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 $i \in \{1, \ldots, n\}$ having a color $c_G(i)$.
- Suppose that we have a graph G on n vertices $\{1, \ldots, n\}$, 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 $1 \leq i \leq n$.
- Suppose that G_1 and G_2 are two graphs on n vertices $\{1, \ldots, n\}$, 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

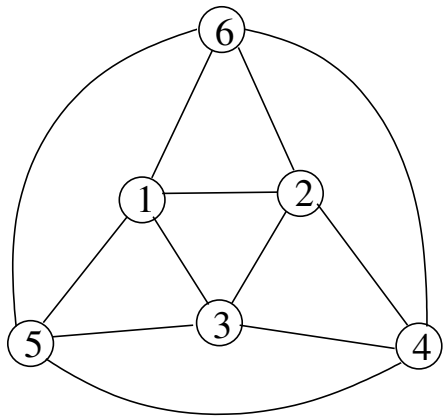
The reduction to a graph

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 $Aut(G(M))$.

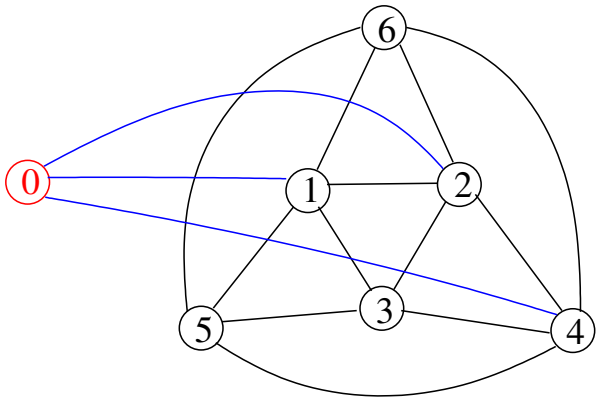
Subset of vertex-set of a graph

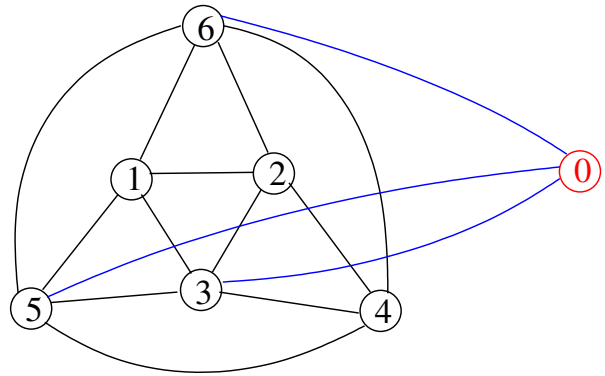
- Suppose that we have a graph G, two subsets S_1, S_2 of G, we want to know if there is an automorphism ϕ of G such that $\phi(S_1) = S_2$.



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

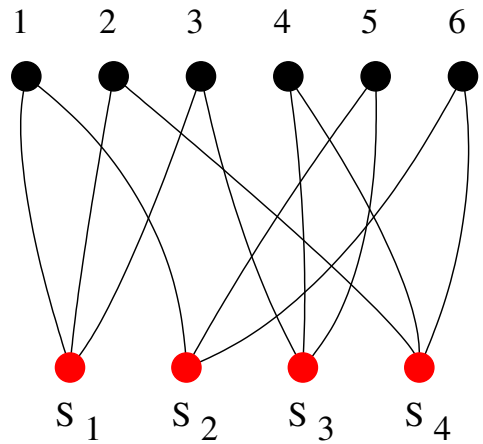
- The method is to define two graphs associated to it:





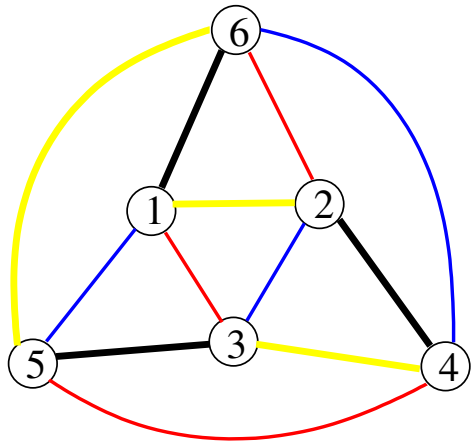
Set systems

- Suppose we have some subsets $S_1, \ldots, S_r$ of $\{1, \ldots, n\}$. We want to find the permutations of $\{1, \ldots, n\}$, 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

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

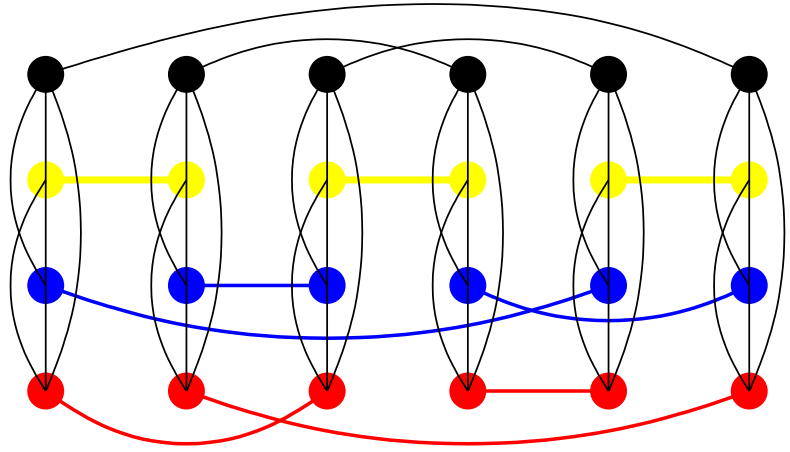


- We want to find automorphisms preserving the graph and the edge colors.
- We form the graph with vertex-set (v_i, C_j) and
  - edges between (v_i, C_j) and $(v_i, C_{j'})$
  - edges between (v_i, C_j) and $(v_{i'}, C_j)$ if there is an edge between v_i and $v_{i'}$ of color C_j

  We get a graph with kN vertices.

Edge colored graphs

- The picture obtained is:

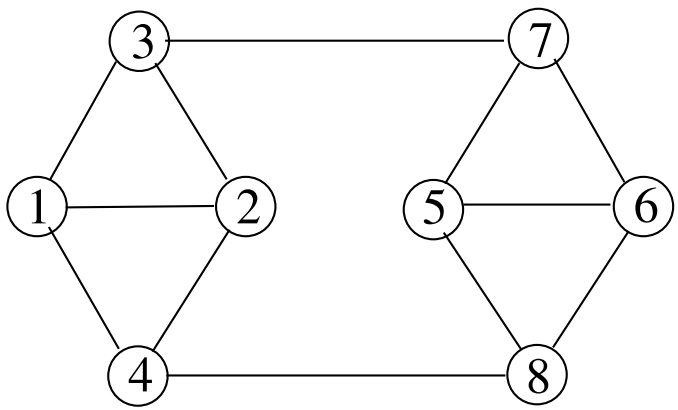


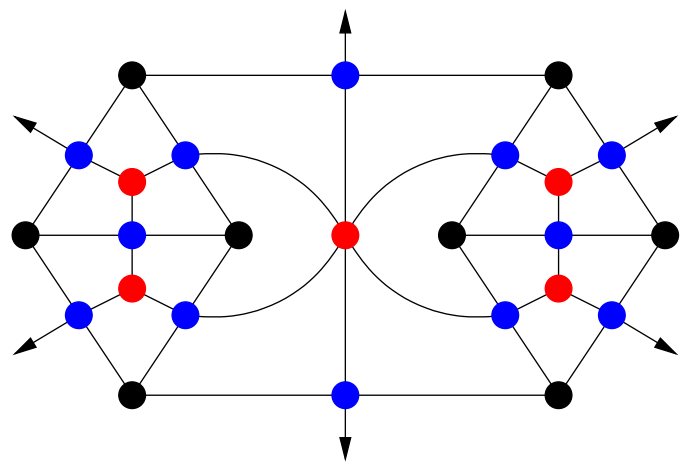
- Actually, one can do better, if the binary expression of j is $b_1 \dots b_r$ with $b_i = 0$ or 1 then we form the graph with vertex-set (v_i, l), $1 \leq l \leq r$ and
  - edges between (v_i, l) and (v_i, l')
  - edges between (v_i, l) and $(v_{i'}, l)$ if the binary number b_l of the expression of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

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

$$(P, A) \to PAP^T$$

  Quadratic forms are used in lattice theory for covering, packing, etc.
- For $x \in \mathbb{R}^n$ define $A[x] = xAx^T$
- There is a canonical form via the Minkowski reduction theory but it is hard to compute.
- 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.

Using shortest vectors

- For $A \in S^n_{>0}$ and $\lambda > 0$ define

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

- Define

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

- For $A \in S^n_{>0}$ define the edge weighted graph $G(A)$ over $Span(A)$ 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 $Span(A)$.

Canonical spanning set

- Given a family of vector $(v_i)_{1 \le i \le N}$ we want to find a $\mathbb{Z}$-basis $\mathcal{B}$ of it.
- We want a function $Basis(\mathcal{V})$ such that $Basis(\mathcal{V}P) = Basis(\mathcal{V})P$ for $P \in \mathrm{GL}_n(\mathbb{Z})$.
- We start with $\mathcal{B} = \emptyset$ and add vectors one by one starting from v_1, ending with v_N.
  - If v_i is $\mathbb{R}$-linearly independent from $\mathcal{B}$ then we insert it into $\mathcal{B}$.
  - If v_i is belongs to the $\mathbb{Z}$-span of $\mathcal{B}$ then we do nothing.
  - If v_i does not belong to the $\mathbb{Z}$-span of $\mathcal{B}$ and is $\mathbb{R}$-linearly independent then take $\beta > 0$ the smallest integer such that

$$\beta v_i = \sum_{v \in \mathcal{B}} \alpha_v v \text{ with } \alpha_v \in \mathbb{Z}$$

    The α and α_v are uniquely determined and invariant under linear transformation.
    We then extract a $\mathbb{Z}$-basis from this and change $\mathcal{B}$ accordingly.
- $Span(A)$ spans $\mathbb{Z}^n$ so $P = (Basis(Span(A)))^{-1} \in \mathrm{GL}_n(\mathbb{Z})$.
- The canonical form is then PAP^T.

Extension 1: Symplectic group

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

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

- So, for a positive definite matrix $A \in S^{2n}_{>0}(\mathbb{R})$ we take the canonical form in $\mathrm{GL}_{2n}(\mathbb{Z})$ and the matrix $P = (Basis(Span(A)))^{-1}$.
- We take $B = PJP^T$ which is a priori different from J.
- The matrix B is integral antisymmetric. We can find a matrix $U \in \mathrm{GL}_{2n}(\mathbb{Z})$ such that $B = UJU^T$ [Hint: Do operations on rows and column at the same time].
- We have $W = U^{-1}P \in \mathrm{Sp}(2n, \mathbb{Z})$ and the canonical form is

$$WAW^T$$

Extension 2: Finite index subgroups of $\mathrm{GL}_n(\mathbb{Z})$

- The approach for the symplectic group can be considered to other subgroups G of $\mathrm{GL}_n(\mathbb{Z})$.
- Since G is of finite index we have a coset decomposition

$$\mathrm{GL}_n(\mathbb{Z}) = \cup_{i=1}^{m} g_i G$$

- Consider a positive definite form $A \in S^n_{>0}$ and take the canonical form PAP^T.
- The element P corresponds to one coset $g_{i_0} G$. The canonical form is then

$$(g_{i_0}^{-1} P) A (g_{i_0}^{-1} P)^T$$

- The problem is that this is not a practical algorithm right now:
  - The coset representatives g_i have to be chosen
  - It is difficult to find which coset the element belongs to.

Other extensions?

- Another aspect is that we would like to use vector sets that are not necessarily $\mathbb{Z}$-basis. This would require a canonical form for group action on sets.
- 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.
- Number rings are also important.
  For number rings which are Euclidean, e.g. Gaussian integers we could have further generalization.

THANK YOU