

Symmetric Computation: Part 3

Anuj Dawar

Department of Computer Science and Technology, University of Cambridge

PhD Open, Warsaw, 19 October 2019

Linear Programming

Linear Programming is an important algorithmic tool for solving a large variety of optimization problems.

It was shown by **(Khachiyan 1980)** that linear programming problems can be solved in polynomial time.

We have a set C of *constraints* over a set V of *variables*.

Each $c \in C$ consists of $a_c \in \mathbb{Q}^V$ and $b_c \in \mathbb{Q}$.

Feasibility Problem: Given a linear programming instance, determine if there is an $x \in \mathbb{Q}^V$ such that:

$$a_c^T x \leq b_c \quad \text{for all } c \in C$$

Optimization Problem: Given a linear programming instance and a linear *objective function* f , find a feasible point x for which $f(x)$ is maximum.

Linear Programs for Hard problems

In the 1980s there was a great deal of excitement at the discovery that *linear programming* could be done in *polynomial time*.

This raised the possibility that linear programming techniques could be used to *efficiently* solve hard problems.

Many proposals were put forth for encoding *hard* problems (such as the *Travelling Salesman Problem*) (TSP) as linear programs.

(Yannakakis 1991) proved that *any* encoding of TSP as a linear program, satisfying natural *symmetry* conditions, must have *exponential size*.

Travelling Salesman Problem

Given a set of V of n vertices and a distance matrix $C = \mathbb{Q}^{V \times V}$, find

$$\min_{\pi \in [n]^{\text{bij}} \rightarrow V} \sum_{i \in [n]} c_{\pi(i)\pi(i+1)} + c_{\pi(n)\pi(1)}$$

To formulate this as a *linear optimization* problem, introduce a set of variables:

$$X = \{x_{ij} \mid i, j \in V\}.$$

So, a graph is a *function* $G : X \rightarrow \{0, 1\}$.

Let $P \subseteq \{0, 1\}^X$ be the collection of simple cycles of length n .

TSP polytope

Let $\text{conv}(P) \subseteq \mathbb{Q}^X$ be the *convex hull* of P .

That is, the set of $\vec{y} \in \mathbb{Q}^X$ such that

$$\vec{y} = \sum_{\vec{x} \in P} \lambda_{\vec{x}} \vec{x} \quad \text{with } \lambda_{\vec{x}} \geq 0 \text{ and } \sum_{\vec{x} \in P} \lambda_{\vec{x}} = 1.$$

TSP: $\min \sum_{i,j \in V} c_{ij} x_{ij}$ over $\vec{x} \in P$.

This is equivalent to minimizing $\sum_{i,j \in V} c_{ij} x_{ij}$ over $\text{conv}(P)$.

We call $\text{conv}(P)$ the *TSP polytope*.

$\text{conv}(P)$ has *exponentially many facets*.

Extended Formulations

Could $\text{conv}(P)$ be obtained as the *projection* of a polytope with a small number of facets?

Is there a *small* $Q \subseteq \mathbb{Q}^X \times \mathbb{Q}^Y$ such that

$$\{\vec{x} \mid \exists \vec{y}(\vec{x}, \vec{y}) \in Q\} = \text{conv}(P)?$$

If a description of such a Q could be obtained in *polynomial time* in n , then $P = NP$.

If such a Q of *polynomial size* exists, then $NP \subseteq P/\text{poly}$.

Also note that by adding inequalities $x \leq G(x)$ for a graph $G : X \rightarrow \{0, 1\}$, we obtain a polytope $Q_G \subseteq \mathbb{Q}^X \times \mathbb{Q}^Y$ which is *non-empty* if, and only if, G contains a Hamiltonian cycle.

Yannakakis

Say $Q \subseteq \mathbb{Q}^X \times \mathbb{Q}^Y$ is *symmetric* if for every $\pi \in S_V$, there is a $\sigma \in S_Y$ such that

$$Q^{(\pi, \sigma)} = Q$$

Here, we extend the action of π to $V \times V$, and hence to \mathbb{Q}^X .
similarly σ to \mathbb{Q}^Y .

Theorem (Yannakakis)

Any symmetric $Q \subseteq \mathbb{Q}^X \times \mathbb{Q}^Y$ whose projection on \mathbb{Q}^X is $\text{conv}(P)$ has *exponentially* many facets.

This is derived from a similar lower bound for the *matching polytope*.

Matching Polytope

Fix V with $|V| = 2n$ and $X = \{x_{ij} \mid i, j \in V\}$

$M \subseteq \{0, 1\}^X$ is the set of graphs that *are* perfect matchings on V .

$\text{conv}(M)$ has an *explicit* description given by **(Edmonds)**:

$$\begin{aligned}x_{ij} &\geq 0, \quad \forall i, j \in V \\ \sum_j x_{ij} &= 1 \quad \forall i \in V \\ \sum_{i \in S; j \notin S} x_{ij} &\geq 1 \quad \forall S \subseteq V \text{ with } |S| \text{ odd,}\end{aligned}$$

This has exponentially many facets.

Lower Bounds

Theorem (Yannakakis)

Any symmetric $Q \subseteq \mathbb{Q}^X \times \mathbb{Q}^Y$ whose projection on \mathbb{Q}^X is $\text{conv}(M)$ has *exponentially* many facets.

The lower bound on the *TSP* polytope is obtained by a reduction from the lower bound on the *matching* polytope.

What if we drop the condition of *symmetry*?

A long line of work since **(Yannakakis 1991)** has looked at *relaxing* the notion of symmetry. This culminated in **(RothvoB 2013)** showing an exponential lower bound even *without* the requirement of symmetry.

But... Linear Programming is P-complete

Any problem in P can be solved by coding it as a *linear program*.

Suppose $L \subseteq \{0, 1\}^*$ is in P .

For any n , let $X = \{x_i \mid i \in [n]\}$.

There is a polytope $Q \subseteq \mathbb{Q}^X \times \mathbb{Q}^Y$ of size $\text{poly}(n)$ whose projection on \mathbb{Q}^X includes all points in $L \cap \{0, 1\}^X$ and excludes all points in $\{0, 1\}^X \setminus L$.

Note: not necessarily the *convex hull* of $L \cap \{0, 1\}^X$.

Circuits to LP

Take a *circuit* C of poly-size deciding $L \cap \{0, 1\}^X$.
Introduce a new variable g for each gate of C .

$$g = \neg u : 0 \leq g = 1 - u \leq 1$$

$$g = u \wedge v : 0 \leq g \leq u \leq 1$$

$$0 \leq g \leq v \leq 1$$

$$g \leq u + v - 1$$

and similarly for other gates.

The argument works for the non-uniform class $P/poly$.

Convex Hulls and Separating Polytopes

For the *matching* and *TSP* polytopes, i.e. the convex hull of solutions, we have exponential lower bounds on both symmetric (by **Yannakakis**) and general (by **Rothvoß**) versions.

For polytopes that *separate* solutions from non-solutions we have poly-size ones for *matching*, and we cannot hope for lower bounds greater than poly-size for *TSP*.

What about *symmetric* polytopes that separate solutions from non-solutions?

Symmetric Linear Programs

Fix $X = \{x_{ij} \mid i, j \in V\}$ for a fixed vertex set V .

Consider a class \mathcal{C} of graphs $G : X \rightarrow \{0, 1\}$.

We say that a polytope $Q \subseteq \mathbb{Q}^X \times \mathbb{Q}^Y$ *decides* \mathcal{C} if its projection on \mathbb{Q}^X includes \mathcal{C} and excludes its complement.

Q is *symmetric* if for each $\pi \in S_V$ there is a $\sigma \in S_Y$ such that $Q = Q^{(\pi, \sigma)}$.

The Power of Symmetric LP

In ([Atserias, D., Ochremiak 2018](#)) we show that the following are equivalent for a class of graphs \mathcal{C} .

1. \mathcal{C} is decided by a family of *polynomial-size, symmetric* linear programs.
2. \mathcal{C} is decided by a family of *polynomial-size, symmetric* threshold circuits.
3. \mathcal{C} is decided by a family of *polynomial-size* formulas of \mathcal{C}^k for some fixed k .

In particular, \mathcal{C} must have bounded counting width.

There *are* poly-size symmetric linear programs that decide the class of graphs with *perfect matchings*.

There are *no* poly-size symmetric linear programs that decide the class of graphs with a *Hamiltonian cycle*.

Linear Programming

We can represent an instance of a linear programming feasibility problem as a *relational structure* over a suitable vocabulary.

We have a set C of *constraints* over a set V of *variables*.

Each $c \in C$ consists of $a_c \in \mathbb{Q}^V$ and $b_c \in \mathbb{Q}$.

The numbers are encoded in *binary* over an ordered set of *bit positions*.

Feasibility Problem: Given a linear programming instance, determine if there is an $x \in \mathbb{Q}^V$ such that:

$$a_c^T x \leq b_c \quad \text{for all } c \in C$$

Representing Rational Numbers

We can take the rational number

$$q = s \frac{n}{d}$$

where $s \in \{1, -1\}$ and $n, d \in \mathbb{N}$
to be given by a structure

$$(B, <, S, N, D)$$

where $<$ is a linear order on the domain B and S , N and D are unary relations.

$S = \emptyset$ iff $s = 1$ and N and D code the binary representation of n and d .

Since the domain is ordered, it is straightforward to see that arithmetic, in the form of addition and multiplication of numbers is definable in **FPC**

Representing Rational Vectors and Matrices

A *rational vector* indexed by a set I :

$$v : I \rightarrow \mathbb{Q}$$

is represented by a structure over domain $I \cup B$ with relations:

- $<$ an order on B ;
- $S, N, D \subseteq I \times B$

Similarly, a *rational matrix* $M \in \mathbb{Q}^{I \times J}$ is given by a structure over domain $I \cup J \cup B$ with relations:

- $<$ an order on B ;
- $S, N, D \subseteq I \times J \times B$

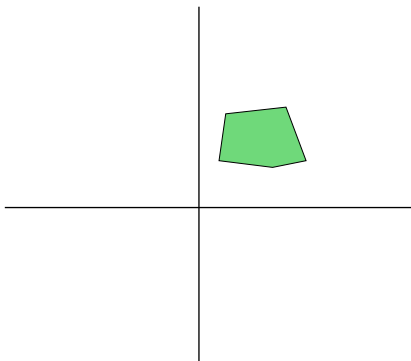
Weighted Graphs

We use a similar encoding to represent problems over *weighted graphs* where the weights may be integer or rational.

For example, a graph with vertex set V with *non-negative rational* weights might be considered as a relational structure over universe $V \cup B$ where B is bigger than the number of bits required to represent any of the rational weights and we have

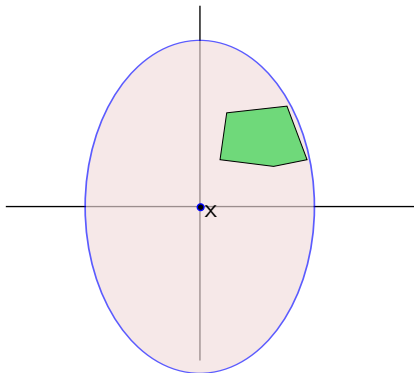
- $<$ an order on B ;
- *weight relations* $W_n, W_d \subseteq V \times V \times B$

Ellipsoid Method



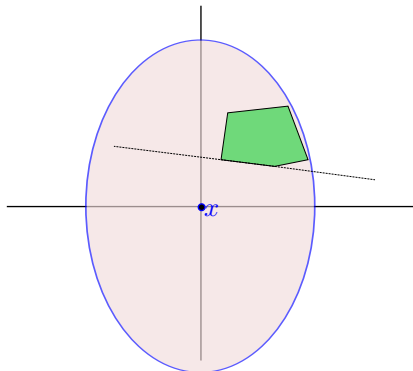
The set of constraints determines a *polytope*

Ellipsoid Method



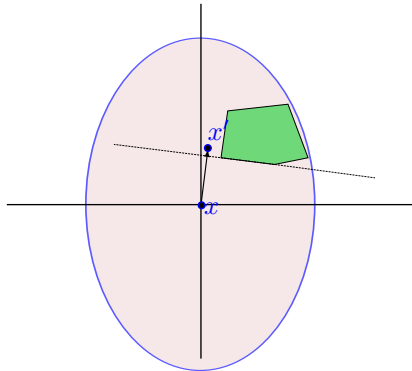
Start at the origin and calculate an *ellipsoid* enclosing it.

Ellipsoid Method



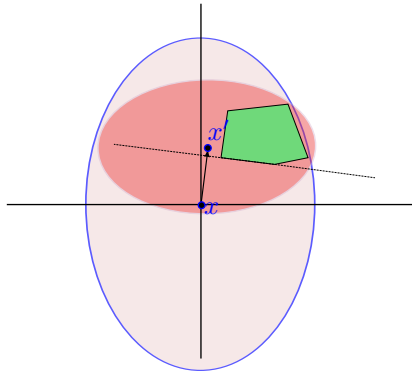
If the centre is not in the polytope, choose a constraint it *violates*.

Ellipsoid Method



Calculate a new *centre*.

Ellipsoid Method



And a new ellipsoid around the centre of at most *half* the volume.

Ellipsoid Method in FPC

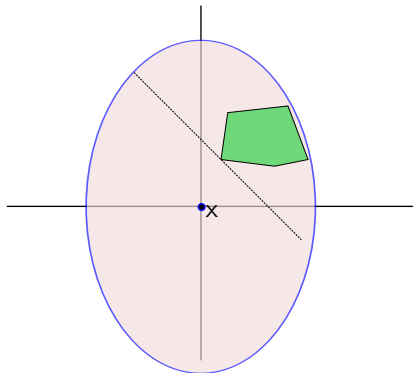
We can encode all the calculations involved in **FPC**.

This relies on expressing algebraic manipulations of *unordered* matrices.

What is not obvious is how to *choose* the violated constraint on which to project.

However, the ellipsoid method works as long as we can find, at each step, some *separating hyperplane*.

Ellipsoid Method in FPC



Ellipsoid Method in FPC

We can encode all the calculations involved in FPC.

This relies on expressing algebraic manipulations of *unordered* matrices.

What is not obvious is how to *choose* the violated constraint on which to project.

However, the ellipsoid method works as long as we can find, at each step, some *separating hyperplane*.

So, we can take:

$$\left(\sum_{c \in S} a_c\right)^T x \leq \sum_{c \in S} b_c$$

where S is the *set* of all violated constraints.

Separation Oracle

More generally, the ellipsoid method can be used, even when the *constraint matrix* is not given explicitly, as long as we can always determine a *separating hyperplane*.

In particular, the polytope represented may have *exponentially many* facets.

(Anderson, D., Holm 2015) shows that as long as the *separation oracle* can be defined in FPC, the corresponding *optimization problem* can be solved in FPC.

Representations of Polytopes

A *representation* of a class \mathcal{P} of *polytopes* is a *relational vocabulary* τ along with a surjective function ν taking τ -structures to polytopes in \mathcal{P} , which is isomorphism invariant.

A *separation oracle* for a representation ν, \mathcal{P} is definable in FPC if there is an FPC formula that given a τ -structure \mathbb{A} and a vector $v \in \mathbb{Q}^V$ either

- determines that $v \in \nu(\mathbb{A})$; or
- defines a hyperplane separating v from $\nu(\mathbb{A})$.

Folding Polytopes

We use the separation oracle to define an *ordered equivalence relation* on the set V of variables.

We also define a *projection* operation on polytopes which either

- preserves feasibility; or
- refines the equivalence relation further.

Folding and Unfolding

Suppose we have $\sigma : V \rightarrow [n]$, for $n \leq |V|$.

We say $c \in \mathbb{Q}^V$ agrees with σ , if $\sigma(u) = \sigma(v) \Rightarrow c_u = c_v$.

Fold $P \subseteq \mathbb{Q}^V$ into $P^\sigma \subseteq \mathbb{Q}^n$.

For $i \in [n]$,

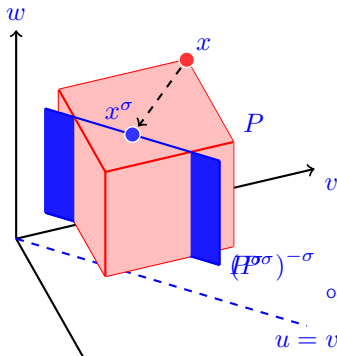
$$(x^{\tilde{\sigma}})_i := \sum_{\{v \in V \mid \sigma(v)=i\}} x_v;$$

$$(x^\sigma)_i := \frac{(x^{\tilde{\sigma}})_i}{|\{v \in V \mid \sigma(v)=i\}|}.$$

Unfold $P^\sigma \subseteq \mathbb{Q}^n$ into $(P^\sigma)^{-\sigma} \subseteq \mathbb{Q}^V$.

For $v \in V$,

$$(x^{-\sigma})_v := x_{\sigma(v)}.$$



Folding and Unfolding

Suppose we have $\sigma : V \rightarrow [n]$, for $n \leq |V|$.

We say $c \in \mathbb{Q}^V$ agrees with σ , if $\sigma(u) = \sigma(v) \Rightarrow c_u = c_v$.

Fold $P \subseteq \mathbb{Q}^V$ into $P^\sigma \subseteq \mathbb{Q}^n$.

For $i \in [n]$,

$$(x^{\tilde{\sigma}})_i := \sum_{\{v \in V \mid \sigma(v)=i\}} x_v;$$

$$(x^\sigma)_i := \frac{(x^{\tilde{\sigma}})_i}{|\{v \in V \mid \sigma(v)=i\}|}.$$

Unfold $P^\sigma \subseteq \mathbb{Q}^n$ into $(P^\sigma)^{-\sigma} \subseteq \mathbb{Q}^V$.

For $v \in V$,

$$(x^{-\sigma})_v := x_{\sigma(v)}.$$

Properties

- P^σ is a polytope.
- $\langle P^\sigma \rangle = \text{poly}(\langle P \rangle)$.
- An optimum of P^σ gives an optimum of P .
- $\text{SEP}(P^\sigma, x)$ reduces to $\text{SEP}(P, x^{-\sigma})$, but...
only if output c agrees with σ .

Graph Matching

Recall, in a *graph* $G = (V, E)$ a matching $M \subset E$ is a set of edges such that each vertex is incident on *at most* one edge in M .

We saw that the existence of a *perfect matching* is not definable in FP.

(Blass, Gurevich, Shelah 1999) showed that for *bipartite* graphs this is definable in FPC.

They conjectured that this was *not* the case for general graphs.

We consider the more general problem of determining the *maximum weight* of a matching in a *weighted graph*:

$$G = (V, E) \quad w : E \rightarrow \mathbb{Q}_{\geq 0}$$

The Matching Polytope

(Edmonds 1965) showed that the problem of finding a *maximum weight matching* in $G = (V, E)$ $w : \mathbb{Q}_{\geq 0}^E$ can be expressed as the following linear programming problem

$$\begin{aligned} \max w^\top y \quad & \text{subject to} \\ Ay &\leq 1^V, \\ y_e &\geq 0, \quad \forall e \in E, \\ \sum_{e \in E \cap W^2} y_e &\leq \frac{1}{2}(|W| - 1), \quad \forall W \subseteq V \text{ with } |W| \text{ odd}, \end{aligned} \tag{1}$$

Matching in FPC

A *separation oracle* for this polytope is definable by an FPC formula interpreted in the weighted graph G .

As a consequence, there is an FPC formula defining the *size* of the maximum matching in G .

Note that this does not allow us to define an *actual* matching.

Maximum Flow

MAXFLOW

Given: A capacitated graph $G = (V, c)$, with $c : V \times V \rightarrow \mathbb{Q}_{\geq 0}$
and $s, t \in V$.

Determine: $f : V \times V \rightarrow \mathbb{Q}_{\geq 0}$ optimising

$$\max \sum_{v \in V} (f(v, t) - f(t, v)) \quad \text{subject to}$$

$$\sum_{v \in V} (f(v, u) - f(u, v)) = 0, \quad \forall u \in V \setminus \{s, t\}$$

$$0 \leq f(u, v) \leq c(u, v), \quad \forall u \neq v \in V.$$

Lemma

MAXFLOW \in FPC.

Proof: Polytope is explicit. Use explicit SEP with FPC reduction.

Minimum Cut

MINCUT

Given: A capacitated graph $G = (V, c)$, with $c : V \times V \rightarrow \mathbb{Q}_{\geq 0}$ and $s, t \in V$.

Determine: A set $C \subseteq V$ with $s \in C$, $t \notin C$, and minimising

$$\sum_{u \in C, v \in V \setminus C} c(u, v).$$

Lemma

MINCUT \in FPC.

Proof:

- Compute max flow f in FPC.
- $C_f = \{v \in V \mid \text{non-0 capacity } s \rightsquigarrow v \text{ in residual graph } G|_f\}$

Lemma

C_f is independent of f . Its the canonical minimum (s, t) -cut of G .

Minimum Odd Cut

MINODDCUT

Given: A capacitated graph $G = (V, c)$, with $c : V \times V \rightarrow \mathbb{Q}_{\geq 0}$ and $|V|$ even.

Determine: A set $C \subseteq V$ with $|C|$ odd, and minimising

$$\sum_{u \in C, v \in V \setminus C} c(u, v).$$

Lemma

For some $s, t \in V$, the canonical min (s, t) -cut is a min odd cut.

Proof Idea: Collapse sets of vertices while preserving existence of some min odd cut.

Lemma

FPC can define a small set of min odd cuts.

Matching

b -MATCHING

Given: $G = (V, E)$ and $A \in \{0, 1\}^{V \times E}$, $b \in \mathbb{N}^V$, $c \in \mathbb{Q}_{\geq 0}^E$. **Determine:**
 $y \in \mathbb{N}_{\geq 0}^E$ optimising

$$\max c^T y \quad \text{subject to} \quad Ay \leq b, y \geq 0^E.$$

Specialises to MAXMATCHING when $b = 1^V, c = 1^E$.

Relax LP (i.e., $y \in \mathbb{Q}_{\geq 0}^E$) and add constraints consistent with integral solutions:

$$y(W) \leq \frac{1}{2}(b(W) - 1), \forall W \subseteq V \text{ with } b(W) \text{ odd.}$$

where $y(W) = \sum_{e \in E, e \subseteq W} y_e$ and $b(W) = \sum_{v \in W} b_v$.

Theorem (Edmonds '65)

The extremal points of the relaxed LP are integral.

Matching, contd.

Lemma (Padberg-Rao '82)

Given $y \in \mathbb{Q}_{\geq 0}^E$. There is exists a capacitated graph H such that y violates an odd set constraint iff H has a min odd cut of value < 1 .

- FPC can define H from y .
- FPC can define a small set of min odd cuts of H .
- FPC can define a small set of violated odd set constraints.
- FPC can define a canonical violated constraint (by linearity).

Lemma

There is an FPC interpretation $\text{fin}[\tau_{\text{match}} \uplus \tau_{\text{vec}}] \rightarrow \text{fin}[\tau_{\text{vec}}]$ expressing the separation problem for b -MATCHING polytopes with respect to their natural representation as τ_{match} -structures.

Symmetric LPs

For $s = O(2^{n^{1-\epsilon}})$, $\epsilon > 0$:

1. a symmetric circuit of size s translates to a symmetric LP of size $\text{poly}(s)$; and
2. a symmetric LP of size s translates to a formula of C^k with $k = O(\frac{\log s}{\log n})$.

So, *polynomial-size* families of symmetric circuits and symmetric LPs are *equivalent*.

Translations

The translation from circuits to linear programs starts from the one given by **Yannakakis**, but we have to

- account for *majority* (or *threshold*) gates; and
- preserve *symmetry*

To achieve these two feats simultaneously requires some work.

Linear Programs to Formulas

Starting with a linear program P defining a *symmetric polytope* $Q \subseteq \mathbb{Q}^X \times \mathbb{Q}^Y$, where $X = [n] \times [n]$, we can:

Partition Y into *orbits* under the induced action of S_n ;

replace the orbits with single variables by *linearity*.

This gives us an equivalent *reduced* linear program \hat{P} that is *rigid*.

We do not know if this can be done in polynomial-time, so we can't guarantee we get a uniform family.

Evaluating Symmetric LPs

We have \hat{P} , which defines a *rigid* symmetric polytope $Q \subseteq \mathbb{Q}^X \times \mathbb{Q}^{\hat{Y}}$, where $X = [n] \times [n]$

And a graph G on n vertices.

Any bijection $\beta : V(G) \rightarrow [n]$ gives a polytope $Q_\beta \subseteq \mathbb{Q}^{\hat{Y}}$. By symmetry, these are all the same up to a permutation of \hat{Y} .

We show that we can obtain an LP equivalent to Q_β by a *C^k -interpretation* (for $k = \frac{\log s}{\log n}$) from the graph G , with advice \hat{P} .

Supports

We can show that, under the action of S_n on \hat{P} , the *stabilizer* of each variable in Y and each constraint in \hat{P} has a *support* of size $k = O(\frac{\log s}{\log n})$.

Theorem

If $n > 8$, $1 \leq k \leq n/4$, and G is a subgroup of S_n with $[S_n : G] < \binom{n}{k}$, then there is a set $S \subseteq [n]$ with $|S| < k$ such that $A_{(S)} \leq G$.

Alternating Groups

To show that we can replace the alternating group by the *symmetric group*, we cannot rely on an induction on depth, as we did with circuits.

Instead, we show that if some variable in Y does *not* have small support, we can construct a small (i.e. size $\text{poly}(s)$) graph whose *automorphism group* is isomorphic to $A_{(s)}$.

Theorem

If $n > 22$, then the number of vertices of any graph whose full automorphism group is isomorphic to A_n is at least $1/2 \binom{n}{\lfloor n/2 \rfloor} \sim 2^n / \sqrt{2\pi n}$.