The Parameterized Complexity of Geometric Graph Isomorphism

Gaurav Rattan

The Institute of Mathematical Sciences, Chennai, India

Workshop on New Developments in Exact Algorithms and Lower Bounds, IIT Delhi, Dec. 2014

Outline

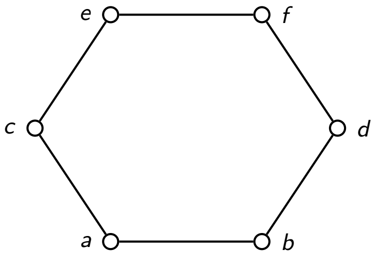
Table of Contents

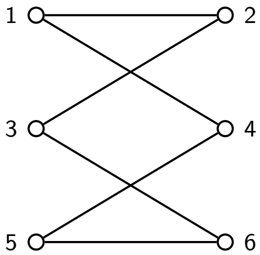
Graph Isomorphism

GI

Input: Graphs G and H

Question: Is G isomorphic to H? I.e., is there an adjacency-preserving bijection between the vertex sets of G and H?





Complexity of GI

Best known algorithm for GI runs in time $2^{\mathcal{O}(\sqrt{n \log n})}$ [Babai, Luks 1983]. Polynomial time algorithms known for restricted graph classes

- bounded genus graphs — Miller 1980
- bounded degree graphs — Luks 1982
- bounded eigenvalue multiplicity graphs — Babai et al. 1982
- bounded treewidth graphs — Bodlaender 1990
- graphs with excluded topological minors — Grohe, Marx 2012

$\mathrm{GI} \in$ FPT, parameterized by

- eigenvalue multiplicity — Evdikomov et al. 1997
- treewidth — Lokshtanov et al. 2014

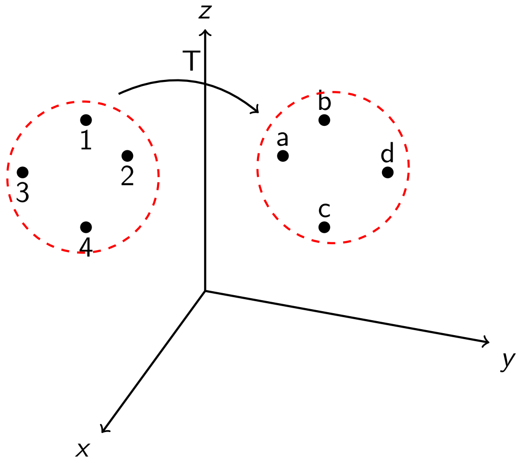
Approaches: Graph-theoretic, group-theoretic, combinatorial, geometrical . . .

Table of Contents

Geometric Graph Isomorphism

GEOM-GI
Input: Point sets $A, B \subseteq Q^k$ of size n
Parameter: k, dimension of the host vector space
Question: Is there is a distance preserving bijection from A to B?

Geometric Graph Isomorphism



Geometric Graph Isomorphism

(Equivalent Formulation:) Does there exist a transformation T of the host space ($T : \mathbb{R}^k \to \mathbb{R}^k$) such that $TA = B$?

- T preserves lengths: $\|Tx\| = \|x\|$ and $\|Tx - Ty\| = \|x - y\|$
- T preserves dot product: $(Tx)^t(Ty) = x^t y$ for all $x, y \in \mathbb{R}^k$

Formally, we call such a T to be an *orthogonal* transformation. We call A and B to be *geomterically-isomorphic via* T.

Fixed Parameter Tractability of GEOM-GI

Dimension of the host space is an important parameter.

Lemma

GEOM-GI *can be solved in polynomial time for bounded dimension k.*

Lemma

$\mathrm{GI} \leq_p \mathrm{GEOM\text{-}GI}_n$.

The reduction maps graphs on n vertices to point-sets in n dimensional host space.

Fixed Parameter Tractability of Geom-GI

Parameterized Algorithms?

- $\mathcal{O}^*(2^{\mathcal{O}(k^4)})$ time complexity, uses cellular algebras [Evdikomov, Ponomarenko]

Theorem (Arvind, R.)

Given point-sets $A, B \in \mathbb{Q}^k$, there is a deterministic $\mathcal{O}^*(k^{\mathcal{O}(k)})$ time algorithm which decides whether A is isomorphic to B.

P time algorithm for bounded dimension

For a k dimensional space, the transformation T can be uniquely described by its action on k independent vectors.

P time algorithm for bounded dimension

For a k dimensional space, the transformation T can be uniquely described by its action on k independent vectors.

Exhaustive Search Algorithm: On input sets A and B in $\mathbb{Q}^k$, size n,

1. Fix k linearly independent vectors $\{a_1, \ldots, a_k\}$ in A.
2. Branch on the possible images $\{b_1, \ldots, b_k\}$ inside B.
3. Let T be the unique transformation which sends $\{a_i\}$ to $\{b_i\}$.
4. Check if T sends A to B in a distance preserving manner.

The algorithm runs in $\mathcal{O}(n^k)$ time.

Table of Contents

Improving over exhaustive search

Build sets S_A and S_B with the following properties

- $|S_A| = |S_B| = f(k)$ *(small-sized)*.
- If A and B are isomorphic via an orthogonal T, then S_A and S_B are also isomorphic via T *(isomorphism-invariant)*.

Improvement: Would imply a $(f(k))^k$ branching, instead of n^k branching.

Lattices

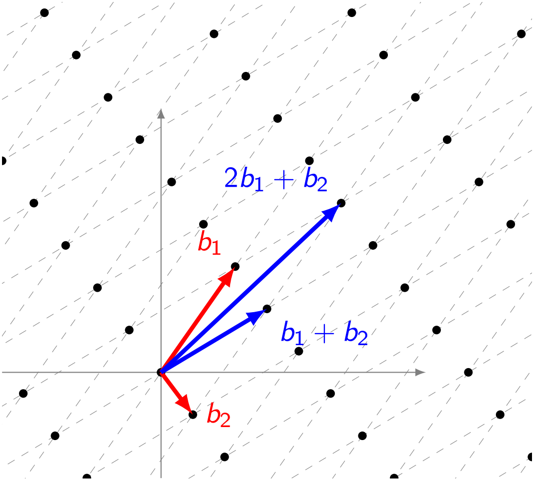


Figure: Integer lattice generated by vectors b_1 and b_2.

Lattices

Definition

The *lattice* generated by a set $A \in \mathbb{R}^k$, denoted by $\mathcal{L}_A$, is the set of all *integer* linear combinations of the set A.

$$\mathcal{L}_A = \{\alpha_1 a_1 + \cdots + \alpha_n a_n \,|\, \alpha_i \in \mathbb{Z}\}$$

Lattices

Definition

The *lattice* generated by a set $A \in \mathbb{R}^k$, denoted by $\mathcal{L}_A$, is the set of all *integer* linear combinations of the set A.

$$\mathcal{L}_A = \{\alpha_1 a_1 + \cdots + \alpha_n a_n \,|\, \alpha_i \in \mathbb{Z}\}$$

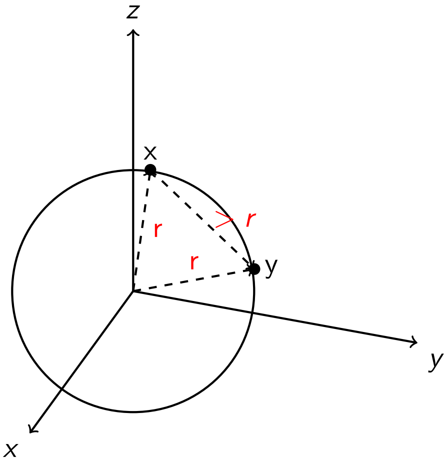
Claim

Let S_A be the set of shortest vectors in $\mathcal{L}_A$. Then, S_A is isomorphism-invariant.

Proof.

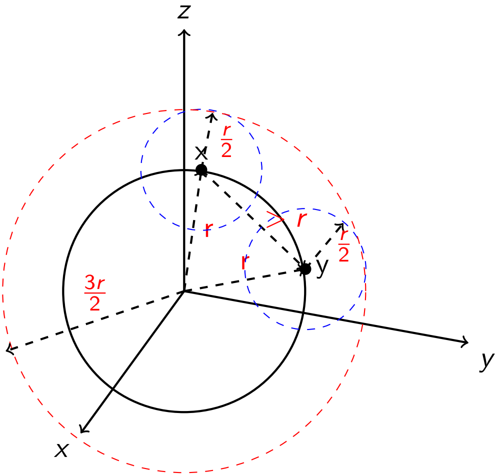
Suppose A and B are isomorphic via T, i.e. $TA = B$. Then, $T(\mathcal{L}_A) = \mathcal{L}_B$. Since T preserves lengths, $T(S_A) = S_B$. □

Shortest vectors in lattices



$$x, y \in \mathcal{L} \Rightarrow x - y \in \mathcal{L}$$

Shortest vectors in lattices



$$x, y \in \mathcal{L} \Rightarrow x - y \in \mathcal{L}$$

$$f(k) \leq \frac{\mathcal{O}\left(\left(\frac{3}{2}r\right)^k\right)}{\mathcal{O}\left(\left(\frac{r}{2}\right)^k\right)} = 3^k$$

Improved Algorithm for GEOM-GI

Given point sets $A, B \subset \mathbb{Q}^k$ as input,

1. Compute sets S_A, S_B of shortest vectors in $\mathcal{L}_A$, $\mathcal{L}_B$ using the SVP algorithm of [MV10].

Improved Algorithm for Geom-GI

Given point sets $A, B \subset \mathbb{Q}^k$ as input,

1. Compute sets S_A, S_B of shortest vectors in $\mathcal{L}_A$, $\mathcal{L}_B$ using the SVP algorithm of [MV10].
2. If $dim(Span(S_A)) = k$, *follow the enumerative strategy*
   - Fix basis in S_A, branch in S_B, ...

Improved Algorithm for Geom-GI

Given point sets $A, B \subset \mathbb{Q}^k$ as input,

1. Compute sets S_A, S_B of shortest vectors in $\mathcal{L}_A$, $\mathcal{L}_B$ using the SVP algorithm of [MV10].
2. If $dim(Span(S_A)) = k$, *follow the enumerative strategy*
   - Fix basis in S_A, branch in S_B, ...
3. If $dim(Span(S_A)) = k_1 < k$, *recursively construct the isomorphism*
   - Construct all $T_1 : Span(S_A) \to Span(S_B)$ enumeratively.
   - *Project sets* A *and* B *out of* $Span(A)$ *and* $Span(B)$.
   - Construct all $T_2 : Span(S_A)^{\perp} \to Span(S_B)^{\perp}$ recursively.
   - Check if $T = T_1 \oplus T_2$ sends A to B.

Improved Algorithm for GEOM-GI

Given point sets $A, B \subset \mathbb{Q}^k$ as input,

1. Compute sets S_A, S_B of shortest vectors in $\mathcal{L}_A$, $\mathcal{L}_B$ using the SVP algorithm of [MV10].
2. If $dim(Span(S_A)) = k$, *follow the enumerative strategy*
   - Fix basis in S_A, branch in S_B, ...
3. If $dim(Span(S_A)) = k_1 < k$, *recursively construct the isomorphism*
   - Construct all $T_1 : Span(S_A) \rightarrow Span(S_B)$ enumeratively.
   - *Project sets* A *and* B *out of* $Span(A)$ *and* $Span(B)$.
   - Construct all $T_2 : Span(S_A)^\perp \rightarrow Span(S_B)^\perp$ recursively.
   - Check if $T = T_1 \oplus T_2$ sends A to B.

Branching: $B(k) = (3^{k_1})^{k_1} \cdot B(k - k_1)$, which yields a $\mathcal{O}^*(2^{\mathcal{O}(k^2)})$ branching.

Table of Contents

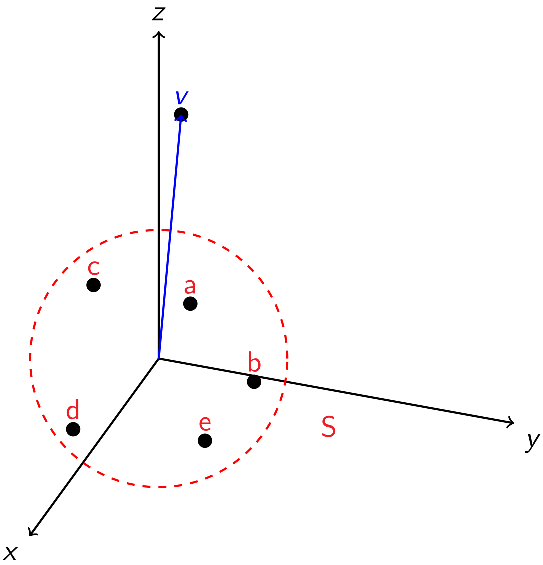
Exploiting the structure of the basis

We are searching for an isomorphism T which maps A to B

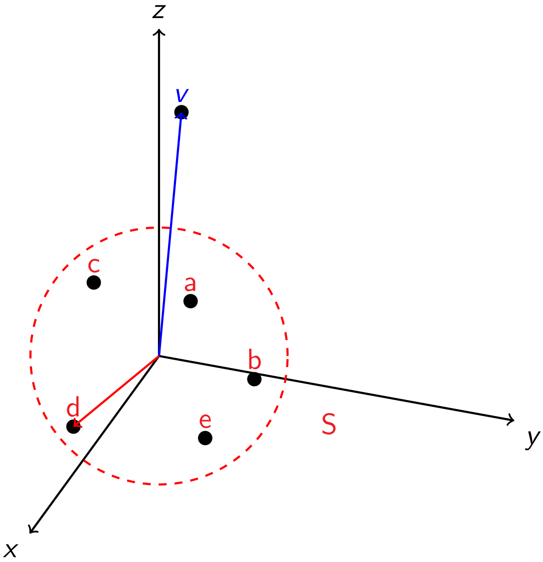
- length-preserving, i.e. $\|Tx\| = \|x\|$
- preserves dot product, i.e. $(Tx)^t(Ty) = x^t y$

Improvement: An isomorphism preserves the geometry of the basis set. (e.g. consider orthogonal bases ...)

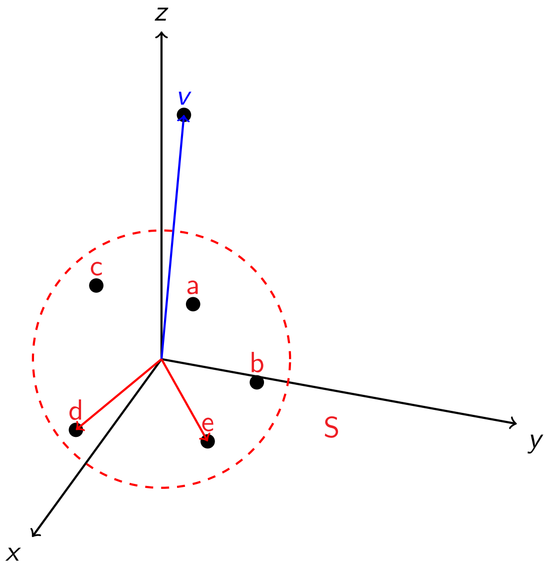
Exploiting the structure of the basis



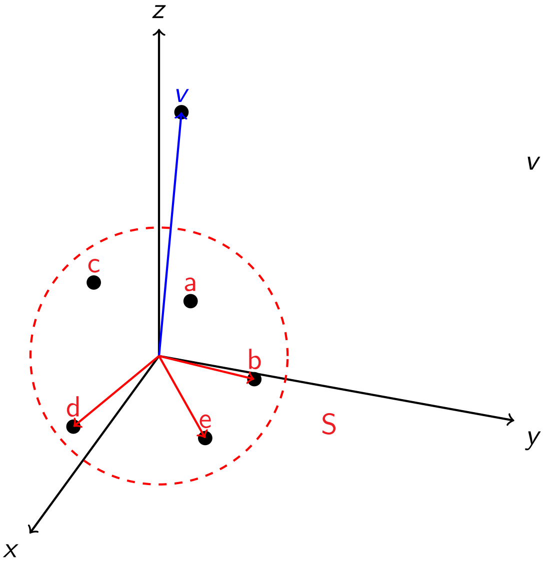
Exploiting the structure of the basis



Exploiting the structure of the basis



Exploiting the structure of the basis



$$v^t d < v^t e < v^t b$$

Exploiting the structure of the basis

Let S be a set of vectors.

Definition (Haviv Regev)

A vector v defines a *chain* of length k in S if for some lin. ind. vectors $a_1, \ldots, a_k \in S$

- a_1 uniquely minimizes $x^t v$ over all $x \in S$
- a_2 uniquely minimizes $x^t v$ over all $x \in S \setminus Span(a_1)$
- and in general, a_i uniquely minimizes $x^t v$ over all $x \in S \setminus Span(a_1, \ldots, a_{i-1})$.

I.e. $v^t a_1 < \cdots < v^t a_k$

Exploiting the structure of the basis

Let S be a set of vectors.

Definition (Haviv Regev)

A vector v defines a *chain* of length k in S if for some lin. ind. vectors $a_1, \ldots, a_k \in S$

- a_1 uniquely minimizes $x^t v$ over all $x \in S$
- a_2 uniquely minimizes $x^t v$ over all $x \in S \backslash Span(a_1)$
- and in general, a_i uniquely minimizes $x^t v$ over all $x \in S \backslash Span(a_1, \ldots, a_{i-1})$.

I.e. $v^t a_1 < \cdots < v^t a_k$ *Remark:* A random v defines a chain of length 1 w.h.p. (Isolation Lemma [MVV]).

Chain isolation in lattices

Given a lattice $\mathcal{L}$, call a vector *dual* if

- it has *integral* dot product with every vector in $\mathcal{L}$.

The set of all dual vectors forms a lattice, the dual lattice $\mathcal{L}_A^*$.

Chain isolation in lattices

Given a lattice $\mathcal{L}$, call a vector *dual* if

- it has *integral* dot product with every vector in $\mathcal{L}$.

The set of all dual vectors forms a lattice, the dual lattice $\mathcal{L}_A^*$.

There are short dual vectors in $\mathcal{L}_A^*$ which define chains inside the set S_A of shortest vectors.

Chain isolation in lattices

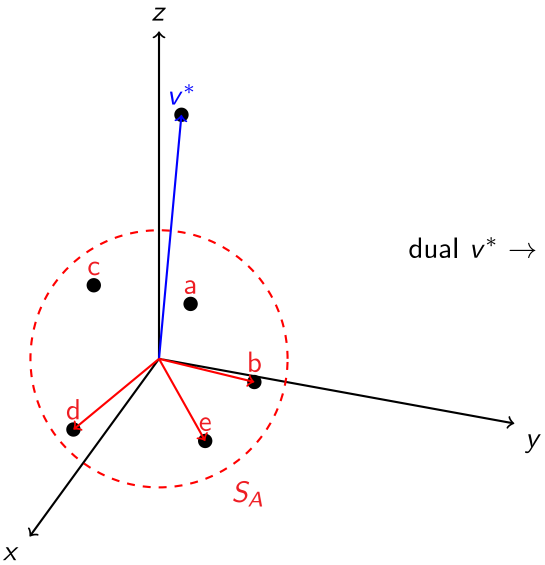
Theorem (Haviv, Regev 14)

There exists a dual vector $v \in \mathcal{L}_A^*$ *such that*

- *(defines chain)* v *defines a chain of length* k *in* S_A *and*
- *(small length)* $\|v\| \leq k^{\mathcal{O}(1)} \cdot \lambda(\mathcal{L}^*)$

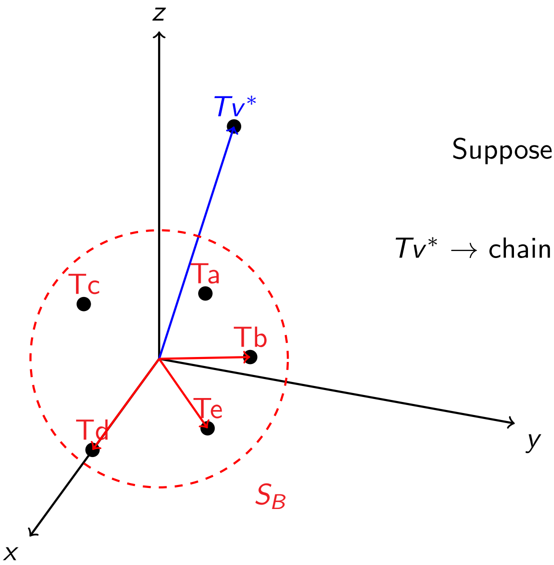
Moreover, the set of all such *isolating* vectors has size at most $\mathcal{O}^*(k^{\mathcal{O}(k)})$, and can be computed in time $\mathcal{O}^*(k^{\mathcal{O}(k)})$.

A faster $\mathcal{O}^*(k^{\mathcal{O}(k)})$ isomorphism algorithm



dual $v^* \to$ chain d-e-b in S_A

A faster $\mathcal{O}^*(k^{\mathcal{O}(k)})$ isomorphism algorithm



Suppose $A \equiv B$ via T

$Tv^* \to$ chain Td-Te-Tb in TS_A

Outline of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ isomorphism algorithm

Algorithm: On input sets $A, B \in \mathbb{Q}^k$,

1. Compute the set of shortest vectors S_A, S_B in $\mathcal{L}_A$, $\mathcal{L}_B$.
2. Compute the set Γ_A, Γ_B of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ dual vectors which induce chains.

Outline of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ isomorphism algorithm

Algorithm: On input sets $A, B \in \mathbb{Q}^k$,

1. Compute the set of shortest vectors S_A, S_B in $\mathcal{L}_A$, $\mathcal{L}_B$.
2. Compute the set Γ_A, Γ_B of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ dual vectors which induce chains.
3. Pick a dual vector $u \in \Gamma_A$. Let $\{a_1, \ldots, a_k\}$ be the chain-basis defined by u inside S_A.
4. For every dual vector $v \in \Gamma_B$,
    - Let $\{b_1, \ldots, b_k\}$ be the chain-basis defined by v inside S_B.
    - Check if $T : \{a_i\} \to \{b_i\}$ is an orthogonal map which sends A to B.

Outline of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ isomorphism algorithm

Algorithm: On input sets $A, B \in \mathbb{Q}^k$,

1. Compute the set of shortest vectors S_A, S_B in $\mathcal{L}_A$, $\mathcal{L}_B$.
2. Compute the set Γ_A, Γ_B of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ dual vectors which induce chains.
3. Pick a dual vector $u \in \Gamma_A$. Let $\{a_1, \dots, a_k\}$ be the chain-basis defined by u inside S_A.
4. For every dual vector $v \in \Gamma_B$,
   - Let $\{b_1, \dots, b_k\}$ be the chain-basis defined by v inside S_B.
   - Check if $T : \{a_i\} \to \{b_i\}$ is an orthogonal map which sends A to B.
5. In case S_A is not k-dimensional, recurse ...

Outline of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ canonization algorithm

Given a point set $A \subset \mathbb{Q}^k$, $f : \mathbb{Q}^k \to \mathbb{Q}^k$ is a *canonizing function* if it has the following properties

- $f(A)$ is isomorphic to A
- if A is isomorphic to B, then $f(A) = f(B)$.

Computing such a f is least as hard as the isomorphism problem.

Outline of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ canonization algorithm

Algorithm: On input set $A \subset \mathbb{Q}^k$,

- Compute the set of shortest vectors S_A in $\mathcal{L}_A$.
- Compute a set Γ_A of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ special bases inside S_A.
- For each basis $J = \{u_1, \ldots, u_k\} \in \Gamma_A$,
  - compute the Gram matrix $G_{i,j} = u_i^t u_j$.
  - compute the coordinates $C(a_i)$ of every point $a_i \in A$ in the basis J.
- Output the lexicographically least description $\sigma = (G, K)$ obtained above.

  If $A \equiv_B T$, then TJ generates same description for B as J generates for A.

Outline of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ canonization algorithm

Algorithm: On input set $A \subset \mathbb{Q}^k$,

- Compute the set of shortest vectors S_A in $\mathcal{L}_A$.
- Compute a set Γ_A of $\mathcal{O}^*(k^{\mathcal{O}(k)})$ special bases inside S_A.
- For each basis $J = \{u_1, \ldots, u_k\} \in \Gamma_A$,
  - compute the Gram matrix $G_{i,j} = u_i^t u_j$.
  - compute the coordinates $C(a_i)$ of every point $a_i \in A$ in the basis J.
- Output the lexicographically least description $\sigma = (G, K)$ obtained above.

  If $A \equiv_B T$, then TJ generates same description for B as J generates for A.
- Compute a canonical A^* using σ as follows.
  - Find unique lower triangular matrix L such that $LL^T = G$.
  - Use row vectors of L to get a basis $J^* = \{v_1, \ldots, v_k\}$.
  - Compute the point set A^* s.t. a_i^* is $K(a_i)$-linear combination of J^*.

Future Directions

- A $\mathcal{O}^*(2^{\mathcal{O}(k)})$ algorithm for geometric graph isomorphism and canonization in Euclidean metric?
  - Faster canonization would give a $2^{\mathcal{O}(n)}$ algorithm for hypergraph canonization
- GEOM-GI in other l_p metrics?
  - Linear algebra breaks down for non-Euclidean metrices; combinatorial algorithms work.
  - Two-dimensional case is polynomial time.
  - Reductions between GEOM-GI for various metrics, similar to embeddings.

Thank you!