Complexity of generating

Vladimir Gurvich

RUTCOR & RBS
Rutgers, the State Univ of NJ
&
Research University,
Higher School of Economics
Moscow, Russia

CSR17 June 6-10, 2017, Moscow
Based on joint works with

Leonid Khachigan
Endre Boros
Kazuhisa Makino
Khaled Elbassioni
Konrad Borys
How a generation problem could be hard, in principle?

Generate all Hamiltonian cycles of a graph.

Here we can't even start but we won't consider so difficult problems.

Generate all vertices of the binary cube \([-1, 1]^n\).

It will take a long time, because there are \(2^n\) vertices. But this is not what is called hard.
3 standard concepts of the efficiency of generation

@ **Total Polynomial Time**

II = initial input
(di)graphs, matrices, vectors etc

O = \{0,1,...\} total output

total generation time

\[ T(O) = \text{poly}(|\text{III}|, |\text{O}|) \]

i.e. polynomial in both input and output size
(8) **Incremental Poly Time**

\[ O_k = \{ O_{i_1}, \ldots, O_{i_k} \} \text{ partial output (in some order)} \]

\[ T(O_{i_{k+1}}) = \text{poly}(|\text{III}|, O_k) \]

or equiv.

\[ T(O_k) = \text{poly}(|\text{III}|, k) \]

the more we output, the more time we have for the next one

(9) **Polynomial delay**

\[ T(O_{i_k}) = \text{poly}(|\text{III}|) \quad \forall k \]

or

\[ T(O_k) = k \text{poly}(|\text{III}|) \]

a trick: we can delay output if necessary
For many generation problems poly is replaced by slower quasi-poly: \[ T = 2 \cdot \text{poly} \log n \]

For example, \[ n \log n \approx 2 \log^2 n \]

while \[ \text{poly}(n) = 2^c \log n \]

Not a big difference, while \[ \exp = 2^n \]

\[ n^c = 2^c \log n < \log n = 2 \log n \ll 2^n \]

"almost" the same; both \( 2^\text{polylogn} \) \( n^c \) for practical reasons, but \( c \) should be small.
When generation is hard?

We got a partial output $O_k = \{ o_1, \ldots, o_k \}$ and obtain a "standard decision problem":

\[ Q: \text{whether the generation is complete (} n = k \text{), or } \exists \text{ more objects to output: } o_{k+1}, \ldots \]

If $Q$ is NP-hard then incremental generation is hardly possible. No easy way to output objects in any order.

Total polynomial generation theoretically still might be possible, but only artificial examples.
In this case there is no efficient total polynomial \(\mathcal{P}\) or quasi-polynomial generation algorithm, either.

Indeed, suppose such a \(\mathcal{P}\)-algorithm exists. Let it work \(\mathcal{O}(P(k))\) time.
Then, either it stops or we get the \(\mathcal{O}(k^k)\) thus solving an NP-hard problem, resp. in the negative or positive.

It may be NP-hard just to start generation. For example: generate all Hamiltonian cycles.
But we avoid so difficult problems.

On the other hand, if we already got \(\exp(k)\) many objects then generation becomes easy, because we are given a lot of time for every next object.
In a hard generation problem typically we get easily poly(II) objects, but then "suddenly" get stuck. We will give several such examples.

Remark. Thus, there are methods to show hardness of incremental generating (and hence, total generating, too) but we do not know how to demonstrate hardness that poly delay generation is hard if possible.
A NP-complete generation problem

Given a positive integer $k$ and a (multi-) hypergraph $H = (V, E)$, generate all minimal vertex-sets $V' \subseteq V$ that contain $2k$ hyperedges of $H$.

This problem is NP-complete.

Reduction from Maximal Independent Set (MIS) of size $\geq k$

Consider a graph $G = (V, E')$ and define a multi-hypergraph $H = (V, E)$ as follows. The same vertices.

Take each edge $e = (v', v'')$, $v', v'' \in V$ with multiplicity $1$. And take each vertex $v \in V$ with multiplicity only $1$. These are all hyperedges of $H$.

All $(m)$ edges of $E'$ and $(n)$ vertices of $V$ are OK. Totally, $m+n$ sets are generated.

More? Yes! Exactly MIS of size $k$.

$\exists$? NP-complete question.
Circulation cone and polyhedron of a directed graph $G = (V, E)$

$$\text{sign} (v, e) = \begin{cases} +1 & \text{if } v \xrightarrow{e} w \\ -1 & \text{if } v \xleftarrow{e} w \\ 0 & \text{if } v \xleftrightarrow{e} w \end{cases}$$

Circulation cone:
$$\sum_{e \in E} x_e \text{sign} (v, e) = 0 \quad \forall v \in V$$

To get the 0 circulation polyhedron add to (1):
$$\sum_{e \in E} x_e = -1$$

Its vertices are exactly the directed cycles of $G$, of negative weight.

Thus generating all negative (directed) cycles of a (di)graph $G$ is NP-complete.

Boros, VG, Elbassioni, Borys & Khachian 2004
\[ G = (V, E) \text{ a digraph} \]
\[ w : E \to \mathbb{R} \text{ a weight function} \]
\[ \text{Circulation cone } \text{Cone}(G, w) = \]
\[ = \left\{ x \in \mathbb{R}^E \left| \sum_{(u, v) \in E} x_{u,v} = \sum_{(v, u') \in E} x_{v,u'} \quad \forall v \in V \right. \right\} \]
\[ x_{u,v} \geq 0 \quad \forall (u, v) \in E \]

Intersect with the hyperplane
\[ \sum_{(u, v) \in E} w_{u,v} x_{u,v} = -1 \]

to get the circulation polyhedron \( P(G, w) \)

The vertices of \( P(G, w) \) are in one-to-one correspondence with the negative cycles of \( (G, w) \), while the extreme rays correspond to the zero-cycles and pairs of the negative and positive cycles.
Sausage NP-completeness technique

Let $C$ be a CNF and $D$ a DNF.

$C_0 = (x_1 \lor y_1) \land \ldots \land (x_k \lor y_k)$

$D_0 = x_1 y_1 \lor \ldots \lor x_k y_k$

The following 10 claims are equivalent to SAT:

$C \rightarrow D_0$ | $D \geq C$ | $D_0 \lor C \equiv D_0$ | $D_0 \land C \equiv C$ | $\overline{C \overline{D}_0} \equiv 0$

$D \leq C_0$ | $C_0 \leq D$ | $C \lor \overline{D} \equiv C_0$ | $C \lor \overline{D} \equiv D$ | $\overline{C \lor D_0} \equiv 1$

$C$ is satisfiable

$D$ is tautological

$C \equiv D$ is exactly Dualization

Note that $D \leq C$, $\overline{C \lor D_0} \equiv 0$ Simple
Sausage Lemma

Khachiyan & VG, 1995

[One of many versions.]

\[ G' = (V', E') \; ; \; G'' = (V'', E'') \; \; V' \cup V'' = \{8\} \]

\[ E' \cong E'' \cong \{e_1, \ldots, e_m\} \]

One-to-one correspondence between the edges.

Q: Is there a path \( p(a, c) \) that contains no pair of identified edges?

Q is NP-complete
Generating negative directed cycles is NP-complete. \textbf{Gadget.}

\[ e = (v^1, v^4) \]
\[ \overline{e} = (u^1, u^4) \]

\[ u \xrightarrow{+1} o \xrightarrow{-1} o \xrightarrow{-1} o \xrightarrow{+1} u \]

\[ u \xrightarrow{+1} o \xrightarrow{-1} o \xrightarrow{-1} o \xrightarrow{+1} u \]

There are \( m \) negative dicycles are obvious but \( \exists \) more negative dicycles IFF \( \exists \) a path \( p(a,c) \) that contains NO identified pair \((e, \overline{e})\).
Generating Minimal Dicuts is also difficult; NP-complete

Given a strongly connected digraph $G=(V,E)$

Generate all:

1) Minimal $E' \subseteq E$ such that $G'=(V,E')$ is still strongly connected

incremental polynomial [later]
can be solved by the Supergraph method

2) Minimal $E'' \subseteq E$ such that $G'=(V,E \setminus E'')$ is NOT already strongly connected; MIN dicuts

NP-complete. Again we use the "Sausage" method
Flash light method of generation results in a poly delay algorithm if successful...

Generation all (directed) paths from $S$ to $T$

Flash light problem: Given a subpath [the beginning] $S \circ 01, 35, \ldots, 35_k$, is it still possible to reach? Polynomial and easy.

The same for cycle, spanning trees etc. (Reed and Tavjan 1975 "Networks") Results but the flash light method might be hard for some problems that are not necessarily hard.
Dualization

For a given hypergraph $H = (V, E)$ generate all minimal transversals, that is, subsets $T \subseteq V$ such that $\bigcap_{E \in E} T \neq \emptyset \forall E \in E$.

Let us try to apply Flashlight:

Subtransversal = a subset $S \subseteq V$ such that $S \subseteq t$ for a minimal transversal $T \subseteq T$.

$H = \{ab, bc, cd\}$

$T = \{ac, cb, bd\}$

$\{ad\}$ is not a subtransversal. Why?

There is the following simple criterion
If \( S \subseteq V \) is a subtransversal?

\[ S \]

\[ E_1 = \{ e \in E \mid \text{len} s = 1 \} \]

\[ E_0 = \{ e \in E \mid \text{len} s = 0 \} \]

All other arcs are irrelevant.

Selection: choose \( e \in E_1 \), such that \( \forall e \in S \text{ for each } e \in S \),

Selection \( \exists e \in E_1; \forall e \in S \) is called covering if \( \exists e \in E_0 \mid e = U e \in S \).

Theorem [Boros, VG, Hammer 1998]

\( S \) is a subtransversal if \( \exists \) a non-covering selection.

For example: \( \{ a, b, c \} \) is not because \( \exists ! \) selection \( \{ a, b, c \} \) and it is covering.
By the above criterion, verifying whether \( S \) is a subtransversal is simple (polynomial) whenever \( |S| \leq \text{const} \), or large \( |V \setminus S| \leq \text{const} \).

But in general, it is \( \text{NP-hard} \) already for graphs \( |e| = 2 \neq E(G) \).

Proof. Reduction from \( \text{SAT} \)

1. Assign a clause to \( \text{HES} \)

2. Assign a variable \( \forall v \mid (v, v') \in E \) and \( v \in S \)

3. \( \neq (v', v'') \in E_0 \) assign \( x_{v'} = x_{v''} \)

Then we get a one-to-one correspondence between the satisfying assignments and non-covering selections.
Dualization itself is quasi-polynomial, \( n \log n \).

Fredman & Khachian 1986

But the corresponding flash light problem [sub transversal] is NP-complete.

The same for many other generation min problems; cuts etc.

Recognition of "subobjects" (sub cuts sub transv) is NP-complete.

It is not surprising, because when it is (quasi-) polynomial, the corresponding generation problems can be solved with poly delay not only incrementally.
Crl 1. Superclique (or Super MIS)
Given a graph $G = (V, E)$ and $V' \subseteq V$, whether $V'$ contains a MAXIMAL clique or a MIS?

This question is NP-complete
Proof by complementation

Crl 2. For hypergraphs of bounded dimension $|E| \leq \text{const}$, dualization is polynomial and, moreover, it can be efficiently done in parallel

ENC if $\text{dim} \leq 3$
ERNC if $\text{dim} = 4, 5, ...$
Supergraph approach

The objects to generate are encoded by the vertices \( V \) of a "supergraph" \( G = (V, E) \). Then the edges \( E \) show "connections".

For example, the bases or circuits of a Matroid; spanning trees, strongly connected components, etc.

Then, the connectivity of \( G \) must be verified.
Incremental quasi-polynomial generation

A powerful method is based on the following three parts:

1. **Dualization**
   - \( n \log(n) \) incremental algorithm
   - by Fredman & Khachiyan, 1994-6

2. Joint generation of dual hypergraphs
   - Khachiyan & VG, 1995, 1999
   - Bioch & Ilparaki, 1995

3. Dual boundedness
   - Inequalities: \( |H| \leq f(|H^d|) \)
   - between the sizes of dual hypergraphs
   - Khachiyan, Boros, Elbassioni, VG.
Dual hypergraphs and dualization

Given a hypergraph \( H = (V; E) \)

\[ H^d = \{ \text{all minimal transversals of } H \} \]

\[ = \{ \text{all inclusion-minimal } W \subseteq V \mid \forall e \in E \exists w \in W \} \]

By this dfn, \( H^d \) is Sperner; \( W \neq W' \)

\[ H/d/d = H \] whenever \( H \) is Sperner

or Equivalently in boolean terms:

\[ F^d(x_1, \ldots, x_n) \overset{df}{=} \overline{F(x_1, \ldots, \overline{x_n})} \]

\( F \) is monotone iff \( F^d \) is monotone

\[ (x \lor y)^d = x \land y \quad (F \lor G)^d = F^d \land G^d \]

\[ (x \land y)^d = x \lor y \quad (F \land G)^d = F^d \lor G^d \]

Given a DNF, replace \( \lor \) \( \lor \) \( \land \) \( \land \)

by

and get

the dual DNF

Berge's Algorithm of dualization
Dualization by Berge's Multiplication Algorithm

\[ D = x_1 x_2 x_3 \lor x_3 x_4 \lor x_4 x_5 x_6 \quad \text{de Morgan} \]

\[ G = (x_1 x_2 x_3) (x_3 x_4) (x_4 x_5 x_6) \quad \text{rule} \]

Multiply successively, each time by the next clause; delete absorbed implicants.

The algorithm might be exponential w.r.t. both \( D \) and \( D^d \) \( \text{current size of the product} \).

Examples such that the size grows exp. w.r.t. \( D \) and \( D^d \) for ANY order of clauses.

Even if \( D \) an efficient order of multiplying, how to find it out?
F and G are dual, \( F^d = G \), iff 
\[ H = (xF \lor yG \lor xy) \] is self-dual, \( H^d = H \)

Paul Seymour (1971) Master's Thesis

Thus, verifying duality and self-duality are poly equivalent.

Recursive procedure by F. Khachiyan (1986)

\[
F = x_i F_0(y) \lor F_1(y) \\
G = x_i G_0(y) \lor G_1(y) \\
\text{variables} \ (x_i, x_{i-1}, y, x_{i+1}, \ldots, x_n) \\
\]

F and G are dual iff
\[ F_1 \] is dual to \( G_0 \lor G_1 \) and 
\[ G_1 \] is dual to \( F_0 \lor F_1 \)

\( F^d = G \) iff \( F_1^d = G_0 \lor G_1 \) and \( G_1^d = F_0 \lor F_1 \)

Choosing a frequent variable \( x_i \),

they get \( n \log^2 n / \log \log n \times 2 \log^2 n \)

and a faster version \( n o(\log n) \)
$H = (V, E)$

$H$ is a Sperner Hypergraph

$H^c$ the complementary one

$H_d$ the dual (or transversal) one

$H_d d = H^c c = H$ and $c$ are involutions

$H_{cd}$ the minimal sets that are not contained in an edge of $H$

$H_{dc}$ the maximal sets that contain no edge of $H$

(MIS)

$i.e.$ maximal independent sets

$H \subset H_{cd} \subset H_{dc} \subset H_{dcd} \subset H_{dcd} \subset \ldots$

These sequences may be very long

$H_{cd}$, $H_{dc}$ and $H_d$ can be generated in quasi-poly time

$H^c$ trivially
In a matroid $M$

B bases = maximal independent sets = MIS ... of which hypergraph?

$B^c$ the complements = bases of the dual matroid $M^d$.

B transversals to bases (may be of different sizes, hence, not bases)

B cd minimal sets not contained in any base of $M = \text{Circuits of } M$

= Minimal dependent sets

B dc maximal sets that do not contain a base of $M$

All can be generated in polynomial time due to matroid (exchange) axioms.

Circuits Paul Seymour 1984

$C = B^{cd} = \text{circuits}$. Hence,

$C^{dc} = B^{cdde} = B = \text{bases}$. 
Joint generation of a pair of dual hypergraphs $H$ and $H^d$ is reduced to dualization (repeated many times).

Given $(H' \subseteq H$ and $H'' \subseteq H^d)$, we verify duality $H' \cap H'' = \emptyset$.

If yes, we stop and conclude that $H' = H$ and $H'' = H^d$.

If not, we conclude that $H' \neq H$ or $H'' \neq H^d$ or both.

and we can get an edge to extend either $H'$ or $H''$, but we can't control which one.

Khachiyan & VG, 1995, 1999
Bioch and Ibaraki 1995

Thus, joint gen $(H, H^d)$ is quasi-polynomial,
while gen$(H)$ or gen$(H^d)$ or both may be NP complete.
Joint generation of dual hypergraphs \( H \) and \( H^d \) is equivalent to dualization repeated many times. Hence, it is incremental quasi-polynomial. (Very unlikely, \( NP \)-hard, since otherwise any quasi-linear problem would be.)

Yet, separate generation of each \( H \) or \( H^d \) or both might be hard.

Suppose we want only \( H \) and \( H^d \) is just garbage for us, then \( |H^d| \) might be \( \gg |H| \). Then each useful object, a \( e \in H \), may be interrupted by output of many \( e \in H^d \).

Somewhat surprisingly, to generate jointly is simpler than separately.
Dual boundedness (DB) Yet, this can't happen if the size of $H^d$ is bounded by a (quasi-) polynomial in the sizes of $H$ and the initial input (II)

$|H^d| \leq \text{(quasi-)poly}(|H|, |I|)$

Then, joint generation of $H$ and $H^d$ is also efficient for $H$ alone. Although $H^d$ is just garbage for us, yet, by $\Box$ we won't have too much of it.

So, the joint generation of $H$ and $H^d$ provides a total polynomial time generation algorithm for $H$ alone. But what about incremental (quasi-) polynomial generation?
Uniform dual-Boundedness UDB

A class of hypergraphs \( \mathcal{H} \) is called UDB by a quasi-polynomial \( \mathbb{Q} \) if \( \forall H \in \mathcal{H} \) and \( \forall \) partial hypergraph \( H' \leq H \)

\[
|H'| \leq \mathbb{Q}(|H|', |\mathcal{I}|)
\]

\( \mathcal{I} \) = initial input = an oracle

Property UDB is stronger than DB and implies incremental (not only total) (quasi-)polynomial generation.

Proof is easy, like joint generation.

Given \( H' \leq H \) and \( H'' \leq H'' \), verify duality.

\( H' = H'' \). If yes then \( H' = H \) and \( H'' = H \). Stop.

If not then we get a set \( E \subseteq V \) such that

- either \( E \) can be reduced to a new edge of \( H \)
- or \( E \) can be expanded to a new MIS of \( H \).

And we do not control what happens \( \circ \) or \( \square \).
Somewhat surprisingly, many classes \( \mathcal{H} \) of hypergraphs appear to be DB.
And, even more surprisingly, all these classes are also UDB.
Only some artificial classes are DB but not UDB.

We will give many examples below.
Dual monotone generating problems

Given a monotone property $P$, generate all
a) minimal sets satisfying $P$
b) maximal sets that do not satisfy $P$

The corresponding hypergraphs are $H$ and $H^{dc} = MIS(H)$

We will consider several example UDB or NOT UDB.
Monotone integer programming

Generate $H(A, b, c) =$ all minimal integer solutions of

\[ Ax \geq b \; ; \; 0 \leq x \leq c \]

\[ A \in \mathbb{R}^{m \times n}, \; x \in \mathbb{Z}^n \; ; \; c \in \mathbb{R}^n \]

for arbitrary $A$, already solvability of (5) is NP-complete. So we consider $A \geq 0$, or a little bit more generally

\[ Ax \geq b \Rightarrow Ax' \geq b \; \forall x, x' \mid 0 \leq x \leq x' \leq c \]

generate $MIS H(A, b, c) = H^{dc}(A, b, c)$ all maximal $x$ not satisfying (5)

$(a, b)$ is a uniformly dual-bounded pair

\[ |H^{dc}(A, b, c)| \leq m \cdot n \cdot |H(A, b, c)| \]

Example

\[ |H^{dc}(A, b, c)| \geq \frac{mn}{2 \log^2 m} |H(A, b, c)| \]

Incremental polynomial, although many special cases were conjectured NP-hard earlier
Maximal frequent and minimal infrequent sets

Given a binary $m \times n$ matrix $A$.

Set of columns $C$ is called $t$-frequent if $\exists \geq t$ rows such that $= 1$

Otherwise $C$ is called $t$-infrequent

$L = \# t$-frequent sets

$\beta = \# t$-infrequent sets

$L \leq (m-t+1)\beta$  UDB-inequality

Thus, we can generate $t$-infrequent sets by an incremental quasi-polynomial generation algorithm. In contrast, generating $t$-frequent sets is NP-hard.
Minimal strongly connected subgraphs and minimal dicuts

Given a (strongly connected) digraph $G=(V,E)$, generate all
a) minimal subsets $E' \subseteq E$ such that the digraph $G'=(V,E')$ is still strongly connected.

b) minimal subsets $E'' \subseteq E$ such that the digraph $G''=(V, E \setminus E'')$ is no longer strongly connected.

Pair $(a,b)$ is NOT DB. Each set may be EXP in the size of the other.

(a) is incremental polynomial. Proof by a supergraph approach.

(b) is NP-complete. Proof by the Sean's age method.
Generalized paths, cuts, and spanning sets in graphs

Given a non-directed (connected) graph $G = (V, E)$, two poles $s$ and $t \in V$, $k$ subsets $E_1, \ldots, E_k \subseteq E$, and consider two pairs of generation problems.

Generate all

\( A1 \) minimal spanning sets, i.e., all minimal $I \subseteq [k] = \{1, \ldots, k\}$ such that $G_I = (V, U_{i \in I} E_i)$ is connected

\( B1 \) maximal complementary cuts, i.e., all maximal $I \subseteq [k]$ such that graph $G_I = (V, U_{i \in I} E_i)$ is not connected

This pair is quasi-polynomial $\text{CUB}\dagger$

\( a) \) can be incrementally $qp$ generated

\( b) \) is $\text{NP}$-complete
also given two poles $s, t \in V$, generate all:

(a) generalized $(s, t)$-paths, i.e., all minimal $I \subseteq [k]$ such that $s, t$ belong to one connected component of the graph $G_I = (V; U \cup E_i)$

(b) generalized $(s, t)$-cuts, that is, all maximal sets $I \subseteq [k]$ such that $s, t$ are in different connected components of the graph $G_I = (V; U \cup E_i)$

The pair $(a_1, b_1)$ is $U \cup B$ quasi-polynomially. Hence, $a_1$ is incrementally quasi-polynomially solvable. Three other problems $(a_2, b_2, b_1)$ are NP-complete.

The pair $(a_2, b_2)$ is NOT $U \cup B$
Spanning Linear Spaces

Let \( L = (L_1, \ldots, L_k) \) be a family of linear subspaces of a linear space \( L = F^d \) of dimension \( d \) over a field \( F \) and \( t \leq d \) be a threshold, generate all

a) minimal subsets \( I \subseteq \{1, \ldots, k\} \) such that \( \dim(L_I) \geq t \), where \( L_I = \bigcup_{i \in I} L_i \)

(b) maximal subsets \( I \subseteq \{k\} \) such that \( \dim(L_I) < t \).

If \( t = d \) then we span the whole space \( L \).

The pair \((a,b)\) is quasi-polynomially \#P

a) can be incrementally \#P generated

b) is \#P complete

Note that a) is exactly the minimization (gen. minimal set covers) of a fixed basis \( B \). If \( L_i \) is defined by a subset \( B_i \subseteq B \), \( \forall i \in \{1, \ldots, k\} \).
Polymatroid functions and systems of polymatroid inequalities

An Integer Non-negative Set-function

\[ g: 2^V \rightarrow \mathbb{Z}_+ \] is called polymatroid if

1. \( g \) is submodular
   \[ g(I' \cup I'') + g(I' \cap I'') \leq g(I') + g(I'') \]

2. \( g \) is non-decreasing
   \[ g(I') \leq g(I'') \text{ whenever } I' \preceq I'' \preceq V \]

3. \( g(\emptyset) = 0 \)

For example, \( g(I) = \dim(V \setminus \bigcup_{i \in I} T_i) \) (from the previous slide) is polymatroid.

Another example, \( g'(I) = |V| - C(V, U \setminus \bigcup_{i \in I} E_i) = |V| - \# \text{ of connected components of } (V, U \setminus \bigcup_{i \in I} E_i) \)
Given a system of polymatroid inequalities:
\[
g_j(I) \geq t_j \quad j \in \{1, \ldots, r\}
\]

$g_j$ are polymatroid set-functions and $t_j$ are thresholds quasi-polynomial in $|N| = r$. Generate all:

- [ ] minimal $I \subseteq V$ satisfying (a)
- [ ] maximal $I \subseteq V$ not satisfying (a)

Pair (a,b) is quasi-polynomially UOB.

- [ ] incrementally polynomial
- [ ] NP-complete

**UOB inequalities**

Denote by $H(g,t)$ the set of all minimal subsets of $V$ satisfying (a) and by $\text{MIS}(H(g,t))$ the set of all maximal subsets of $V$ not satisfying (a)
UB inequalities for polymatroid system

Let $H(g,t)$ be the set of all minimal subsets of $V$ satisfying

and

$\text{MIS}(H(g,t))$ be the set of all maximal subsets of $V$ not satisfying

\[ \beta = |H(g,t)| \quad \lambda = |\text{MIS}(H(g,t))| = |H^d(g,t)| \]

\[ \lambda \leq \beta \quad \log t / c_{n,\beta} \quad \text{for } \beta \geq 2 \quad (1) \]

where $c = c_{n,\beta}$ is the (unique) root of

\[ 2^c \left( c / \log \beta - 1 \right) = 1 \quad (2) \]

Note that ($\lambda \leq n$ for $\beta = 1$) (3)

(1,2,3) imply

\[ 1 = n^{-c / \log \beta} + (n\beta)^{-c / \log \beta} \geq 2 (n\beta)^{-c / \log \beta} \]

Hence, \( (\beta / c_{n,\beta}) \leq n \beta \)

Thus, we can replace (1,2) by

\[ \lambda \leq (n\beta)^{\log t} \quad (4) \]

which is a bit weaker but simpler.
For a system of $v$ inequalities we generalize as follows:

$$N \leq v \max \left( N, \beta^{\log t / c(n, \beta)} \right)$$

where $t = \max (t_1, \ldots, t_v)$ and $c(n, \beta)$ again is the (unique) root of

$$2^{c(n, \beta)} \left( n^{\frac{c(n, \beta)}{\log \beta}} \right) - 1 = 1$$

Interestingly, the coefficient in the exponent cannot be reduced.

This bound is a generalization of the Balasov bound for the number of MIS in graphs

$$2^p \leq \# \text{MIS}(G) \leq \delta^p + 1$$

$p = p(G)$ is the size of a maximum induced matching of $G$

$\delta = \delta(G) = \# \text{pairs of vertices of } G \text{ at the distance 2}$

$$\delta \leq \frac{(n-1)(n-2)}{2} \quad n = 1, 2, \ldots$$