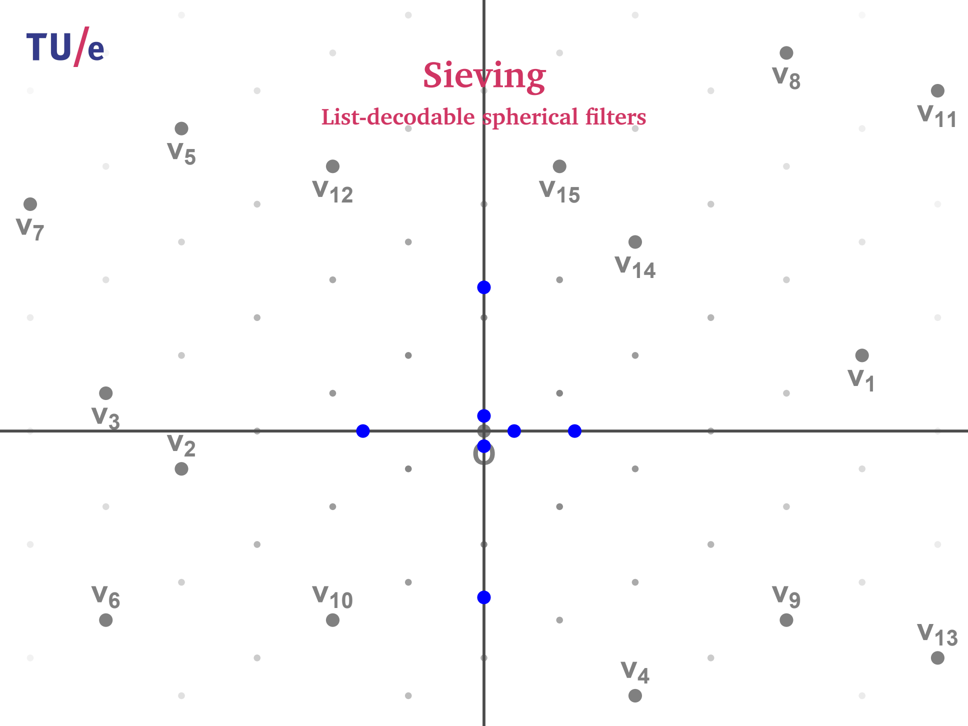
Sieving

List-decodable spherical filters



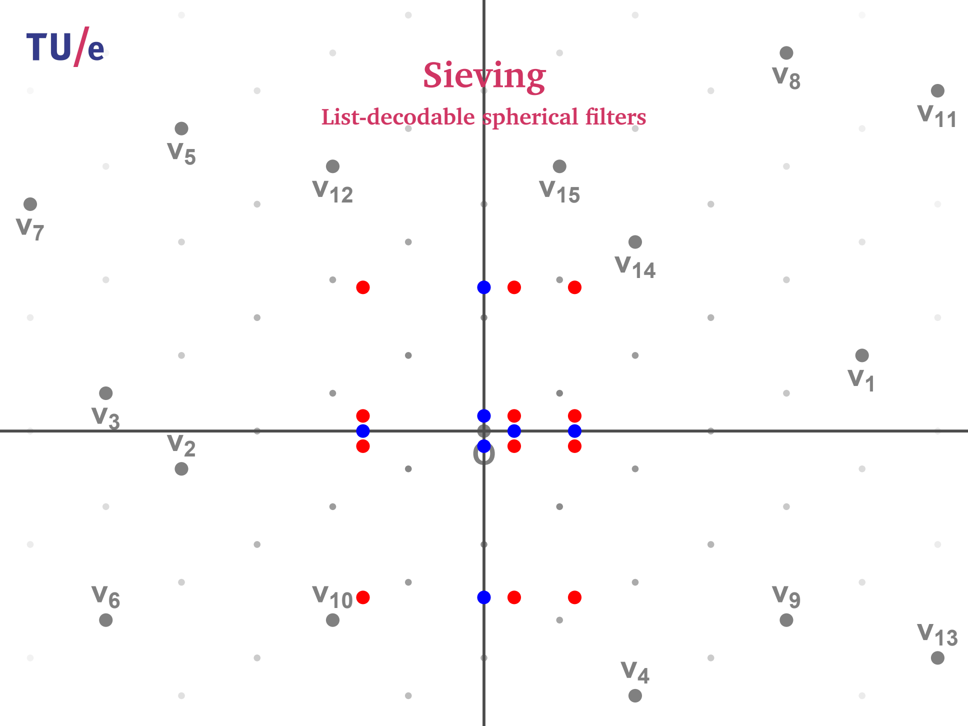
Sieving

List-decodable spherical filters



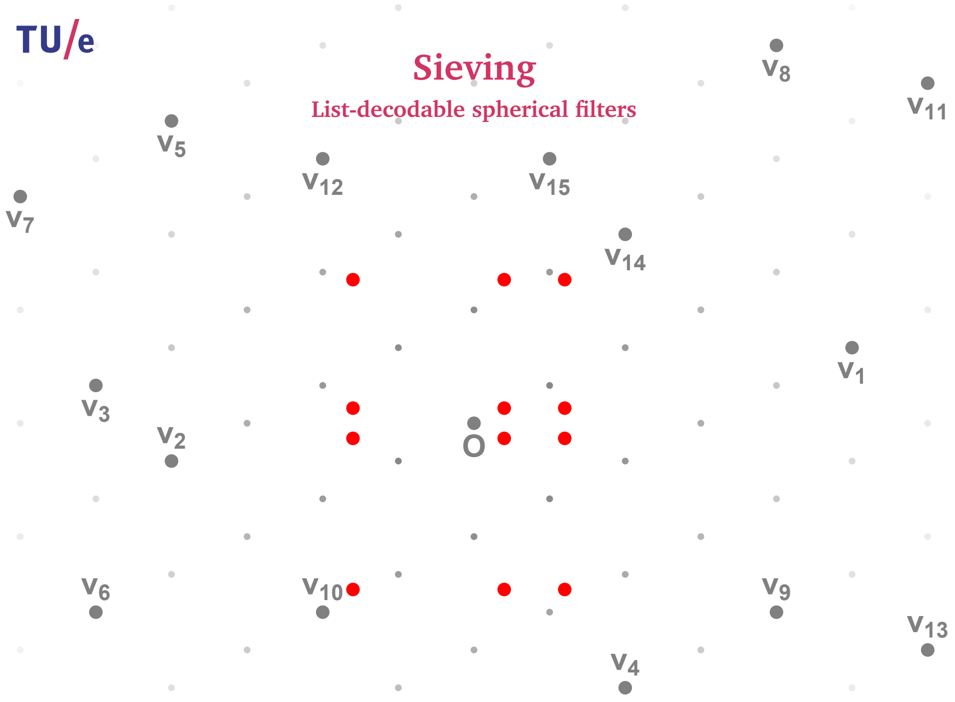
Sieving

List-decodable spherical filters



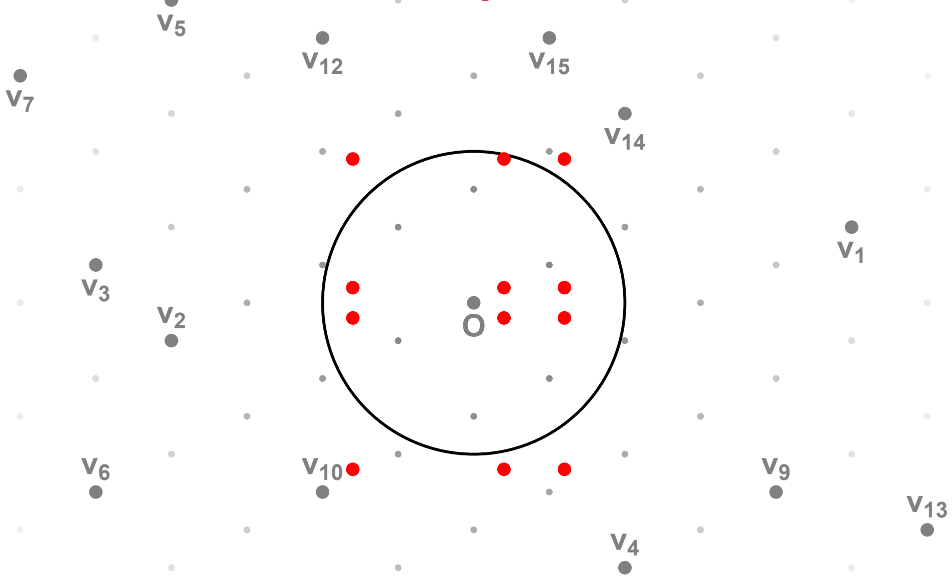
Sieving

List-decodable spherical filters



Sieving

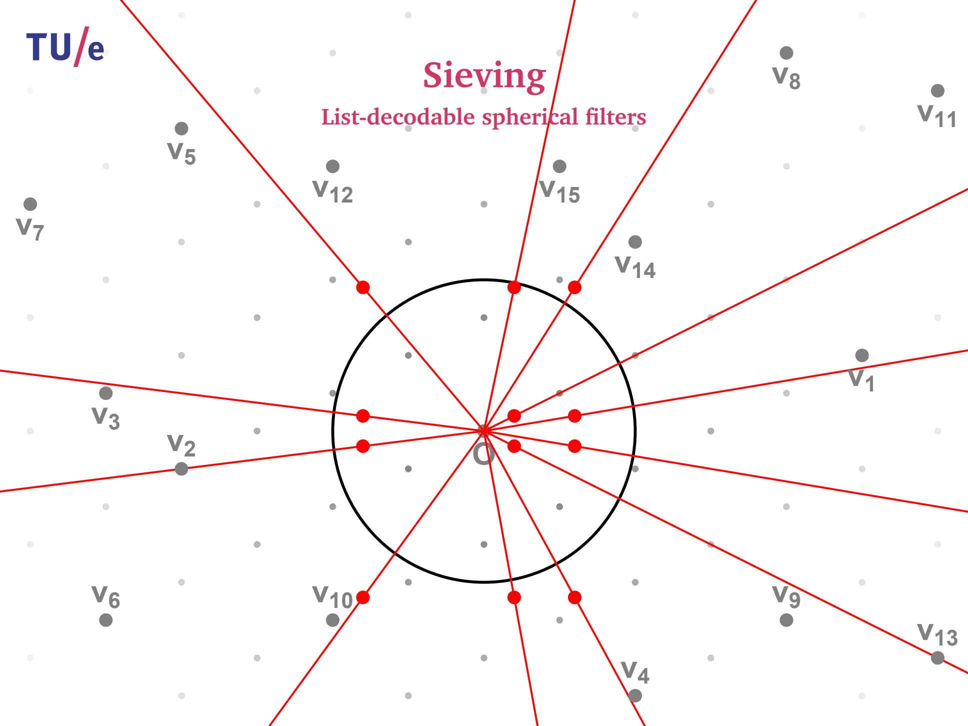
List-decodable spherical filters



TU/e

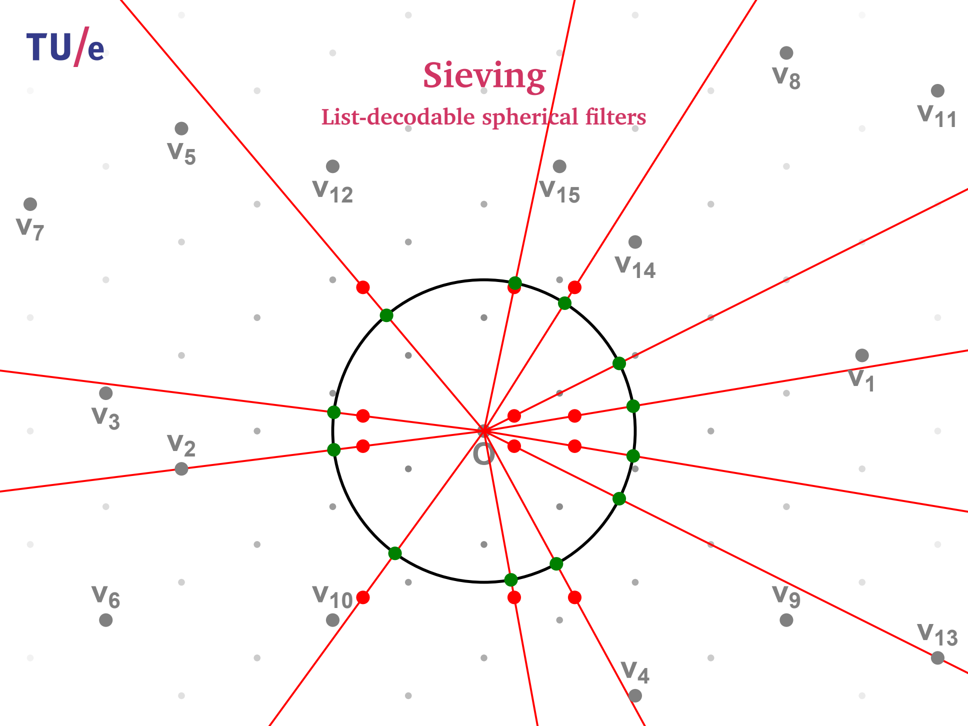
Sieving

List-decodable spherical filters



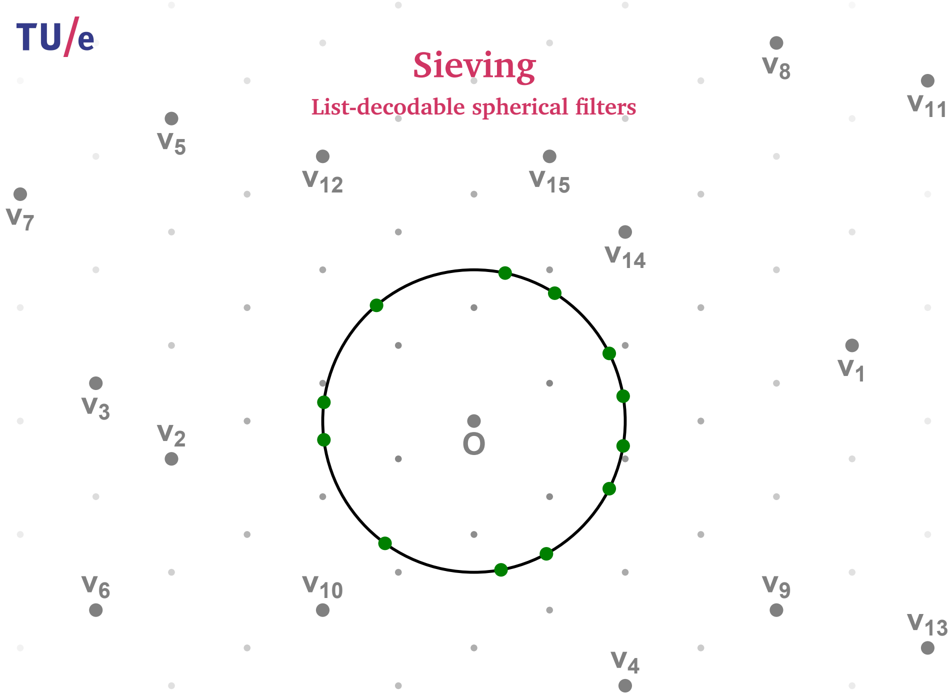
Sieving

List-decodable spherical filters



Sieving

List-decodable spherical filters



Outline

Lattices

SVP algorithms
- Enumeration
- Sieving

SVP hardness
- Theory
- Practice

Conclusion

SVP hardness

Theory (March 2019)

	Algorithm	$\log_2(\text{Time})$	$\log_2(\text{Space})$
Worst-case SVP	Enumeration [Poh81, Kan83, ..., MW15, AN17]	$O(n \log n)$	$O(\log n)$
	AKS-sieve [AKS01, NV08, MV10, HPS11]	$3.398n$	$1.985n$
	ListSieve [MV10, MDB14]	$3.199n$	$1.327n$
	Birthday sieves [PS09, HPS11]	$2.465n$	$1.233n$
	Enumeration/DGS hybrid [CCL17]	$2.048n$	$0.500n$
	Voronoi cell algorithm [AEVZ02, MV10b]	$2.000n$	$1.000n$
	Quantum sieve [LMP13, LMP15]	$1.799n$	$1.286n$
	Quantum enum/DGS [CCL17]	$1.256n$	**0.500n**
	Discrete Gaussian sampling [ADRS15, ADS15, AS18]	**1.000n**	$1.000n$
Average-case SVP	The Nguyen–Vidick sieve [NV08]	$0.415n$	$0.208n$
	GaussSieve [MV10, ..., IKMT14, BNvdP16, YKYC17]	$0.415n$	$0.208n$
	Triple sieve [BLS16, HK17]	$0.396n$	$0.189n$
	Two-level sieve [WLTB11]	$0.384n$	$0.256n$
	Kleinjung sieve [Kle14]	$0.379n$	$0.189n$
	Three-level sieve [ZPH13]	$0.378n$	$0.283n$
	Overlattice sieve [BGJ14]	$0.377n$	$0.293n$
	Triple sieve with NNS [HK17, HKL18]	$0.359n$	**0.189n**
	Single filters [DL17, ADH+19]	$0.349n$	$0.246n$
	Hyperplane LSH [Cha02, FBB+14, Laa15, ..., LM18]	$0.337n$	$0.337n$
	Graph-based NNS [EPY99, DCL11, MPLK14, Laa18]	$0.327n$	$0.282n$
	Hypercube LSH [TT07, Laa17]	$0.322n$	$0.322n$
	May–Ozerov NNS [MO15, BGJ15]	$0.311n$	$0.311n$
	Quantum sieve [LMP13]	$0.311n$	$0.208n$
	Spherical LSH [AINR14, LdW15]	$0.297n$	$0.297n$
	Cross-polytope LSH [TT07, AILRS15, BL16, KW17]	$0.297n$	$0.297n$
	Spherical LSF [BDGL16, MLB17, ALRW17, Chr17]	**0.292n**	$0.292n$
	Quantum NNS sieve [LMP15, Laa16]	**0.265n**	$0.265n$

SVP hardness

Theory (March 2019)

	Algorithm	$\log_2(\text{Time})$	$\log_2(\text{Space})$
Worst-case SVP	Enumeration [Poh81, Kan83, ..., MW15, AN17]	$O(n \log n)$	$O(\log n)$
	AKS-sieve [AKS01, NV08, MV10, HPS11]	$3.398n$	$1.985n$
	ListSieve [MV10, MDB14]	$3.199n$	$1.327n$
	Birthday sieves [PS09, HPS11]	$2.465n$	$1.233n$
	Enumeration/DGS hybrid [CCL17]	$2.048n$	$0.500n$
	Voronoi cell algorithm [AEVZ02, MV10b]	$2.000n$	$1.000n$
	Quantum sieve [LMP13, LMP15]	$1.799n$	$1.286n$
	Quantum enum/DGS [CCL17]	$1.256n$	**0.500n**
	Discrete Gaussian sampling [ADRS15, ADS15, AS18]	**1.000n**	$1.000n$
Average-case SVP	The Nguyen–Vidick sieve [NV08]	$0.415n$	$0.208n$
	GaussSieve [MV10, ..., IKMT14, BNvdP16, YKYC17]	$0.415n$	$0.208n$
	Triple sieve [BLS16, HK17]	$0.396n$	$0.189n$
	Two-level sieve [WLTB11]	$0.384n$	$0.256n$
	Kleinjung sieve [Kle14]	$0.379n$	$0.189n$
	Three-level sieve [ZPH13]	$0.378n$	$0.283n$
	Overlattice sieve [BGJ14]	$0.377n$	$0.293n$
	Triple sieve with NNS [HK17, HKL18]	$0.359n$	**0.189n**
	Single filters [DL17, ADH+19]	$0.349n$	$0.246n$
	Hyperplane LSH [Cha02, FBB+14, Laa15, ..., LM18]	$0.337n$	$0.337n$
	Graph-based NNS [EPY99, DCL11, MPLK14, Laa18]	$0.327n$	$0.282n$
	Hypercube LSH [TT07, Laa17]	$0.322n$	$0.322n$
	May–Ozerov NNS [MO15, BGJ15]	$0.311n$	$0.311n$
	Quantum sieve [LMP13]	$0.311n$	$0.208n$
	Spherical LSH [AINR14, LdW15]	$0.297n$	$0.297n$
	Cross-polytope LSH [TT07, AILRS15, BL16, KW17]	$0.297n$	$0.297n$
	Spherical LSF [BDGL16, MLB17, ALRW17, Chr17]	**0.292n**	$0.292n$
	Quantum NNS sieve [LMP15, Laa16]	**0.265n**	$0.265n$

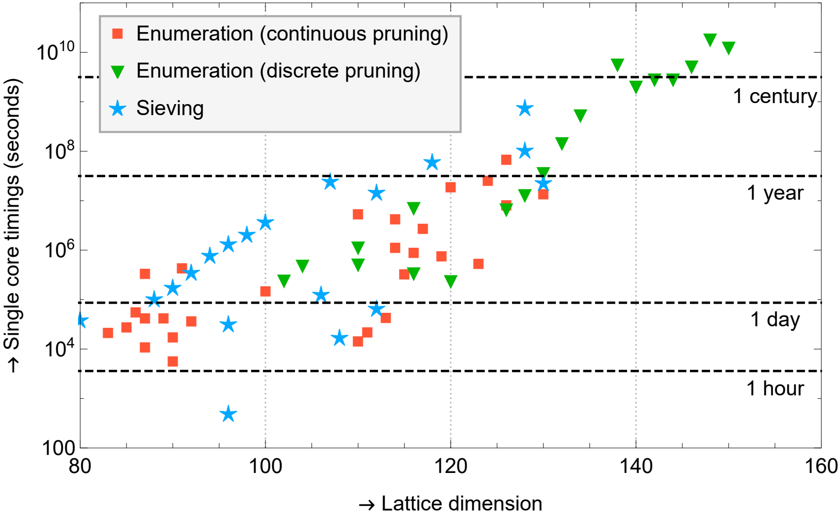
SVP hardness

Theory (March 2019)

	Algorithm	$\log_2$(Time)	$\log_2$(Space)	Max. n
Worst-case SVP	Enumeration [Poh81, Kan83, ..., MW15, AN17]	$O(n \log n)$	$O(\log n)$	150
	AKS-sieve [AKS01, NV08, MV10, HPS11]	$3.398n$	$1.985n$	–
	ListSieve [MV10, MDB14]	$3.199n$	$1.327n$	70
	Birthday sieves [PS09, HPS11]	$2.465n$	$1.233n$	–
	Enumeration/DGS hybrid [CCL17]	$2.048n$	$0.500n$	–
	Voronoi cell algorithm [AEVZ02, MV10b]	$2.000n$	$1.000n$	40
	Quantum sieve [LMP13, LMP15]	$1.799n$	$1.286n$	–
	Quantum enum/DGS [CCL17]	$1.256n$	**0.500n**	–
	Discrete Gaussian sampling [ADRS15, ADS15, AS18]	**1.000n**	$1.000n$	–
Average-case SVP	The Nguyen–Vidick sieve [NV08]	$0.415n$	$0.208n$	50
	GaussSieve [MV10, ..., IKMT14, BNvdP16, YKYC17]	$0.415n$	$0.208n$	130*
	Triple sieve [BLS16, HK17]	$0.396n$	$0.189n$	80
	Two-level sieve [WLTB11]	$0.384n$	$0.256n$	–
	Kleinjung sieve [Kle14]	$0.379n$	$0.189n$	116
	Three-level sieve [ZPH13]	$0.378n$	$0.283n$	–
	Overlattice sieve [BGJ14]	$0.377n$	$0.293n$	90
	Triple sieve with NNS [HK17, HKL18]	$0.359n$	**0.189n**	76
	Single filters [DL17, ADH+19]	$0.349n$	$0.246n$	155
	Hyperplane LSH [Cha02, FBB+14, Laa15, ..., LM18]	$0.337n$	$0.337n$	107
	Graph-based NNS [EPY99, DCL11, MPLK14, Laa18]	$0.327n$	$0.282n$	–
	Hypercube LSH [TT07, Laa17]	$0.322n$	$0.322n$	–
	May–Ozerov NNS [MO15, BGJ15]	$0.311n$	$0.311n$	–
	Quantum sieve [LMP13]	$0.311n$	$0.208n$	–
	Spherical LSH [AINR14, LdW15]	$0.297n$	$0.297n$	–
	Cross-polytope LSH [TT07, AILRS15, BL16, KW17]	$0.297n$	$0.297n$	80
	Spherical LSF [BDGL16, MLB17, ALRW17, Chr17]	**0.292n**	$0.292n$	92
	Quantum NNS sieve [LMP15, Laa16]	**0.265n**	$0.265n$	–

SVP hardness

Practice (July 2017)



TU/e

The General Sieve Kernel and New Records in Lattice Reduction

Martin R. Albrecht[1], Léo Ducas[2], Gottfried Herold[3], Elena Kirshanova[3], Eamonn W. Postlethwaite[1], Marc Stevens[2⋆]

[1] Information Security Group, Royal Holloway, University of London
[2] Cryptology Group, CWI, Amsterdam, The Netherlands
[3] ENS Lyon

Abstract. We propose the General Sieve Kernel (G6K, pronounced /ʒe.si.ka/), an abstract stateful machine supporting a wide variety of lattice reduction strategies based on sieving algorithms. Using the basic instruction set of this abstract stateful machine, we first give concise formulations of previous sieving strategies from the literature and then propose new ones. We then also give a light variant of BKZ exploiting the features of our abstract stateful machine. This encapsulates several recent suggestions (Ducas at Eurocrypt 2018; Laarhoven and Mariano at PQCrypto 2018) to move beyond treating sieving as a blackbox SVP oracle and to utilise strong lattice reduction as preprocessing for sieving. Furthermore, we propose new tricks to minimise the sieving computation required for a given reduction quality with mechanisms such as recycling vectors between sieves, on-the-fly lifting and flexible insertions akin to Deep LLL and recent variants of Random Sampling Reduction.

Moreover, we provide a highly optimised, multi-threaded and tweakable implementation of this machine which we make open-source. We then illustrate the performance of this implementation of our sieving strategies by applying G6K to various lattice challenges. In particular, our approach allows us to solve previously unsolved instances of the Darmstadt SVP (151, 153, 155) and LWE (e.g. (75, 0.005)) challenges. Our solution for the SVP-151 challenge was found 400 times faster than the time reported for the SVP-150 challenge, the previous record. For exact SVP, we observe a performance crossover between G6K and FPLLL's state of the art implementation of enumeration at dimension 70.

TU/e

The General Sieve Kernel and New Records in Lattice Reduction

Martin R. Albrecht[1], Léo Ducas[2], Gottfried Herold[3], Elena Kirshanova[3], Eamonn W. Postlethwaite[1], Marc Stevens[2*]

[1] Information Security Group, Royal Holloway, University of London
[2] Cryptology Group, CWI, Amsterdam, The Netherlands
[3] ENS Lyon

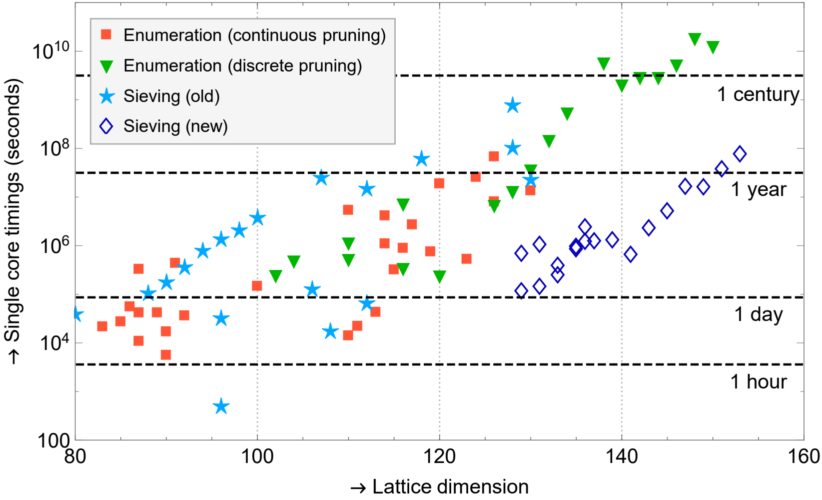
Abstract. We propose the General Sieve Kernel (G6K, pronounced /ʒe.si.ka/), an abstract stateful machine supporting a wide variety of lattice reduction strategies based on sieving algorithms. Using the basic instruction set of this abstract stateful machine, we first give concise formulations of previous sieving strategies from the literature and then propose new ones. We then also give a light variant of BKZ exploiting the features of our abstract stateful machine. This encapsulates several recent suggestions (Ducas at Eurocrypt 2018; Laarhoven and Mariano at PQCrypto 2018) to move beyond treating sieving as a blackbox SVP oracle and to utilise strong lattice reduction as preprocessing for sieving. Furthermore, we propose new tricks to minimise the sieving computation required for a given reduction quality with mechanisms such as recycling vectors between sieves, on-the-fly lifting and flexible insertions akin to Deep LLL and recent variants of Random Sampling Reduction.

Moreover, we provide a highly optimised, multi-threaded and tweakable implementation of this machine which we make open-source. We then illustrate the performance of this implementation of our sieving strategies

(151, 153, 155) and LWE (e.g. (75, 0.005)) challenges. Our solution for the SVP-151 challenge was found 400 times faster than the time reported for the SVP-150 challenge, the previous record. For exact SVP, we observe a performance crossover between G6K and FPLLL's state of the art implementation of enumeration at dimension 70.

SVP hardness

Practice (February 2019)



SVP hardness

NIST submissions – Round 1 (December 2017)

Title	S	E	O	Submitters
CRYSTALS–Dilithium	•			**Lyubashevsky**, Ducas, Kiltz, Lepoint, Schwabe, Seiler, Stehlé
CRYSTALS–Kyber	•			**Schwabe**, Avanzi, Bos, Ducas, Kiltz, Lepoint, Lyubashevsky, Schanck, ...
Ding Key Exchange	•			**Ding**, Takagi, Gao, Wang
DRS			•	**Plantard**, Sipasseuth, Dumondelle, Susilo
(R.)EMBLEM	•			**Seo**, Park, Lee, Kim, Lee
FALCON	•			**Prest**, Fouque, Hoffstein, Kirchner, Lyubashevsky, Pornin, Ricosset, ...
FrodoKEM	•			**Naehrig**, Alkim, Bos, Ducas, Easterbrook, LaMacchia, Longa, Mironov, ...
Giophantus	•			**Akiyama**, Goto, Okumura, Takagi, Nuida, Hanaoka, Shimizu, Ikematsu
HILA5	•			**Saarinen**
KCL	•			**Zhao**, Jin, Gong, Sui
KINDI	•			**El Bansarkhani**
LAC	•			**Lu**, Liu, Jia, Xue, He, Zhang
LIMA	•			**Smart**, Albrecht, Lindell, Orsini, Osheter, Paterson, Peer
Lizard	•			**Cheon**, Park, Lee, Kim, Song, Hong, Kim, Kim, Hong, Yun, Kim, Park, ...
LOTUS		•		**Phong**, Hayashi, Aono, Moriai
NewHope	•			**Pöppelmann**, Alkim, Avanzi, Bos, Ducas, De La Piedra, Schwabe, Stebila
NTRUEncrypt	◦	◦		**Zhang**, Chen, Hoffstein, Whyte
NTRU-HRSS-KEM	•			**Schanck**, Hülsing, Rijneveld, Schwabe
NTRU Prime		•		**Bernstein**, Chuengsatiansup, Lange, Van Vredendaal
Odd Manhattan			•	**Plantard**
pqNTRUSign	◦	◦		**Zhang**, Chen, Hoffstein, Whyte
qTESLA	•			**Bindel**, Akleylek, Alkim, Barreto, Buchmann, Eaton, Gutoski, Krämer, ...
Round2	•			**Garcia-Morchon**, Zhang, Bhattacharya, Rietman, Tolhuizen, Torre-Arce
SABER	•			**D'Anvers**, Karmakar, Roy, Vercauteren
Three Bears	•			**Hamburg**
Titanium	•			**Steinfeld**, Sakzad, Zhao
Totals:	**24**	**4**	**2**	**Total: 26 proposals with SVP hardness estimates**

*Not included in the overview: Compact LWE, Mersenne, Ramstake, ...

SVP hardness

NIST submissions – Round 1 (merges)

Title	S	E	O	Submitters
CRYSTALS–Dilithium	•			**Lyubashevsky**, Ducas, Kiltz, Lepoint, Schwabe, Seiler, Stehlé
CRYSTALS–Kyber	•			**Schwabe**, Avanzi, Bos, Ducas, Kiltz, Lepoint, Lyubashevsky, Schanck, ...
Ding Key Exchange	•			**Ding**, Takagi, Gao, Wang
DRS			•	**Plantard**, Sipasseuth, Dumondelle, Susilo
(R.)EMBLEM	•			**Seo**, Park, Lee, Kim, Lee
FALCON	•			**Prest**, Fouque, Hoffstein, Kirchner, Lyubashevsky, Pornin, Ricosset, ...
FrodoKEM	•			**Naehrig**, Alkim, Bos, Ducas, Easterbrook, LaMacchia, Longa, Mironov, ...
Giophantus	•			**Akiyama**, Goto, Okumura, Takagi, Nuida, Hanaoka, Shimizu, Ikematsu
KCL	•			**Zhao**, Jin, Gong, Sui
KINDI	•			**El Bansarkhani**
LAC	•			**Lu**, Liu, Jia, Xue, He, Zhang
LIMA	•			**Smart**, Albrecht, Lindell, Orsini, Osheter, Paterson, Peer
Lizard	•			**Cheon**, Park, Lee, Kim, Song, Hong, Kim, Kim, Hong, Yun, Kim, Park, ...
LOTUS		•		**Phong**, Hayashi, Aono, Moriai
NewHope	•			**Pöppelmann**, Alkim, Avanzi, Bos, Ducas, De La Piedra, Schwabe, Stebila
NTRU	◦	◦		**Zhang**, Chen, Hoffstein, Hülsing, Rijneveld, Schanck, Schwabe, Whyte
NTRU Prime		•		**Bernstein**, Chuengsatiansup, Lange, Van Vredendaal
Odd Manhattan			•	**Plantard**
pqNTRUSign	◦	◦		**Zhang**, Chen, Hoffstein, Whyte
qTESLA	•			**Bindel**, Akleylek, Alkim, Barreto, Buchmann, Eaton, Gutoski, Krämer, ...
Round5	•			**Garcia-Morchon**, Saarinen, Zhang, Bhattacharya, Rietman, Tolhuizen, ...
SABER	•			**D'Anvers**, Karmakar, Roy, Vercauteren
Three Bears	•			**Hamburg**
Titanium	•			**Steinfeld**, Sakzad, Zhao
Totals:	**20**	**4**	**2**	**Total: 24 proposals with SVP hardness estimates**

*Not included in the overview: Compact LWE, Mersenne, Ramstake, ...

SVP hardness

NIST submissions – Round 2 (February 2019)

Title	S	E	O	Submitters
CRYSTALS–Dilithium	•			**Lyubashevsky**, Ducas, Kiltz, Lepoint, Schwabe, Seiler, Stehlé
CRYSTALS–Kyber	•			**Schwabe**, Avanzi, Bos, Ducas, Kiltz, Lepoint, Lyubashevsky, Schanck, ...
Ding Key Exchange	•			**Ding**, Takagi, Gao, Wang
DRS			•	**Plantard**, Sipasseuth, Dumondelle, Susilo
(R.)EMBLEM	•			**Seo**, Park, Lee, Kim, Lee
FALCON	•			**Prest**, Fouque, Hoffstein, Kirchner, Lyubashevsky, Pornin, Ricosset, ...
FrodoKEM	•			**Naehrig**, Alkim, Bos, Ducas, Easterbrook, LaMacchia, Longa, Mironov, ...
Giophantus	•			**Akiyama**, Goto, Okumura, Takagi, Nuida, Hanaoka, Shimizu, Ikematsu
KCL	•			**Zhao**, Jin, Gong, Sui
KINDI	•			**El Bansarkhani**
LAC	•			**Lu**, Liu, Jia, Xue, He, Zhang
LIMA	•			**Smart**, Albrecht, Lindell, Orsini, Osheter, Paterson, Peer
Lizard	•			**Cheon**, Park, Lee, Kim, Song, Hong, Kim, Kim, Hong, Yun, Kim, Park, ...
LOTUS		•		**Phong**, Hayashi, Aono, Moriai
NewHope	•			**Pöppelmann**, Alkim, Avanzi, Bos, Ducas, De La Piedra, Schwabe, Stebila
NTRU	◦	◦		**Zhang**, Chen, Hoffstein, Hülsing, Rijneveld, Schanck, Schwabe, Whyte
NTRU Prime		•		**Bernstein**, Chuengsatiansup, Lange, Van Vredendaal
Odd Manhattan			•	**Plantard**
pqNTRUSign	◦	◦		**Zhang**, Chen, Hoffstein, Whyte
qTESLA	•			**Bindel**, Akleylek, Alkim, Barreto, Buchmann, Eaton, Gutoski, Krämer, ...
Round5	•			**Garcia-Morchon**, Saarinen, Zhang, Bhattacharya, Rietman, Tolhuizen, ...
SABER	•			**D'Anvers**, Karmakar, Roy, Vercauteren
Three Bears	•			**Hamburg**
Titanium	•			**Steinfeld**, Sakzad, Zhao
Totals:	**11**	**2**	**0**	**Total: 12 proposals with SVP hardness estimates**

*Not included in the overview: Compact LWE, Mersenne, Ramstake, ...

Estimate all the {LWE, NTRU} schemes! 

Model	Schemes
0.292β 0.265β	CRYSTALS [LDK+17,SAB+17] SABER [DKRV17] Falcon [PFH+17] ThreeBears [Ham17] HILA5 [Saa17] Titanium [SSZ17] KINDI [Ban17] NTRU HRSS [SHRS17] LAC [LLJ+17] NTRUEncrypt [ZCHW17a] New Hope [PAA+17] pqNTRUSign [ZCHW17b]
$0.292\beta + 16.4$ $0.265\beta + 16.4$	LIMA [SAL+17]
0.368β 0.2975β	NTRU HRSS [SHRS17]
$0.292\beta + \log(\beta)$ $0.265\beta + \log(\beta)$	Frodo [NAB+17] KCL [ZjGS17] Lizard [CPL+17] Round2 [GMZB+17]
$0.292\beta + 16.4 + \log(8d)$	Ding Key Exchange [DTGW17] EMBLEM [SPL+17]
$0.265\beta + 16.4 + \log(8d)$	qTESLA [BAA+17]
$0.187\beta \log \beta - 1.019\beta + 16.1$	NTRU HRSS [SHRS17] pqNTRUSign [ZCHW17b] NTRUEncrypt [ZCHW17a]
$\frac{1}{2}(0.187\beta \log \beta - 1.019\beta + 16.1)$	NTRU HRSS [SHRS17]
$0.000784\beta^2 + 0.366\beta - 0.9 + \log(8d)$	NTRU Prime [BCLvV17]
$0.125\beta \log \beta - 0.755\beta + 2.25$	LOTUS [PHAM17]

Estimate all the {LWE, NTRU} schemes! 

Model	Schemes
	CRYSTALS [LDK+17,SAB+17]
	SABER [DKRV17]
	Falcon [PFH+17]
	ThreeBears [Ham17]
	HILA5 [Saa17]
0.292β	
0.265β	
	NTRU HRSS [SHRS17]
	LAC [LLJ+17]
	NTRUEncrypt [ZCHW17a]
	New Hope [PAA+17]
	pqNTRUSign [ZCHW17b]
$0.292\beta + 16.4$	
$0.265\beta + 16.4$	
0.368β	NTRU HRSS [SHRS17]
0.2975β	
	Frodo [NAB+17]
$0.292\beta + \log(\beta)$	
$0.265\beta + \log(\beta)$	
	Round2 [GMZB+17]
$0.292\beta + 16.4 + \log(8d)$	
$0.265\beta + 16.4 + \log(8d)$	qTESLA [BAA+17]
	NTRU HRSS [SHRS17]
$0.187\beta \log \beta - 1.019\beta + 16.1$	
	NTRUEncrypt [ZCHW17a]
$\frac{1}{2}(0.187\beta \log \beta - 1.019\beta + 16.1)$	NTRU HRSS [SHRS17]
$0.000784\beta^2 + 0.366\beta - 0.9 + \log(8d)$	NTRU Prime [BCLvV17]
$0.125\beta \log \beta - 0.755\beta + 2.25$	

SVP hardness

NIST submissions

Most common hardness estimates:

- Cost of BKZ(β) $\geq$ Cost of SVP(β)
- Ignore space complexity, ignore $o(n)$ in time complexity
- Classical sieving: $2^{0.292n}$ time [BDGL16]
- Quantum sieving: $2^{0.265n}$ time [Laa16]
- "Paranoid bound": $2^{0.208n}$ time

SVP hardness

NIST submissions

Most common hardness estimates:

- Cost of BKZ(β) $\geq$ Cost of SVP(β)
- Ignore space complexity, ignore $o(n)$ in time complexity
- Classical sieving: $2^{0.292n}$ time [BDGL16]
- Quantum sieving: $2^{0.265n}$ time [Laa16]
- "Paranoid bound": $2^{0.208n}$ time

Classical lower bound: $2^{0.277n}$ time

Conclusion

Lattice-based cryptography

- Security relies on hardness of finding short vectors
- State-of-the-art approach: BKZ with fast SVP subroutine
- Cost of BKZ dominated by cost of exact SVP algorithm

SVP algorithms

- Lattice enumeration: Brute-force approach
- Lattice sieving: Memory-intensive approach

SVP hardness

- Theory: Sieving superior in high dimensions
- Practice: Sieving superior in moderate/high dimensions
- Hardness estimates: Commonly based on sieving

Questions?