Enumerator polynomials: Completeness and Intermediate Complexity

Meena Mahajan

The Institute of Mathematical Sciences, Chennai.

Joint work with Nitin Saurabh

The Classification Program of Counting Complexity
Workshop at Simons Institute for the Theory of Computing, 28 March - 1 April 2016

31 March, 2016

Hamiltonian cycles

- Input: Graph G
- Object of interest: Hamiltonian cycle
- Q1: Does G have a Ham cycle? NP-complete

Hamiltonian cycles

- Input: Graph G
- Object of interest: Hamiltonian cycle
- Q1: Does G have a Ham cycle? NP-complete
- Q2: How many Ham cycles? #P-complete

Hamiltonian cycles

- Input: Graph G
- Object of interest: Hamiltonian cycle
- Q1: Does G have a Ham cycle? NP-complete
- Q2: How many Ham cycles? #P-complete
- Q3: Describe all cycles. Enumerate them symbolically.

Hamiltonian cycles

- Input: Graph G
- Object of interest: Hamiltonian cycle
- Q1: Does G have a Ham cycle? NP-complete
- Q2: How many Ham cycles? #P-complete
- Q3: Describe all cycles. Enumerate them symbolically.
  New variable for each edge.

$$\mathrm{HamC}_n([X_{i,j}]) \triangleq \sum_{\sigma:\ n\text{-cycle}} \left(\prod_{i=1}^{n} X_{i,\sigma(i)} \right)$$

Hamiltonian cycles

- Input: Graph G
- Object of interest: Hamiltonian cycle
- Q1: Does G have a Ham cycle? NP-complete
- Q2: How many Ham cycles? #P-complete
- Q3: Describe all cycles. Enumerate them symbolically.
  New variable for each edge.

$$\mathrm{HamC}_n([X_{i,j}]) \triangleq \sum_{\sigma:\ n\text{-cycle}} \left(\prod_{i=1}^{n} X_{i,\sigma(i)} \right)$$

Hamiltonian cycles $\iff$ Monomials of HamC.

eg K_4:
1-2-3-4-1: $X_{12}X_{23}X_{34}X_{41}$
1-2-4-3-1: $X_{12}X_{24}X_{43}X_{31}$
1-3-2-4-1: $X_{13}X_{32}X_{24}X_{41}$

Hamiltonian cycles

- Input: Graph G
- Object of interest: Hamiltonian cycle
- Q1: Does G have a Ham cycle? NP-complete
- Q2: How many Ham cycles? #P-complete
- Q3: Describe all cycles. Enumerate them symbolically.
  New variable for each edge.

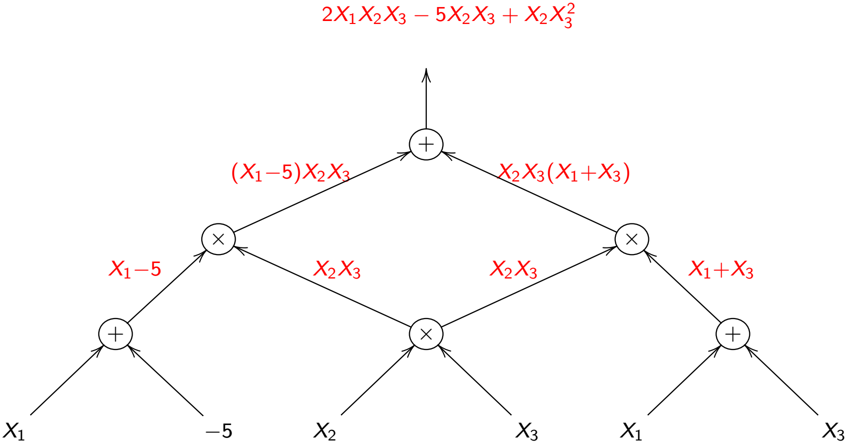
$$\mathrm{HamC}_n([X_{i,j}]) \triangleq \sum_{\sigma:\ n\text{-cycle}} \left(\prod_{i=1}^{n} X_{i,\sigma(i)} \right)$$

Hamiltonian cycles $\Longleftrightarrow$ Monomials of HamC.

eg K_4: 1-2-3-4-1: $X_{12}X_{23}X_{34}X_{41}$
1-2-4-3-1: $X_{12}X_{24}X_{43}X_{31}$
1-3-2-4-1: $X_{13}X_{32}X_{24}X_{41}$

- HamC must be "hard". In what computation model?

Algebraic computation models: Circuits



Arithmetic Circuit Families

Circuit family (C_n) computes polynomial family (p_n).

Family $\{f_n\}_{n>0}$ is a p-family if degree and number of variables in f_n grows polynomially in n.

Now onwards, only p-families.

Algebraic Complexity Classes

- VP: *p-computability*; polynomial size circuits.
- VNP: *p-definability*; exponential sums of partial Boolean instantiations of polynomials in VP.
  $(f_n) \in$ VNP if there exist $(g_m) \in VP$ and polynomial $r(n)$:
  $$f_n(\tilde{x}) = \sum_{\tilde{y} \in \{0,1\}^{t(n)}} g_{r(n)}(\tilde{x}, \tilde{y})$$

(Defined by Valiant in 1979; algebraic analogues of P, NP.)

Algebraic Reductions: Projections

- $(\mathrm{HamC}_n) \in$ VNP.
- (HamC_n) hard for VNP with respect to p-projections.

Algebraic Reductions: Projections

- $(\mathrm{HamC}_n) \in$ VNP.
- (HamC_n) hard for VNP with respect to p-projections.
- projections – Example: $g(x_1, x_2, x_3, x_4) = x_1x_2 + x_3x_4$.

projections of g		not projections of g
$y_1 + y_2$	$= g(y_1, 1, y_2, 1)$	$y_1^2 y_2$
$y_1y_2 + 5$	$= g(y_1, y_2, 1, 5)$	(too high degree)
$y_1y_2 + y_2y_3$	$= g(y_1, y_2, y_2, y_3)$	$y_1 + y_2 + y_3$
$2y^2$	$= g(y, y, y, y)$	(too many terms)

Algebraic Reductions: Projections

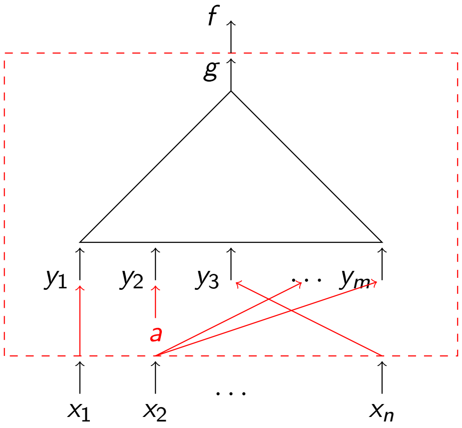
- $(\mathrm{HamC}_n) \in$ VNP.
- (HamC_n) hard for VNP with respect to p-projections.
- projections – Example: $g(x_1, x_2, x_3, x_4) = x_1x_2 + x_3x_4$.

projections of g		not projections of g
$y_1 + y_2$	$= g(y_1, 1, y_2, 1)$	$y_1^2 y_2$
$y_1y_2 + 5$	$= g(y_1, y_2, 1, 5)$	(too high degree)
$y_1y_2 + y_2y_3$	$= g(y_1, y_2, y_2, y_3)$	$y_1 + y_2 + y_3$
$2y^2$	$= g(y, y, y, y)$	(too many terms)

  $f \leq_{proj} g$ if circuit for g can be used to compute f, with no extra gates.

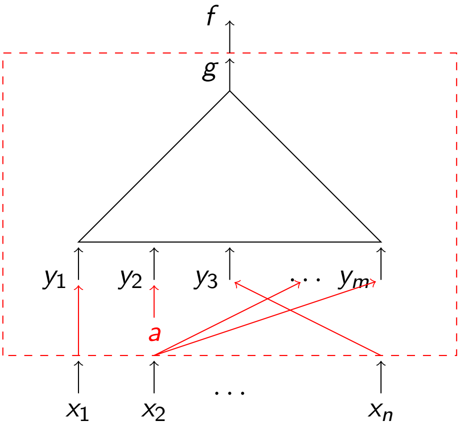
- p-projection: $f_n \leq_{proj} g_{m(n)}$ for some poly $m(.)$.

Algebraic Reductions: Projections



f is a projection of g

Algebraic Reductions: Projections



f is a projection of g

f is a p-projection of g if $m(n) \in n^{O(1)}$.

Other Hard "Enumerator" Polynomials

- Enumerating Cliques:

$$\mathrm{Clique}_n \triangleq \sum_{A \subseteq [n]} \left(\prod_{i,j \in A, i<j} X_{i,j} \right)$$

Other Hard "Enumerator" Polynomials

- Enumerating Cliques:

$$\text{Clique}_n \triangleq \sum_{A \subseteq [n]} \left(\prod_{i,j \in A, i<j} X_{i,j} \right) = \sum_{\substack{T \subseteq E_n : (V_n, T) \text{ is clique} \\ + \text{ isolated vertices}}} \left(\prod_{e \in T} X_e \right)$$

VNP-complete with respect to p-projections

Other Hard "Enumerator" Polynomials

- Enumerating Cliques:

$$\text{Clique}_n \triangleq \sum_{A\subseteq[n]} \left(\prod_{i,j\in A, i<j} X_{i,j} \right) = \sum_{\substack{T\subseteq E_n:(V_n,T) \text{ is clique} \\ + \text{ isolated vertices}}} \left(\prod_{e\in T} X_e \right)$$

VNP-complete with respect to p-projections

- Enumerating Bipartite Perfect Matchings:

$$\text{Perm}_n \triangleq \sum_{\substack{M \text{ a perfect} \\ \text{matching in } K_{n,n}}} \left(\prod_{(u_i,v_j)\in M} X_{i,j} \right) = \sum_{\sigma\in S_n} \left(\prod_{i\in[n]} X_{i,\sigma(i)} \right)$$

VNP-complete with respect to p-projections
(over fields of characteristic $\neq 2$).

A remarkable enumerator polynomial

$$\mathsf{Cut}_n(X) \triangleq \sum_{(A,B) \text{ partition of } [n]} \left(\prod_{i \in A, j \in B} X_{i,j} \right).$$

eg: $\mathsf{Cut}_3(X) = 1 + X_{1,2}X_{1,3} + X_{1,2}X_{2,3} + X_{1,3}X_{2,3}$.

A remarkable enumerator polynomial

$$\mathrm{Cut}_n(X) \triangleq \sum_{(A,B) \text{ partition of } [n]} \left(\prod_{i \in A, j \in B} X_{i,j} \right).$$

eg: $\mathrm{Cut}_3(X) = 1 + X_{1,2}X_{1,3} + X_{1,2}X_{2,3} + X_{1,3}X_{2,3}$.

(Cut_n) is in VNP. What's remarkable?

A remarkable enumerator polynomial

$$\mathrm{Cut}_n(X) \triangleq \sum_{(A,B) \text{ partition of } [n]} \left(\prod_{i \in A, j \in B} X_{i,j} \right).$$

eg: $\mathrm{Cut}_3(X) = 1 + X_{1,2}X_{1,3} + X_{1,2}X_{2,3} + X_{1,3}X_{2,3}$.

(Cut_n) is in VNP. What's remarkable?

Theorem (Bürgisser (1999))

Over the field GF[2],
(Cut_n) *is neither in* VP, *nor* VNP-*hard (with respect to p-projections),*
unless all languages in $\oplus$P *(Mod*$_2$*P) have polynomial-size circuits*
and hence PH *collapses to second level.*

Intermediate Complexity

- (Boolean world) Ladner's theorem (1975): If $P \neq NP$, then there is a language in NP that is neither in P nor NP-hard.
- (Algebraic world) Bürgisser (1999): Over every field, if $VP \neq VNP$, then there is a polynomial family in VNP that is neither in VP nor VNP-hard.

Intermediate Complexity

- (Boolean world) Ladner's theorem (1975): If P $\neq$ NP, then there is a language in NP that is neither in P nor NP-hard.
- (Algebraic world) Bürgisser (1999): Over every field, if VP $\neq$ VNP, then there is a polynomial family in VNP that is neither in VP nor VNP-hard.
- **Existence** of intermediate-complexity demonstrated (using diagonalisation).

Intermediate Complexity

- (Boolean world) Ladner's theorem (1975): If P $\neq$ NP, then there is a language in NP that is neither in P nor NP-hard.
- (Algebraic world) Bürgisser (1999): Over every field, if VP $\neq$ VNP, then there is a polynomial family in VNP that is neither in VP nor VNP-hard.
- **Existence** of intermediate-complexity demonstrated (using diagonalisation).
- Over GF[2], **explicit** polynomial: the cut enumerator. (using an additional assumption about $\oplus$P)

Intermediate Complexity

- (Boolean world) Ladner's theorem (1975): If P $\neq$ NP, then there is a language in NP that is neither in P nor NP-hard.
- (Algebraic world) Bürgisser (1999): Over every field, if VP $\neq$ VNP, then there is a polynomial family in VNP that is neither in VP nor VNP-hard.
- **Existence** of intermediate-complexity demonstrated (using diagonalisation).
- Over GF[2], **explicit** polynomial: the cut enumerator. (using an additional assumption about $\oplus$P)

  Over other fields?

Intermediate Complexity

- (Boolean world) Ladner's theorem (1975): If P $\neq$ NP, then there is a language in NP that is neither in P nor NP-hard.
- (Algebraic world) Bürgisser (1999): Over every field, if VP $\neq$ VNP, then there is a polynomial family in VNP that is neither in VP nor VNP-hard.
- **Existence** of intermediate-complexity demonstrated (using diagonalisation).
- Over GF[2], **explicit** polynomial: the cut enumerator. (using an additional assumption about $\oplus$P)

  Over other fields?
- Over $\mathbb{R}$, Cut_n is in fact VNP-complete. [deRugy-Altherre 2012]

Intermediate Complexity over finite fields

Fix field $\mathbb{F}_q$ of size q, characteristic p.

$$\mathrm{Cut^q}_n(X) \triangleq \sum_{(A,B) \text{ partition of } [n]} \left(\prod_{i \in A, j \in B} (X_{i,j})^{q-1} \right)$$

Intermediate Complexity over finite fields

Fix field $\mathbb{F}_q$ of size q, characteristic p.

$$\mathrm{Cut^q}_n(X) \triangleq \sum_{(A,B) \text{ partition of } [n]} \left(\prod_{i \in A, j \in B} (X_{i,j})^{q-1} \right)$$

Theorem (Bürgisser (1999))

Over the field $\mathbb{F}_q$, $(\mathrm{Cut^q}_n)$ *is in* VNP. *It is*

- *not* VNP*-hard with respect to p-projections, and*
- *not in* VP,

unless all languages in $\mathrm{Mod}_p\mathrm{P}$ *have polynomial-size circuits (and hence* PH *collapses to second level).*

Intermediate Complexity over finite fields

Fix field $\mathbb{F}_q$ of size q, characteristic p.

$$\mathrm{Cut}^{\mathsf{q}}{}_n(X) \triangleq \sum_{(A,B) \text{ partition of } [n]} \left(\prod_{i \in A, j \in B} (X_{i,j})^{q-1} \right)$$

Theorem (Bürgisser (1999))

Over the field $\mathbb{F}_q$, $(\mathrm{Cut}^{\mathsf{q}}{}_n)$ is in VNP. It is

- *not VNP-hard with respect to p-projections, and*
- *not in VP,*

unless all languages in $\mathrm{Mod}_p\mathrm{P}$ have polynomial-size circuits (and hence PH collapses to second level).

Since 1999, these were the only known intermediate-complexity polynomials.

New Intermediate Polynomials!

- Why HamC, Clique are hard:
  monomials encode (weights of) hard-to-find combinatorial objects

New Intermediate Polynomials!

- Why HamC, Clique are hard:
  monomials encode (weights of) hard-to-find combinatorial objects
- We put even more information into the encoding. Surprisingly, this gives easier polynomials, of intermediate complexity!

New Intermediate Polynomials!

- Why HamC, Clique are hard:
  monomials encode (weights of) hard-to-find combinatorial objects
- We put even more information into the encoding. Surprisingly, this gives easier polynomials, of intermediate complexity!
  - Clique encoded differently.
  - Vertex Cover
  - Closed Walks
  - 3-dimensional matchings
  - 3-SAT

Clique polynomial, redefined

Old definition:

$$\text{Clique}_n \triangleq \sum_{\substack{T \subseteq E_n : (V_n, T) \text{ is clique} \\ + \text{isolated vertices}}} \left(\prod_{e \in T} X_e \right)$$

Clique polynomial, redefined

Old definition:

$$\mathrm{Clique}_n \triangleq \sum_{\substack{T \subseteq E_n : (V_n, T) \text{ is clique} \\ + \text{isolated vertices}}} \left(\prod_{e \in T} X_e \right)$$

Our definition for GF[2]:

$$\mathsf{CIS}_n \triangleq \sum_{T \subseteq E_n} \left(\prod_{e \in T} X_e \right) \left(\prod_{v \text{ incident on } T} Y_v \right)$$

Clique polynomial, redefined

Old definition:

$$\text{Clique}_n \triangleq \sum_{\substack{T \subseteq E_n : (V_n, T) \text{ is clique} \\ + \text{isolated vertices}}} \left(\prod_{e \in T} X_e \right)$$

Our definition for GF[2]:

$$\mathsf{CIS}_n \triangleq \sum_{T \subseteq E_n} \left(\prod_{e \in T} X_e \right) \left(\prod_{v \text{ incident on } T} Y_v \right)$$

In K_3, T	$\emptyset$	$\{12\}$	$\{12, 23\}$	E
Monomial	1	$X_{1,2} Y_1 Y_2$	$X_{1,2} X_{2,3} Y_1 Y_2 Y_3$	$X_{1,2} X_{2,3} X_{1,3} Y_1 Y_2 Y_3$

Clique polynomial, redefined

Old definition:

$$\text{Clique}_n \triangleq \sum_{\substack{T \subseteq E_n : (V_n, T) \text{ is clique} \\ + \text{isolated vertices}}} \left(\prod_{e \in T} X_e \right)$$

Our definition for GF[2]:

$$\mathsf{CIS}_n \triangleq \sum_{T \subseteq E_n} \left(\prod_{e \in T} X_e \right) \left(\prod_{v \text{ incident on } T} Y_v \right)$$

In K_3, T	$\emptyset$	$\{12\}$	$\{12, 23\}$	E
Monomial	1	$X_{1,2} Y_1 Y_2$	$X_{1,2} X_{2,3} Y_1 Y_2 Y_3$	$X_{1,2} X_{2,3} X_{1,3} Y_1 Y_2 Y_3$

For other fields $\mathbb{F}_q$:

$$\mathsf{CIS^q}_n \triangleq \sum_{T \subseteq E_n} \left(\prod_{e \in T} (X_e)^{q-1} \right) \left(\prod_{v \text{ incident on } T} (Y_v)^{q-1} \right)$$

3Sat polynomial (over GF[2])

Cl_n: Set of all possible 3-literal clauses on n variables.

$$\mathrm{Sat}_n \triangleq \sum_{a \in \{0,1\}^n} \left(\prod_{i \in [n] : a_i = 1} X_i \right) \left(\prod_{\substack{c \in \mathrm{Cl}_n : \\ a \text{ satisfies } c}} Y_c \right)$$

Closed-Walk polynomial (over GF[2])

Clow: Closed walk, not necessarily simple.
Smallest vertex visited exactly once.

Closed-Walk polynomial (over GF[2])

Clow: Closed walk, not necessarily simple.
Smallest vertex visited exactly once.

$$\mathrm{Clow}_n \triangleq \sum_{\substack{w=\langle v_0,v_1,\ldots,v_{n-1}\rangle: \\ \forall j>0, \quad v_0<v_j}} \left(\prod_{i\in[n]} X_{(v_{i-1},v_{i \bmod n})} \right) \left(\prod_{v\in\{v_0,v_1,\ldots,v_{n-1}\}} Y_v \right)$$

Closed-Walk polynomial (over GF[2])

Clow: Closed walk, not necessarily simple.
Smallest vertex visited exactly once.

$$\text{Clow}_n \triangleq \sum_{\substack{w=\langle v_0,v_1,\ldots,v_{n-1}\rangle: \\ \forall j>0, \quad v_0<v_j}} \left(\prod_{i\in[n]} X_{(v_{i-1},v_{i \bmod n})} \right) \left(\prod_{v\in\{v_0,v_1,\ldots,v_{n-1}\}} Y_v \right)$$

Clow 1-2-3-2-3-1: $X_{1,2}X_{2,3}^2X_{3,2}X_{3,1}Y_1Y_2Y_3$

Clow 1-2-2-2-2-1: $X_{1,2}X_{2,2}^3X_{2,1}Y_1Y_2$

Vertex Cover polynomial (over GF[2])

$$\mathsf{VC}_n \triangleq \sum_{S \subseteq V_n} \left(\prod_{e \in E_n :\ e \text{ is incident on } S} X_e \right) \left(\prod_{v \in S} Y_v \right)$$

3-Dimensional Matching polynomial (over GF[2])

$$\mathrm{3DM^q}_n := \sum_{M \subseteq A_n \times B_n \times C_n} \left(\prod_{e \in M} X_e \right) \left(\prod_{\substack{v \in M \\ \text{(counted only once)}}} Y_v \right)$$

Why these are intermediate ...

Following Bürgisser's strategy,

For h any of the polynomials (Cut, CIS, Sat, Clow, VC, 3DM), show that:

Why these are intermediate ...

Following Bürgisser's strategy,

For h any of the polynomials (Cut, CIS, Sat, Clow, VC, 3DM), show that:

M: Membership. h is in VNP.

Why these are intermediate ...

Following Bürgisser's strategy,

For h any of the polynomials (Cut, CIS, Sat, Clow, VC, 3DM), show that:

M: Membership. h is in VNP.

E: Ease. Over GF[2], h can be evaluated in P.
(Hence, if h is VNP-hard, then $\oplus$P has small circuits.)

Why these are intermediate ...

Following Bürgisser's strategy,

For h any of the polynomials (Cut, CIS, Sat, Clow, VC, 3DM), show that:

M: Membership. h is in VNP.

E: Ease. Over GF[2], h can be evaluated in P.
(Hence, if h is VNP-hard, then $\oplus$P has small circuits.)

H: Hardness. The monomials of h encode solutions to a problem that is #P-hard via parsimonious reductions.
(Hence, if h is in VP, then $\oplus$P has small circuits.)

Why Sat is intermediate

$$\mathsf{Sat}_n \triangleq \sum_{a \in \{0,1\}^n} \left(\prod_{i \in [n] : a_i = 1} X_i \right) \left(\prod_{\substack{c \ \in \mathsf{Cl}_n : \\ a \text{ satisfies } c}} Y_c \right)$$

Why Sat is intermediate

$$\mathsf{Sat}_n \triangleq \sum_{a\in\{0,1\}^n} \left(\prod_{i\in[n]:a_i=1} X_i \right) \left(\prod_{\substack{c\ \in \mathsf{Cl}_n: \\ a \text{ satisfies } c}} Y_c \right)$$

Ease: Given a 0-1 assignment to $\tilde{X}$ and $\tilde{Y}$, $\mathsf{Sat}_n(\tilde{x}, \tilde{y})$ equals $\# \{a\colon x_i = 0 \implies a_i = 0 \text{ and } y_c = 0 \implies a \text{ does not satisfy } c\}$.

Why Sat is intermediate

$$\mathrm{Sat}_n \triangleq \sum_{a\in\{0,1\}^n} \left(\prod_{i\in[n]:a_i=1} X_i \right) \left(\prod_{\substack{c\ \in \mathrm{Cl}_n: \\ a \text{ satisfies } c}} Y_c \right)$$

Ease: Given a 0-1 assignment to $\tilde{X}$ and $\tilde{Y}$, $\mathrm{Sat}_n(\tilde{x}, \tilde{y})$ equals $\# \{a\colon x_i = 0 \implies a_i = 0 \text{ and } y_c = 0 \implies a \text{ does not satisfy } c\}$.
This equals $2^{\text{number of unconstrained bits}}$.

Why Sat is intermediate

$$\mathrm{Sat}_n \triangleq \sum_{a\in\{0,1\}^n} \left(\prod_{i\in[n]:a_i=1} X_i \right) \left(\prod_{\substack{c\ \in \mathrm{Cl}_n: \\ a \text{ satisfies } c}} Y_c \right)$$

Ease: Given a 0-1 assignment to $\tilde{X}$ and $\tilde{Y}$, $\mathrm{Sat}_n(\tilde{x}, \tilde{y})$ equals
$\# \{a: x_i = 0 \implies a_i = 0 \text{ and } y_c = 0 \implies a \text{ does not satisfy } c\}$.
This equals $2^{\text{number of unconstrained bits}}$.

Hard: Given any 3-CNF formula F on n variables with m clauses,
For clauses $c \in F$, set all $Y_c = t$; set other Y_c to 1. Set all X_i to 1.

$$\mathrm{Sat}_n(t) = \sum_{a\in\{0,1\}^n} \left(\prod_{\substack{c\ \in F: \\ a \text{ satisfies } c}} t \right) = \sum_{a\in\{0,1\}^n} t^{(\text{number of clauses sat by } a)}$$

Coefficient of t^m equals $\#F \pmod 2$.

Why CIS is intermediate

$$\mathsf{CIS}_n \triangleq \sum_{T \subseteq E_n} \left(\prod_{e \in T} X_e \right) \left(\prod_{v \text{ incident on } T} Y_v \right)$$

Why CIS is intermediate

$$\mathsf{CIS}_n \triangleq \sum_{T \subseteq E_n} \left(\prod_{e \in T} X_e \right) \left(\prod_{v \text{ incident on } T} Y_v \right)$$

Ease: Given a 0-1 assignment to $\tilde{X}$ and $\tilde{Y}$,

Discard vertices v with $Y_v = 0$; discard edges e touching discarded vertices or with $X_e = 0$.

ℓ edges remain. Each subset of these edges contributes 1.
Value: $2^\ell \pmod 2$; 1 iff $\ell = 0$.

Why CIS is intermediate

$$\mathsf{CIS}_n \triangleq \sum_{T \subseteq E_n} \left(\prod_{e \in T} X_e \right) \left(\prod_{v \text{ incident on } T} Y_v \right)$$

Ease: Given a 0-1 assignment to $\tilde{X}$ and $\tilde{Y}$,

Discard vertices v with $Y_v = 0$; discard edges e touching discarded vertices or with $X_e = 0$.

ℓ edges remain. Each subset of these edges contributes 1.
Value: $2^\ell \pmod 2$; 1 iff $\ell = 0$.

Hard: Given any graph $G = (V, E)$,

Set all $Y_v = t$; Set $X_e = z$ if $e \in E$, $X_e = 1$ otherwise.

$$\mathsf{CIS}(z, t) = \sum_{T \subseteq E_n} z^{|T \cap E(G)|} t^{(\text{number of vertices incident on } T)}$$

Coefficient of $z^{\binom{k}{2}} t^k$ = Number of cliques of size k, $\pmod 2$.

Why Clow is intermediate

$$\mathrm{Clow}_n \triangleq \sum_{\substack{w=\langle v_0,v_1,\ldots,v_{n-1}\rangle: \\ \forall j>0, \quad v_0<v_j}} \left(\prod_{i\in[n]} X_{(v_{i-1},v_{i \bmod n})} \right) \left(\prod_{v\in\{v_0,v_1,\ldots,v_{n-1}\}} Y_v \right)$$

Why Clow is intermediate

$$\text{Clow}_n \triangleq \sum_{\substack{w=\langle v_0,v_1,\ldots,v_{n-1}\rangle: \\ \forall j>0, \quad v_0<v_j}} \left(\prod_{i\in[n]} X_{(v_{i-1},v_{i \bmod n})} \right) \left(\prod_{v\in\{v_0,v_1,\ldots,v_{n-1}\}} Y_v \right)$$

Ease: Given a 0-1 assignment to $\tilde{X}$ and $\tilde{Y}$,

Discard vertices v with $Y_v = 0$; discard edges e with $X_e = 0$.

In resulting graph, find number of clows of length n, modulo 2, by powering the adjacency matrix.

Why Clow is intermediate

$$\text{Clow}_n \triangleq \sum_{\substack{w=\langle v_0,v_1,\ldots,v_{n-1}\rangle: \\ \forall j>0, \quad v_0<v_j}} \left(\prod_{i\in[n]} X_{(v_{i-1},v_{i \bmod n})} \right) \left(\prod_{v\in\{v_0,v_1,\ldots,v_{n-1}\}} Y_v \right)$$

Ease: Given a 0-1 assignment to $\tilde{X}$ and $\tilde{Y}$,

Discard vertices v with $Y_v = 0$; discard edges e with $X_e = 0$.

In resulting graph, find number of clows of length n, modulo 2, by powering the adjacency matrix.

Hard: Given any graph $G = (V, E)$,

Set all $Y_v = t$; Set $X_e = z$ if $e \in E$, $X_e = 1$ otherwise.

$$\text{Clow}(z, t) = \sum_{w\text{: clow of length } n} z^{|w\cap E|} t^{(\text{number of vertices in } w)}$$

Coefficient of $z^n t^n$ = Number of Hamilton cycles $\pmod 2$.

Enumerating Graph Homomorphisms

Graphs G, H.

Homomorphism from G to H:
a map $\phi : V(G) \to V(H)$ preserving adjacencies.

- Object of interest: Homomorphism from G to H
- Q1: Is there a homomorphism $G \to H$?
- Q2: How many homomorphisms?
- Q3: Describe all homomorphisms; Enumerate them symbolically.

Enumerator Polynomial for Homomorphisms

Graphs G, H.
Variables on edges of H. (Think of G as fixed.)

$$f_{G,H} \triangleq \sum_{\phi:\text{homomorphism } G \to H} \left(\prod_{(u,v)\in E(G)} Y_{(\phi(u),\phi(v))} \right)$$

Enumerator Polynomial for Homomorphisms

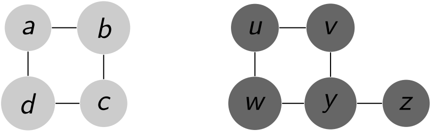
Graphs G, H.
Variables on edges of H. (Think of G as fixed.)

$$f_{G,H} \triangleq \sum_{\phi\text{:homomorphism } G \to H} \left(\prod_{(u,v) \in E(G)} Y_{(\phi(u),\phi(v))} \right)$$

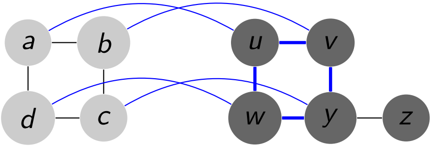
(G_n), (H_n): p-families of graphs. (size grows polynomially with n)

$f_n = f_{G_n,H_n}$.

Homomorphism Polynomials (continued)

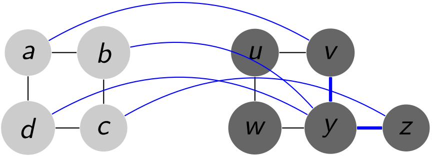


Homomorphism Polynomials (continued)



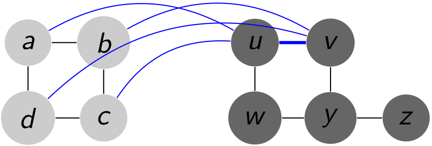
Homomorphism	Monomial
$a \to u$ $b \to v$ $c \to y$ $d \to w$	$Y_{u,v} Y_{v,y} Y_{y,w} Y_{u,w}$

Homomorphism Polynomials (continued)



Homomorphism	Monomial
$a \to v$	
$b \to y$	$Y_{v,y}^2 Y_{y,z}^2$
$c \to z$	
$d \to y$	

Homomorphism Polynomials (continued)



Homomorphism	Monomial
$a \to u$	
$b \to v$	$Y_{u,v}^4$
$c \to u$	
$d \to v$	

Rigid, incomparable graphs

- A rigid: the only homomorphism from A to A is the identity. Asymptotically, almost all graphs are rigid.

Rigid, incomparable graphs

- A rigid: the only homomorphism from A to A is the identity. Asymptotically, almost all graphs are rigid.
- $A \rightarrow B$: there exists a homomorphism from A to B.

  $A \not\rightarrow B$: there exists no homomorphism from A to B.

  A, B, incomparable: $A \not\rightarrow B$ and $B \not\rightarrow A$.

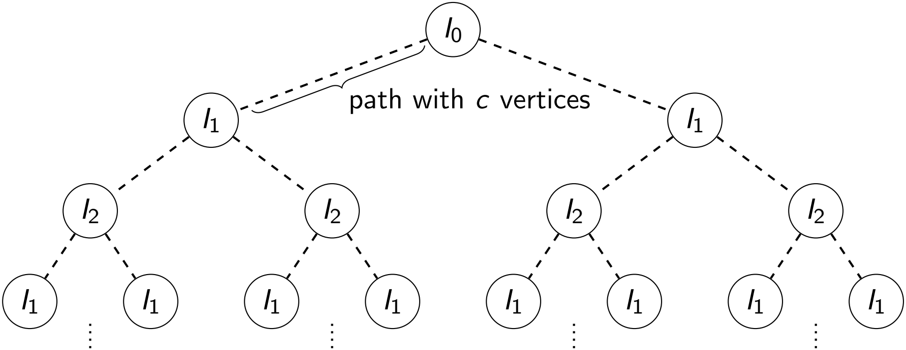
  Asymptotically, almost all pairs of graphs are incomparable.

Describing our graph families

The family (G_n):

I_0, I_1, I_2: any three rigid pairwise incomparable graphs.

Mark three nodes in each as attachment points. $c = |I_0| + |I_1| + |I_2|$.



What we show:

- The family (G_n): complete binary tree with $2^{\lceil \log n \rceil}$ leaves, "inflated" by three rigid pairwise-incomparable graphs, and "stretched" with long paths.
- The family (H_n): complete graph on n^6 vertices.

$$f_{G,H} = \sum_{\psi : V(G) \to n^6} \left(\prod_{(u,v) \in E(G)} Y_{(\psi(u),\psi(v))} \right)$$

- The family $(f_{G,H})$ is complete for VP w.r.t. p-projections.

What we show:

- The family (G_n): complete binary tree with $2^{\lceil \log n \rceil}$ leaves, “inflated” by three rigid pairwise-incomparable graphs, and “stretched” with long paths.
- The family (H_n): complete graph on n^6 vertices.

$$f_{G,H} = \sum_{\psi : V(G) \to n^6} \left(\prod_{(u,v) \in E(G)} Y_{(\psi(u),\psi(v))} \right)$$

- The family $(f_{G,H})$ is complete for VP w.r.t. p-projections.

- The family (G_n): simple path, “stretched”, endpoints “inflated” to rigid pairwise-incomparable graphs.
- The family (H_n): complete graph on n^2 vertices.
- The family $(f_{G,H})$ is complete for VBP w.r.t. p-projections.

What's the big deal?

- For VP, first natural complete family whose definition is independent of circuits and where completeness is w.r.t. p-projections.

  (Earlier work by Durand,Malod,M,Rugy-Altherre,Saurabh (2014) gave completeness w.r.t. oracle reductions, or for more artificial homomorphisms with labels and weights.)

What's the big deal?

- For VP, first natural complete family whose definition is independent of circuits and where completeness is w.r.t. p-projections.

  (Earlier work by Durand,Malod,M,Rugy-Altherre,Saurabh (2014) gave completeness w.r.t. oracle reductions, or for more artificial homomorphisms with labels and weights.)
- For VBP, complete polynomials were known – determinant, iterated matrix multiplication. This is one more.

What's the big deal?

- For VP, first natural complete family whose definition is independent of circuits and where completeness is w.r.t. p-projections.

  (Earlier work by Durand,Malod,M,Rugy-Altherre,Saurabh (2014) gave completeness w.r.t. oracle reductions, or for more artificial homomorphisms with labels and weights.)
- For VBP, complete polynomials were known – determinant, iterated matrix multiplication. This is one more.
- Our upper bounds hold whenever G_n is bounded tree-width / path-width and H_n is complete.

  (Dynamic programming approach using nice normal-form tree-width/path-width decompositions of G_n.)

Monotone p-projections

- Even more restrictive than p-projections.
- Recall projection: $f \leq_{proj} g$ if circuit for g can be used to compute f, with no extra gates.

  Now monotone projections: $f \leq_{m-proj} g$ if circuit for g can be used to compute f, with no extra gates, without using "negative" constants.

  (Makes sense over totally ordered semi-ring.
  eg $\mathbb{R}$, $\mathbb{Q}$, Boolean semi-ring.)

Why bother?

Goal: to get lower bounds for restricted circuits.

- Jukna: If HamC_n is a monotone p-projection of Perm_n, then monotone Boolean circuits for the Permanent must be of $2^{n^{\Omega(1)}}$ size.
  Current best lower bound: $n^{\log n}$ size. (Razborov 1985)
  Over reals, lower bound $2^{\Omega(n)}$. (Jerrum, Snir 1982)

Why bother?

Goal: to get lower bounds for restricted circuits.

- Jukna: If HamC_n is a monotone p-projection of Perm_n, then monotone Boolean circuits for the Permanent must be of $2^{n^{\Omega(1)}}$ size.
  Current best lower bound: $n^{\log n}$ size. (Razborov 1985)
  Over reals, lower bound $2^{\Omega(n)}$. (Jerrum, Snir 1982)
- Grochow 2015: Any monotone projection from Perm to HamC needs exponential blowup.
  If $\mathrm{HamC}_n \leq_{m-proj} \mathrm{Perm}_{t(n)}$, then $t(n) = 2^{\Omega(n)}$.

What we show:

- Over the reals (or any totally ordered semi-ring), the families Sat and Clow are not monotone p-projections of Perm.
- Any monotone affine projection from Perm to Sat must have a blow-up of at least $2^{\Omega(\sqrt{n})}$.
- Any monotone affine projection from Perm to Clow must have a blow-up of at least $2^{\Omega(n)}$.

What we show:

- Over the reals (or any totally ordered semi-ring), the families Sat and Clow are not monotone p-projections of Perm.
- Any monotone affine projection from Perm to Sat must have a blow-up of at least $2^{\Omega(\sqrt{n})}$.
- Any monotone affine projection from Perm to Clow must have a blow-up of at least $2^{\Omega(n)}$.

- More recently, Nitin Saurabh showed: Any monotone affine projection from Perm to Clique must have a blow-up of at least $2^{\Omega(\sqrt{n})}$.

Proof strategy

- (Following Grochow's idea) Associate polytopes with polynomials / solution sets.

Proof strategy

- (Following Grochow's idea) Associate polytopes with polynomials / solution sets.
- For Perm_t, polytope in $\mathbb{R}^{t^2}$, convex hull of bipartite perfect matchings in $K_{t,t}$.
  Can be described with $O(t)$ inequalities.

Proof strategy

- (Following Grochow's idea) Associate polytopes with polynomials / solution sets.
- For Perm_t, polytope in $\mathbb{R}^{t^2}$, convex hull of bipartite perfect matchings in $K_{t,t}$.
  Can be described with $O(t)$ inequalities.
- For Sat_n, polytope in $\mathbb{R}^{n+8n^3}$, convex hull of assignments+satisfied-clauses. There are formulas for which, even allowing embedding in $\geq n + 8n^3$ dimensions, $2^{\Omega(\sqrt{n})}$ inequalities are needed. (AvisTiwary2013)

Proof strategy

- (Following Grochow's idea) Associate polytopes with polynomials / solution sets.
- For Perm_t, polytope in $\mathbb{R}^{t^2}$, convex hull of bipartite perfect matchings in $K_{t,t}$.
  Can be described with $O(t)$ inequalities.
- For Sat_n, polytope in $\mathbb{R}^{n+8n^3}$, convex hull of assignments+satisfied-clauses. There are formulas for which, even allowing embedding in $\geq n + 8n^3$ dimensions, $2^{\Omega(\sqrt{n})}$ inequalities are needed. (AvisTiwary2013)
- Suppose Sat_n is a monotone projection of Perm_t. Then Sat_n polytope can be described with $O(t(n))$ inequalities. (using Grochow 2015)

Proof strategy

- (Following Grochow's idea) Associate polytopes with polynomials / solution sets.
- For Perm_t, polytope in $\mathbb{R}^{t^2}$, convex hull of bipartite perfect matchings in $K_{t,t}$.
  Can be described with $O(t)$ inequalities.
- For Sat_n, polytope in $\mathbb{R}^{n+8n^3}$, convex hull of assignments+satisfied-clauses. There are formulas for which, even allowing embedding in $\geq n + 8n^3$ dimensions, $2^{\Omega(\sqrt{n})}$ inequalities are needed. (AvisTiwary2013)
- Suppose Sat_n is a monotone projection of Perm_t. Then Sat_n polytope can be described with $O(t(n))$ inequalities. (using Grochow 2015)
- For Clow_n, polytope in $\mathbb{R}^{n^2}$, convex hull of clows.
  The Travelling SalesPerson (TSP) polytope is embedded in it.
  Any extension of TSP needs $2^{\Omega(n)}$ inequalities. (Rothvoss 2014)

Summary

1. Over finite fields, five families of enumerator polynomials shown to have complexity intermediate between VP and VNP, assuming the PH does not collapse to second level.
2. Over $\mathbb{R}$ and $\mathbb{Q}$, two of these families proved to require exponential blowup when expressed as monotone p-projections of the permanent.
3. Enumerator polynomials for graph homomorphisms: Rich canvas.
   - First natural family of polynomials defined independent of circuits and shown VP-complete w.r.t. p-projections.
   - Smooth transition to VBP-complete family.
   - VNP-complete variants also exist.

Future Directions

- Can we find polynomials with intermediate complexity over all fields? all fields with non-0 characteristic? all finite fields? all finite fields with characteristic p? finite fields with infinitely many different characteristics?
- Are there polynomials with intermediate complexity over some finite fields but obtainable as monotone p-projections of the permanent?
- Can we find polynomials enumerating homomorphisms, with intermediate complexity?

Thank You!