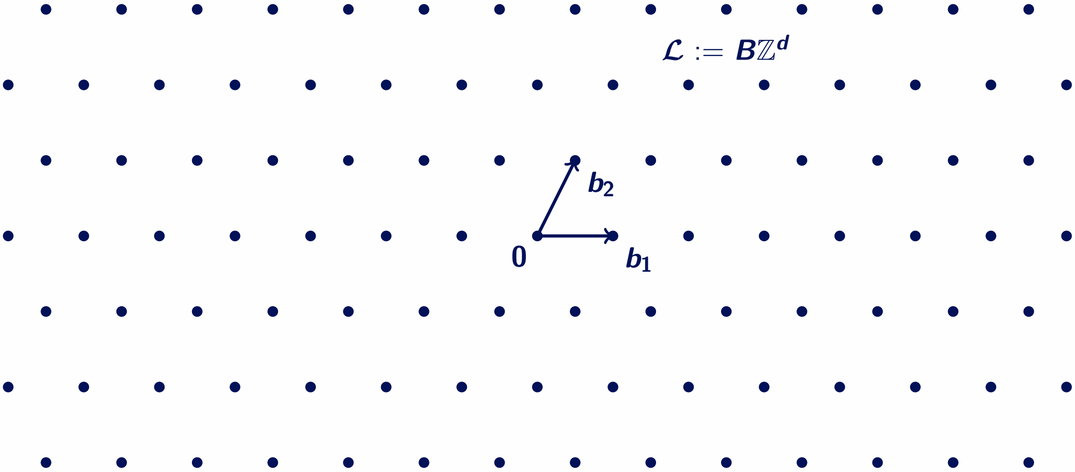
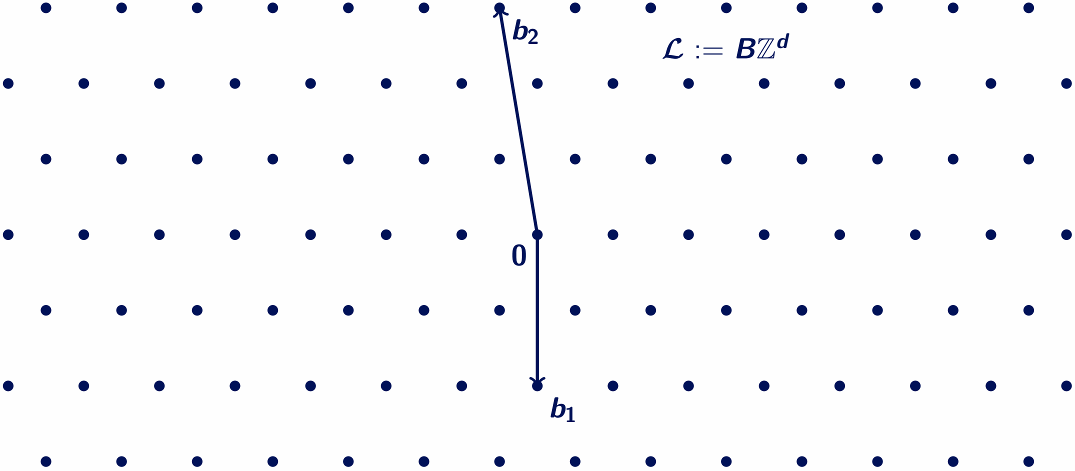
A canonical form for positive definite matrices

Mathieu Dutour Sikirić (Rudjer Bosković Institute),
Anna Haensch (Duquesne University),
John Voight (Dartmouth College),
Wessel van Woerden (CWI).

Lattice


$\mathcal{L} := \boldsymbol{B}\mathbb{Z}^{d}$
$\boldsymbol{b}_2$
$\boldsymbol{0}$
$\boldsymbol{b}_1$

Lattice



Lattice Equality

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$

$$\iff$$

$$\exists \boldsymbol{U} \in \mathsf{GL}_{\boldsymbol{d}}(\mathbb{Z}) : \boldsymbol{B}_1 \boldsymbol{U} = \boldsymbol{B}_2$$

Lattice Equality

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$
$$\iff$$
$$\exists \boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z}) : \boldsymbol{B}_1 \boldsymbol{U} = \boldsymbol{B}_2$$

- Linear algebra: $\boldsymbol{U} := \boldsymbol{B}_1^{\dagger} \boldsymbol{B}_2$.

Lattice Equality

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$
$$\iff$$
$$\exists \boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z}) : \boldsymbol{B}_1 \boldsymbol{U} = \boldsymbol{B}_2$$

- Linear algebra: $\boldsymbol{U} := \boldsymbol{B}_1^{\dagger} \boldsymbol{B}_2$.
- What if we have many bases $\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_m$?

Lattice Equality

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$
$$\iff$$
$$\exists \boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z}) : \boldsymbol{B}_1 \boldsymbol{U} = \boldsymbol{B}_2$$

- Linear algebra: $\boldsymbol{U} := \boldsymbol{B}_1^{\dagger} \boldsymbol{B}_2$.
- What if we have many bases $\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_m$?
- $\boldsymbol{O}(\boldsymbol{m}^2)$ pairwise checks.

Hermite normal form

- $\boldsymbol{B} \in \mathbb{Z}^{\boldsymbol{d}' \times \boldsymbol{d}}$ of full column rank.

Hermite normal form

- $\boldsymbol{B} \in \mathbb{Z}^{\boldsymbol{d'} \times \boldsymbol{d}}$ of full column rank.
- $\boldsymbol{H} = \mathrm{HNF}(\boldsymbol{B})$ is the ***unique*** basis of $\mathcal{L}(\boldsymbol{B})$ s.t.
  - $\boldsymbol{H}$ is in column echelon form.
  - Left of pivot: non-negative and strictly smaller than pivot.

Hermite normal form

- $\boldsymbol{B} \in \mathbb{Z}^{d' \times d}$ of full column rank.
- $\boldsymbol{H} = \mathrm{HNF}(\boldsymbol{B})$ is the ***unique*** basis of $\mathcal{L}(\boldsymbol{B})$ s.t.
  - $\boldsymbol{H}$ is in column echelon form.
  - Left of pivot: non-negative and strictly smaller than pivot.
- Example:

$$\begin{pmatrix} 5 & 0 & 0 \\ 5 & 6 & 0 \\ 0 & 0 & 1 \\ 4 & 3 & 0 \end{pmatrix}$$

Hermite normal form

- $\boldsymbol{B} \in \mathbb{Z}^{d' \times d}$ of full column rank.
- $\boldsymbol{H} = \mathrm{HNF}(\boldsymbol{B})$ is the ***unique*** basis of $\mathcal{L}(\boldsymbol{B})$ s.t.
  - $\boldsymbol{H}$ is in column echelon form.
  - Left of pivot: non-negative and strictly smaller than pivot.
- Example:

$$\begin{pmatrix} 5 & 0 & 0 \\ 5 & 6 & 0 \\ 0 & 0 & 1 \\ 4 & 3 & 0 \end{pmatrix}$$

- Polytime algorithm to compute HNF (using LLL to prevent coefficient blow-up).

Hermite normal form

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$

$$\iff$$

$$\text{HNF}(\boldsymbol{B}_1) = \text{HNF}(\boldsymbol{B}_2)$$

Hermite normal form

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$
$$\Longleftrightarrow$$
$$\mathrm{HNF}(\boldsymbol{B}_1) = \mathrm{HNF}(\boldsymbol{B}_2)$$

- $\boldsymbol{B} \mapsto \mathrm{HNF}(\boldsymbol{B})$ is what we call a **canonical** function.

Hermite normal form

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$
$$\iff$$
$$\mathrm{HNF}(\boldsymbol{B}_1) = \mathrm{HNF}(\boldsymbol{B}_2)$$

- $\boldsymbol{B} \mapsto \mathrm{HNF}(\boldsymbol{B})$ is what we call a **canonical** function.
- $\mathrm{HNF}(\boldsymbol{BU}) = \mathrm{HNF}(\boldsymbol{B})$ for all $\boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$.

Hermite normal form

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$
$$\iff$$
$$\mathrm{HNF}(\boldsymbol{B}_1) = \mathrm{HNF}(\boldsymbol{B}_2)$$

- $\boldsymbol{B} \mapsto \mathrm{HNF}(\boldsymbol{B})$ is what we call a **canonical** function.
- $\mathrm{HNF}(\boldsymbol{BU}) = \mathrm{HNF}(\boldsymbol{B})$ for all $\boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$.
- Compute $\mathrm{HNF}(\boldsymbol{B}_1), \ldots, \mathrm{HNF}(\boldsymbol{B}_{\boldsymbol{m}})$.

Hermite normal form

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$
$$\iff$$
$$\mathrm{HNF}(\boldsymbol{B}_1) = \mathrm{HNF}(\boldsymbol{B}_2)$$

- $\boldsymbol{B} \mapsto \mathrm{HNF}(\boldsymbol{B})$ is what we call a **canonical** function.
- $\mathrm{HNF}(\boldsymbol{BU}) = \mathrm{HNF}(\boldsymbol{B})$ for all $\boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$.
- Compute $\mathrm{HNF}(\boldsymbol{B}_1), \ldots, \mathrm{HNF}(\boldsymbol{B}_{\boldsymbol{m}})$.
- Only $\boldsymbol{O}(\boldsymbol{m})$ queries/insertions in a hash table.

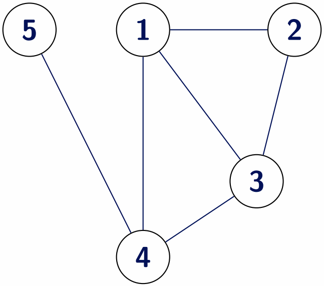
Hermite normal form

$$\mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2)$$
$$\iff$$
$$\mathrm{HNF}(\boldsymbol{B}_1) = \mathrm{HNF}(\boldsymbol{B}_2)$$

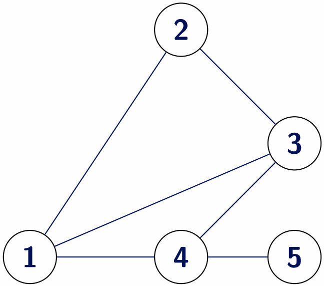
- $\boldsymbol{B} \mapsto \mathrm{HNF}(\boldsymbol{B})$ is what we call a **canonical** function.
- $\mathrm{HNF}(\boldsymbol{BU}) = \mathrm{HNF}(\boldsymbol{B})$ for all $\boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$.
- Compute $\mathrm{HNF}(\boldsymbol{B}_1), \ldots, \mathrm{HNF}(\boldsymbol{B}_{\boldsymbol{m}})$.
- Only $\boldsymbol{O}(\boldsymbol{m})$ queries/insertions in a hash table.
- Variant can be used for left action: $\mathrm{HNF}_{\boldsymbol{L}}(\boldsymbol{UB}^{\boldsymbol{t}}) = \mathrm{HNF}_{\boldsymbol{L}}(\boldsymbol{B}^{\boldsymbol{t}})$.

Graph Isomorphism

Graph $\boldsymbol{G} = (\boldsymbol{V} = [\boldsymbol{n}], \boldsymbol{E} \subset \boldsymbol{V} \times \boldsymbol{V})$

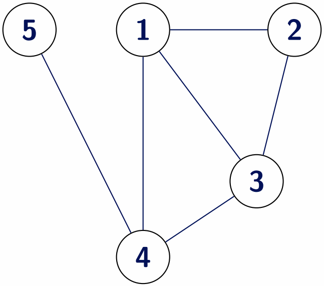


Graph $\boldsymbol{G'} = (\boldsymbol{V} = [\boldsymbol{n}], \boldsymbol{E'})$

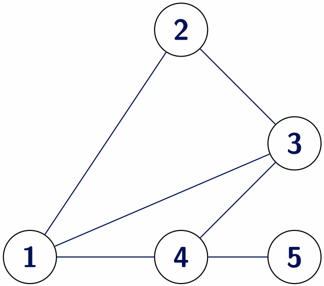


Graph Isomorphism

Graph $G = (V = [n], E \subset V \times V)$



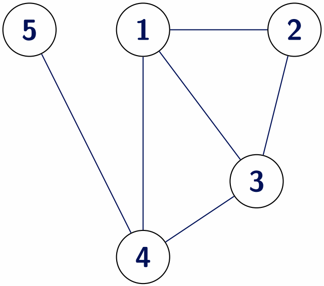
Graph $G' = (V = [n], E')$



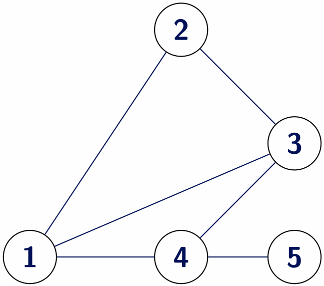
- Graph equality: $E = E'$.

Graph Isomorphism

Graph $G = (V = [n], E \subset V \times V)$



Graph $G' = (V = [n], E')$

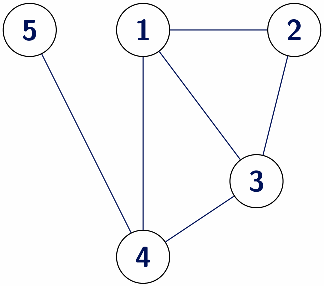


- Graph equality: $E = E'$.
- Graph Automorphisms: $\text{Stab}(G) = \{\sigma \in \text{Sym}_{|V|} : \sigma(E) = E\}$.

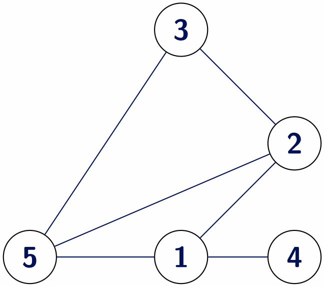
$$\sigma(E) := \{(\sigma(i), \sigma(j)) : (i, j) \in E\}$$

Graph Isomorphism

Graph $G = (V = [n], E \subset V \times V)$



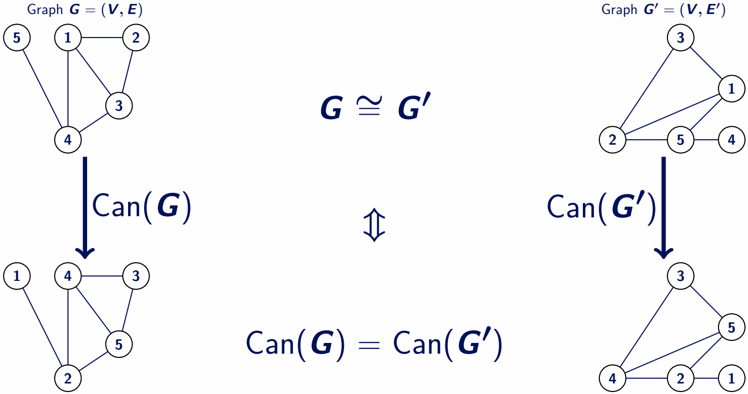
Graph $G' = (V = [n], E')$



- Graph Isomorphism: $G \cong G' \Leftrightarrow \sigma(E) = E'$ for some $\sigma \in \mathrm{Sym}_n$.

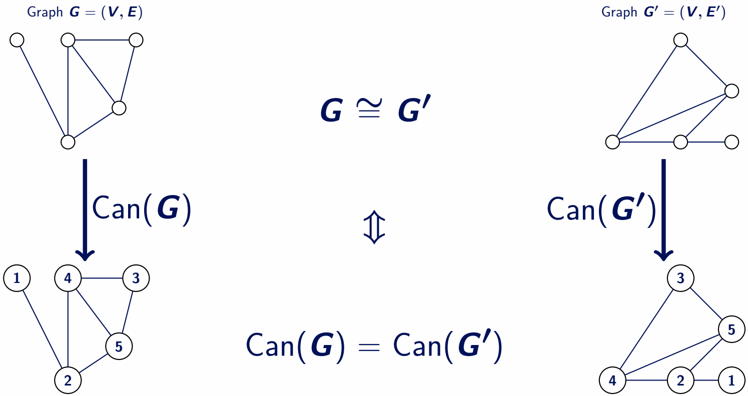
Canonical Graph Ordering

- Give a vertex ordering of the graph purely based on the graph structure.



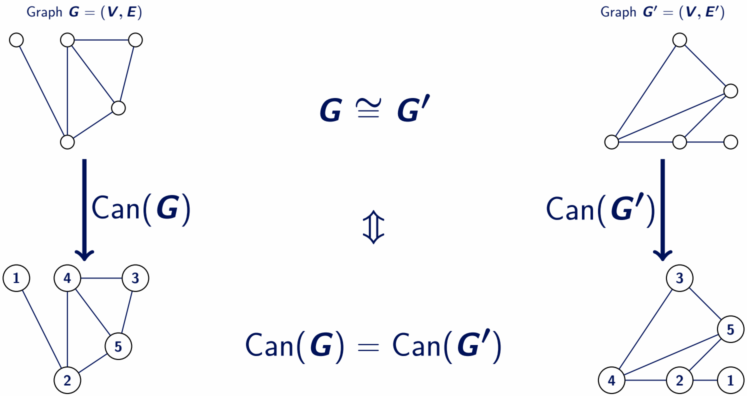
Canonical Graph Ordering

- Give a vertex ordering of the graph purely based on the graph structure.



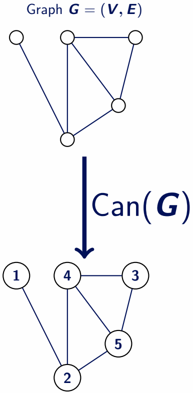
Canonical Graph Ordering

- Give a vertex ordering of the graph purely based on the graph structure.
- Example: vertex order that minimizes $\boldsymbol{E}$ under a lexicographic ordering.



Canonical Graph Ordering

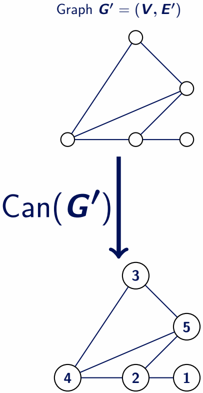
- Give a vertex ordering of the graph purely based on the graph structure.
- Example: vertex order that minimizes $\boldsymbol{E}$ under a lexicographic ordering.
- Permutation (relative to input) is unique up to $\mathrm{Stab}(\boldsymbol{G})$.



$$G \cong G'$$

$$\Updownarrow$$

$$\mathrm{Can}(G) = \mathrm{Can}(G')$$



Complexity

- More generally for weighted complete graphs $\boldsymbol{G}$ with weights $\boldsymbol{W} = (\boldsymbol{w_{ij}})_{\boldsymbol{ij}}$:

$$G \cong G' \iff \exists \sigma \in \mathrm{Sym}_n : \; \forall i, j \;\; w_{\sigma(i)\sigma(j)} = w'_{ij}$$

Complexity

- More generally for weighted complete graphs G with weights $W = (w_{ij})_{ij}$:

$$G \cong G' \iff \exists \sigma \in \mathrm{Sym}_n : \ \forall i, j \ \ w_{\sigma(i)\sigma(j)} = w'_{ij}$$

- Several canonical graph ordering implementations exist: nauty, bliss, traces.

Complexity

- More generally for weighted complete graphs G with weights $W = (w_{ij})_{ij}$:

$$G \cong G' \iff \exists \sigma \in \mathrm{Sym}_n : \; \forall i, j \;\; w_{\sigma(i)\sigma(j)} = w'_{ij}$$

- Several canonical graph ordering implementations exist: nauty, bliss, traces.
- **Practical Complexity:** Most graphs up to thousands of vertices are no problem.

Complexity

- More generally for weighted complete graphs G with weights $W = (w_{ij})_{ij}$:

$$G \cong G' \iff \exists \sigma \in \mathrm{Sym}_n : \ \forall i, j \ \ w_{\sigma(i)\sigma(j)} = w'_{ij}$$

- Several canonical graph ordering implementations exist: nauty, bliss, traces.
- **Practical Complexity:** Most graphs up to thousands of vertices are no problem.
- **Theoretical:**

Complexity

- More generally for weighted complete graphs G with weights $W = (w_{ij})_{ij}$:

$$G \cong G' \iff \exists \sigma \in \mathrm{Sym}_n : \ \forall i, j \ \ w_{\sigma(i)\sigma(j)} = w'_{ij}$$

- Several canonical graph ordering implementations exist: nauty, bliss, traces.
- **Practical Complexity:** Most graphs up to thousands of vertices are no problem.
- **Theoretical:**
  - L. Babai, *Graph isomorphism in quasipolynomial time*, 2015.

Complexity

- More generally for weighted complete graphs G with weights $W = (w_{ij})_{ij}$:

$$G \cong G' \iff \exists \sigma \in \mathrm{Sym}_n : \ \forall i, j \ \ w_{\sigma(i)\sigma(j)} = w'_{ij}$$

- Several canonical graph ordering implementations exist: nauty, bliss, traces.
- **Practical Complexity:** Most graphs up to thousands of vertices are no problem.
- **Theoretical:**
  - L. Babai, *Graph isomorphism in quasipolynomial time*, 2015.
  - $\exp(\log(|V|)^{O(1)})$.

Complexity

- More generally for weighted complete graphs G with weights $W = (w_{ij})_{ij}$:

$$G \cong G' \iff \exists \sigma \in \mathrm{Sym}_n : \ \forall i, j \ \ w_{\sigma(i)\sigma(j)} = w'_{ij}$$

- Several canonical graph ordering implementations exist: nauty, bliss, traces.
- **Practical Complexity:** Most graphs up to thousands of vertices are no problem.
- **Theoretical:**
  - L. Babai, *Graph isomorphism in quasipolynomial time*, 2015.
  - $\exp(\log(|V|)^{O(1)})$.
  - Retracted in 2017 after H.A. Helfgott found a flaw in the proof.

Complexity

- More generally for weighted complete graphs G with weights $W = (w_{ij})_{ij}$:

$$G \cong G' \iff \exists \sigma \in \mathrm{Sym}_n : \ \forall i, j \ \ w_{\sigma(i)\sigma(j)} = w'_{ij}$$

- Several canonical graph ordering implementations exist: nauty, bliss, traces.
- **Practical Complexity:** Most graphs up to thousands of vertices are no problem.
- **Theoretical:**
  - L. Babai, *Graph isomorphism in quasipolynomial time*, 2015.
  - $\exp(\log(|V|)^{O(1)})$.
  - Retracted in 2017 after H.A. Helfgott found a flaw in the proof.
  - Almost immediately fixed, confirmed by H.A. Helfgott.

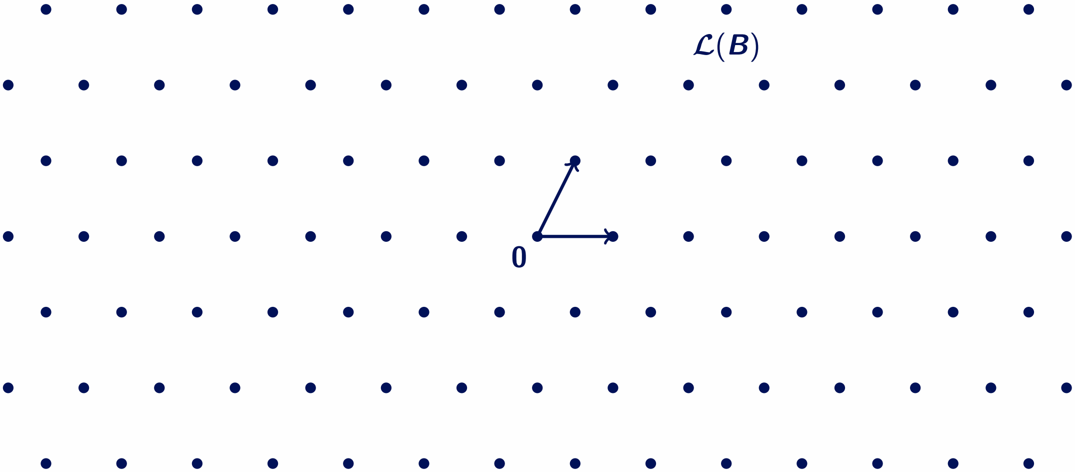
Complexity

- More generally for weighted complete graphs G with weights $W = (w_{ij})_{ij}$:

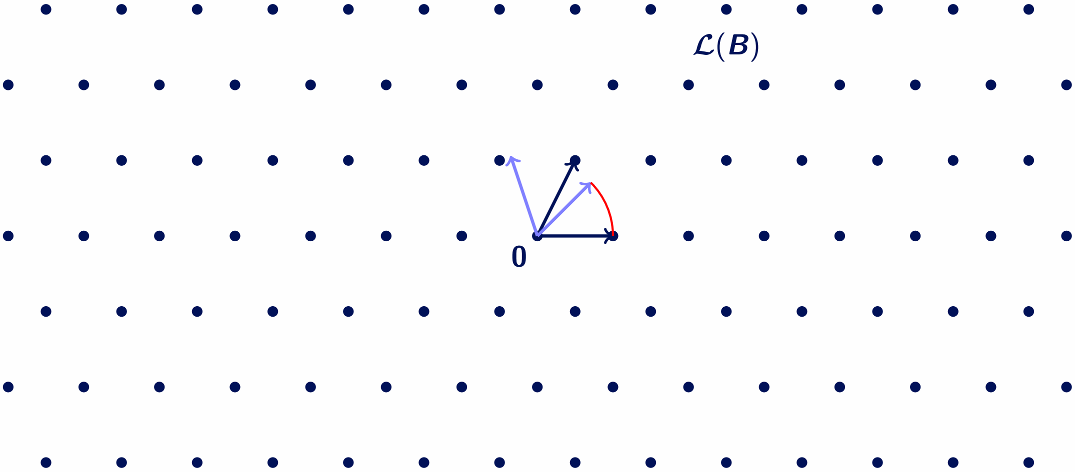
$$G \cong G' \iff \exists \sigma \in \mathrm{Sym}_n : \ \forall i, j \ \ w_{\sigma(i)\sigma(j)} = w'_{ij}$$

- Several canonical graph ordering implementations exist: nauty, bliss, traces.
- **Practical Complexity:** Most graphs up to thousands of vertices are no problem.
- **Theoretical:**
  - L. Babai, *Graph isomorphism in quasipolynomial time*, 2015.
  - $\exp(\log(|V|)^{O(1)})$.
  - Retracted in 2017 after H.A. Helfgott found a flaw in the proof.
  - Almost immediately fixed, confirmed by H.A. Helfgott.
  - L. Babai, *Canonical form for graphs in quasipolynomial time*, 2019.

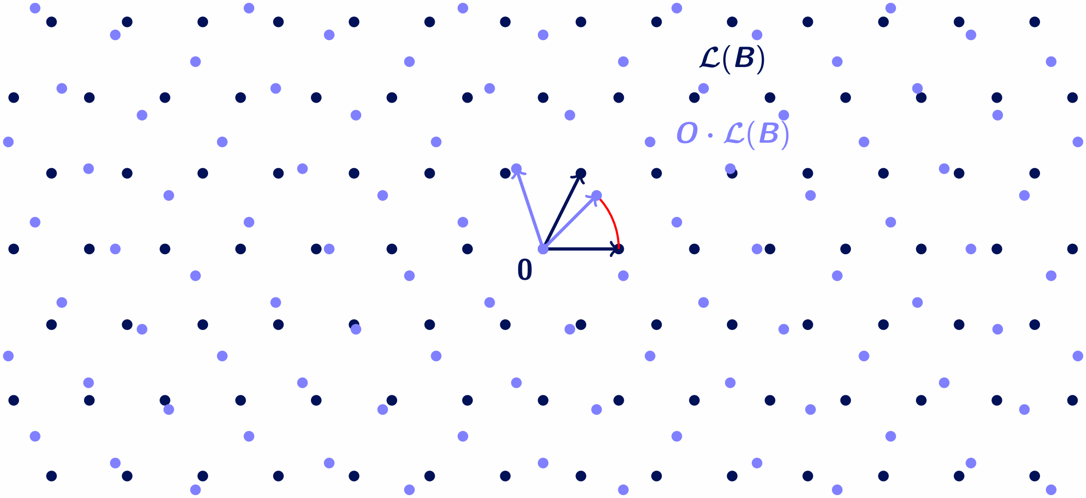
Lattice Isomorphism



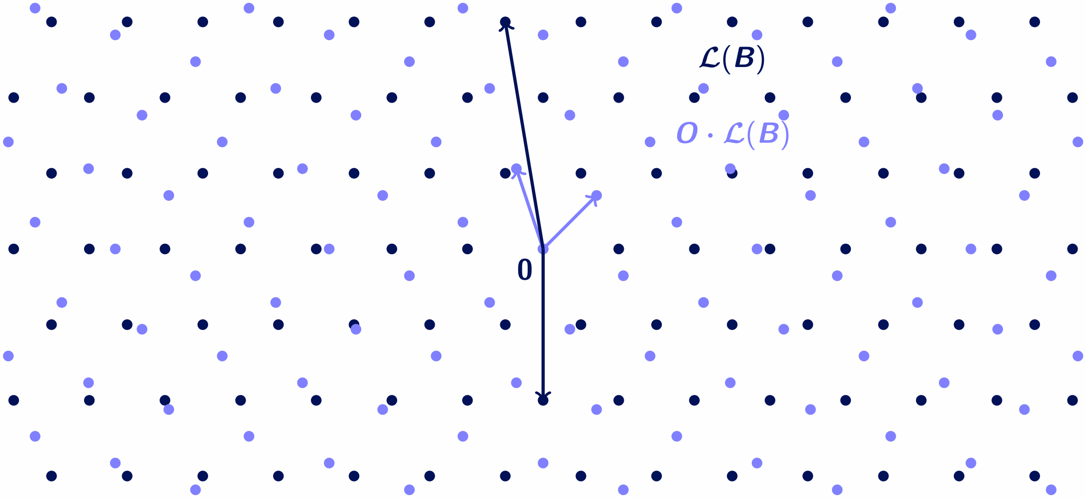
Lattice Isomorphism



Lattice Isomorphism



Lattice Isomorphism



Lattice Isomorphism

$$\mathcal{L}(\boldsymbol{B}_1) \cong \mathcal{L}(\boldsymbol{B}_2)$$

$$\Longleftrightarrow$$

$$\boldsymbol{O} \cdot \mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2) \qquad \text{for some } \boldsymbol{O} \in \boldsymbol{O}_{\boldsymbol{d}}(\mathbb{R})$$

$$\Longleftrightarrow$$

$$\boldsymbol{O} \cdot \boldsymbol{B}_1 \cdot \boldsymbol{U} = \boldsymbol{B}_2 \qquad \text{for some } \boldsymbol{O} \in \boldsymbol{O}_{\boldsymbol{d}}(\mathbb{R}), \boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$$

Lattice Isomorphism

$$\mathcal{L}(\boldsymbol{B}_1) \cong \mathcal{L}(\boldsymbol{B}_2)$$

$$\Longleftrightarrow$$

$$\boldsymbol{O} \cdot \mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2) \qquad \text{for some } \boldsymbol{O} \in \boldsymbol{O}_{\boldsymbol{d}}(\mathbb{R})$$

$$\Longleftrightarrow$$

$$\boldsymbol{O} \cdot \boldsymbol{B}_1 \cdot \boldsymbol{U} = \boldsymbol{B}_2 \qquad \text{for some } \boldsymbol{O} \in \boldsymbol{O}_{\boldsymbol{d}}(\mathbb{R}), \boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$$

- If either $\boldsymbol{O}$ or $\boldsymbol{U}$ is trivial: linear algebra.

Lattice Isomorphism

$$\mathcal{L}(\boldsymbol{B}_1) \cong \mathcal{L}(\boldsymbol{B}_2)$$

$$\Longleftrightarrow$$

$$\boldsymbol{O} \cdot \mathcal{L}(\boldsymbol{B}_1) = \mathcal{L}(\boldsymbol{B}_2) \qquad \text{for some } \boldsymbol{O} \in \boldsymbol{O}_{\boldsymbol{d}}(\mathbb{R})$$

$$\Longleftrightarrow$$

$$\boldsymbol{O} \cdot \boldsymbol{B}_1 \cdot \boldsymbol{U} = \boldsymbol{B}_2 \qquad \text{for some } \boldsymbol{O} \in \boldsymbol{O}_{\boldsymbol{d}}(\mathbb{R}), \boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$$

$$\Longleftrightarrow$$

$$\boldsymbol{U}^t \boldsymbol{B}_1^t \boldsymbol{B}_1 \boldsymbol{U} = \boldsymbol{B}_2^t \boldsymbol{B}_2 \qquad \text{for some } \boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$$

- If either $\boldsymbol{O}$ or $\boldsymbol{U}$ is trivial: linear algebra.
- Use $\boldsymbol{O}^t \boldsymbol{O} = \boldsymbol{I}$ to remove the orthonormal transformation.

Quadratic Forms

- The gram matrix $\boldsymbol{A} = \boldsymbol{B}^t \boldsymbol{B} \in \mathcal{S}^d_{>0}$ induces a quadratic form:

$$\boldsymbol{A} : \boldsymbol{x} \mapsto \boldsymbol{x}^t \boldsymbol{A} \boldsymbol{x} \qquad \text{for } \boldsymbol{x} \in \mathbb{Z}^d$$

Quadratic Forms

- The gram matrix $\boldsymbol{A} = \boldsymbol{B}^t\boldsymbol{B} \in \mathcal{S}^d_{>0}$ induces a quadratic form:

$$\boldsymbol{A} : \boldsymbol{x} \mapsto \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} \qquad \text{for } \boldsymbol{x} \in \mathbb{Z}^d$$

- Geometric information remains: $\langle \boldsymbol{Bx}, \boldsymbol{By} \rangle = \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{y}$.

Quadratic Forms

- The gram matrix $A = B^t B \in \mathcal{S}^d_{>0}$ induces a quadratic form:

$$A : x \mapsto x^t A x \qquad \text{for } x \in \mathbb{Z}^d$$

- Geometric information remains: $\langle Bx, By \rangle = x^t A y$.
- A_1 is arithmetically equivalent ($\sim$) to A_2 if

$$U^t A_1 U = A_2 \qquad \text{for some } U \in \mathrm{GL}_d(\mathbb{Z}).$$

Quadratic Forms

- The gram matrix $\boldsymbol{A} = \boldsymbol{B}^t\boldsymbol{B} \in \mathcal{S}^d_{>0}$ induces a quadratic form:

$$\boldsymbol{A} : \boldsymbol{x} \mapsto \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} \qquad \text{for } \boldsymbol{x} \in \mathbb{Z}^d$$

- Geometric information remains: $\langle \boldsymbol{Bx}, \boldsymbol{By} \rangle = \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{y}$.
- $\boldsymbol{A}_1$ is arithmetically equivalent ($\sim$) to $\boldsymbol{A}_2$ if

$$\boldsymbol{U}^t\boldsymbol{A}_1\boldsymbol{U} = \boldsymbol{A}_2 \qquad \text{for some } \boldsymbol{U} \in \mathrm{GL}_d(\mathbb{Z}).$$

- $\boldsymbol{U}$ above is unique up to $\mathrm{Stab}(\boldsymbol{A}_1) := \{\boldsymbol{S} \in \mathrm{GL}_d(\mathbb{Z}) : \boldsymbol{S}^t\boldsymbol{A}_1\boldsymbol{S} = \boldsymbol{A}_1\}$.

Quadratic Forms

- The gram matrix $\boldsymbol{A} = \boldsymbol{B}^t\boldsymbol{B} \in \mathcal{S}^d_{>0}$ induces a quadratic form:

$$\boldsymbol{A} : \boldsymbol{x} \mapsto \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} \qquad \text{for } \boldsymbol{x} \in \mathbb{Z}^d$$

- Geometric information remains: $\langle \boldsymbol{B}\boldsymbol{x}, \boldsymbol{B}\boldsymbol{y} \rangle = \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{y}$.
- $\boldsymbol{A}_1$ is arithmetically equivalent ($\sim$) to $\boldsymbol{A}_2$ if

$$\boldsymbol{U}^t\boldsymbol{A}_1\boldsymbol{U} = \boldsymbol{A}_2 \qquad \text{for some } \boldsymbol{U} \in \mathrm{GL}_d(\mathbb{Z}).$$

- $\boldsymbol{U}$ above is unique up to $\mathrm{Stab}(\boldsymbol{A}_1) := \{\boldsymbol{S} \in \mathrm{GL}_d(\mathbb{Z}) : \boldsymbol{S}^t\boldsymbol{A}_1\boldsymbol{S} = \boldsymbol{A}_1\}$.
- Can we construct a canonical function $\mathrm{Can} : \mathcal{S}^d_{>0} \to \mathcal{S}^d_{>0}$ such that
  (1) $\mathrm{Can}(\boldsymbol{A}) \sim \boldsymbol{A}$.

Quadratic Forms

- The gram matrix $\boldsymbol{A} = \boldsymbol{B}^t\boldsymbol{B} \in \mathcal{S}^d_{>0}$ induces a quadratic form:

$$\boldsymbol{A} : \boldsymbol{x} \mapsto \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} \qquad \text{for } \boldsymbol{x} \in \mathbb{Z}^d$$

- Geometric information remains: $\langle \boldsymbol{Bx}, \boldsymbol{By} \rangle = \boldsymbol{x}^t\boldsymbol{Ay}$.
- $\boldsymbol{A}_1$ is arithmetically equivalent ($\sim$) to $\boldsymbol{A}_2$ if

$$\boldsymbol{U}^t\boldsymbol{A}_1\boldsymbol{U} = \boldsymbol{A}_2 \qquad \text{for some } \boldsymbol{U} \in \mathrm{GL}_d(\mathbb{Z}).$$

- $\boldsymbol{U}$ above is unique up to $\mathrm{Stab}(\boldsymbol{A}_1) := \{\boldsymbol{S} \in \mathrm{GL}_d(\mathbb{Z}) : \boldsymbol{S}^t\boldsymbol{A}_1\boldsymbol{S} = \boldsymbol{A}_1\}$.
- Can we construct a canonical function $\mathrm{Can} : \mathcal{S}^d_{>0} \to \mathcal{S}^d_{>0}$ such that
  (1) $\mathrm{Can}(\boldsymbol{A}) \sim \boldsymbol{A}$.
  (2) $\boldsymbol{A}_1 \sim \boldsymbol{A}_2 \iff \mathrm{Can}(\boldsymbol{A}_1) = \mathrm{Can}(\boldsymbol{A}_2)$.

Quadratic Forms

- The gram matrix $\boldsymbol{A} = \boldsymbol{B}^t\boldsymbol{B} \in \mathcal{S}^d_{>0}$ induces a quadratic form:

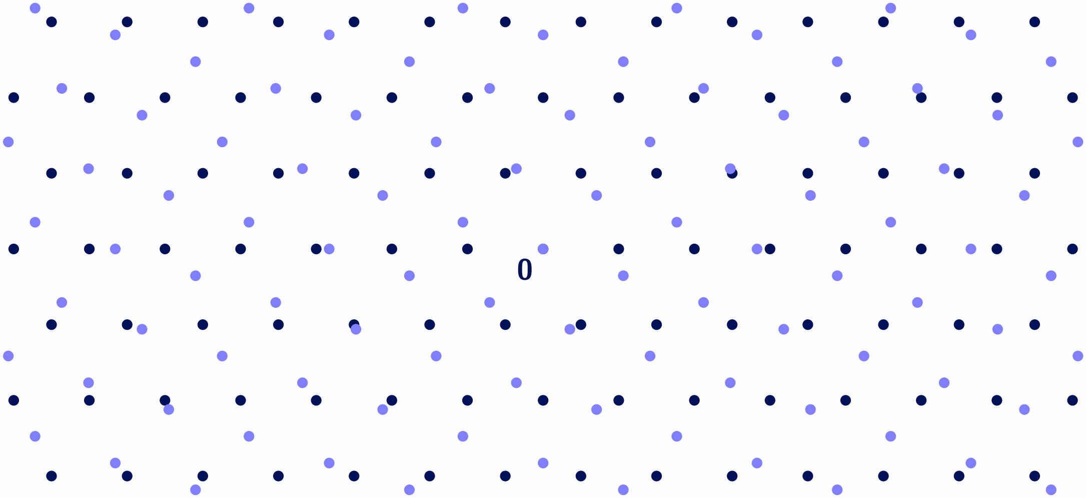
$$\boldsymbol{A} : \boldsymbol{x} \mapsto \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} \qquad \text{for } \boldsymbol{x} \in \mathbb{Z}^d$$

- Geometric information remains: $\langle \boldsymbol{Bx}, \boldsymbol{By} \rangle = \boldsymbol{x}^t\boldsymbol{A}\boldsymbol{y}$.
- $\boldsymbol{A}_1$ is arithmetically equivalent ($\sim$) to $\boldsymbol{A}_2$ if

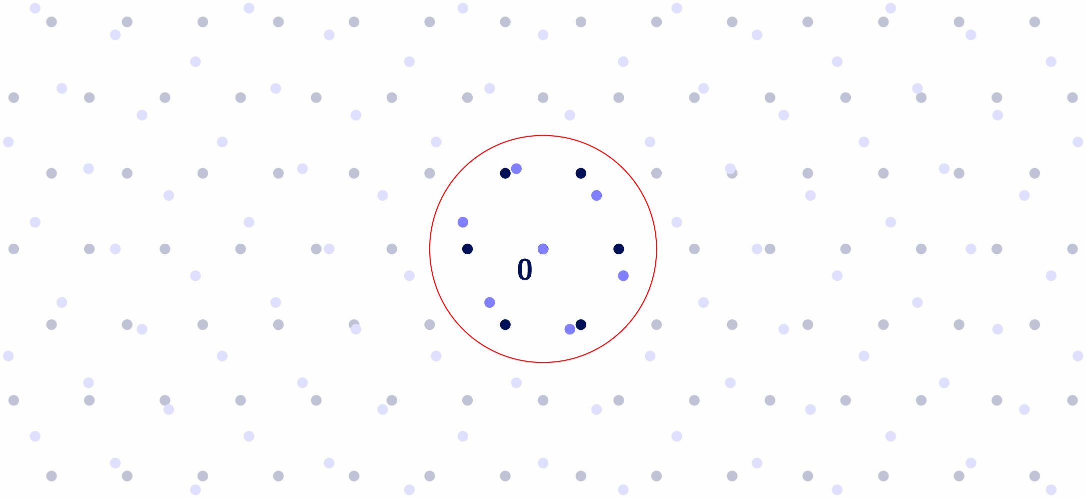
$$\boldsymbol{U}^t\boldsymbol{A}_1\boldsymbol{U} = \boldsymbol{A}_2 \qquad \text{for some } \boldsymbol{U} \in \mathrm{GL}_d(\mathbb{Z}).$$

- $\boldsymbol{U}$ above is unique up to $\mathrm{Stab}(\boldsymbol{A}_1) := \{\boldsymbol{S} \in \mathrm{GL}_d(\mathbb{Z}) : \boldsymbol{S}^t\boldsymbol{A}_1\boldsymbol{S} = \boldsymbol{A}_1\}$.
- Can we construct a canonical function $\mathrm{Can} : \mathcal{S}^d_{>0} \to \mathcal{S}^d_{>0}$ such that
  (1) $\mathrm{Can}(\boldsymbol{A}) \sim \boldsymbol{A}$.
  (2) $\mathrm{Can}(\boldsymbol{U}^t\boldsymbol{A}\boldsymbol{U}) = \mathrm{Can}(\boldsymbol{A})$ for all $\boldsymbol{U} \in \mathrm{GL}_d(\mathbb{Z})$.

Characteristic Vector Set



Characteristic Vector Set



Characteristic Vector Set

- $\mathcal{V} : \boldsymbol{A} \mapsto \mathcal{V}(\boldsymbol{A}) \subset \mathbb{Z}^{\boldsymbol{d}}$ is a **characteristic vector set** function if

Characteristic Vector Set

- $\mathcal{V} : \boldsymbol{A} \mapsto \mathcal{V}(\boldsymbol{A}) \subset \mathbb{Z}^{\boldsymbol{d}}$ is a **characteristic vector set** function if

  (1) $\mathcal{V}(\boldsymbol{A})$ generates $\mathbb{Z}^{\boldsymbol{d}}$ (as a $\mathbb{Z}$-module).

Characteristic Vector Set

- $\mathcal{V} : A \mapsto \mathcal{V}(A) \subset \mathbb{Z}^d$ is a **characteristic vector set** function if
  (1) $\mathcal{V}(A)$ generates $\mathbb{Z}^d$ (as a $\mathbb{Z}$-module).
  (2) $\mathcal{V}(U^t A U) = U^{-1}\mathcal{V}(A)$ for all $U \in \mathrm{GL}_d(\mathbb{Z})$.

Characteristic Vector Set

- $\mathcal{V} : A \mapsto \mathcal{V}(A) \subset \mathbb{Z}^d$ is a **characteristic vector set** function if
  (1) $\mathcal{V}(A)$ generates $\mathbb{Z}^d$ (as a $\mathbb{Z}$-module).
  (2) $\mathcal{V}(U^t A U) = U^{-1}\mathcal{V}(A)$ for all $U \in \mathrm{GL}_d(\mathbb{Z})$.
- Property (**2**) is satisfied e.g. by $\mathrm{Min}(A, \lambda) := \{x \in \mathbb{Z}^n : x^t A x \leq \lambda\}$.

Characteristic Vector Set

- $\mathcal{V} : A \mapsto \mathcal{V}(A) \subset \mathbb{Z}^d$ is a **characteristic vector set** function if
  (1) $\mathcal{V}(A)$ generates $\mathbb{Z}^d$ (as a $\mathbb{Z}$-module).
  (2) $\mathcal{V}(U^t A U) = U^{-1}\mathcal{V}(A)$ for all $U \in \mathrm{GL}_d(\mathbb{Z})$.
- Property (**2**) is satisfied e.g. by $\mathrm{Min}(A, \lambda) := \{x \in \mathbb{Z}^n : x^t A x \leq \lambda\}$.
- $V_{\mathrm{ms}}(A) := \mathrm{Min}(A, \lambda_{\min}(A))$ with $\lambda_{\min}(A)$ minimal such that (1) is satisfied.

Characteristic Vector Set

- $\mathcal{V} : A \mapsto \mathcal{V}(A) \subset \mathbb{Z}^d$ is a **characteristic vector set** function if
  (1) $\mathcal{V}(A)$ generates $\mathbb{Z}^d$ (as a $\mathbb{Z}$-module).
  (2) $\mathcal{V}(U^t A U) = U^{-1}\mathcal{V}(A)$ for all $U \in \mathrm{GL}_d(\mathbb{Z})$.
- Property **(2)** is satisfied e.g. by $\mathrm{Min}(A, \lambda) := \{x \in \mathbb{Z}^n : x^t A x \leq \lambda\}$.
- $V_{\mathrm{ms}}(A) := \mathrm{Min}(A, \lambda_{\min}(A))$ with $\lambda_{\min}(A)$ minimal such that (1) is satisfied.
- Can be used as a proxy:

$$A_1 = U^t A_2 U \qquad \text{for some } U \in \mathrm{GL}_d(\mathbb{Z})$$

$$\Longleftrightarrow$$

$$U \cdot \mathcal{V}(A_1) \underbrace{=}_{\text{As a Set}} \mathcal{V}(A_2) \qquad \text{for some } U \in \mathrm{GL}_d(\mathbb{Z})$$

Characteristic Vector Set

- $\mathcal{V} : \boldsymbol{A} \mapsto \mathcal{V}(\boldsymbol{A}) \subset \mathbb{Z}^{d}$ is a **characteristic vector set** function if
  (1) $\mathcal{V}(\boldsymbol{A})$ generates $\mathbb{Z}^{d}$ (as a $\mathbb{Z}$-module).
  (2) $\mathcal{V}(\boldsymbol{U}^t \boldsymbol{A} \boldsymbol{U}) = \boldsymbol{U}^{-1}\mathcal{V}(\boldsymbol{A})$ for all $\boldsymbol{U} \in \mathrm{GL}_d(\mathbb{Z})$.
- Property (**2**) is satisfied e.g. by $\mathrm{Min}(\boldsymbol{A}, \boldsymbol{\lambda}) := \{\boldsymbol{x} \in \mathbb{Z}^{n} : \boldsymbol{x}^t \boldsymbol{A} \boldsymbol{x} \leq \boldsymbol{\lambda}\}$.
- $\boldsymbol{V}_{\mathrm{ms}}(\boldsymbol{A}) := \mathrm{Min}(\boldsymbol{A}, \boldsymbol{\lambda}_{\min}(\boldsymbol{A}))$ with $\boldsymbol{\lambda}_{\min}(\boldsymbol{A})$ minimal such that (1) is satisfied.
- Can be used as a proxy:

$$\boldsymbol{A}_1 = \boldsymbol{U}^t \boldsymbol{A}_2 \boldsymbol{U} \qquad \text{for some } \boldsymbol{U} \in \mathrm{GL}_d(\mathbb{Z})$$

$$\iff$$

$$\boldsymbol{U} \cdot \mathcal{V}(\boldsymbol{A}_1) \underbrace{=}_{\text{As a Set}} \mathcal{V}(\boldsymbol{A}_2) \qquad \text{for some } \boldsymbol{U} \in \mathrm{GL}_d(\mathbb{Z})$$

- Used by W. Plesken and B. Souvignier (1997) to compute lattice automorphisms and isomorphisms.

Permutation Game

- Suppose $\boldsymbol{A}_1 = \boldsymbol{U}^t \boldsymbol{A}_2 \boldsymbol{U}$.

Permutation Game

- Suppose $\boldsymbol{A}_1 = \boldsymbol{U}^t \boldsymbol{A}_2 \boldsymbol{U}$.
- $\mathcal{V}(\boldsymbol{A}_1) = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$.

Permutation Game

- Suppose $\boldsymbol{A}_1 = \boldsymbol{U}^t \boldsymbol{A}_2 \boldsymbol{U}$.
- $\mathcal{V}(\boldsymbol{A}_1) = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$.
- $\mathcal{V}(\boldsymbol{A}_2) = \boldsymbol{U}\mathcal{V}(\boldsymbol{A}_1) = \{\boldsymbol{w}_1 = \boldsymbol{U}\boldsymbol{v}_1, \ldots, \boldsymbol{w}_n = \boldsymbol{U}\boldsymbol{v}_n\}$.

Permutation Game

- Suppose $\boldsymbol{A}_1 = \boldsymbol{U}^t \boldsymbol{A}_2 \boldsymbol{U}$.
- $\mathcal{V}(\boldsymbol{A}_1) = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$.
- $\mathcal{V}(\boldsymbol{A}_2) = \boldsymbol{U}\mathcal{V}(\boldsymbol{A}_1) = \{\boldsymbol{w}_1 = \boldsymbol{U}\boldsymbol{v}_1, \ldots, \boldsymbol{w}_n = \boldsymbol{U}\boldsymbol{v}_n\}$.
- Under this ordering we *necessarily* have equal pairwise inner products:

$$\boldsymbol{v}_i^t \boldsymbol{A}_1 \boldsymbol{v}_j = (\boldsymbol{U}\boldsymbol{v}_i)^t \boldsymbol{A}_2 (\boldsymbol{U}\boldsymbol{v}_j) = \boldsymbol{w}_i^t \boldsymbol{A}_2 \boldsymbol{w}_j \text{ for all } \boldsymbol{i}, \boldsymbol{j}$$

Permutation Game

- Suppose $\boldsymbol{A}_1 = \boldsymbol{U}^t \boldsymbol{A}_2 \boldsymbol{U}$.
- $\mathcal{V}(\boldsymbol{A}_1) = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$.
- $\mathcal{V}(\boldsymbol{A}_2) = \boldsymbol{U}\mathcal{V}(\boldsymbol{A}_1) = \{\boldsymbol{w}_1 = \boldsymbol{U}\boldsymbol{v}_1, \ldots, \boldsymbol{w}_n = \boldsymbol{U}\boldsymbol{v}_n\}$.
- Under this ordering we *necessarily* have equal pairwise inner products:

$$\boldsymbol{v}_i^t \boldsymbol{A}_1 \boldsymbol{v}_j = (\boldsymbol{U}\boldsymbol{v}_i)^t \boldsymbol{A}_2 (\boldsymbol{U}\boldsymbol{v}_j) = \boldsymbol{w}_i^t \boldsymbol{A}_2 \boldsymbol{w}_j \text{ for all } \boldsymbol{i}, \boldsymbol{j}$$

- $\boldsymbol{v}_i^t \boldsymbol{A}_1 \boldsymbol{v}_j = \boldsymbol{w}_i^t \boldsymbol{A}_2 \boldsymbol{w}_j$ for all $\boldsymbol{i}, \boldsymbol{j}$ is also *sufficient* for such a $\boldsymbol{U}$ to exist.

Permutation Game

- Suppose $\boldsymbol{A}_1 = \boldsymbol{U}^t \boldsymbol{A}_2 \boldsymbol{U}$.
- $\mathcal{V}(\boldsymbol{A}_1) = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$.
- $\mathcal{V}(\boldsymbol{A}_2) = \boldsymbol{U}\mathcal{V}(\boldsymbol{A}_1) = \{\boldsymbol{w}_1 = \boldsymbol{U}\boldsymbol{v}_1, \ldots, \boldsymbol{w}_n = \boldsymbol{U}\boldsymbol{v}_n\}$.
- Under this ordering we *necessarily* have equal pairwise inner products:

$$\boldsymbol{v}_i^t \boldsymbol{A}_1 \boldsymbol{v}_j = (\boldsymbol{U}\boldsymbol{v}_i)^t \boldsymbol{A}_2 (\boldsymbol{U}\boldsymbol{v}_j) = \boldsymbol{w}_i^t \boldsymbol{A}_2 \boldsymbol{w}_j \text{ for all } \boldsymbol{i}, \boldsymbol{j}$$

- $\boldsymbol{v}_i^t \boldsymbol{A}_1 \boldsymbol{v}_j = \boldsymbol{w}_i^t \boldsymbol{A}_2 \boldsymbol{w}_j$ for all $\boldsymbol{i}, \boldsymbol{j}$ is also *sufficient* for such a $\boldsymbol{U}$ to exist.
- $\mathcal{V}(\boldsymbol{A}_2) = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$.

Permutation Game

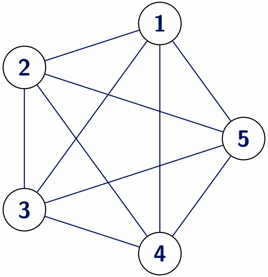
- Suppose $A_1 = U^t A_2 U$.
- $\mathcal{V}(A_1) = \{v_1, v_2, \ldots, v_n\}$.
- $\mathcal{V}(A_2) = U\mathcal{V}(A_1) = \{w_1 = Uv_1, \ldots, w_n = Uv_n\}$.
- Under this ordering we *necessarily* have equal pairwise inner products:

$$v_i^t A_1 v_j = (Uv_i)^t A_2 (Uv_j) = w_i^t A_2 w_j \text{ for all } i, j$$

- $v_i^t A_1 v_j = w_i^t A_2 w_j$ for all i, j is also *sufficient* for such a U to exist.
- $\mathcal{V}(A_2) = \{w_1, \ldots, w_n\}$.
- We want to find a permutation σ such that $v_i A_1 v_j = w_{\sigma(i)} A_2 w_{\sigma(j)}$ for all i, j.

Back to Graph Isomorphism

$G(\mathcal{V}(A_1))$



weights $w_{ij} = v_i^t A_1 v_j$

$$A_1 \cong A_2$$

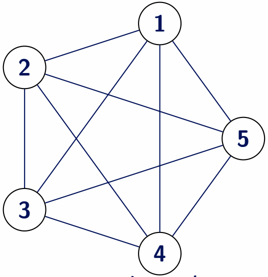
$$\Updownarrow$$

$$\exists \sigma : \forall i, j \; v_i^t A_1 v_j = w_{\sigma(i)} A_2 w_{\sigma(j)}$$

$$\Updownarrow$$

$$G(\mathcal{V}(A_1)) \cong G(\mathcal{V}(A_2))$$

$G(\mathcal{V}(A_2))$

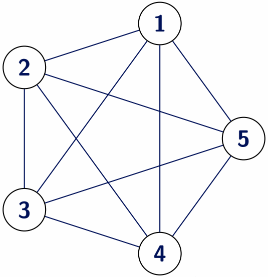


weights $w'_{ij} = w_i^t A_2 w_j$

- It becomes a graph isomorphism problem.

Back to Graph Isomorphism

$G(\mathcal{V}(A_1))$



weights $w_{ij} = v_i^t A_1 v_j$

$$A_1 \cong A_2$$

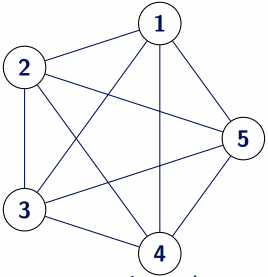
$$\Updownarrow$$

$$\exists \sigma : \forall i, j \ v_i^t A_1 v_j = w_{\sigma(i)} A_2 w_{\sigma(j)}$$

$$\Updownarrow$$

$$G(\mathcal{V}(A_1)) \cong G(\mathcal{V}(A_2))$$

$G(\mathcal{V}(A_2))$



weights $w'_{ij} = w_i^t A_2 w_j$

- It becomes a graph isomorphism problem.
- $\mathrm{Stab}(A_i) \cong \mathrm{Stab}(G(\mathcal{V}(A_i)))$.

Canonical Matrix

- From the graph we obtain some canonical ordering of $\mathcal{V}(\boldsymbol{A}) = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$, say

$$\begin{bmatrix} \vdots & \vdots & & \vdots & \vdots \\ \boldsymbol{v}_{23} & \boldsymbol{v}_{16} & \cdots & \boldsymbol{v}_{3} & \boldsymbol{v}_{7} \\ \vdots & \vdots & & \vdots & \vdots \end{bmatrix} \in \mathbb{Z}^{d \times n}$$

Canonical Matrix

- From the graph we obtain some canonical ordering of $\mathcal{V}(\boldsymbol{A}) = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$, say

$$\begin{bmatrix} \vdots & \vdots & & \vdots & \vdots \\ \boldsymbol{S}\boldsymbol{v}_{23} & \boldsymbol{S}\boldsymbol{v}_{16} & \cdots & \boldsymbol{S}\boldsymbol{v}_{3} & \boldsymbol{S}\boldsymbol{v}_{7} \\ \vdots & \vdots & & \vdots & \vdots \end{bmatrix} \in \mathbb{Z}^{d \times n}$$

- Unique up to some $\boldsymbol{S} \in \mathrm{Stab}(\boldsymbol{A})$.

Canonical Matrix

- From the graph we obtain some canonical ordering of $\mathcal{V}(A) = \{v_1, \ldots, v_n\}$, say

$$M(A) :\equiv \begin{pmatrix} \vdots & \vdots & & \vdots & \vdots \\ Sv_{23} & Sv_{16} & \cdots & Sv_3 & Sv_7 \\ \vdots & \vdots & & \vdots & \vdots \end{pmatrix} \in \mathrm{Stab}(A) \setminus \mathbb{Z}^{d \times n}$$

- Unique up to some $S \in \mathrm{Stab}(A)$.
- Defines a matrix $M(A) \in \mathrm{Stab}(A) \setminus \mathbb{Z}^{d \times n}$ with the (canonical) property:

$$M(U^t A U) \equiv U^{-1} M(A) \in \mathrm{Stab}(U^t A U) \setminus \mathbb{Z}^{d \times n}$$

Canonical Matrix

- From the graph we obtain some canonical ordering of $\mathcal{V}(A) = \{v_1, \ldots, v_n\}$, say

$$M(A) :\equiv \left[\begin{array}{ccccc} \vdots & \vdots & & \vdots & \vdots \\ Sv_{23} & Sv_{16} & \cdots\cdots\cdots & Sv_3 & Sv_7 \\ \vdots & \vdots & & \vdots & \vdots \end{array}\right] \in \mathrm{Stab}(A) \setminus \mathbb{Z}^{d\times n}$$

- Unique up to some $S \in \mathrm{Stab}(A)$.
- Defines a matrix $M(A) \in \mathrm{Stab}(A) \setminus \mathbb{Z}^{d\times n}$ with the (canonical) property:

$$M(U^t A U) \equiv U^{-1} M(A) \in \mathrm{Stab}(U^t A U) \setminus \mathbb{Z}^{d\times n}$$

- Now we can apply HNF: $A_1 \sim A_2 \iff \mathrm{HNF}_L(M(A_1)) = \mathrm{HNF}_L(M(A_2))$

Canonical Form

- Let $\boldsymbol{T_A} \in \mathrm{Stab}(\boldsymbol{A}) \setminus \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$ be a transformation s.t.

$$\boldsymbol{M}(\boldsymbol{A}) \equiv \boldsymbol{T_A} \cdot \mathrm{HNF}_{\boldsymbol{L}}(\boldsymbol{M}(\boldsymbol{A})).$$

Canonical Form

- Let $\boldsymbol{T_A} \in \mathrm{Stab}(\boldsymbol{A}) \setminus \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$ be a transformation s.t.

$$\boldsymbol{M}(\boldsymbol{A}) \equiv \boldsymbol{T_A} \cdot \mathrm{HNF}_{\boldsymbol{L}}(\boldsymbol{M}(\boldsymbol{A})).$$

- Note that $\mathrm{Can}(\boldsymbol{A}) := \boldsymbol{T_A^t} \boldsymbol{A} \boldsymbol{T_A}$ is well defined.

Canonical Form

- Let $\boldsymbol{T_A} \in \mathrm{Stab}(\boldsymbol{A}) \setminus \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$ be a transformation s.t.

$$\boldsymbol{M}(\boldsymbol{A}) \equiv \boldsymbol{T_A} \cdot \mathrm{HNF}_{\boldsymbol{L}}(\boldsymbol{M}(\boldsymbol{A})).$$

- Note that $\mathrm{Can}(\boldsymbol{A}) := \boldsymbol{T}_{\boldsymbol{A}}^{\boldsymbol{t}} \boldsymbol{A} \boldsymbol{T_A}$ is well defined.
- Note that $\boldsymbol{T}_{\boldsymbol{U^t A U}} = \boldsymbol{U}^{-1} \boldsymbol{T_A} \in \mathrm{Stab}(\boldsymbol{U^t A U}) \setminus \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$.

Canonical Form

- Let $\boldsymbol{T_A} \in \mathrm{Stab}(\boldsymbol{A}) \setminus \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$ be a transformation s.t.

$$\boldsymbol{M}(\boldsymbol{A}) \equiv \boldsymbol{T_A} \cdot \mathrm{HNF}_{\boldsymbol{L}}(\boldsymbol{M}(\boldsymbol{A})).$$

- Note that $\mathrm{Can}(\boldsymbol{A}) := \boldsymbol{T_A^t A T_A}$ is well defined.
- Note that $\boldsymbol{T_{U^t AU}} = \boldsymbol{U}^{-1}\boldsymbol{T_A} \in \mathrm{Stab}(\boldsymbol{U^t AU}) \setminus \mathrm{GL}_{\boldsymbol{d}}(\mathbb{Z})$.
- Then we have:

$$\begin{aligned} \mathrm{Can}(\boldsymbol{U^t AU}) &= \boldsymbol{T^t_{U^t AU}}(\boldsymbol{U^t AU})\boldsymbol{T_{U^t AU}} \\ &= \boldsymbol{T_A U^{-t} U^t A U U^{-1} T_A} = \boldsymbol{T_A^t A T_A} = \mathrm{Can}(\boldsymbol{A}) \end{aligned}$$

Theoretical Complexity

- Suppose $\boldsymbol{A}$ is defined over a computable subfield $\boldsymbol{F} \subset \mathbb{R}$.

Theoretical Complexity

- Suppose $\boldsymbol{A}$ is defined over a computable subfield $\boldsymbol{F} \subset \mathbb{R}$.
- We define a characteristic set $\mathcal{V}_{\text{vor}}$ function such that:

Theoretical Complexity

- Suppose $\boldsymbol{A}$ is defined over a computable subfield $\boldsymbol{F} \subset \mathbb{R}$.
- We define a characteristic set $\mathcal{V}_{\text{vor}}$ function such that:
  - $|\mathcal{V}_{\text{vor}}(\boldsymbol{A})| \leq \boldsymbol{O}(2^{\boldsymbol{d}})$.

Theoretical Complexity

- Suppose $\boldsymbol{A}$ is defined over a computable subfield $\boldsymbol{F} \subset \mathbb{R}$.
- We define a characteristic set $\mathcal{V}_{\text{vor}}$ function such that:
  - $|\mathcal{V}_{\text{vor}}(\boldsymbol{A})| \leq \boldsymbol{O}(2^{\boldsymbol{d}})$.
  - $\mathcal{V}_{\text{vor}}(\boldsymbol{A})$ can be computed in $2^{\boldsymbol{O}(\boldsymbol{d})}$ arithmetical operations over $\boldsymbol{F}$.

Theoretical Complexity

- Suppose $\boldsymbol{A}$ is defined over a computable subfield $\boldsymbol{F} \subset \mathbb{R}$.
- We define a characteristic set $\mathcal{V}_{\text{vor}}$ function such that:
  - $|\mathcal{V}_{\text{vor}}(\boldsymbol{A})| \leq O(2^d)$.
  - $\mathcal{V}_{\text{vor}}(\boldsymbol{A})$ can be computed in $2^{O(d)}$ arithmetical operations over $\boldsymbol{F}$.
- Quasi-polytime canonical graph algorithm on a graph of size $|\mathcal{V}_{\text{vor}}(\boldsymbol{A})| \leq 2^{O(d)}$.

Theoretical Complexity

- Suppose $\boldsymbol{A}$ is defined over a computable subfield $\boldsymbol{F} \subset \mathbb{R}$.
- We define a characteristic set $\mathcal{V}_{\text{vor}}$ function such that:
  - $|\mathcal{V}_{\text{vor}}(\boldsymbol{A})| \leq O(2^d)$.
  - $\mathcal{V}_{\text{vor}}(\boldsymbol{A})$ can be computed in $2^{O(d)}$ arithmetical operations over $\boldsymbol{F}$.
- Quasi-polytime canonical graph algorithm on a graph of size $|\mathcal{V}_{\text{vor}}(\boldsymbol{A})| \leq 2^{O(d)}$.
- We can compute $\text{Can}(\boldsymbol{A})$ in

$$\exp(\log(|\mathcal{V}_{\text{vor}}(\boldsymbol{A})|)^{O(1)}) + 2^{O(d)} \leq \exp(d^{O(1)}),$$

arithmetical operations over $\boldsymbol{F} \subset \mathbb{R}$.

Theoretical Complexity

- Suppose $\boldsymbol{A}$ is defined over a computable subfield $\boldsymbol{F} \subset \mathbb{R}$.
- We define a characteristic set $\mathcal{V}_{\mathrm{vor}}$ function such that:
  - $|\mathcal{V}_{\mathrm{vor}}(\boldsymbol{A})| \leq O(2^d)$.
  - $\mathcal{V}_{\mathrm{vor}}(\boldsymbol{A})$ can be computed in $2^{O(d)}$ arithmetical operations over $\boldsymbol{F}$.
- Quasi-polytime canonical graph algorithm on a graph of size $|\mathcal{V}_{\mathrm{vor}}(\boldsymbol{A})| \leq 2^{O(d)}$.
- We can compute $\mathrm{Can}(\boldsymbol{A})$ in

$$\exp(\log(|\mathcal{V}_{\mathrm{vor}}(\boldsymbol{A})|)^{O(1)}) + 2^{O(d)} \leq \exp(d^{O(1)}),$$

  arithmetical operations over $\boldsymbol{F} \subset \mathbb{R}$.
- For $\boldsymbol{F} = \mathbb{Q}$ these operations are polynomially bounded in the input size of $\boldsymbol{A}$.

Practical Complexity

- Efficient in practice.

			Time (s)			$\#\mathcal{V}_{\mathsf{ms}}$		
Type	Samples	n	min	avg	max	min	avg	max
Perfect	10 963	2–8	0.00041	0.0032	0.086	6	73.74	240
	524 288	9	0.0039	0.00594	0.11	90	94.04	272
Random	100	10	0.0015	0.08	2.03	20	100.36	988
	100	20	0.016	0.17	4.18	40	114.34	812
	100	30	2.43	23.41	511.42	60	93.46	310
	100	40	5.18	24.91	251.51	82	107.7	240
Catalogue	107	2-16	0.00018	2.12	36.71	4	630.47	4320

Table: Timings of our implementation

Practical Complexity

- Efficient in practice.
- In the order of (milli)seconds up to dimension **20**.

			Time (s)			$\#\mathcal{V}_{\text{ms}}$		
Type	Samples	n	min	avg	max	min	avg	max
Perfect	10 963	2–8	0.00041	0.0032	0.086	6	73.74	240
	524 288	9	0.0039	0.00594	0.11	90	94.04	272
Random	100	10	0.0015	0.08	2.03	20	100.36	988
	100	20	0.016	0.17	4.18	40	114.34	812
	100	30	2.43	23.41	511.42	60	93.46	310
	100	40	5.18	24.91	251.51	82	107.7	240
Catalogue	107	2-16	0.00018	2.12	36.71	4	630.47	4320

Table: Timings of our implementation

Applications

- Removing redundant forms in a large collection.

Applications

- Removing redundant forms in a large collection.
- Often a critical bottleneck in enumeration algorithms (Kneser's method, Voronoi's algorithm).

Applications

- Removing redundant forms in a large collection.
- Often a critical bottleneck in enumeration algorithms (Kneser's method, Voronoi's algorithm).
- Enumeration of forms of bounded discriminant, or small (spinor) class number.

Applications

- Removing redundant forms in a large collection.
- Often a critical bottleneck in enumeration algorithms (Kneser's method, Voronoi's algorithm).
- Enumeration of forms of bounded discriminant, or small (spinor) class number.
  - Brandt–Intrau tables of reduced ternary forms with discriminant $\leq \mathbf{1000}$

Applications

- Removing redundant forms in a large collection.
- Often a critical bottleneck in enumeration algorithms (Kneser's method, Voronoi's algorithm).
- Enumeration of forms of bounded discriminant, or small (spinor) class number.
  - Brandt–Intrau tables of reduced ternary forms with discriminant $\leq \mathbf{1000}$
  - Nipp's tables of positive definite primitive quaternary forms with discriminant $\leq \mathbf{1732}$.

Applications

- Removing redundant forms in a large collection.
- Often a critical bottleneck in enumeration algorithms (Kneser's method, Voronoi's algorithm).
- Enumeration of forms of bounded discriminant, or small (spinor) class number.
  - Brandt–Intrau tables of reduced ternary forms with discriminant $\leq \mathbf{1000}$
  - Nipp's tables of positive definite primitive quaternary forms with discriminant $\leq \mathbf{1732}$.
  - Complete table of lattices with class number $\mathbf{1}$ due to Kirschmer–Lorch

Applications

- Removing redundant forms in a large collection.
- Often a critical bottleneck in enumeration algorithms (Kneser's method, Voronoi's algorithm).
- Enumeration of forms of bounded discriminant, or small (spinor) class number.
  - Brandt–Intrau tables of reduced ternary forms with discriminant $\leq \mathbf{1000}$
  - Nipp's tables of positive definite primitive quaternary forms with discriminant $\leq \mathbf{1732}$.
  - Complete table of lattices with class number $\mathbf{1}$ due to Kirschmer–Lorch
- Perfect form enumeration to solve the lattice packing theorem.

Applications

- Removing redundant forms in a large collection.
- Often a critical bottleneck in enumeration algorithms (Kneser's method, Voronoi's algorithm).
- Enumeration of forms of bounded discriminant, or small (spinor) class number.
  - Brandt–Intrau tables of reduced ternary forms with discriminant $\leq \mathbf{1000}$
  - Nipp's tables of positive definite primitive quaternary forms with discriminant $\leq \mathbf{1732}$.
  - Complete table of lattices with class number $\mathbf{1}$ due to Kirschmer–Lorch
- Perfect form enumeration to solve the lattice packing theorem.
  - Estimates are in the order of a **billion** perfect forms in dimension $\mathbf{9}$.

Applications

- Removing redundant forms in a large collection.
- Often a critical bottleneck in enumeration algorithms (Kneser's method, Voronoi's algorithm).
- Enumeration of forms of bounded discriminant, or small (spinor) class number.
  - Brandt–Intrau tables of reduced ternary forms with discriminant $\leq \mathbf{1000}$
  - Nipp's tables of positive definite primitive quaternary forms with discriminant $\leq \mathbf{1732}$.
  - Complete table of lattices with class number **1** due to Kirschmer–Lorch
- Perfect form enumeration to solve the lattice packing theorem.
  - Estimates are in the order of a **billion** perfect forms in dimension **9**.
  - Canonical form computed in an average of **6** milliseconds.

Applications

- Removing redundant forms in a large collection.
- Often a critical bottleneck in enumeration algorithms (Kneser's method, Voronoi's algorithm).
- Enumeration of forms of bounded discriminant, or small (spinor) class number.
  - Brandt–Intrau tables of reduced ternary forms with discriminant $\leq \mathbf{1000}$
  - Nipp's tables of positive definite primitive quaternary forms with discriminant $\leq \mathbf{1732}$.
  - Complete table of lattices with class number $\mathbf{1}$ due to Kirschmer–Lorch
- Perfect form enumeration to solve the lattice packing theorem.
  - Estimates are in the order of a **billion** perfect forms in dimension $\mathbf{9}$.
  - Canonical form computed in an average of $\mathbf{6}$ milliseconds.
- Algebraic Modular Forms related to Kneser's method.

Conclusions

- We show an **explicit** and **deterministic** algorithm for finding a canonical form for a positive definite matrix under unimodular integral transformations.

Conclusions

- We show an **explicit** and **deterministic** algorithm for finding a canonical form for a positive definite matrix under unimodular integral transformations.
- Based on Canonical Graph algorithms and Characteristic Vector sets.

Conclusions

- We show an **explicit** and **deterministic** algorithm for finding a canonical form for a positive definite matrix under unimodular integral transformations.
- Based on Canonical Graph algorithms and Characteristic Vector sets.
- It is efficient in practice and has many applications.

Bibliography

- Babai, L., 2016. *Graph isomorphism in quasipolynomial time*. In Proceedings of the forty-eighth annual ACM symposium on Theory of Computing.
- Helfgott, H.A., 2019. *Isomorphismes de graphes en temps quasi-polynomial*, Séminaire Bourbaki. Vol. 2016/2017, no. 407.
- Babai, L., 2019. *Canonical form for graphs in quasipolynomial time: preliminary report*. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing.
- Plesken, W. and Souvignier, B., 1997. *Computing isometries of lattices*. Journal of Symbolic Computation, 24(3-4).
- Implementation at: `https://github.com/MathieuDutSik/polyhedral_common`