Canonical form of positive definite matrix

Anna Haensch

Duquesne University

John Voight

Dartmouth College

Mathieu Dutour Sikirić

Rudjer Bošković Institute

Wessel van Woerden

CWI Amsterdam

I. The automorphism and isomorphism problems

The graph isomorphism problem

- We consider vertex colored graphs G on n vertices with each vertex $i \in \{1, \dots, n\}$ having a color $c_G(i)$.
- Suppose that we have a graph G on n vertices $\{1, \dots, n\}$, we want to compute its automorphism group $Aut(G)$.
  g is formed of all elements in $Sym(n)$ such that
  $$\{g(i), g(j)\} \in E(G) \text{ if and only if } \{i, j\} \in E(G)$$
  and $c(g(i)) = c(i)$ for $1 \le i \le n$.
- Suppose that G_1 and G_2 are two graphs on n vertices $\{1, \dots, n\}$, we want to test if G_1 and G_2 are isomorphic, i.e. if there is $g \in Sym(n)$ such that
  $$\{g(i), g(j)\} \in E(G_2) \text{ if and only if } \{i, j\} \in E(G_1)$$
  and $c_{G_2}(g(i)) = c_{G_1}(i)$.

Complexity: Theoretical and Practical

Theoretical

- The theoretical complexity of the Graph isomorphism problem was unknown for a long time.
- Then in 2015 following happened
  - László Babai, *Graph Isomorphism in Quasipolynomial Time*, arXiv:1512.03547

  that is running time is $\exp((\log n)^{O(1)})$.

Practical

- Since the 70s we have very efficient graph isomorphism programs.
- They can compute the automorphisms of graphs with thousands of vertices.
- Some hard graphs from Projective planes with about 100 vertices can be problematic.

The program nauty

- The program nauty by Brendan McKay solves the graph isomorphism and the automorphism problems.

  http://cs.anu.edu.au/people/bdm/nauty/
- nauty is extremely efficient in doing those computations.
- nauty can deal with directed graph but this is not recommended.
- nauty can deal with vertex colors.
- nauty iterates over all $n!$ permutation but it prunes the search tree so as to obtain a fast running time.
- nauty has exponential runtime in worst case.
- There are alternatives such as bliss or traces with the same performance features.

II. Vertex colored graph reductions

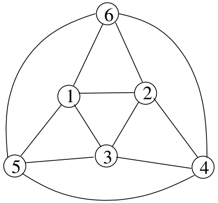
The reduction to a graph

Why focus on graph?

- We have many other combinatorial problems:
  - subset of vertex-set of a graph,
  - set system,
  - edge weighted graph,
  - plane graph,
  - partially ordered set, etc.
- If M is a "combinatorial structure", then we have to define a graph $G(M)$, such that:
  - If M_1 and M_2 are two "combinatorial structure", then M_1 and M_2 are isomorphic if and only if $G(M_1)$ and $G(M_2)$ are isomorphic.
  - If M is a "combinatorial structure", then $Aut(M)$ is isomorphic to $Aut(G(M))$.

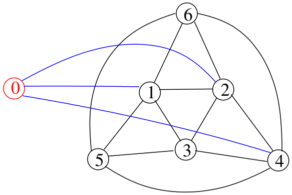
Subset of vertex-set of a graph

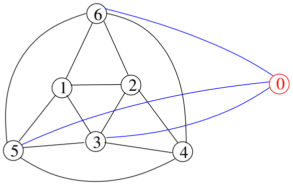
- Suppose that we have a graph G, two subsets S_1, S_2 of G, we want to know if there is an automorphism ϕ of G such that $\phi(S_1) = S_2$.



$S_1 = \{1, 2, 4\}$
$S_2 = \{3, 5, 6\}$

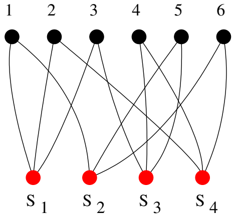
- The method is to define two graphs associated to it:





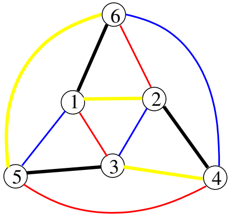
Set systems

- Suppose we have some subsets S_1, ..., S_r of $\{1, \dots, n\}$. We want to find the permutations of $\{1, \dots, n\}$, which permutes the S_i.
- We define a graph with $n + r$ vertices j and S_i with j adjacent to S_i if and only if $j \in S_i$
- Example $\mathcal{S} = \{\{1, 2, 3\}, \{1, 5, 6\}, \{3, 4, 5\}, \{2, 4, 6\}\}$:



Edge colored graphs I

- G is a graph with vertex-set $(v_i)_{1\leq i\leq N}$, edges are colored with k colors $C_1, \dots, C_k$:

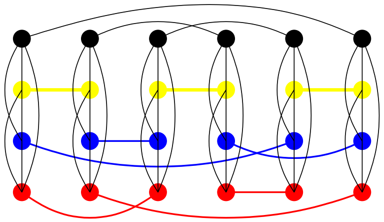


- We want to find automorphisms preserving the graph and the edge colors.
- We form the graph with vertex-set (v_i, C_j) and
  - edges between (v_i, C_j) and $(v_i, C_{j'})$
  - edges between (v_i, C_j) and $(v_{i'}, C_j)$ if there is an edge between v_i and $v_{i'}$ of color C_j

  We get a graph with kN vertices.

Edge colored graphs II

- The picture obtained is:

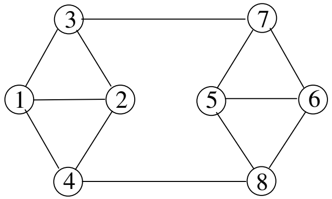


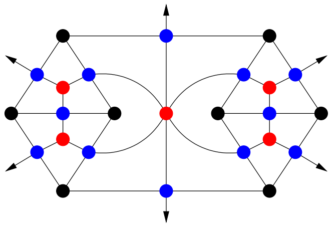
- Actually, one can do better, if the binary expression of j is $b_1 \dots b_r$ with $b_i = 0$ or 1 then we form the graph with vertex-set (v_i, l), $1 \leq l \leq r$ and
  - edges between (v_i, l) and (v_i, l')
  - edges between (v_i, l) and $(v_{i'}, l)$ if the binary number b_l of the expression of C_j is 1.

  This makes a graph with $\lceil \log_2(k) \rceil N$ vertices.

Plane graphs

- If G is a simple 3-connected plane graph then the skeleton determine the embedding, we can forget the faces.
- If G has multiple edge and/or is not 3-connected we consider the graph formed by its vertices, edges and faces with adjacency given by incidence





- This idea extends to partially ordered sets, face lattices, etc.

III. Canonical forms

Canonical form

- One possible canonical form of a graph is obtained by taking the lexicographic minimum of all possible adjacency matrix of a given graph.
- Partition backtrack algorithms provide a way to get a canonical form of a given graph. This will varies from program to program and with chosen options.
- Suppose that one has N different graphs from which we want to select the non-isomorphic ones:
  - If one do isomorphism tests then at worst we have $\frac{N(N-1)}{2}$ tests.
  - If one computes canonical forms, then we have N canonical form computation and then string equality tests.

  This is a key to many computer enumeration goals.
- The runtime of canonical form computation is about the same as computing the automorphism group.
- The problem is how to lift the canonical form of the graph to the canonical form of the original object.

IV. Positive definite form

Problem setting

- For given $n \geq 2$ define $S^n_{>0}$ the set of positive definite quadratic forms.
- The group $\mathrm{GL}_n(\mathbb{Z})$ acts on $S^n_{>0}$ by
$$(P, A) \to P^t A P$$
Quadratic forms are used in lattice theory for covering, packing, etc.
- For a column vector $x \in \mathbb{R}^n$ define $A[x] = x^t A x$
- A canonical form function is just as useful in this field of computational mathematics.
- There is an existing program (AUTO/ISOM) by Plesken & Souvignier for computing the stabilizer and isomorphism.

Minkowski Approach

- Minkowski reduction consists of considering shortest vectors and successive minima of quadratic forms.
- In a fixed dimension n it yields a polyhedral domain M_n on which we can map any positive definite form by an arithmetic equivalence.

Dimension	Nr. facets	Nr. Ext. Rays
1	1	1
2	3	3
3	9	11
4	26	109
5	117	4105
6	1086	?

- For a generic form, we have a unique form in this domain. By ordering the facets of the Minkowski domain, we can obtain a unique form.
- Practical implementation of the Minkowski reduction have been done up to dimension 4, so much fewer to what we really need.

Invariant vector configurations

- We need an invariant vector function V such that for every positive definite $n \times n$-dimensional form A we have
  - $V(A)$ is a set of vector of $\mathbb{Z}^n$ that $\mathbb{Z}$-spans $\mathbb{Z}^n$.
  - If $P \in \mathrm{GL}_n(\mathbb{Z})$ then $V(P^t A P) = P^{-1} V(A)$.
- For $A \in S^n_{>0}$ and $\lambda > 0$ define

$$\mathrm{Min}_\lambda(A) = \{x \in \mathbb{Z}^n \text{ s.t.} A[x] \leq \lambda\},$$

$\min(A) = \min_{x \in \mathbb{Z}^n - \{0\}} A[x]$ and $\mathrm{Min}(A) = \mathrm{Min}_{\min(A)}(A)$.

- Define

$$\mathcal{V}_{\min}(A) = \{\text{smallest } \lambda \text{ s.t. } \mathrm{Min}_\lambda(A)\ \mathbb{Z}\text{-spans } \mathbb{Z}^n\}$$

- For example for the matrix $A_\lambda = \begin{pmatrix} 1 & 0 \\ 0 & \lambda \end{pmatrix}$ this gives us

$$\mathcal{V}_{\min}(A) = \{\pm e_2\} \cup \left\{\pm e_1, \pm 2e_1, \dots, \pm \left\lfloor \sqrt{\lambda} \right\rfloor e_1\right\}$$

Invariant family for well-rounded lattices

- Suppose that A is well-rounded.
- Let $v_1, \ldots, v_n$ be a $\mathbb{Z}$-basis of the full rank lattice $\mathcal{L}_{\min}(A)$ spanned by $\mathrm{Min}(A)$ and let $B \in \mathrm{M}_{n\times n}(\mathbb{Z})$ be the matrix with columns $v_1, \ldots, v_n$. We then define

$$\mathcal{V}_{\text{wr-cvp}}(A) := \mathrm{Min}(A) \cup \bigcup_{c \in \mathbb{Z}^n/\mathcal{L}_{\min}(A)} \left(c - B\,\mathrm{CVP}(B^t AB, B^{-1}c)\right). \tag{1}$$

- The set $\mathcal{V}_{\text{wr-cvp}}(A)$ consists of the union of the shortest vectors together with the set of points in each coset closest to the origin.
- The advantage of this construction is that the size of $\mathcal{V}_{\text{wr-cvp}}(A)$ is bounded from above by $n^{O(n)}$.

Using the subspace filtration

- Compute the set $\mathrm{Min}(A)$ and the saturated sublattice $L_1 := \mathrm{satspan}(\mathrm{Min}(A))$ spanned by these vectors.
- Compute a $\mathbb{Z}$-basis $v_1, \dots, v_r$ of L_1, where r is its rank. Let $B_1 \in \mathrm{M}_{n,r}(\mathbb{R})$ be the matrix with columns $v_1, \dots, v_r$, and let $A_1 := B_1^T A B_1 \in \mathcal{S}^r_{>0}$. Note that A_1 is well-rounded by construction.
- Let
$$\mathrm{proj}\colon \mathbb{Z}^n \to \mathbb{R}^n$$
be the orthogonal projection with respect to A away from L_1.
- Compute a basis $w_1, \dots, w_{n-r}$ of $L_2 := \mathrm{proj}(\mathbb{Z}^n)$ and let $B_2 \in \mathrm{M}_{n,(n-r)}(\mathbb{R})$ the matrix with columns $w_1, \dots, w_{n-r}$. Let $A_2 = B_2^t A_2 B_2$.
- If $r = n$, let $\mathcal{V}_{cvp}(A_2) := \emptyset$; otherwise, compute $\mathcal{V}_{\mathrm{cvp}}(A_2)$ recursively and then define
$$\mathcal{V}_{\mathrm{cvp}}(A) := B_1 \mathcal{V}_{\text{wr-cvp}}(A_1) \cup \bigcup_{v \in B_2 \mathcal{V}_{\mathrm{cvp}}(A_2)} \mathrm{CVP}(A, v).$$

Properties of $\mathcal{V}_{\text{cvp}}(A)$

- It is an invariant vector family.
- We have
$$\mathcal{V}_{\text{cvp}}(A_\lambda) = \{\pm e_1\} \cup \{\pm e_2\}$$
- We have a bound
$$|\mathcal{V}_{\text{cvp}}(A)| \leq n^{O(n)}$$
- Theorem: Given as input a positive definite symmetric matrix A and a corresponding characteristic vector set $\mathcal{V}(A)$ we can compute a canonical form for A in time $\exp(O(\log(N)^c)) \cdot s^{O(1)}$ where N is the input size of $\mathcal{V}(A)$, s is the input size of A and $c > 1$ is some fixed constant.
- Remark: One invariant vector family of size at most $2(2^n - 1)$ can be built by taking all the relevant vectors of the Voronoi cell. This can be computed in time $n^{O(n)}$.

Using invariant vector families

- We choose a specific invariant vector functions V.
- For $A \in S^n_{>0}$ define the edge weighted graph $G(A)$ over $V(A)$ with edge weight $w(v, v') = v' A v^T$.
- The edge weighted graph can be converted into a vertex colored graph $G_2(A)$.
- The vertices of $G(A)$ correspond to disjoint sets of vertices in $G_2(A)$.
- Thus we can order the sets of vertices in $G_2(A)$ by $\min(S) < \min(S')$.
- And so we have a canonical ordering of the vectors $V(A)$.

Canonical spanning set

- Given a family of vector $(v_i)_{1\leq i\leq N}$ we want to find a $\mathbb{Z}$-basis $\mathcal{B}$ of it.
- We want a function $Basis(\mathcal{V})$ such that $Basis(P\mathcal{V}) = PBasis(\mathcal{V})$ for $P \in \mathrm{GL}_n(\mathbb{Z})$.
- The Hermite Normal Form (HNF) provides what we want. A matrix $A \in M_{m,n}(\mathbb{Z})$ is written uniquely as $A = UH$ with $U \in \mathrm{GL}_m(\mathbb{Z})$, $H \in M_{m,n}(\mathbb{Z})$ and
  - H is upper triangular with $H_{i,j} = 0$ for $i > j$.
  - Every pivot of H, the leading coefficient of a row is strictly to the right of the pivots above
  - The elements below pivot are zero and elements above pivots are nonnegative and strictly smaller than the pivot.

  This is computed with Euclidean Divisions.
- We define $Basis(A) = U$.

V. Extensions and Applications

Extension 1: Symplectic group

- We are interested in the group $G = \mathrm{Sp}(2n, \mathbb{Z})$ defined as

$$G = \left\{ M \in \mathrm{GL}_{2n}(\mathbb{Z}) \text{ s.t. } M^t J M = J \right\} \text{ with } J = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}$$

- So, for a positive definite matrix $A \in S^{2n}_{>0}(\mathbb{R})$ and a generating vector set $\mathcal{V}$ we do the following:
  - Instead of the pairwise weight vAw^T we take the pairwise weight (vAw^T, vJw^T).
  - The same algorithm apply and we can reorder the generating set.
  - We can deterministically build a symplectic basis from the reordered generating set.

Extension 2: Non-generating vector sets?

- There are many examples of lattices for which we can get an invariant full dimensional set of vectors which does not $\mathbb{Z}$-span the lattice.
  Examples: The Niemeier lattices are such.
- If one wants to compute automorphism (and testing for equivalence) this is fine as we can do group action on the possible integral embedding and find stabilizer.
  One needs to use the partition backtrack for finding the set stabilizer of finite groups.
- But this does not work for the canonical form because we do not have a canonical form for the orbit of a set under a finite group.
  But there is no theoretical obstacle for this to be done.

Extension 4: Other groups?

- $\mathrm{GL}_n(\mathbb{Z})$ is just a case for the group action. Conceivably we could consider other group cases.
- It is not likely to obtain an uniform description for finite index subgroups of $\mathrm{GL}_n(\mathbb{Z})$. But what special finite index subgroups could be doable?
- For subgroups G of $\mathrm{GL}_n(\mathbb{Z})$ preserving a finite list of n-dimensional lattices $L_1, \ldots, L_m$ we have generalization of stabilizer/isomorphy formalism.
- If $m = 1$ we are in $\mathrm{GL}_n(\mathbb{Z})$ case.
- But there does not appear to have some generalization of the canonical form for $m > 1$.
- Conceivably we can do the reduced form to $\mathrm{GL}_n(\mathbb{Z}[i])$.

Application 1: Application to perfect forms

- The enumeration of perfect forms in a fixed dimension n is done by using Voronoi algorithm:
  - Take a perfect form A and insert it into the list of perfect forms.
  - For a given perfect form, compute the forms whose perfect domain is adjacent to it.
  - Insert the new found forms in the list if they are new.
  - Terminate when all forms have been treated.
- The enumeration has to be done in parallel with each parallel node having a specific set of perfect forms.
- The canonical forms allow to assign the forms to the parallel node in a reliable way.

Application 2: Genus enumeration

- By this we mean enumeration of forms of bounded discriminant, or small class number or small spinor class number.
- Example of algorithm used is Kneser's neighboring algorithm which takes a form and return its p-neighbors.
- It works similarly to Voronoi's Algorithm and the same technique can be used.
- The Kneser's algorithm for a prime number p can be used to build a Hecke operators. Canonical forms can be used to reduce the number of isomorphy tests.
- This can be used to compute some Algebraic Modular Forms.

Application 3: Database of forms

- The L-functions and Modular Forms Database (LMFDB) contains a lot number theoretic informations.
- This includes some lattices coming from different sources.
- Using canonical forms can make the query faster.

THANK YOU