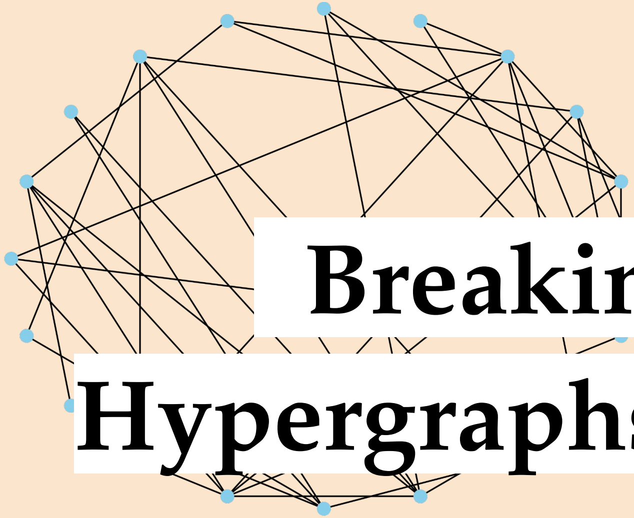
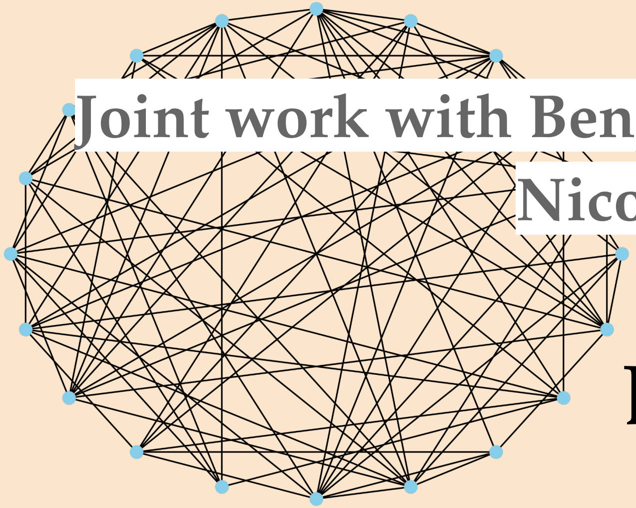
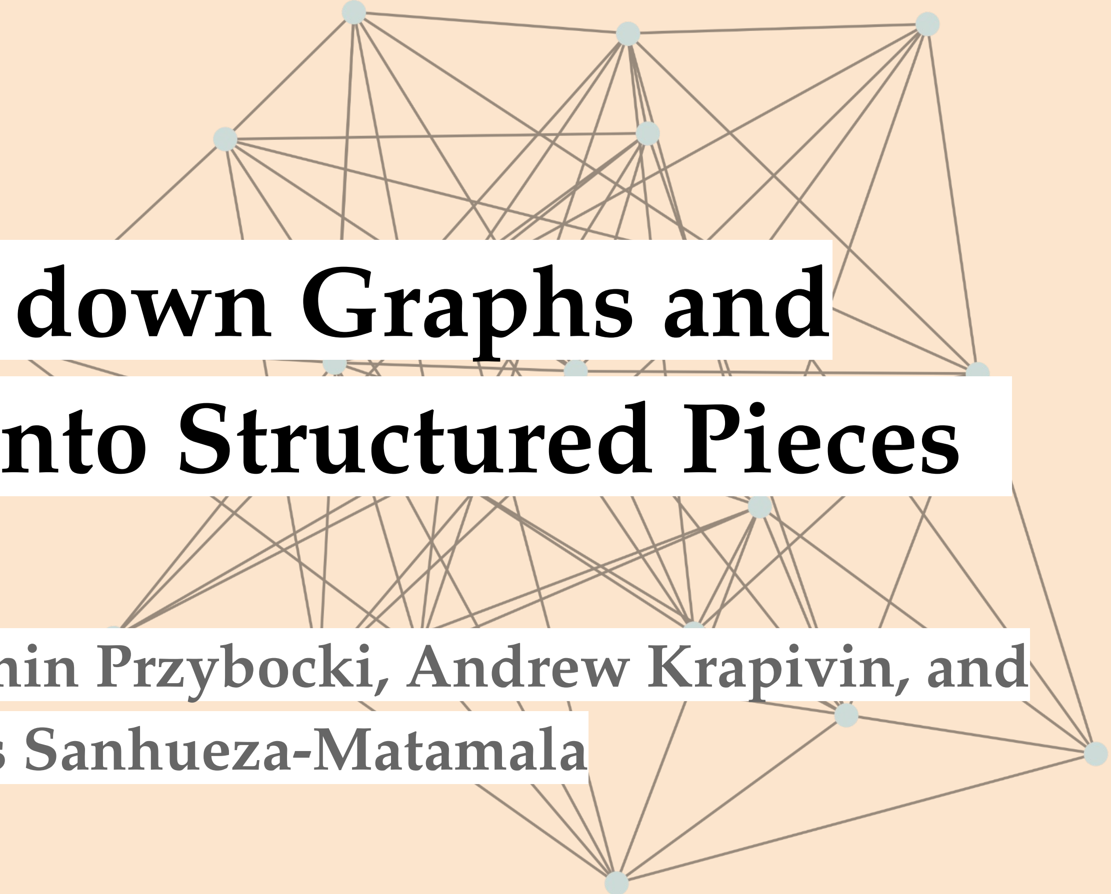




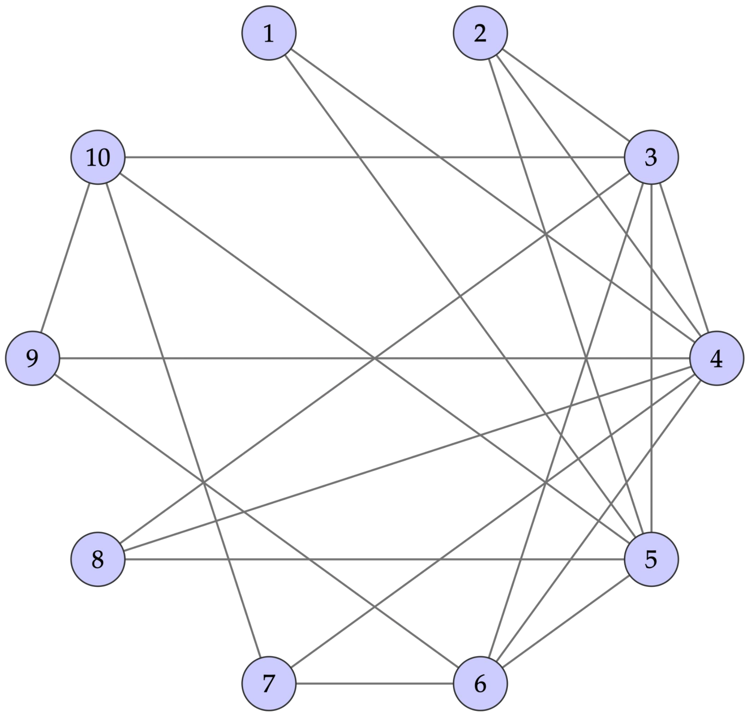


Breaking down Graphs and Hypergraphs into Structured Pieces

Joint work with Benjamin Przybocki, Andrew Krapivin, and Nicolás Sanhueza-Matamala

Bernardo Subercaseaux (CMU)

Independent Set Queries

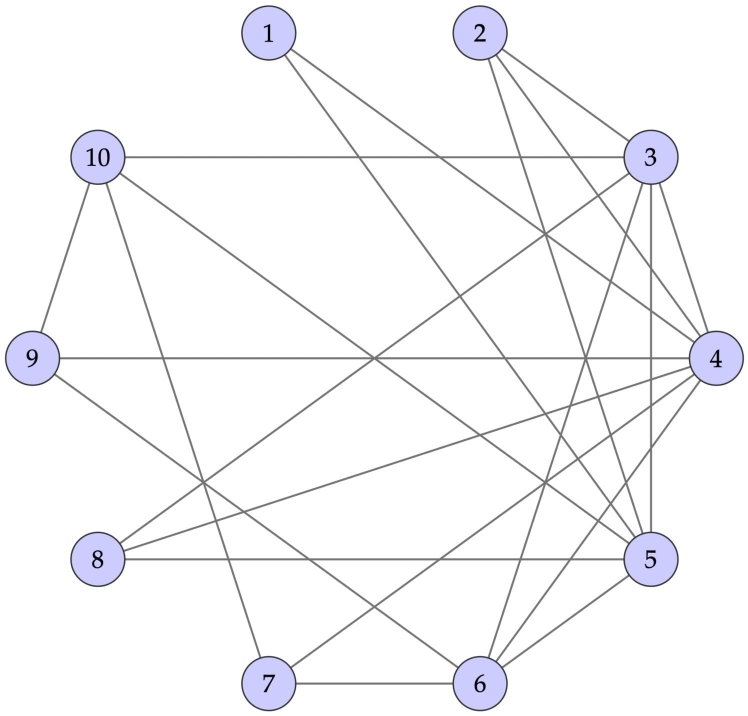


Q: is set *{2, 7, 8, 10}* independent?

⋮

Q: is set *{4, 5, 7, 9, 10}* independent?

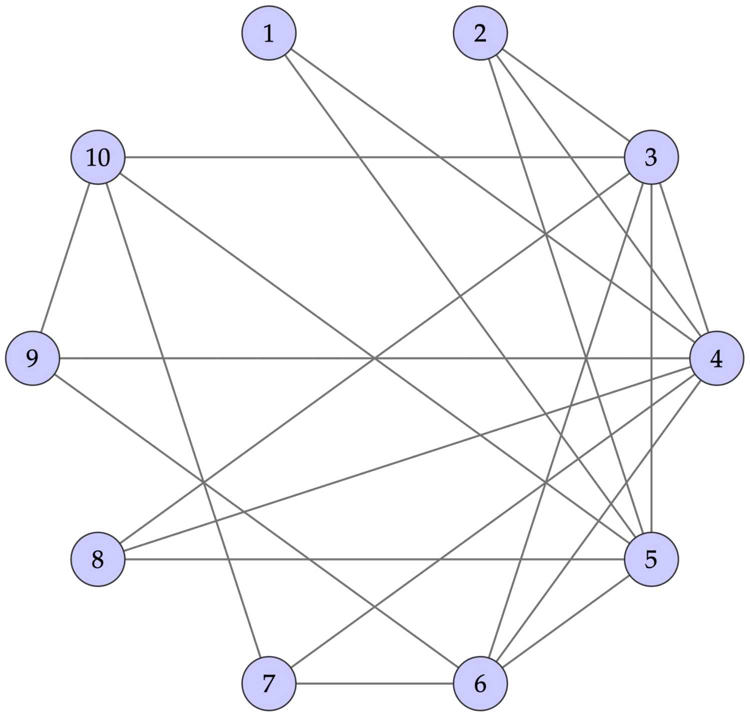
Independent Set Queries



Q: is **S** = *{2, 7, 8, 10}* independent?

Naive idea 1: $\mathbf{\Omega}(|\mathbf{S}|^2)$
*Iterate over all pairs in **S*** and check.

Independent Set Queries



Q: is **S** = *{2, 7, 8, 10}* independent?

Naive idea 1: $\Omega(|S|^2)$

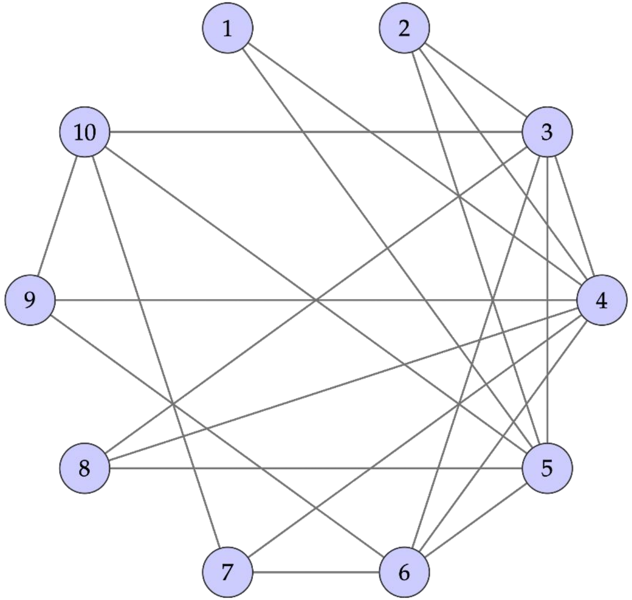
*Iterate over all pairs in **S*** and check.

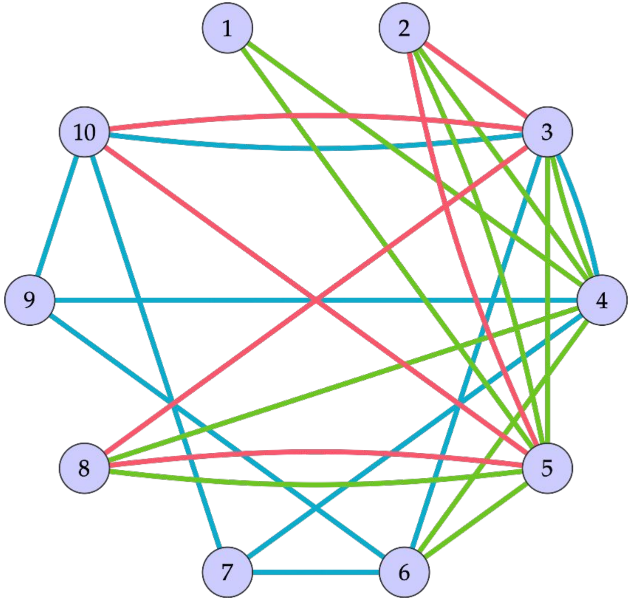
Naive idea 2: $\Omega(|E|)$

*Make bitvector for **S**, then iterate over **E***

1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	1	1	0	1

Biclique Covers



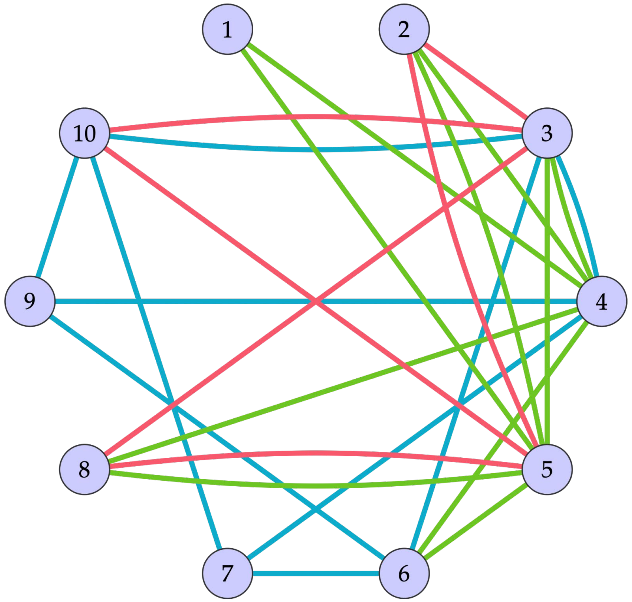


$B_1 = \{3, 7, 9\} \times \{4, 6, 10\}$

$B_2 = \{1, 2, 3, 6, 8\} \times \{4, 5\}$

$B_3 = \{2, 8, 10\} \times \{3, 5\}$

Biclique Covers



$B_1 = \{3,7,9\} \times \{4,6,10\}$

$B_2 = \{1,2,3,6,8\} \times \{4,5\}$

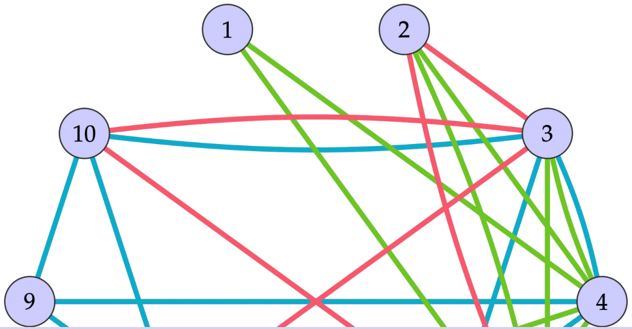
$B_3 = \{2,8,10\} \times \{3,5\}$

Q: is **S** = *{2, 7, 8, 10}* independent?

1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	1	1	0	1

For each $\mathbf{B_i} = (\mathbf{L_i}, \mathbf{R_i})$
Check if $\mathbf{L_i}$ *intersects* **S**
Check if $\mathbf{R_i}$ *intersects* **S**
If both: return ***False***
return ***True***

Biclique Covers



For each $\boldsymbol{B_i = (L_i, R_i)}$

Check if $\boldsymbol{L_i}$ *intersects* $\boldsymbol{S}$

Check if $\boldsymbol{R_i}$ *intersects* $\boldsymbol{S}$

If both: return ***False***

return ***True***

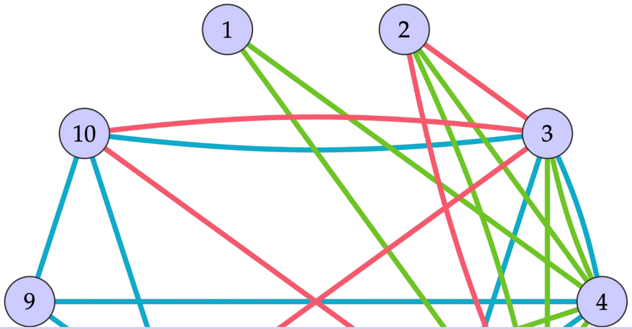
$$O\left(\sum_i |L_i| + |R_i|\right) = O\left(\sum_i |V(B_i)|\right) = O(\text{"biclique cover weight"})$$

$B_1 = \{3,7,9\} \times \{4,6,10\}$

$B_2 = \{1,2,3,6,8\} \times \{4,5\}$

$B_3 = \{2,8,10\} \times \{3,5\}$

Biclique Covers



For each $B_i = (L_i, R_i)$

Check if L_i *intersects* S

Check if R_i *intersects* S

If both: return ***False***

return ***True***

$$O\left(\sum_i |L_i| + |R_i|\right) = O\left(\sum_i |V(B_i)|\right) = O(\text{“biclique cover weight”})$$

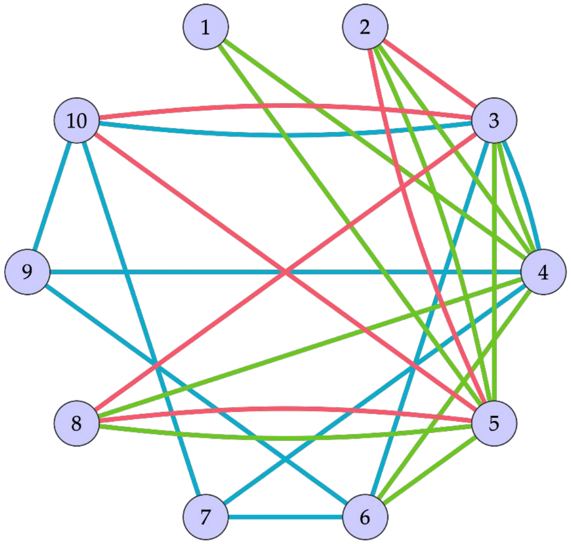
$B_1 = \{3,7,9\} \times \{4,6,10\}$

$B_2 = \{1,2,3,6,8\} \times \{4,5\}$

$B_3 = \{2,8,10\} \times \{3,5\}$

$$|V(B_1)| + |V(B_2)| + |V(B_3)| = 6 + 7 + 5 = 18 < |E| = 21$$

Biclique Cover

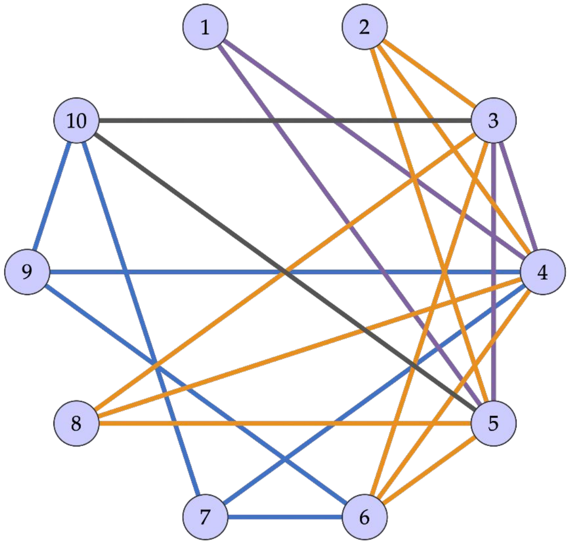


$B_1 = \{3,7,9\} \times \{4,6,10\}$

$B_2 = \{1,2,3,6,8\} \times \{4,5\}$

$B_3 = \{2,8,10\} \times \{3,5\}$

Biclique Partition



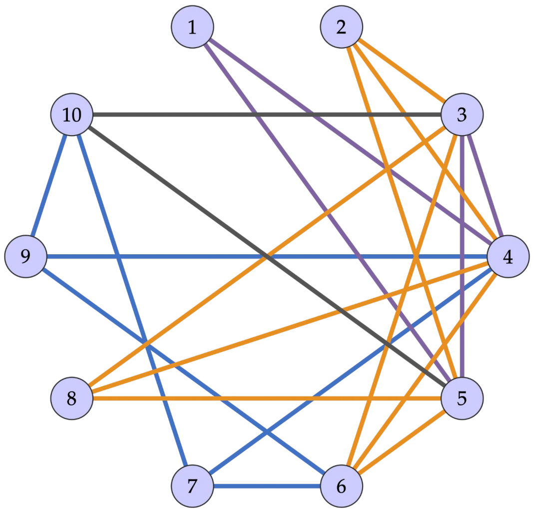
$B_1 = \{4,6,10\} \times \{7,9\}$

$B_2 = \{1,3\} \times \{4,5\}$

$B_3 = \{2,6,8\} \times \{3,4,5\}$

$B_4 = \{3,5\} \times \{10\}$

Biclique Partition



— $B_1 = \{4,6,10\} \times \{7,9\}$

— $B_2 = \{1,3\} \times \{4,5\}$

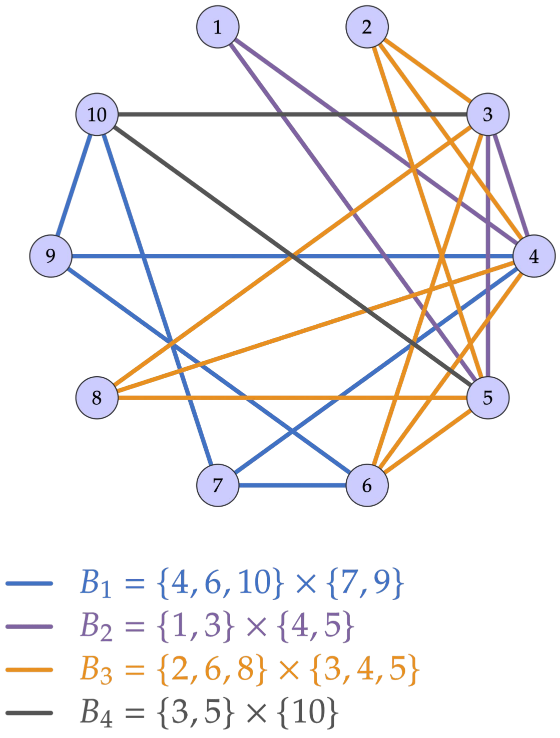
— $B_3 = \{2,6,8\} \times \{3,4,5\}$

— $B_4 = \{3,5\} \times \{10\}$

Cut queries

Q: how many edges between
A = {1, 4, 5} and **B** = {2, 7, 8}?

Biclique Partition



Cut queries

Q: how many edges between **A** = {1, 4, 5} and **B** = {2, 7, 8}?

Ans = 0
For each $\boldsymbol{B_i} = (\boldsymbol{L_i}, \boldsymbol{R_i})$
Compute:
1) $|\boldsymbol{L_i} \cap \boldsymbol{A}|$
2) $|\boldsymbol{R_i} \cap \boldsymbol{B}|$
3) $|\boldsymbol{R_i} \cap \boldsymbol{A}|$
4) $|\boldsymbol{L_i} \cap \boldsymbol{B}|$

*Ans += 1)*2) + 3)*4)*
return Ans

Matrix-Vector Multiplication in Sub-Quadratic Time (Some Preprocessing Required)

Ryan Williams*

Abstract

We show that any $n \times n$ matrix A over any finite semiring can be preprocessed in $O(n^{2+\varepsilon})$ time, such that all subsequent vector multiplications with A can be performed in $O(n^2/(\varepsilon \log n)^2)$ time, for all $\varepsilon > 0$. The approach is combinatorial and can be implemented on a pointer machine or a $(\log n)$-word RAM. Some applications are described.

Matrix-Vector Multiplication in Sub-Quadratic Time (Some Preprocessing Required)

Ryan Williams*

Abstract

We show that any $n \times n$ matrix A over any finite semiring can be preprocessed in $O(n^{2+\varepsilon})$ time, such that all subsequent vector multiplications with A can be performed in $O(n^2/(\varepsilon \log n)^2)$ time, for all $\varepsilon > 0$. The approach is combinatorial and can be implemented on a pointer machine or a $(\log n)$-word RAM. Some applications are described.

$$S \text{ is independent in } G \iff \vec{\mathbb{1}}_S^{\top} A_G \vec{\mathbb{1}}_S = 0$$

$$e_G(X, Y) = \vec{\mathbb{1}}_X^{\top} A_G \vec{\mathbb{1}}_Y$$

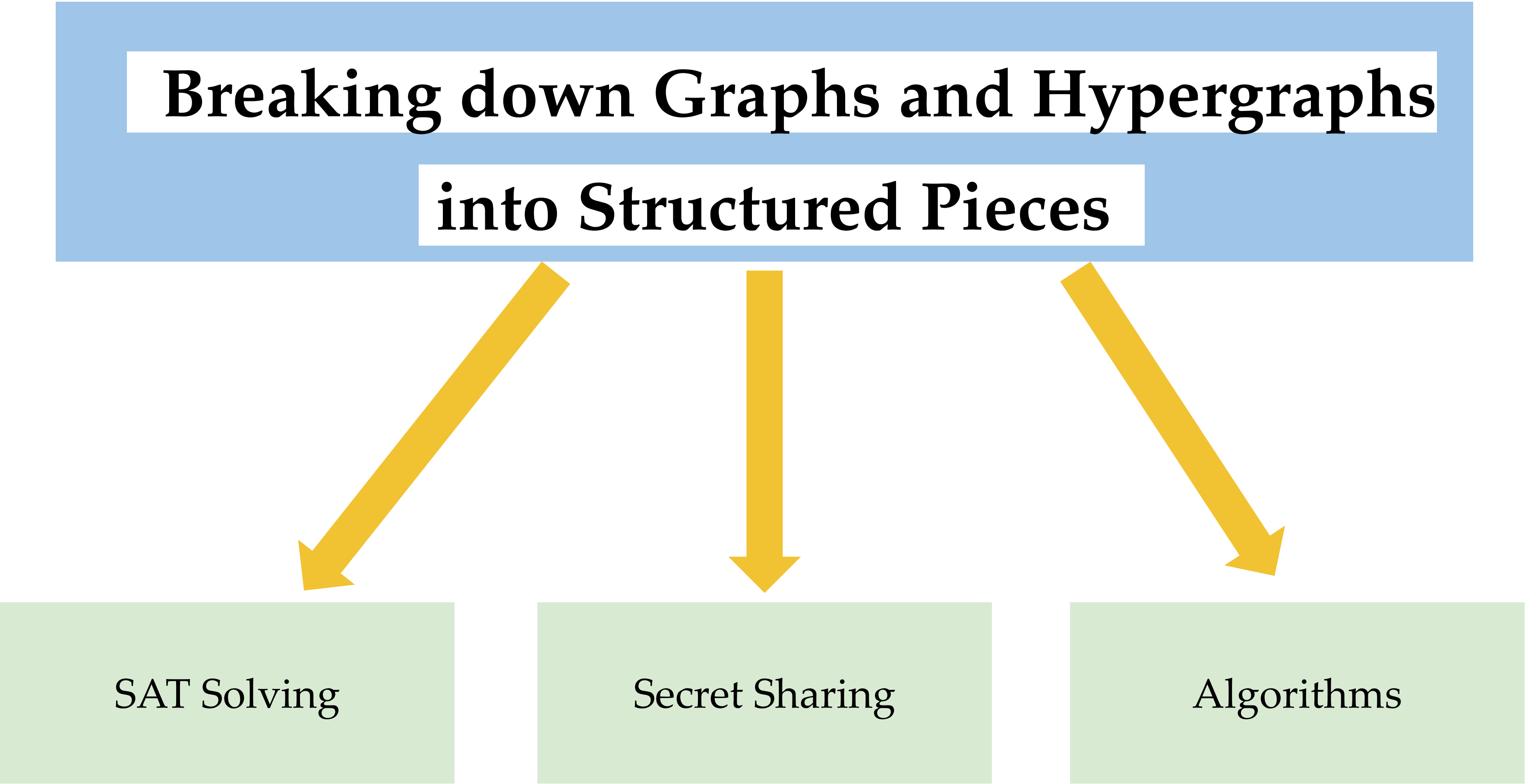
Matrix-Vector Multiplication in Sub-Quadratic Time (Some Preprocessing Required)

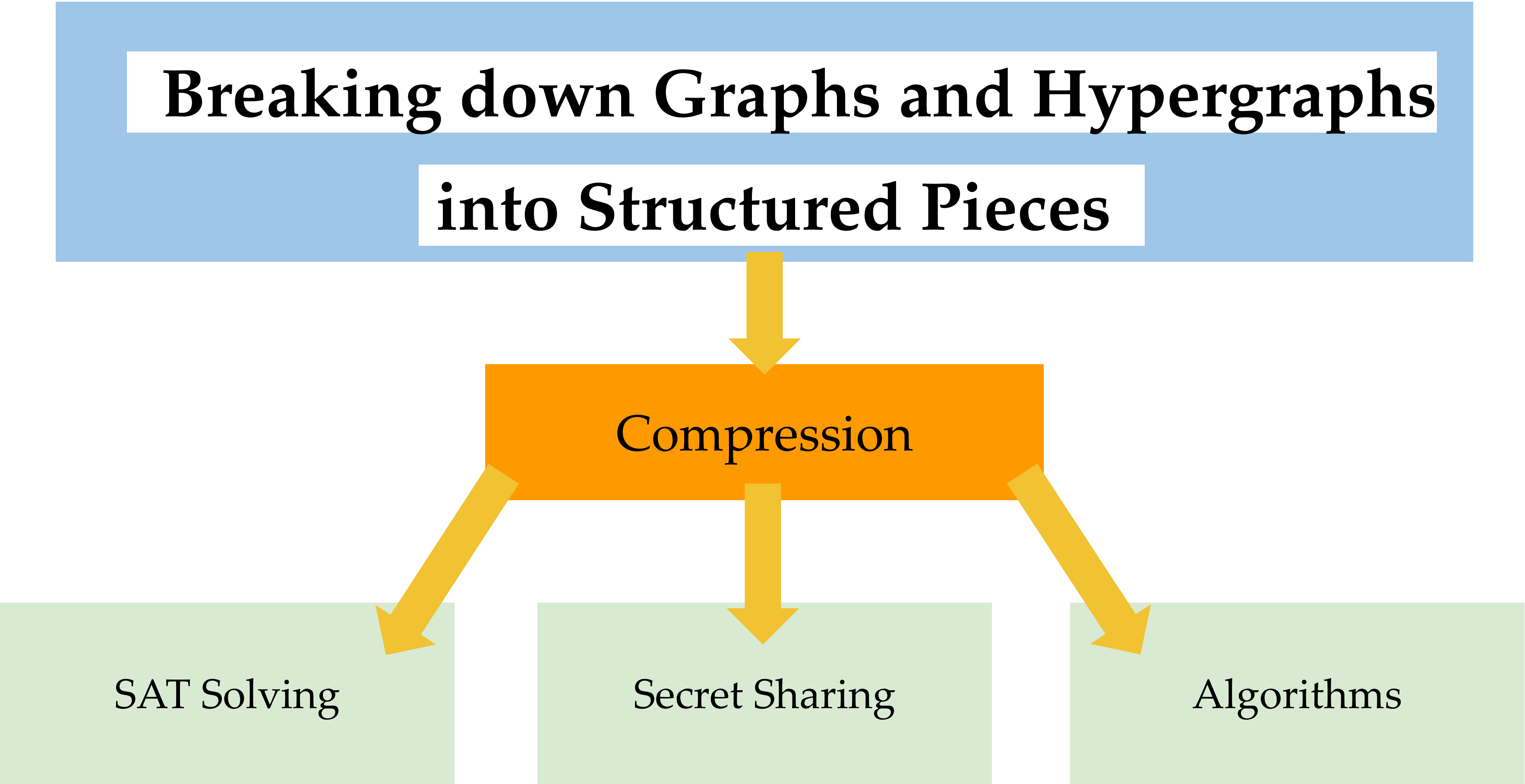
Ryan Williams*

Abstract

We show that any $n \times n$ matrix A over any finite semiring can be preprocessed in $O(n^{2+\varepsilon})$ time, such that all subsequent vector multiplications with A can be performed in $O(n^2/(\varepsilon \log n)^2)$ time, for all $\varepsilon > 0$. The approach is combinatorial and can be implemented on a pointer machine or a $(\log n)$-word RAM. Some applications are described.

Theorem. *Given an n-vertex graph G, we can compute a biclique partition $B_1, \ldots, B_k$ of weight $(\frac{1}{2} + o(1))\frac{n^2}{\lg n}$ in time $O(n^2)$. Then, we can answer independent-set queries and cut-queries in time $O(n^2/\lg n)$ each.*


Breaking down Graphs and Hypergraphs
into Structured Pieces
SAT Solving
Secret Sharing
Algorithms


Breaking down Graphs and Hypergraphs
into Structured Pieces
Compression
SAT Solving
Secret Sharing
Algorithms

My naive view of *compression*

100010101010111111111111111

0101101101010001100100

My naive view of *compression*

pattern

100010101010111111111111111

0101101101010001100100

My naive view of *compression*

pattern

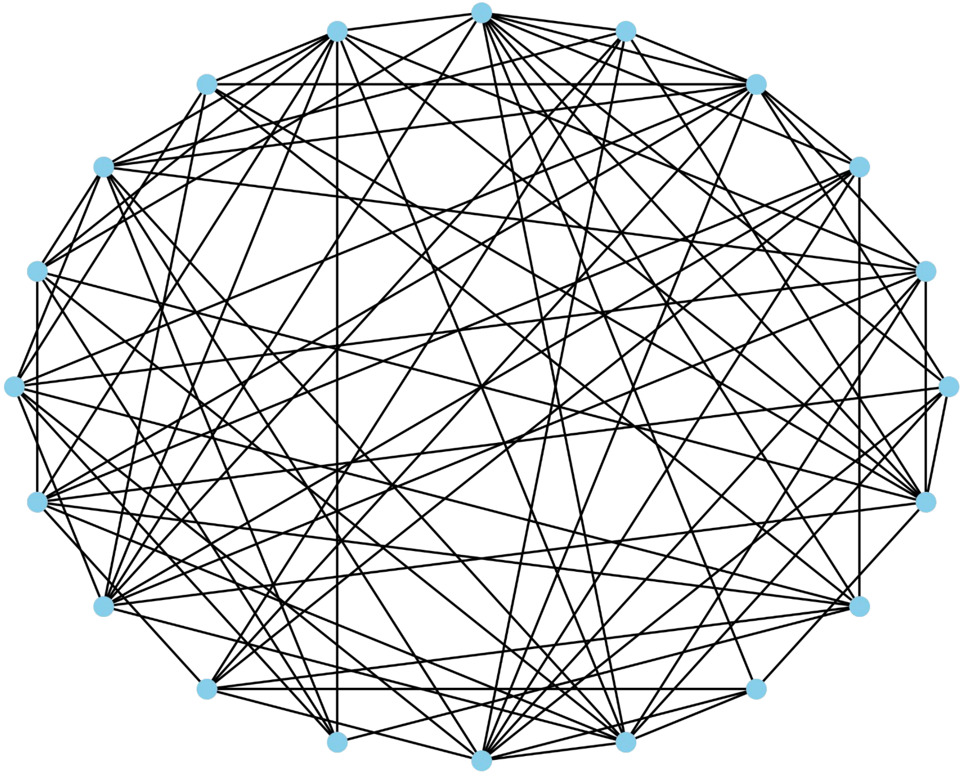
100010101010111111111111111

0101101101010001100100

compression

100010101010**{15x1}**

0101101101010001100100



Say we want to *compress* this graph

What could be a pattern?

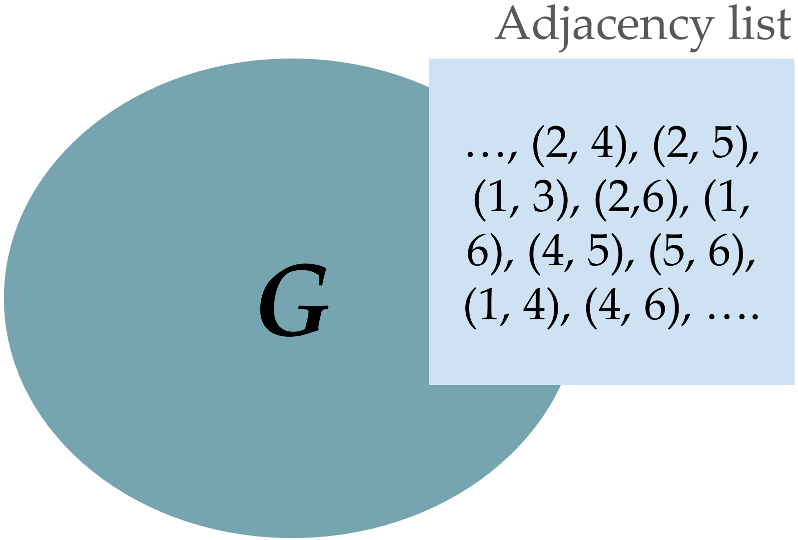
Could there be none?

Theorem (Ramsey, Erdős, Szekeres). *Every graph on n vertices contains either a clique of size $\Omega(\lg n)$ or an independent set of size $\Omega(\lg n)$.*

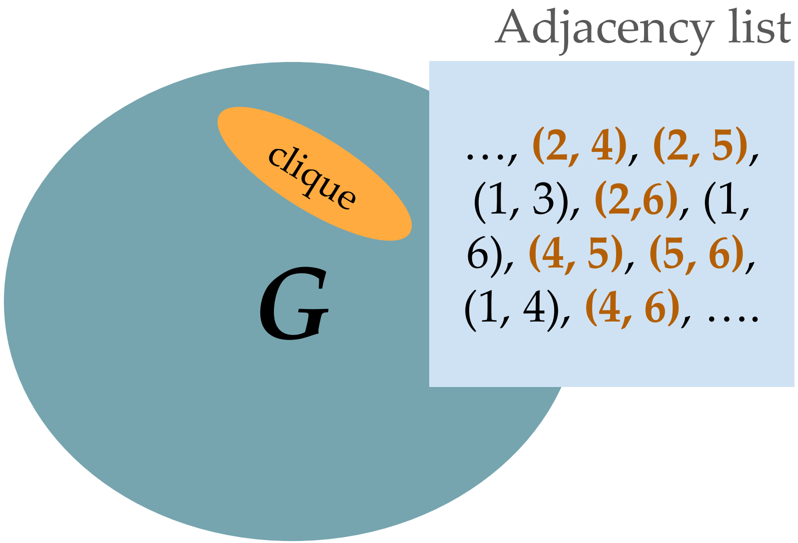
Frank Ramsey

Paul Erdős and George Szekeres

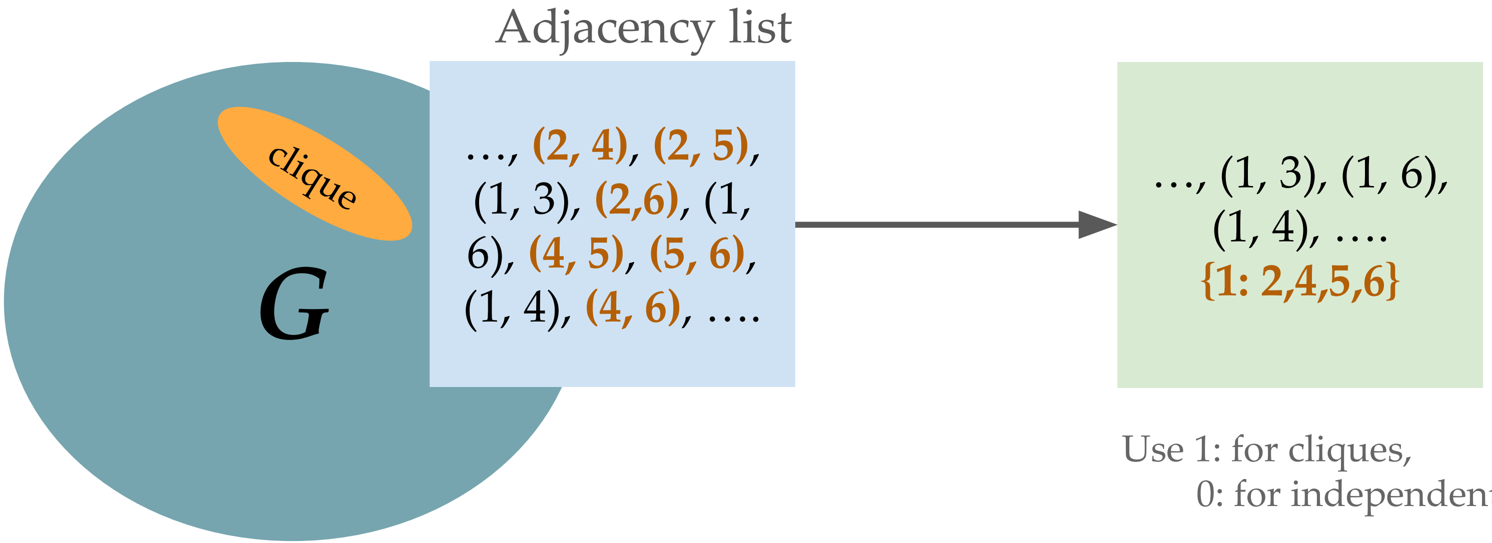
Theorem (Ramsey, Erdős, Szekeres). *Every graph on n vertices contains either a clique of size $\Omega(\lg n)$ or an independent set of size $\Omega(\lg n)$.*



Theorem (Ramsey, Erdős, Szekeres). *Every graph on n vertices contains either a clique of size $\Omega(\lg n)$ or an independent set of size $\Omega(\lg n)$.*



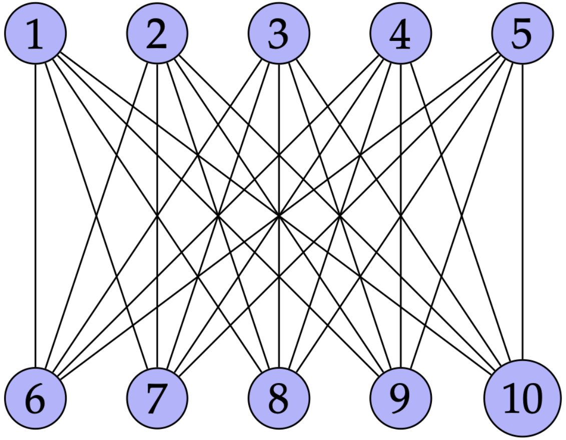
Theorem (Ramsey, Erdős, Szekeres). *Every graph on n vertices contains either a clique of size $\Omega(\lg n)$ or an independent set of size $\Omega(\lg n)$.*



Use 1: for cliques,
0: for independent sets

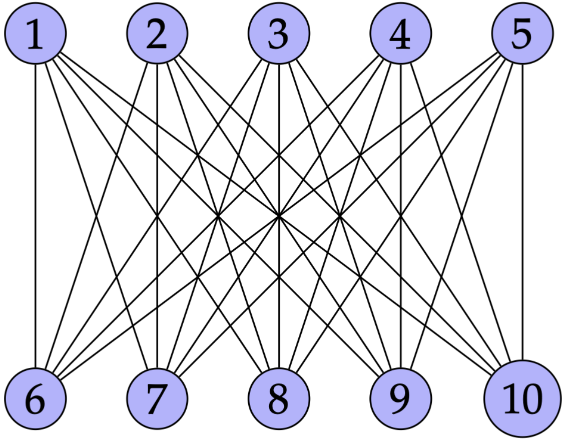
Idea: write our graph **G** as a union of sets $S_1, S_2, \ldots, S_k$ each with 1 bit marking if it's clique or independent, and $(|S_i| \lg n)$ bits for the vertices

Idea: write our graph **G** as a union of sets $S_1, S_2, \ldots, S_k$ each with 1 bit marking if it's clique or independent, and $(|S_i| \lg n)$ bits for the vertices



0: 1,2,3,4,5
0: 6,7,8,9,10
1: 1,6
1: 1,7
…
1: 5,6
…
1: 5,10

Idea: write our graph **G** as a union of sets $S_1, S_2, \ldots, S_k$ each with 1 bit marking if it's clique or independent, and $(|S_i| \lg n)$ bits for the vertices



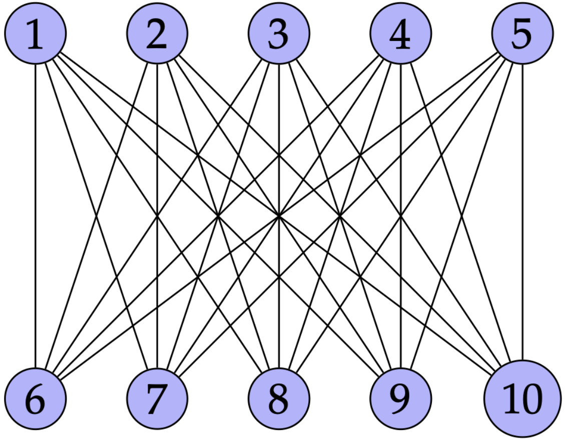
0: 1,2,3,4,5
0: 6,7,8,9,10

$2\ n \lg n$ bits

1: 1,6
1: 1,7
…
1: 5,6
…
1: 5,10

$\Omega(n^2 \lg n)$ bits

Idea: write our graph **G** as a union of sets $S_1, S_2, \ldots, S_k$ each with 1 bit marking if it's clique or independent, and $(|S_i| \lg n)$ bits for the vertices



0: 1,2,3,4,5
0: 6,7,8,9,10

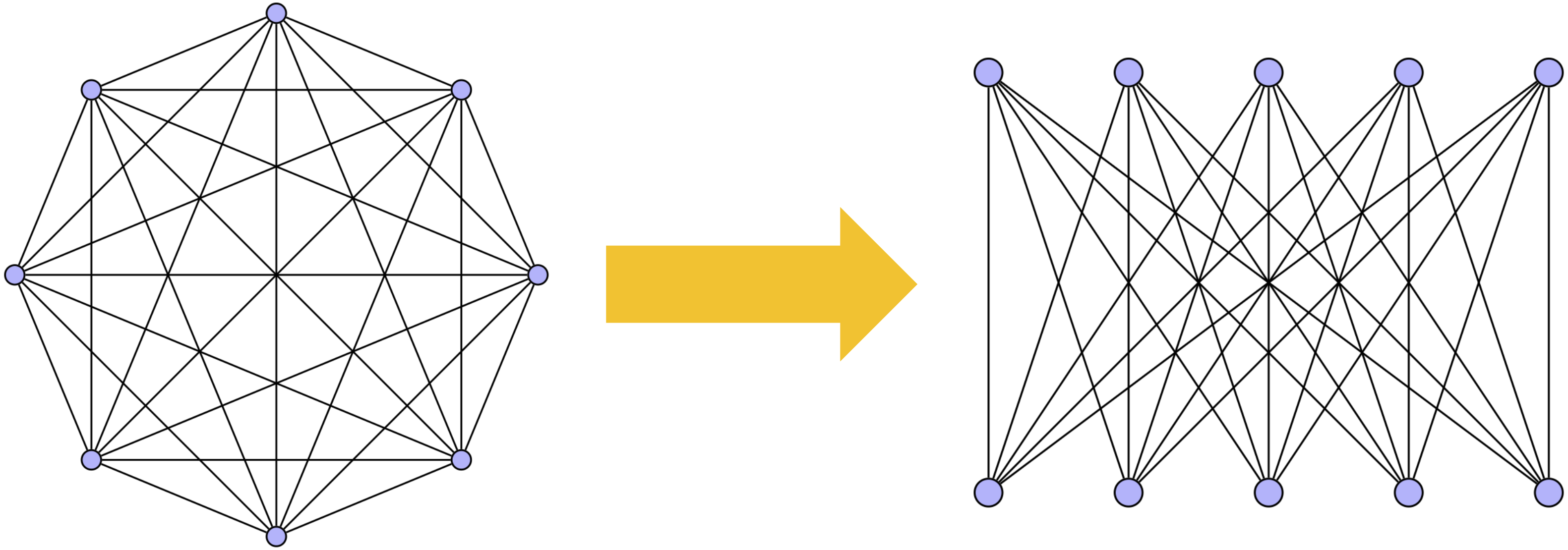
$2\ n \lg n$ bits

1: 1,6
1: 1,7
…
1: 5,6
…
1: 5,10

$\Omega(n^2 \lg n)$ bits

Adjacency matrix uses only $0.5n^2$ bits!

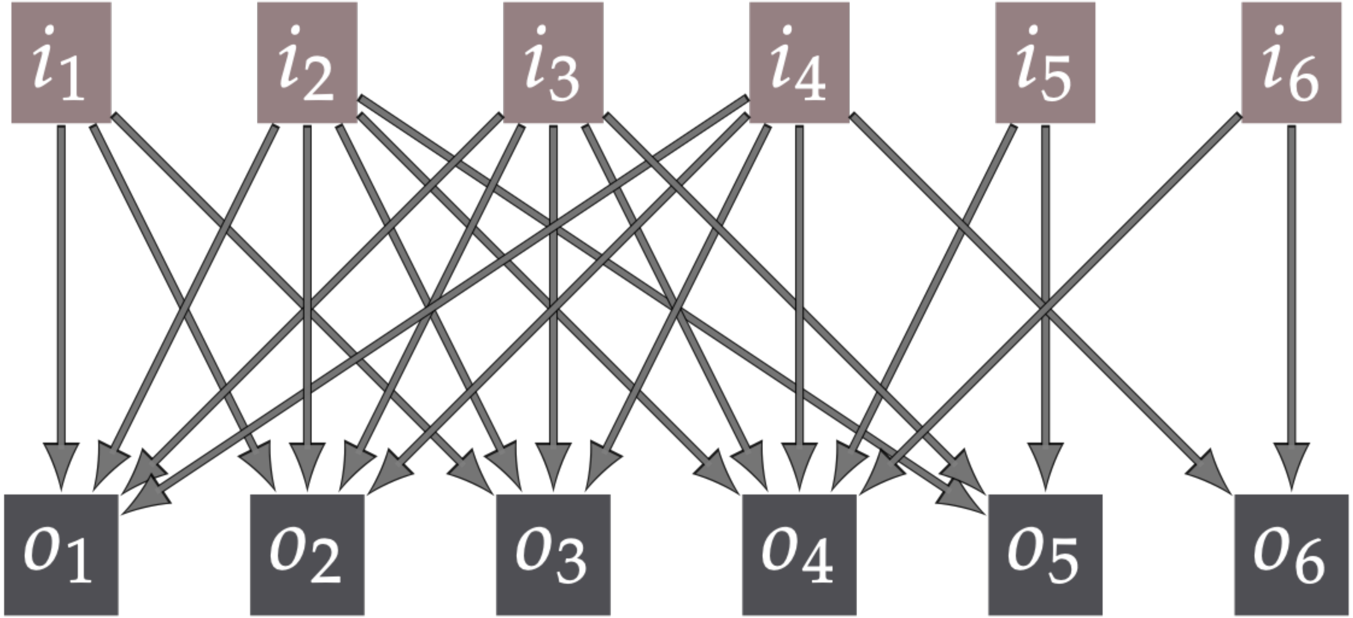
The problem is the pattern!



cliques

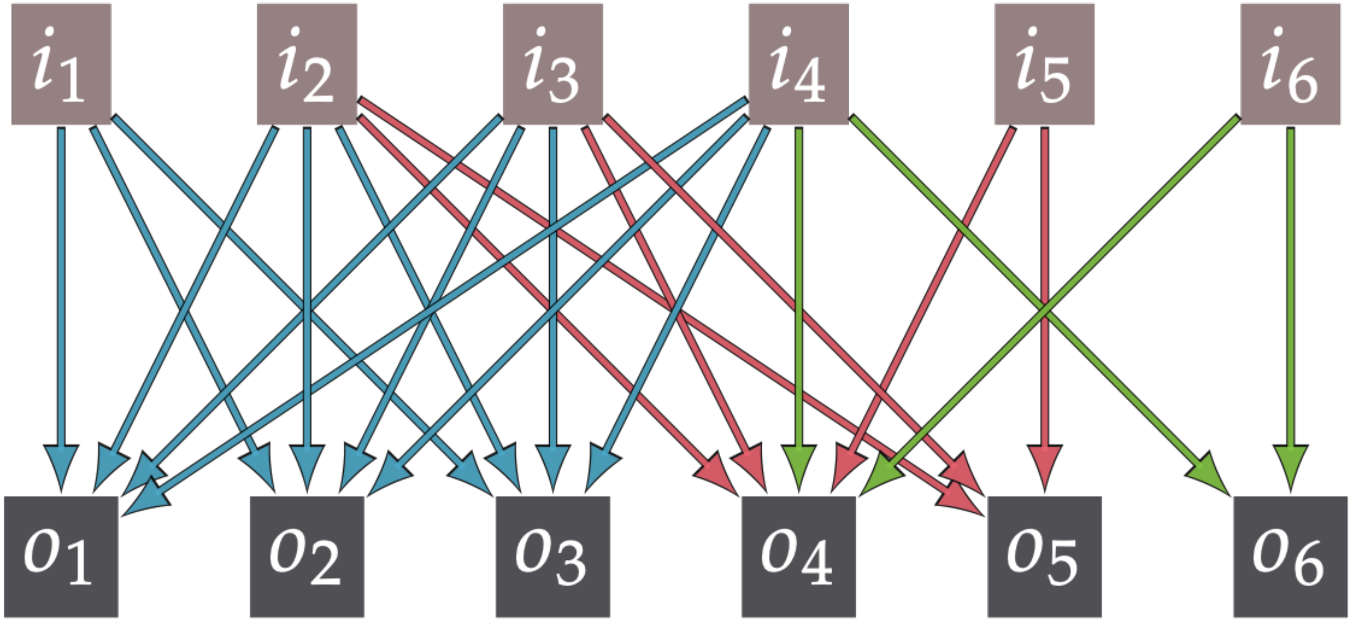
bicliques

Suppose we have a dense electric circuit



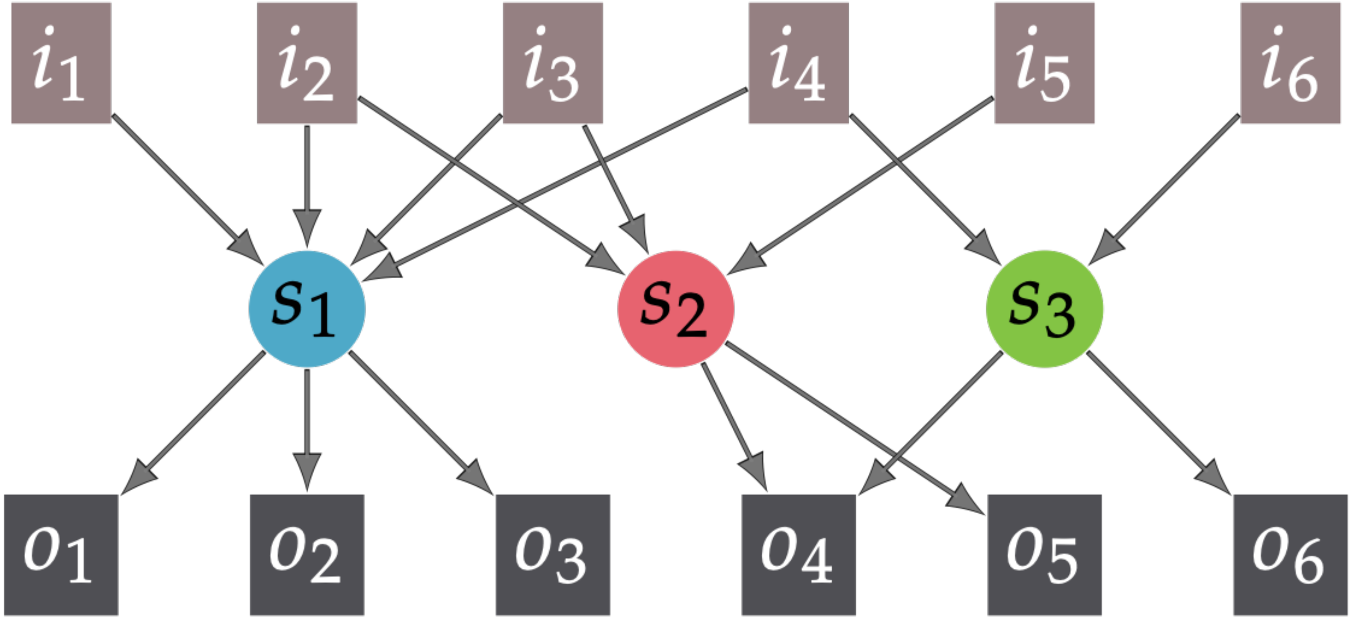
And we want to **reduce the number of wires**…

Lupanov (1956) noted that if we identify ***bicliques***



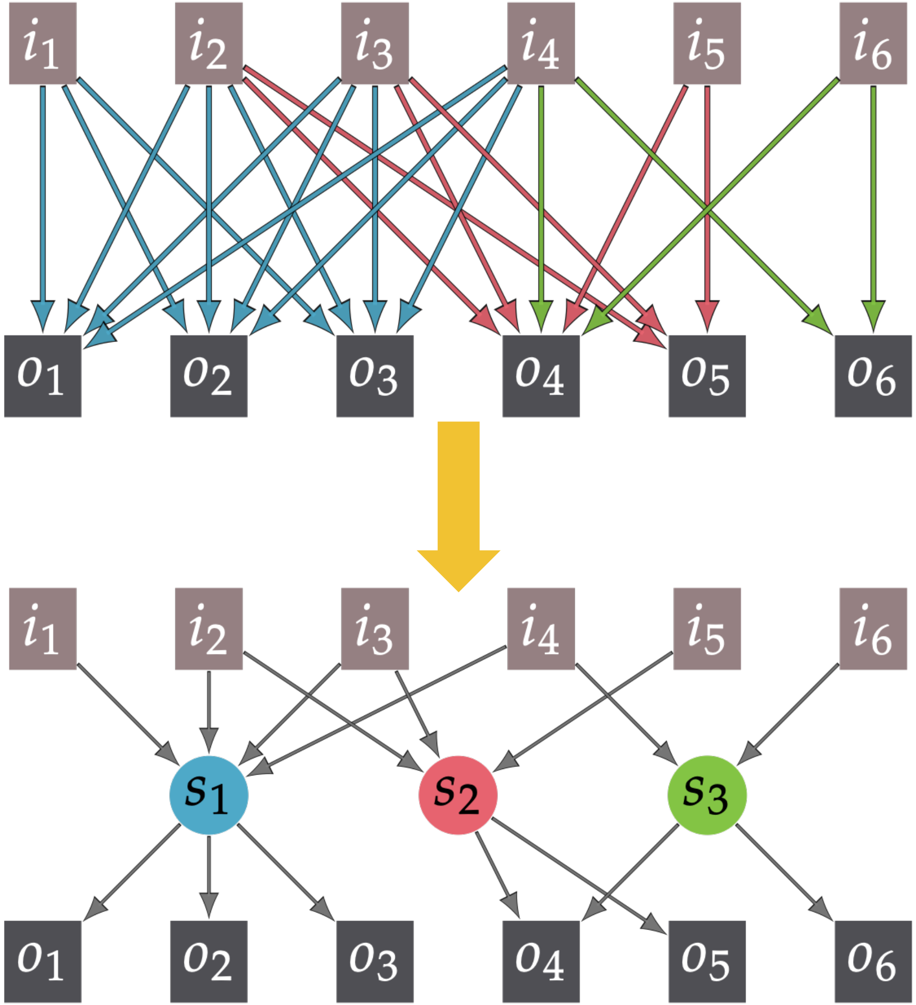
we can easily reduce each of them!

Lupanov (1956) noted that if we identify ***bicliques***

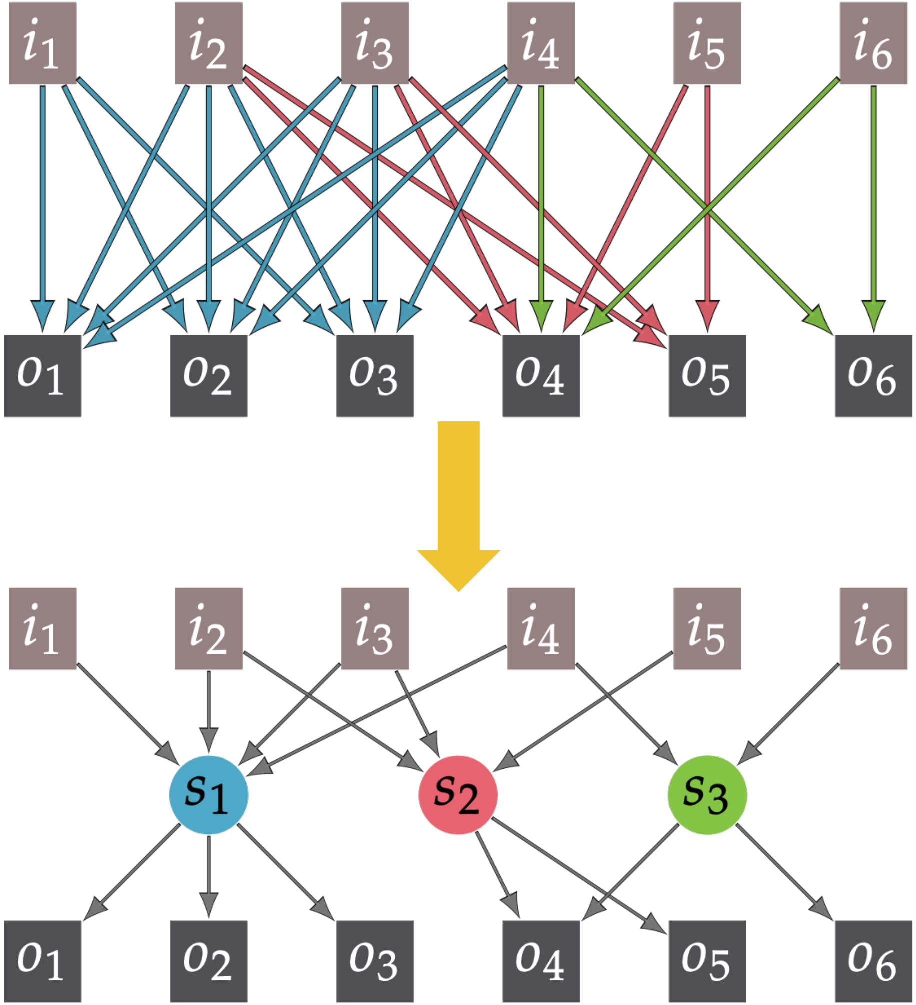


we can easily reduce each of them!

Introducing a ***switch*** we can reduce ***ab*** wires to $a + b$ if we identify a $K_{a,b}$



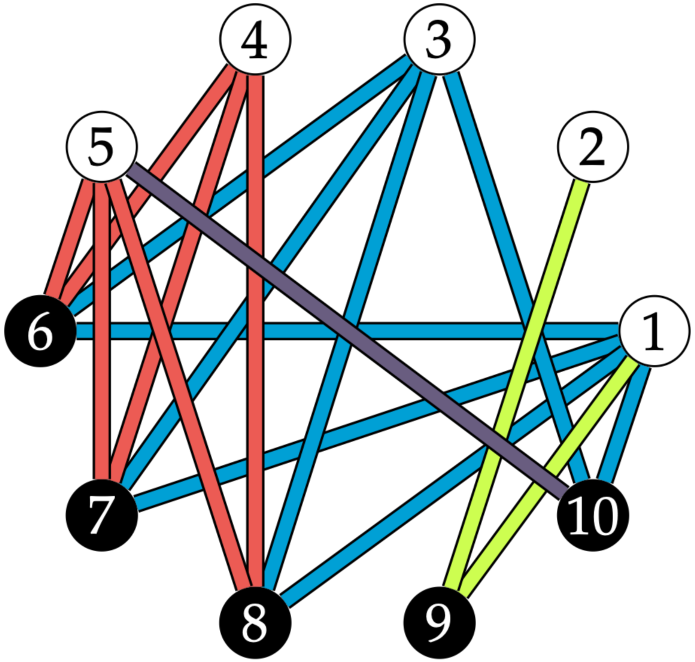
Introducing a ***switch*** we can reduce ***ab*** wires to $a + b$ if we identify a $K_{a,b}$



Reduction factor

$ab/(a+b)$

Biclique Partitions



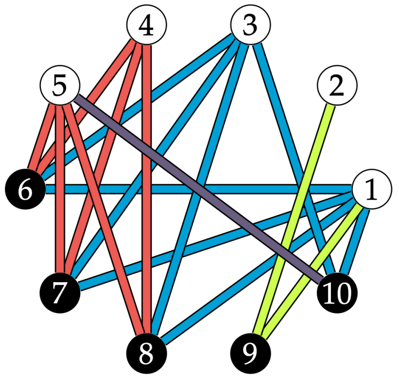
Biclique	Edges	Vertices
$K_{1,1}$	1	2
$K_{1,2}$	2	3
$K_{2,3}$	6	5
$K_{2,5}$	10	7
Total	19	**17**

Weight of the partition

Notation for Biclique Partitions

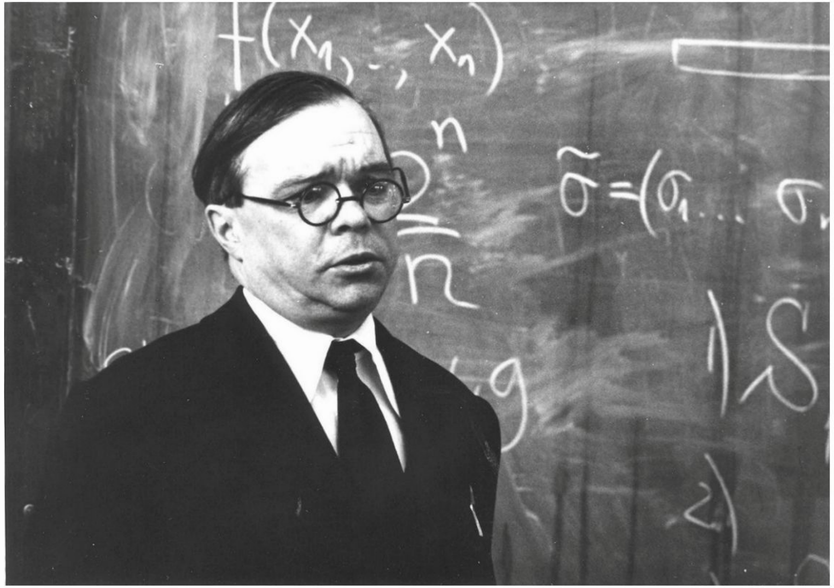
BP(G) := minimum weight of a biclique partition for G

BP(n) := maximum $BP(G)$ over G's on n vertices



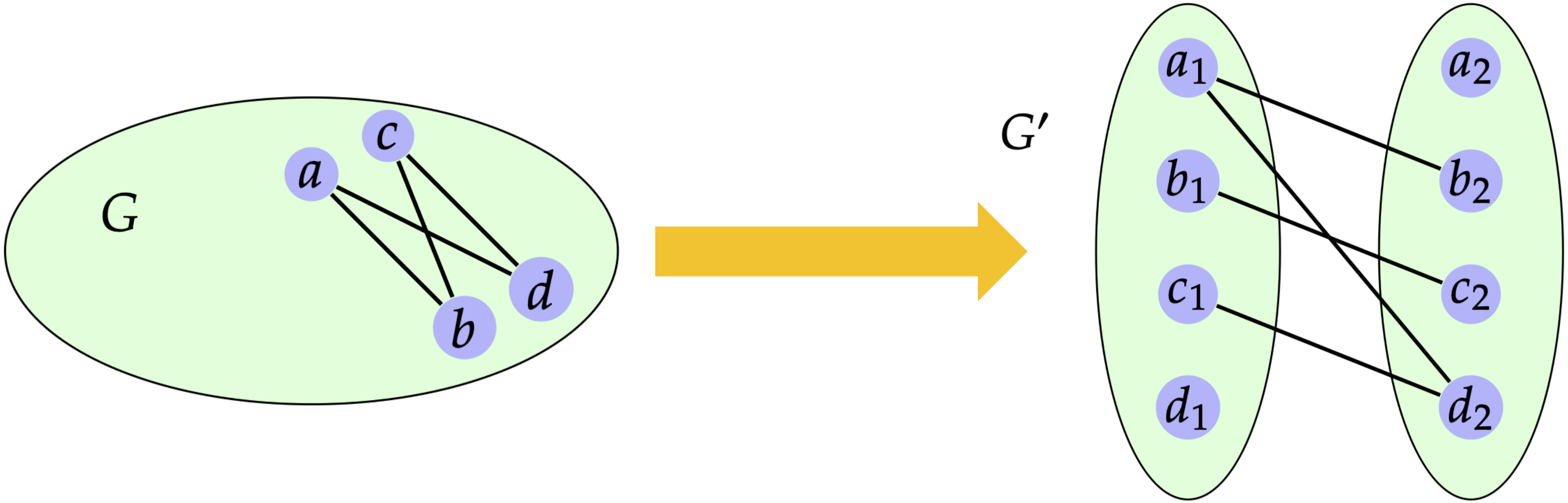
E.g., $BP(G) \leq 17$, $BP(n) \leq n^2$

Theorem (Lupanov 1956). *For every bipartite graph G with n vertices per part we have* $BP(G) \leq (1+o(1))\frac{n^2}{\lg n}$. *Moreover, there are such graphs G with* $BP(G) \geq (1-o(1))\frac{n^2}{\lg n}$.

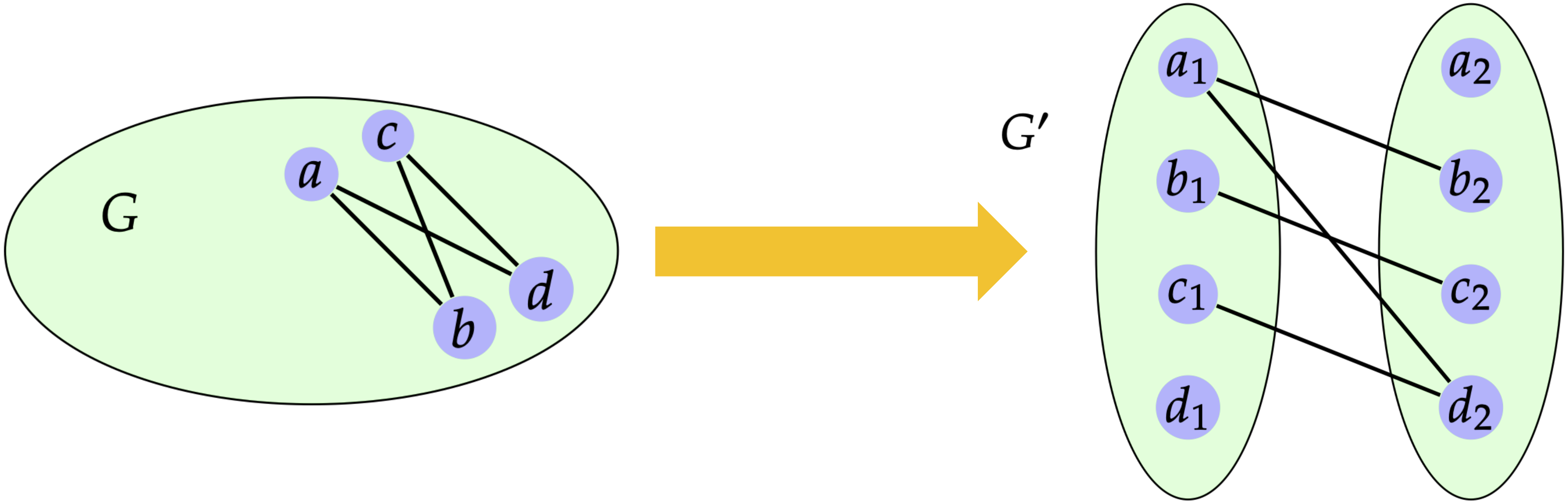


Oleg Borisovich Lupanov (1932-2006)

Lupanov for general graphs

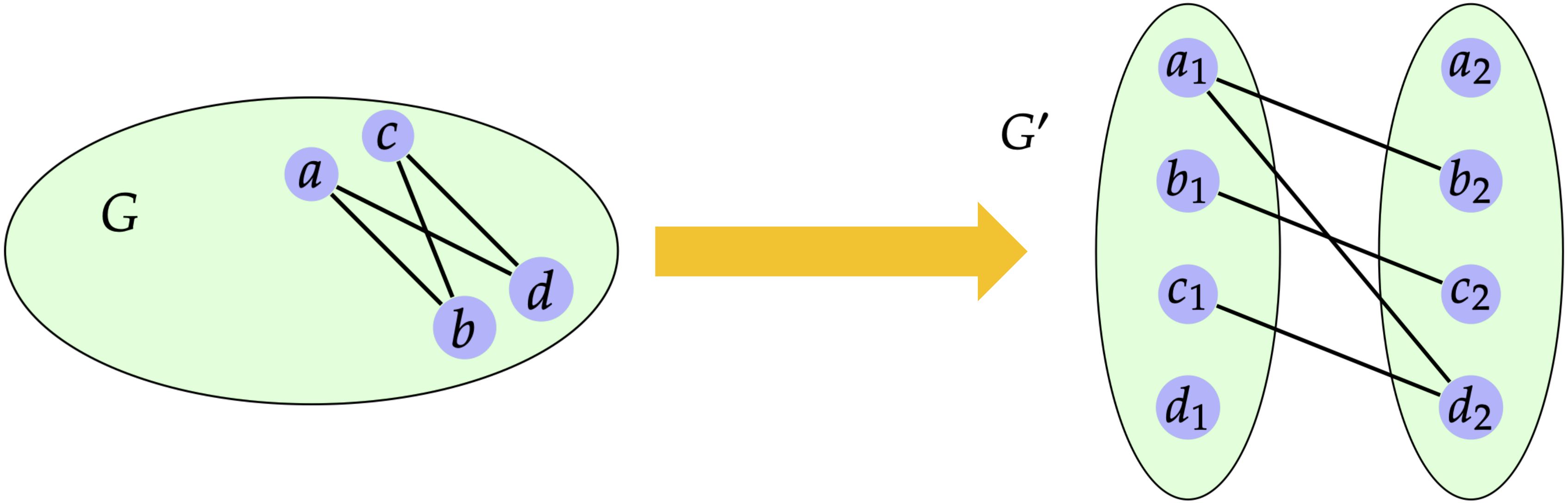


Lupanov for general graphs



Bicliques in G' correspond to bicliques in G (but not conversely)

Lupanov for general graphs



Bicliques in G' correspond to bicliques in G (but not conversely)

$$\left(\tfrac{1}{4} - o(1)\right)\frac{n^2}{\lg n} \leqslant BP(n) \leqslant (1 + o(1))\frac{n^2}{\lg n}.$$

Chung–Erdős–Spencer (1983)

$$\left(0.265 - o(1)\right)\frac{n^2}{\lg n} \leqslant BC(n) \leqslant BP(n) \leqslant \left(0.721 + o(1)\right)\frac{n^2}{\lg n}$$

Lower bound: random graph with edge probability *1/e*

$$\Pr[\text{contain } K_{a,b}] \leq \binom{n}{a}\binom{n}{b}e^{-ab} < e^{(a+b)\ln n - ab}$$

Chung–Erdős–Spencer (1983)

$$(0.265 - o(1))\frac{n^2}{\lg n} \leqslant BC(n) \leqslant BP(n) \leqslant (0.721 + o(1))\frac{n^2}{\lg n}$$

Lower bound: random graph with edge probability *1/e*

$$\Pr[\text{contain } K_{a,b}] \leq \binom{n}{a}\binom{n}{b}e^{-ab} < e^{(a+b)\ln n - ab}$$

Say a $K_{a,\,b}$ is "good" if ab/(a+b) ≥ ln *n* / (1-ε)

Chung–Erdős–Spencer (1983)

$$(0.265 - o(1))\frac{n^2}{\lg n} \leqslant BC(n) \leqslant BP(n) \leqslant (0.721 + o(1))\frac{n^2}{\lg n}$$

Lower bound: random graph with edge probability *1/e*

$$\Pr[\text{contain } K_{a,b}] \leq \binom{n}{a}\binom{n}{b}e^{-ab} < e^{(a+b)\ln n - ab}$$

Say a $K_{a,\,b}$ is "good" if ab/(a+b) ≥ ln n / (1-ε)

$$\Pr[\text{contain some good } K_{a,b}] \leq \sum_{(a,b)\text{ is good}} e^{-\varepsilon ab} \leq n^2 e^{-\varepsilon \ln^2 n} < 1$$

Chung–Erdős–Spencer (1983)

$$\left(0.265 - o(1)\right)\frac{n^2}{\lg n} \leqslant BC(n) \leqslant BP(n) \leqslant \left(0.721 + o(1)\right)\frac{n^2}{\lg n}$$

Upper bound: iteratively peel good bicliques

Chung–Erdős–Spencer (1983)

$$(0.265 - o(1))\frac{n^2}{\lg n} \leqslant BC(n) \leqslant BP(n) \leqslant (0.721 + o(1))\frac{n^2}{\lg n}$$

Upper bound: iteratively peel good bicliques

Theorem 1 (Mini KST). *Let $E(G) = \gamma\binom{n}{2}$, then G contains a $K_{s,t}$ such that the 'reduction factor' $st/(s+t)$ is $\Omega(\lg n \cdot \lg(1/\gamma))$.*

So if G has $O(n^2/\lg n)$ edges, then take each edge as a biclique.
Otherwise peel a biclique with reduction factor $\lg n$ and repeat!

Chung–Erdős–Spencer (1983)

$$(0.265 - o(1))\frac{n^2}{\lg n} \leqslant BC(n) \leqslant BP(n) \leqslant (0.721 + o(1))\frac{n^2}{\lg n}$$

In this paper, we have shown that $\alpha_{f_1}(n;\mathbf{B}) = O(n^2/\log n)$. However, we do not know the existence of

$$\lim_{n\to\infty} \frac{\alpha_{f_1}(n;\mathbf{B})}{n^2/\log n} \quad \text{or} \quad \lim_{n\to\infty} \frac{\beta_{f_1}(n;\mathbf{B})}{n^2/\log n},$$

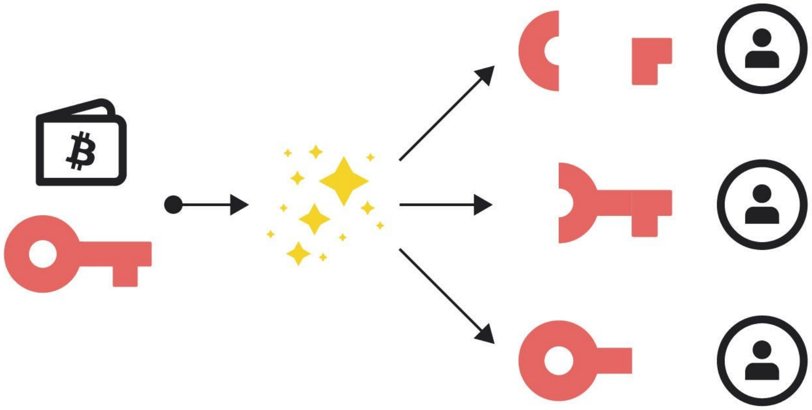
obviously.

Erdős–Pyber (1997)

Theorem (Erdős–Pyber). *Every graph G on n vertices admits a biclique partition in which every vertex v belongs to $O(n/\lg n)$ bicliques.*

Erdős–Pyber (1997)

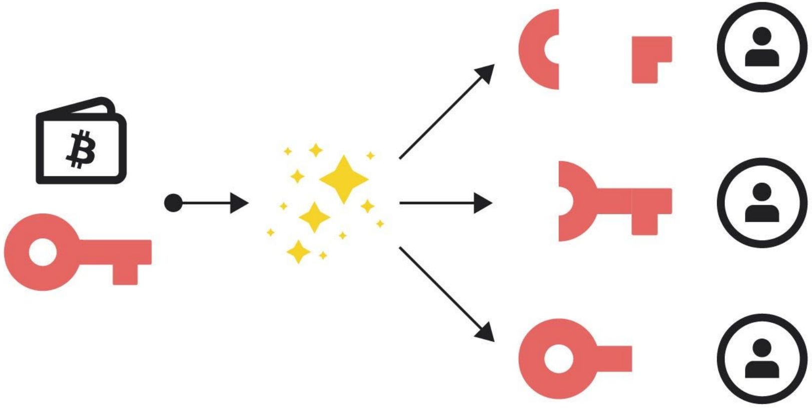
Theorem (Erdős–Pyber). *Every graph G on n vertices admits a biclique partition in which every vertex v belongs to $O(n/\lg n)$ bicliques.*

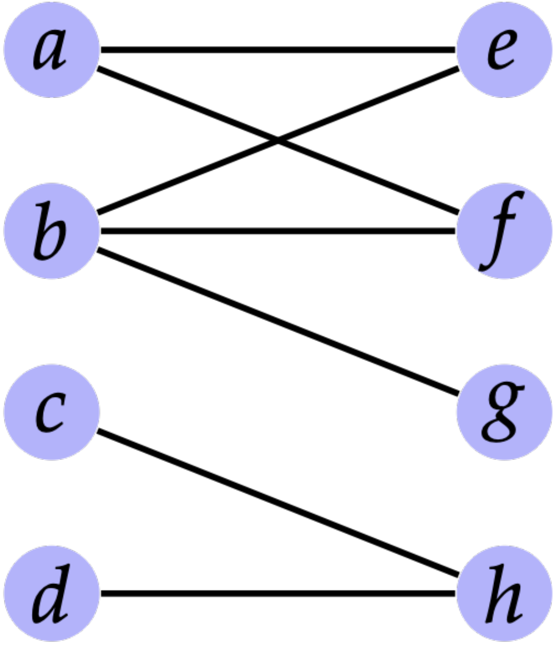


Important for secret sharing!

Erdős–Pyber (1997)

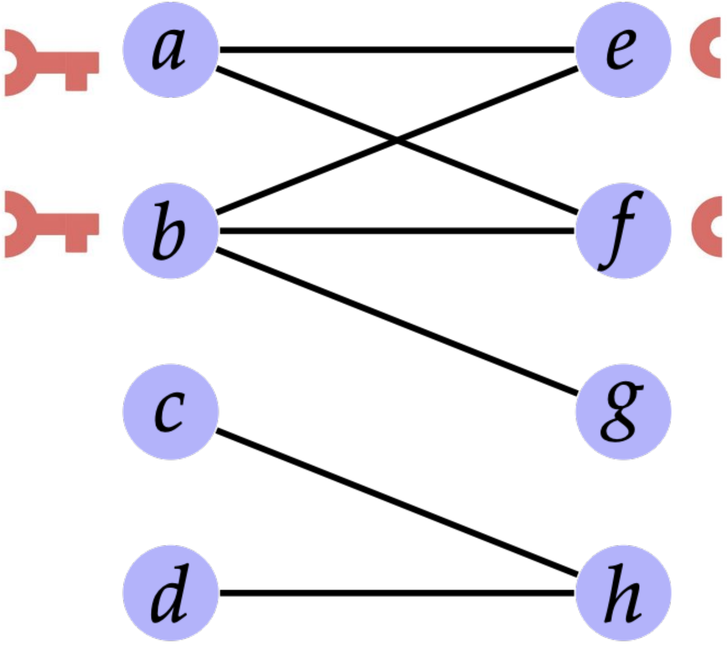
Theorem (Erdős–Pyber). *Every graph G on n vertices admits a biclique partition in which every vertex v belongs to $O(n/\lg n)$ bicliques.*





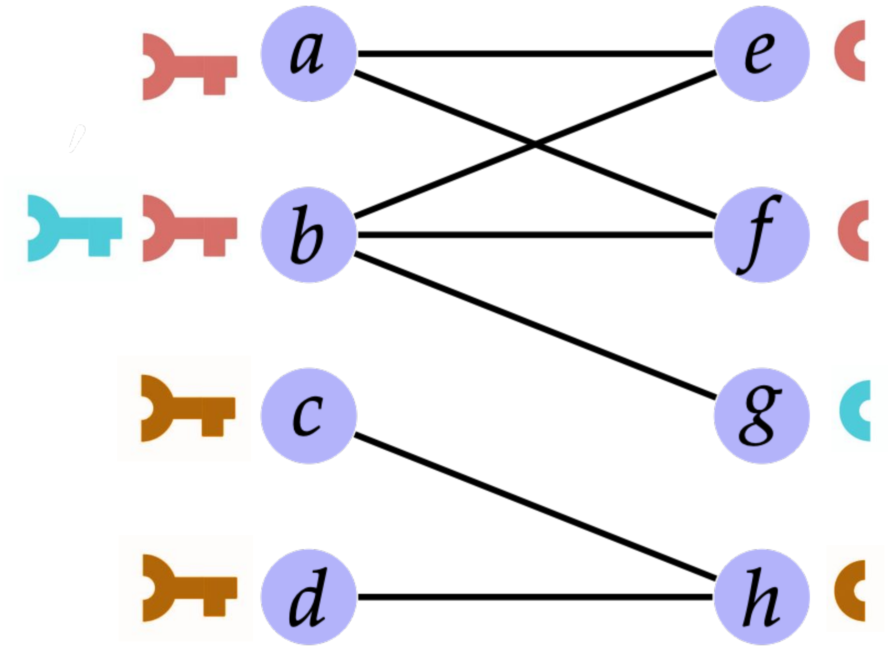
Erdős–Pyber (1997)

Theorem (Erdős–Pyber)**.** *Every graph G on n vertices admits a biclique partition in which every vertex v belongs to $O(n/\lg n)$ bicliques.*



Erdős–Pyber (1997)

Theorem (Erdős–Pyber). *Every graph G on n vertices admits a biclique partition in which every vertex v belongs to $O(n/\lg n)$ bicliques.*



The number of bicliques per vertex determines the amount of secret they have to receive!

Erdős-Pyber theorem for hypergraphs and secret sharing

László Csirmaz · Péter Ligeti · Gábor Tardos

LBP(n) is the maximum *load* per vertex on optimal partition for n-vertex graph.
Note that ***BP(n)*** $\leq$ ***n · LBP(n)***

Erdős-Pyber theorem for hypergraphs and secret sharing

László Csirmaz · Péter Ligeti · Gábor Tardos

LBP(n) is the maximum *load* per vertex on optimal partition for n-vertex graph.
Note that ***BP(n) ≤ n · LBP(n)***

Theorem (Czirmaz, Ligetti, Tardos (2014)).

$$(0.27 - o(1))\frac{n}{\lg n} \leq \mathit{LBP}(n) \leq (1 + o(1))\frac{n}{\lg n}$$

Erdős-Pyber theorem for hypergraphs and secret sharing

László Csirmaz · Péter Ligeti · Gábor Tardos

LBP(n) is the maximum *load* per vertex on optimal partition for n-vertex graph.
Note that $BP(n) \leq n \cdot LBP(n)$

Theorem (Czirmaz, Ligetti, Tardos (2014)).

$$(0.27 - o(1))\frac{n}{\lg n} \leq LBP(n) \leq (1 + o(1))\frac{n}{\lg n}$$

Theorem (Czirmaz, Ligetti, Tardos (2014)). *For d-uniform hypergraphs we have*

$$\left(\frac{0.53}{d!} + o(1)\right)\frac{n^{d-1}}{\lg n} \leq LBP_d(n) \leq \left(\frac{1}{(d-2)!} + o(1)\right)\frac{n^{d-1}}{\lg n}$$

Our main result

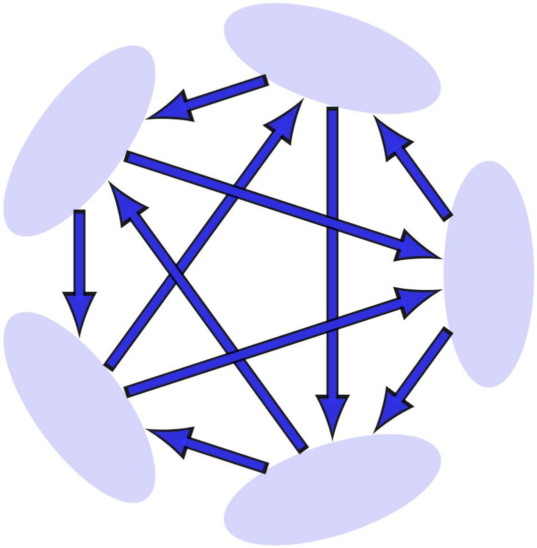
Theorem 1. *For d-uniform hypergraphs we have*

$$\mathsf{LBP}_d(n) \sim \frac{1}{d!} \cdot \frac{n^{d-1}}{\lg n}.$$

Graph Upper bound

Step 1: Partition vertices with part-size $\boxed{r := \lfloor \lg n - 2 \lg \lg n \rfloor}$

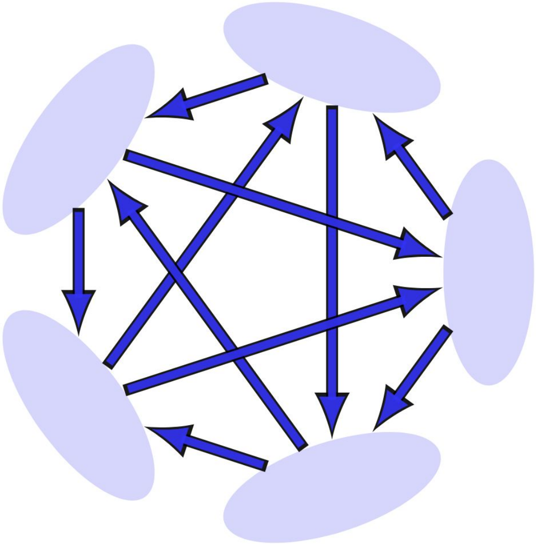
and set up an almost-regular tournament between parts



Graph Upper bound

Step 1: Partition vertices with part-size $\boxed{r := \lfloor \lg n - 2 \lg \lg n \rfloor}$

and set up an almost-regular tournament between parts

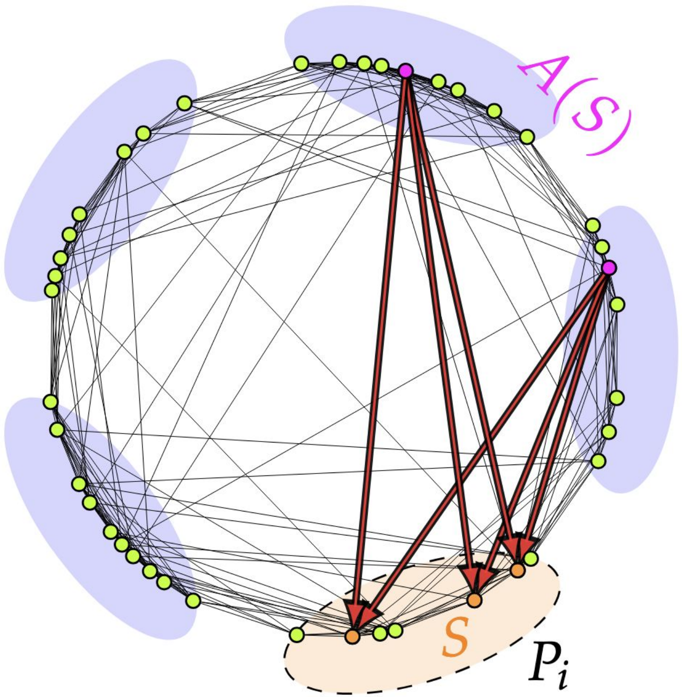


Step 2: Take each edge within a part as its own biclique.
Each vertex participates in at most r - $1 = o(n/ \lg n)$ of these.

Graph Upper bound

$$r := \lfloor \lg n - 2 \lg \lg n \rfloor$$

Step 3: For each part P_i, consider each of its $2^r \approx n / \lg^2 n$ subsets S, and emit biclique $(S, A(S))$ with $A(S)$ the vertices whose directed neighborhood in P_i is exactly S.

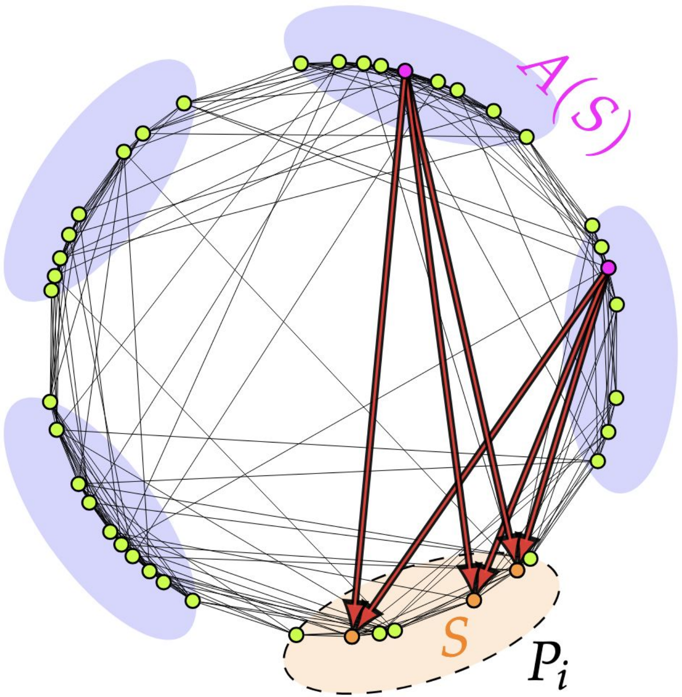


Graph Upper bound

$r := \lfloor \lg n - 2 \lg \lg n \rfloor$

Step 3: For each part P_i, consider each of its $2^r \approx n / \lg^2 n$ subsets S, and emit biclique $(S, A(S))$ with $A(S)$ the vertices whose directed neighborhood in P_i is exactly S.

Q: In how many such bicliques does each vertex v participate?



Graph Upper bound

$r := \lfloor \lg n - 2 \lg \lg n \rfloor$

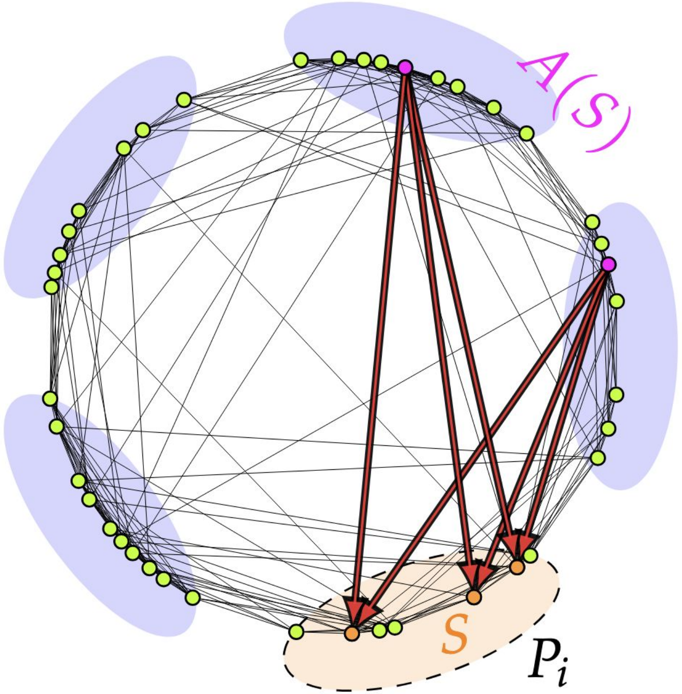
Q: In how many such bicliques does each vertex v participate?

Answer:

On the S-side, to at most $O(n / \lg^2 n)$

On the $A(S)$-side, by almost regularity to at most

$$\left\lceil \frac{n}{2r} \right\rceil \sim \frac{n}{2 \lg n}$$



We have shown:

$$\mathsf{LBP}(n) \le \left(\frac{1}{2} + o(1)\right)\frac{n}{\lg n}$$

and thus:

$$\mathsf{BP}(n) \le \left(\frac{1}{2} + o(1)\right)\frac{n^2}{\lg n}$$

Graph Upper bound

Efficient $O(n^2)$ implementation!

Algorithm 1 Biclique Partition

Require: A graph $G = (V, E)$ with $V = [n]$.

$r \leftarrow \lfloor \lg n - 2 \lg \lg n \rfloor$

function $R(i, j)$ ▷ $R \colon [\lceil n/r \rceil]^2 \to$ *Bool*

  return $(i < j \text{ and } d_n(i,j) \equiv 0 \pmod 2)$ or $(i > j \text{ and } d_n(i,j) \equiv 1 \pmod 2)$

for all $i \in [\lceil n/r \rceil]$ **do**

  $P_i \leftarrow \{(i-1) \cdot r + 1, \ldots, i \cdot r\}$

  $A \leftarrow \varnothing$ ▷ $A \colon \mathcal{P}(P_i) \to \mathcal{P}(V)$, *implemented as an array of linked lists of size* $2^{|P_i|} = O\left(\frac{n}{\lg^2 n}\right)$

  for all $v \in V$ **do**

    if $R(\lceil v/r \rceil, i)$ **then**

      $S \leftarrow \varnothing$ ▷ $S \colon \mathcal{P}(P_i)$

      for all $u \in P_i$ **do**

        if $\{u, v\} \in E$ **then**

          $S \leftarrow S \cup \{u\}$

      $A(S) \leftarrow A(S) \cup \{v\}$ ▷ *We think of S as a binary string that indexes A*

  for all $S \in \mathcal{P}(P_i)$ **do**

    if $|S| > 0$ and $|A(S)| > 0$ **then**

      Emit $(S, A(S))$

  for all $\{u, v\} \in \binom{P_i}{2}$ **do**

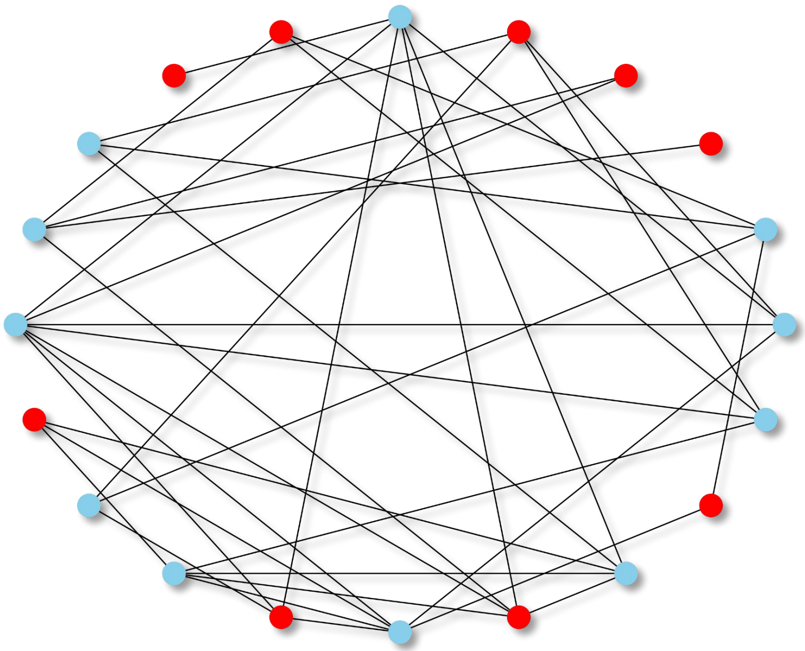
    if $\{u, v\} \in E$ **then**

      Emit $(\{u\}, \{v\})$

Ok but how does this help SAT solving?

Independent Set

How many people here can we select such that no two of them have co-authored a paper?



Independent Set

- Given a graph $G = (V, E)$, can we select k vertices with no edges between them?

Direct encoding

Independence

$$(\overline{x_u} \vee \overline{x_v}), \forall \{u, v\} \in E$$

Cardinality

$$\sum_{v \in V} x_v = k$$

Independent Set

Direct encoding

Independence

$$(\overline{x_u} \vee \overline{x_v}), \forall \{u, v\} \in E$$

Cardinality

$$\sum_{v \in V} x_v = k$$

Can be encoded in $O(|V|)$ clauses *[Sinz05, Warner98]* or $O(|V| \log |V|)$ with propagation *[Asín et al.11]*.

Independent Set

Direct encoding

Independence

$$(\overline{x_u} \vee \overline{x_v}), \forall\{u, v\} \in E$$

Cardinality

$$\sum_{v \in V} x_v = k$$

Bottleneck is here!
$\Theta(|E|)$ clauses, so $\Omega(|V|^2)$ for dense graphs

And it's not only independent sets!

Vertex cover

$(x_u \vee x_v), \forall \{u, v\} \in E$

Clique

$(\overline{x_u} \vee \overline{x_v}), \forall \{u, v\} \notin E$

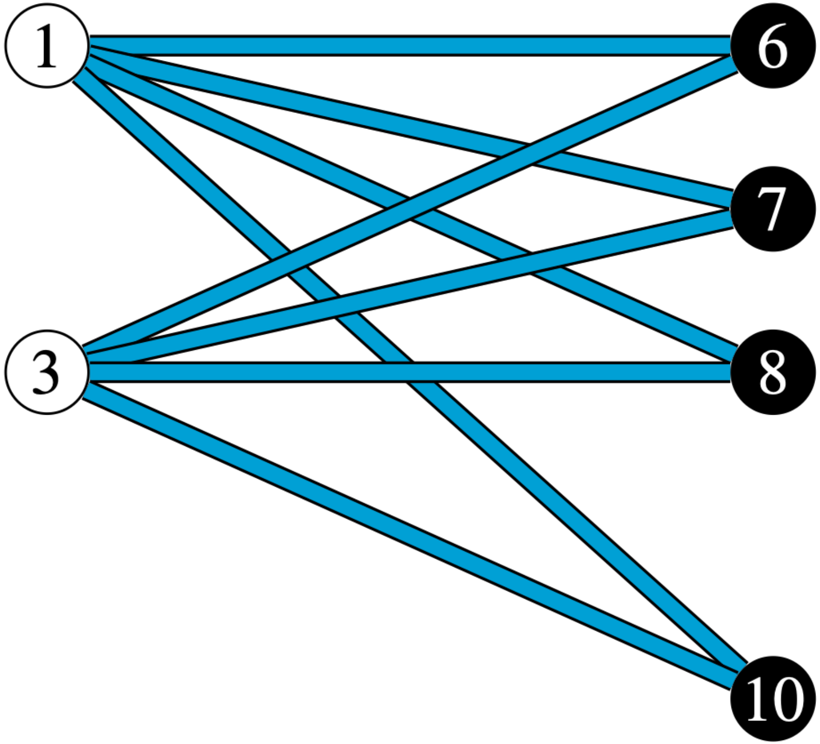
Graph coloring

$(\overline{x_{u,\mathrm{red}}} \vee \overline{x_{v,\mathrm{red}}}), \quad \forall \{u, v\} \in E$

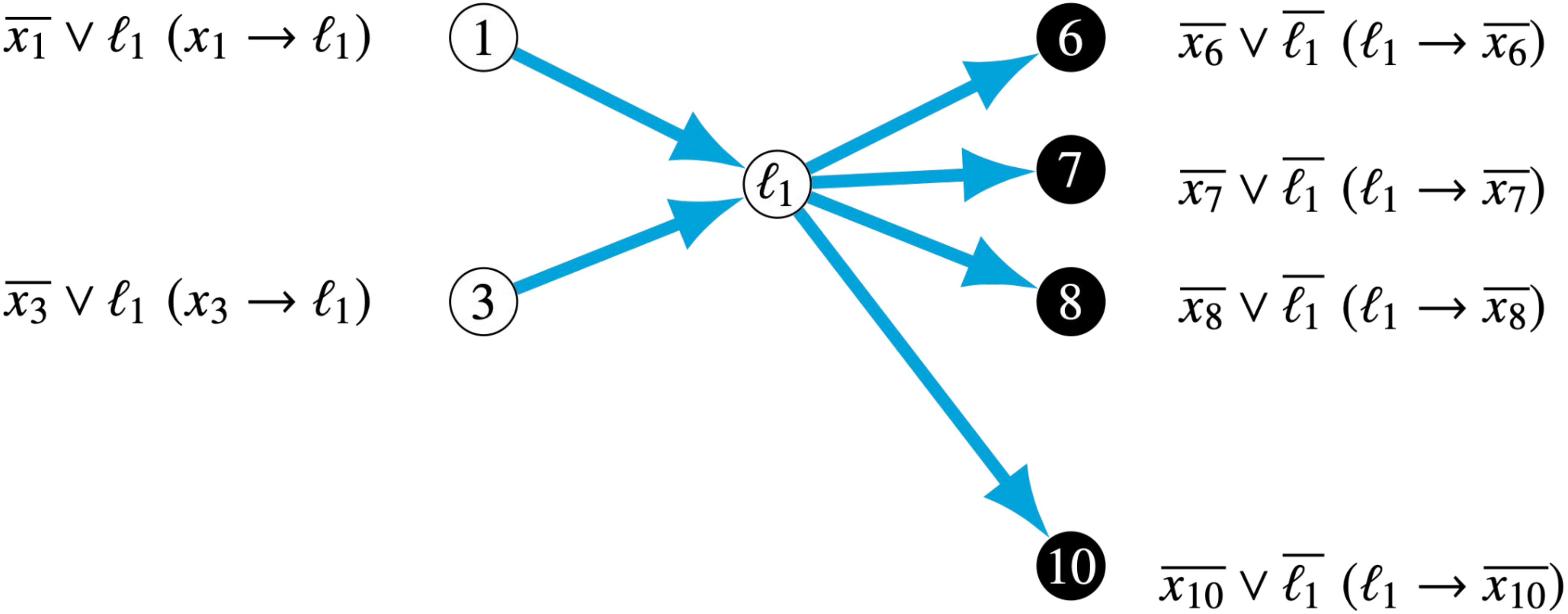
Suppose your graph is a biclique

$$\overline{x_1} \vee \overline{x_6}$$
$$\overline{x_1} \vee \overline{x_7}$$
$$\overline{x_1} \vee \overline{x_8}$$
$$\overline{x_1} \vee \overline{x_{10}}$$

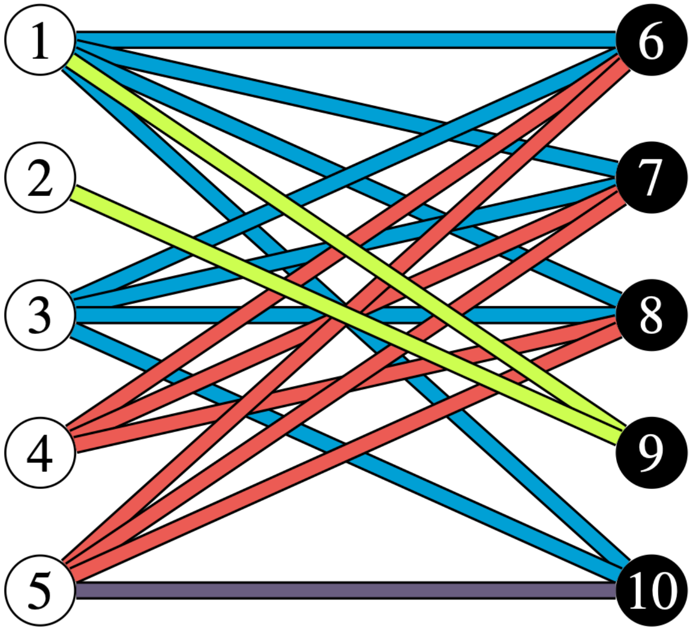
$$\overline{x_3} \vee \overline{x_6}$$
$$\overline{x_3} \vee \overline{x_7}$$
$$\overline{x_3} \vee \overline{x_8}$$
$$\overline{x_3} \vee \overline{x_{10}}$$



Suppose your graph is a biclique



Clauses ≤ BP(G)



$\overline{x_1} \vee \ell_1$ $\quad$ $\overline{x_3} \vee \ell_1$ $\quad$ $\overline{x_6} \vee \overline{\ell_1}$

$\overline{x_7} \vee \overline{\ell_1}$ $\quad$ $\overline{x_8} \vee \overline{\ell_1}$ $\quad$ $\overline{x_{10}} \vee \overline{\ell_1}$

$\overline{x_4} \vee \ell_2$ $\quad$ $\overline{x_6} \vee \overline{\ell_2}$ $\quad$ $\overline{x_8} \vee \overline{\ell_2}$

$\overline{x_5} \vee \ell_2$ $\quad$ $\overline{x_7} \vee \overline{\ell_2}$

$\overline{x_1} \vee \ell_3$ $\quad$ $\overline{x_2} \vee \ell_3$ $\quad$ $\overline{x_9} \vee \overline{\ell_3}$

$\overline{x_5} \vee \ell_4$ $\quad$ $\overline{x_{10}} \vee \overline{\ell_4}$ $\quad$ $\overline{x_5} \vee \overline{x_{10}}$

Information theory lower bound

There are $2^{\binom{n}{2}}$ labeled graphs on n vertices.
So any binary encoding for them
must use $\lg(2^{\binom{n}{2}}) \sim \frac{n^2}{2}$ bits on some graph.

Information theory lower bound

There are $2^{\binom{n}{2}}$ labeled graphs on n vertices.
So any binary encoding for them
must use $\lg(2^{\binom{n}{2}}) \sim \frac{n^2}{2}$ bits on some graph.

But a biclique partition of a graph fully identifies the graph, and as each vertex can be encoded with lg n bits, we can encode every G with BP(G)*(lg n + 1) bits

Information theory lower bound

There are $2^{\binom{n}{2}}$ labeled graphs on n vertices.
So any binary encoding for them
must use $\lg(2^{\binom{n}{2}}) \sim \frac{n^2}{2}$ bits on some graph.

But a biclique partition of a graph fully identifies the graph, and as each vertex can be encoded with lg n bits, we can encode every G with BP(G)*(lg n + 1) bits

So for some graph G, $\mathrm{BP}(G) \geq \left(\frac{1}{2} - o(1)\right) \frac{n^2}{\lg n}$

Optimal CES for ***d*-Hypergraphs**

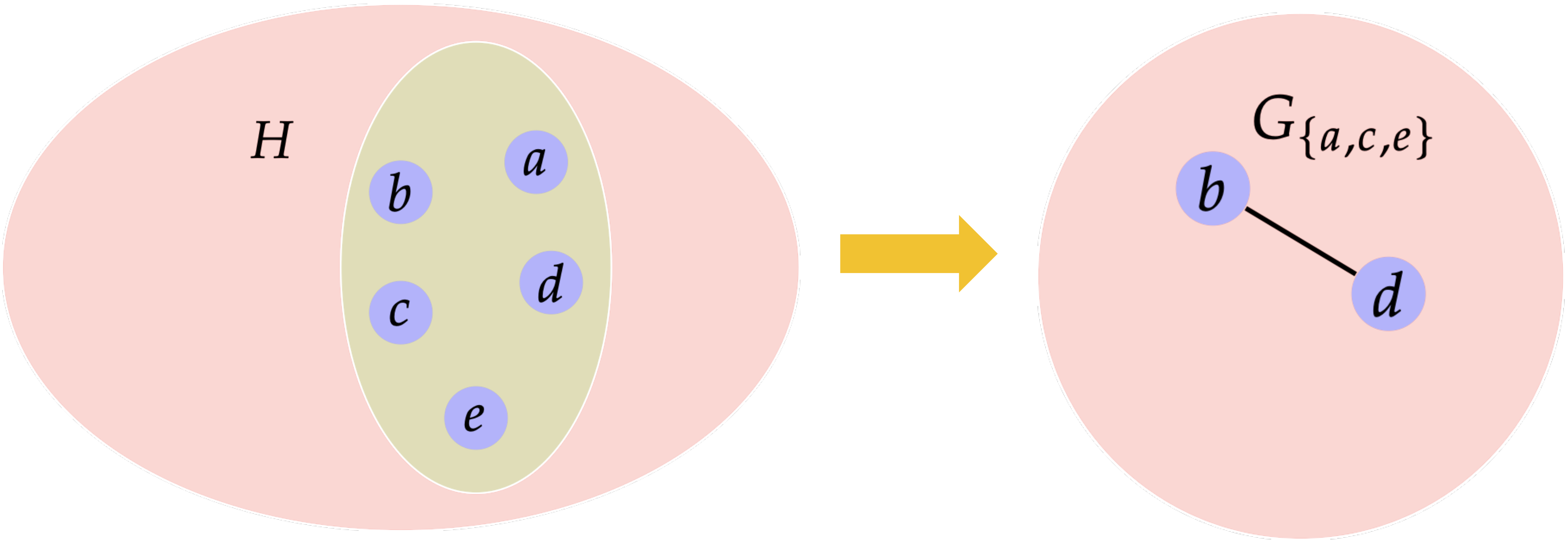
Here we want a d-clique partition of weight at most

$$\left(\frac{1}{d!} + o(1)\right) \frac{n^d}{\lg n}$$

And it turns out this is easy to get after having solved the graph case!

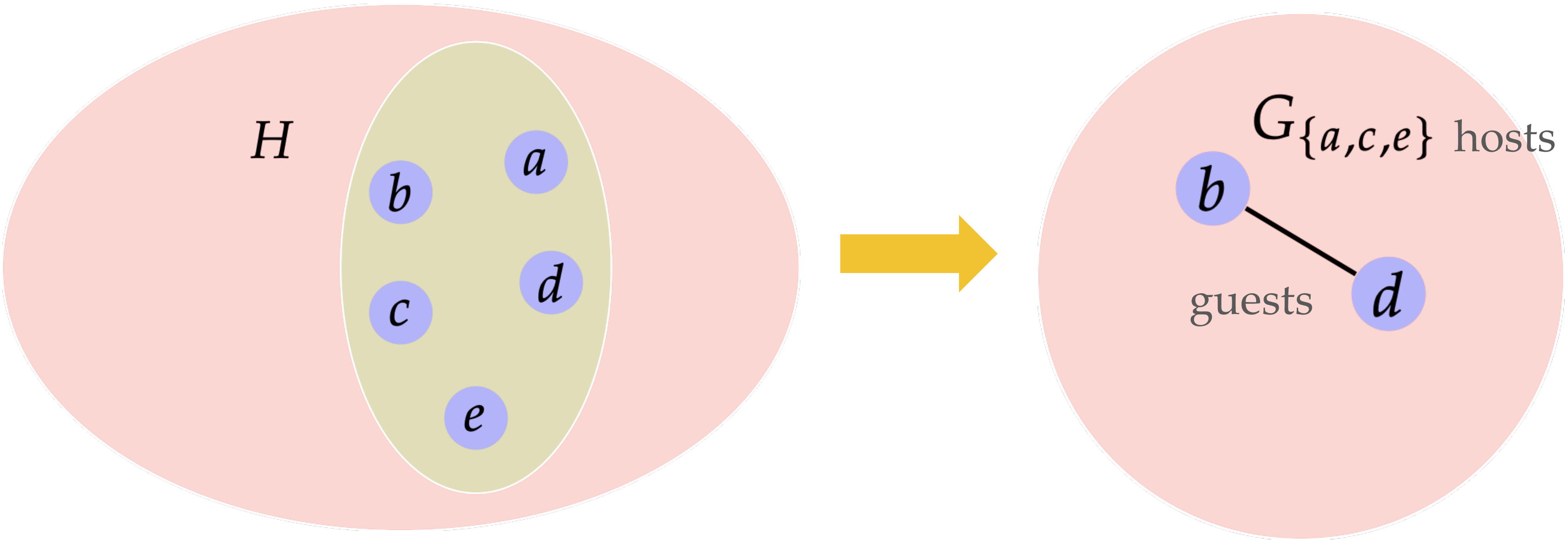
Optimal CES for *d*-Hypergraphs

We will reduce to d=2 through ***link graphs***



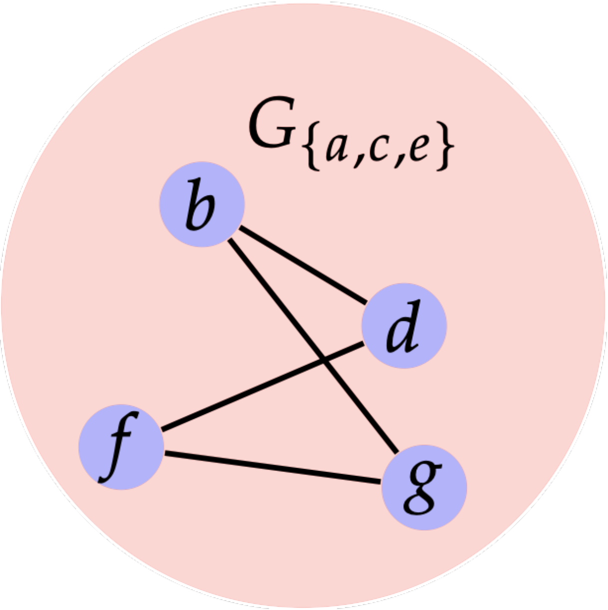
Optimal CES for *d*-Hypergraphs

We will reduce to d=2 through ***link graphs***



Optimal CES for *d*-Hypergraphs

We will reduce to d=2 through ***link graphs***

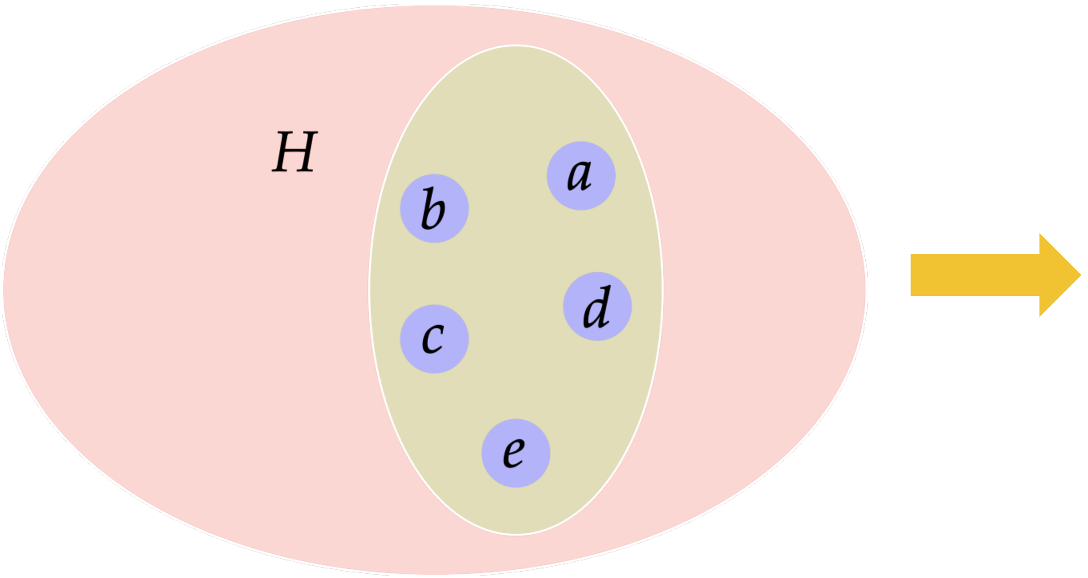


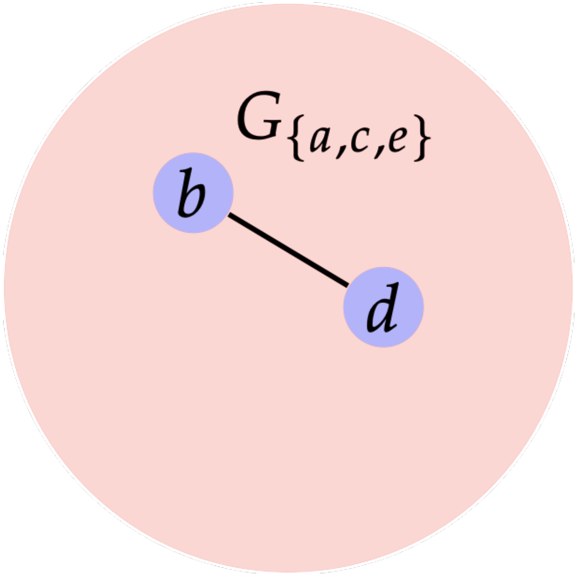
Bicliques in these link graphs are d-cliques in the original graph!

$$(\{b, f\}, \{d, g\}, \{a\}, \{c\}, \{e\})$$

Optimal CES for *d*-Hypergraphs

We will reduce to d=2 through ***link graphs***





We don’t want to also place the edge $\{a,c\}$ on the link graph $G_{\{b,d,e\}}$!

Optimal CES for d-Hypergraphs

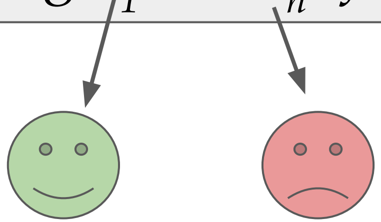
Step 1: Fix an ordering $v_1, \ldots, v_n$ *of the vertices*

Step 2: Separate each d-edge $e := \{x_1, \ldots, x_d\}$ into $S :=$ first d-2 vertices in e according to ordering, and $e \setminus S$, and place edge $e \setminus S$ in link graph G_S

Step 3: Decompose each link graph with an optimal clique partition for graphs

Optimal Erdős–Pyber for *d*-Hypergraphs

Step 1: Fix an ordering $v_1, \ldots, v_n$ *of the vertices*



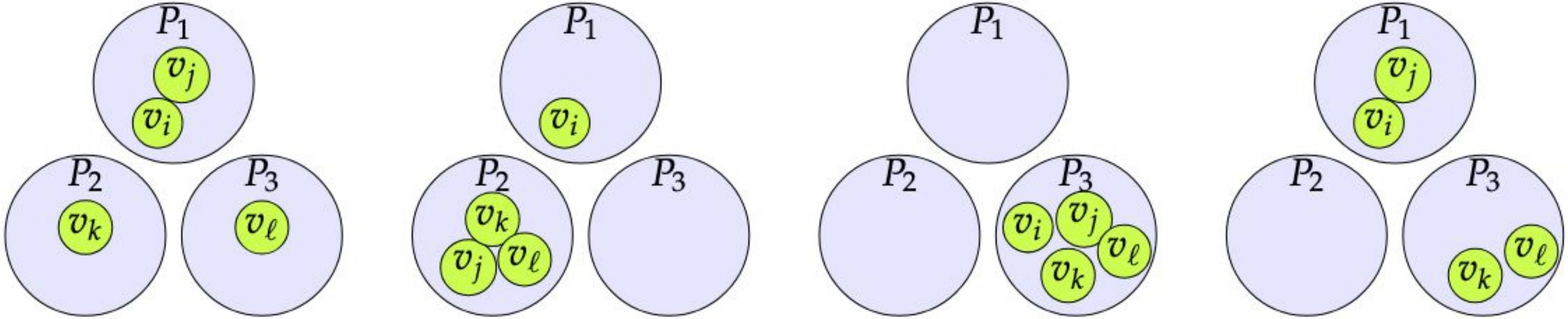
The issue with the previous proof is that any fixed ordering puts much more weight on some vertices than in others!

We need some way to assign edges that is **equitable**!

Optimal Erdős–Pyber for d-Hypergraphs

Step 1: Partition $V(H)$ into d-1 parts

Step 2: Use an *'equitable'* selection strategy for assigning edges to link graphs!



(a) $(2,1,1)$ (b) $(1,3,0)$ (c) $(0,0,4)$ (d) $(2,0,2)$

Optimal Erdős–Pyber for *d*-Hypergraphs

Step 1: Partition $V(H)$ into d-1 parts

Step 2: Use an *'equitable'* selection strategy for assigning edges to link graphs!

How to assign shapes like (2,2,3,3,0,0,0,0,0) or (2,2,2,2,1,1,0,0,0)?

Optimal Erdős–Pyber for d-Hypergraphs

How to assign shapes like (2,2,3,3,0,0,0,0,0) ?

It suffices to prove that their sets of cyclic rotations
{ (0,2,2,3,3,0,0,0,0), … (3,3,0,0,0,0,0,2,2),…}
always have size *d-1 :)*

That way, from each equivalence class under rotation we assign one element to each part.

Optimal Erdős–Pyber for d-Hypergraphs

How to assign shapes like (2,2,3,3,0,0,0,0,0) ?

It suffices to prove that their sets of cyclic rotations
{ (0,2,2,3,3,0,0,0,0), … (3,3,0,0,0,0,0,2,2),…}
always have size *d-1 :)*

In general, something like (1,1,1,1,1) or (2,2,0,2,2,0) has fewer than *d-1* distinct rotations…

Optimal Erdős–Pyber for *d*-Hypergraphs

How to assign shapes like (2,2,3,3,0,0,0,0,0) ?

It suffices to prove that their sets of cyclic rotations

{ (0,2,2,3,3,0,0,0,0), … (3,3,0,0,0,0,0,2,2),…}

always have size *d-1 :)*

In general, something like (1,1,1,1,1) or (2,2,0,2,2,0) has fewer than *d-1* distinct rotations…

This happens only when the shape is of the form w^k *for some* $k > 1$

Optimal Erdős–Pyber for d-Hypergraphs

How to assign shapes like (2,2,3,3,0,0,0,0,0) ?

In general, something like (1,1,1,1,1) or (2,2,0,2,2,0) has fewer than d-1 distinct rotations…

This happens only when the shape is of the form w^k *for some* $k > 1$

But our words have d-1 components, and their sum is d. If they were of the form w^k, then $|w| \cdot k = d-1$. But also the sum would be $k \cdot \mathrm{sum}(w) = d$. **Contradiction**!

Optimal Erdős–Pyber for d-Hypergraphs

Given a uniformly random potential hypergraph edge e, we want the probability that part i pays for it to be $1/(d-1)$

Optimal Erdős–Pyber for d-Hypergraphs

Given a uniformly random potential hypergraph edge e, we want the probability that part i pays for it to be $1/(d-1)$

Note that this depends only on the part-distribution of e, not on its individual vertices. Therefore, if f is the function that assigns part-distributions to parts, we want

Optimal Erdős–Pyber for *d*-Hypergraphs

Given a uniformly random potential hypergraph edge *e*, we want the probability that part *i* pays for it to be *1/(d-1)*

Note that this depends only on the part-distribution of *e*, not on its individual vertices. Therefore, if *f* is the function that assigns part-distributions to parts, we want

$$\Pr_e[f(\text{distribution of } e) = i] = \sum_{(x_1,\ldots,x_{d-1}) \in f^{-1}(i)} \frac{\binom{d}{x_1,\ldots,x_{d-1}}}{(d-1)^d} = \frac{1}{d-1}$$

Optimal Erdős–Pyber for d-Hypergraphs

Step 2: Use an equitable selection strategy for assigning edges to link graphs!

Definition 16. Let $d \geqslant 2$ be an integer. A *d-distribution* is a tuple $(x_1, \ldots, x_{d-1}) \in \mathbb{N}^{d-1}$ such that $x_1 + \ldots + x_{d-1} = d$. Let $\mathcal{S}_d$ be the set of d-distributions. A *selection strategy* is a function $f : \mathcal{S}_d \to [d-1]$ such that $f(x_1, \ldots, x_{d-1}) = i$ for some i such that $x_i \geqslant 2$. Then, we say a selection strategy is *equitable* if for all $i, j \in [d-1]$,

$$\sum_{(x_1,\ldots,x_{d-1}) \in f^{-1}(i)} \binom{d}{x_1, \ldots, x_{d-1}} = \sum_{(x_1,\ldots,x_{d-1}) \in f^{-1}(j)} \binom{d}{x_1, \ldots, x_{d-1}}.$$

Lemma 17. *For every $d \geqslant 2$, there exists an equitable selection strategy.*

Optimal Erdős–Pyber for d-Hypergraphs

How to assign shapes like (2,2,3,3,0,0,0,0,0) ?

It suffices to prove that their sets of cyclic rotations

{ (0,2,2,3,3,0,0,0,0), … (3,3,0,0,0,0,0,2,2),…}

always have size *d-1 :)*

The claim follows from word combinatorics, or the orbit-stabilizer theorem!

Density-aware partitions

Theorem 1. *Let γ such that* $\max\{\gamma^{-1}, (1-\gamma)^{-1}\} = n^{o(1)}$. *Then,*

$$\mathsf{BP}(n,\gamma) \sim \lg \binom{\binom{n}{2}}{\gamma\binom{n}{2}} / \lg n \sim \frac{h_2(\gamma)}{2} \frac{n^2}{\lg n},$$

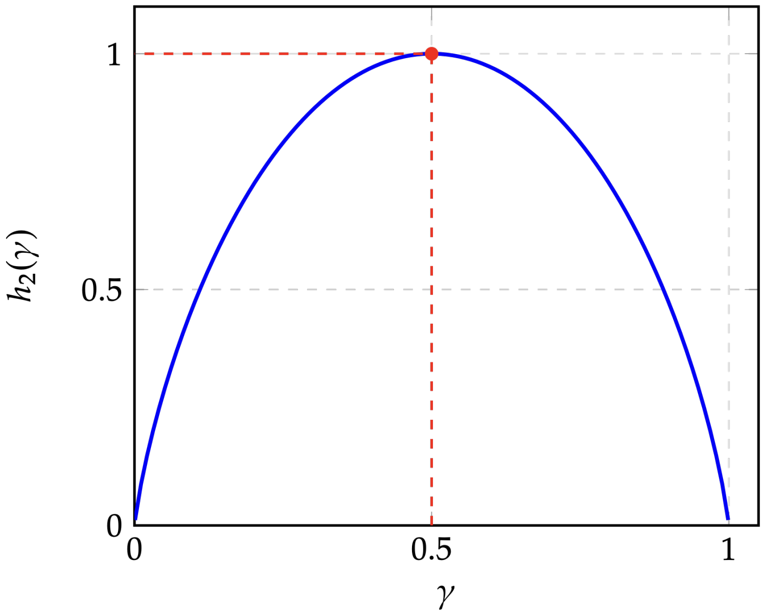
where $h_2(x) := -x\lg(x) - (1-x)\lg(1-x)$ *is the binary entropy function.*

Based on Stirling: $$\binom{n}{n\gamma_n} \sim \frac{2^{h_2(\gamma_n)n}}{\sqrt{2\pi\gamma_n(1-\gamma_n)n}},$$

Theorem 1. *Let γ such that $\max\{\gamma^{-1}, (1-\gamma)^{-1}\} = n^{o(1)}$. Then,*

$$\mathsf{BP}(n,\gamma) \sim \lg \binom{\binom{n}{2}}{\gamma\binom{n}{2}} / \lg n \sim \frac{h_2(\gamma)}{2} \frac{n^2}{\lg n},$$

where $h_2(x) := -x\lg(x) - (1-x)\lg(1-x)$ is the binary entropy function.



Density-aware partitions

Theorem 1. *Let γ such that $\max\{\gamma^{-1}, (1-\gamma)^{-1}\} = n^{o(1)}$. Then,*

$$\mathsf{BP}(n,\gamma) \sim \lg \binom{\binom{n}{2}}{\gamma\binom{n}{2}} / \lg n \sim \frac{h_2(\gamma)}{2} \frac{n^2}{\lg n},$$

where $h_2(x) := -x\lg(x) - (1-x)\lg(1-x)$ is the binary entropy function.

So biclique partitions are an ***information-theoretically optimal representation*** at almost every density!

Density-aware partitions (***Proof Idea***)

On top of splitting into equal-sized parts (now bigger), and setting up the almost-regular tournament, we view the adjacencies of vertex v into part P_i as a binary string like

0011101010000010000000011111111000000100001000010000001

Density-aware partitions (***Proof Idea***)

On top of splitting into equal-sized parts (now bigger), and setting up the almost-regular tournament, we view the adjacencies of vertex v into part P_i as a binary string like

0011101010000010000000111111110000010000100001000000001

And split it into slices of roughly equal ***entropy*** $:= \lg \binom{\ell}{w}$

Density-aware partitions (***Proof Idea***)

001110101000001000000011111111000001000010000100000001

And split it into slices of roughly equal ***entropy*** $:= \lg \binom{\ell}{w}$

Now bicliques will be within slices!

Recall: $$\prod_{i=1}^{k} \binom{a_i}{b_i} \leqslant \binom{a_1 + \cdots + a_k}{b_1 + \cdots + b_k}$$

Density-aware partitions (***Proof Idea***)

0011101010000010000000111111110000010000100001000000001

And split it into slices of roughly equal ***entropy*** := $\lg \binom{\ell}{w}$

Now bicliques will be within slices!

Recall: $$\prod_{i=1}^{k} \binom{a_i}{b_i} \leqslant \binom{a_1 + \cdots + a_k}{b_1 + \cdots + b_k}.$$

$$\sum_{v \in V} \sum_{\text{slice index } i} \lg \binom{\ell_i}{w_i} = \lg \left(\prod_{v \in V} \prod_{\text{slice index } i} \binom{\ell_i}{w_i} \right) \leqslant^{\text{Rem. 24}} (1 + o(1)) \lg \binom{n^2/2}{|E|} \sim^{\text{Rem. 23}} h_2(\gamma) \frac{n^2}{2},$$

My naive view of *compression*

100010101010111111111111111

0101101101010001100100

My naive view of *compression*

pattern

10001010101010111111111111111

0101101101010001100100

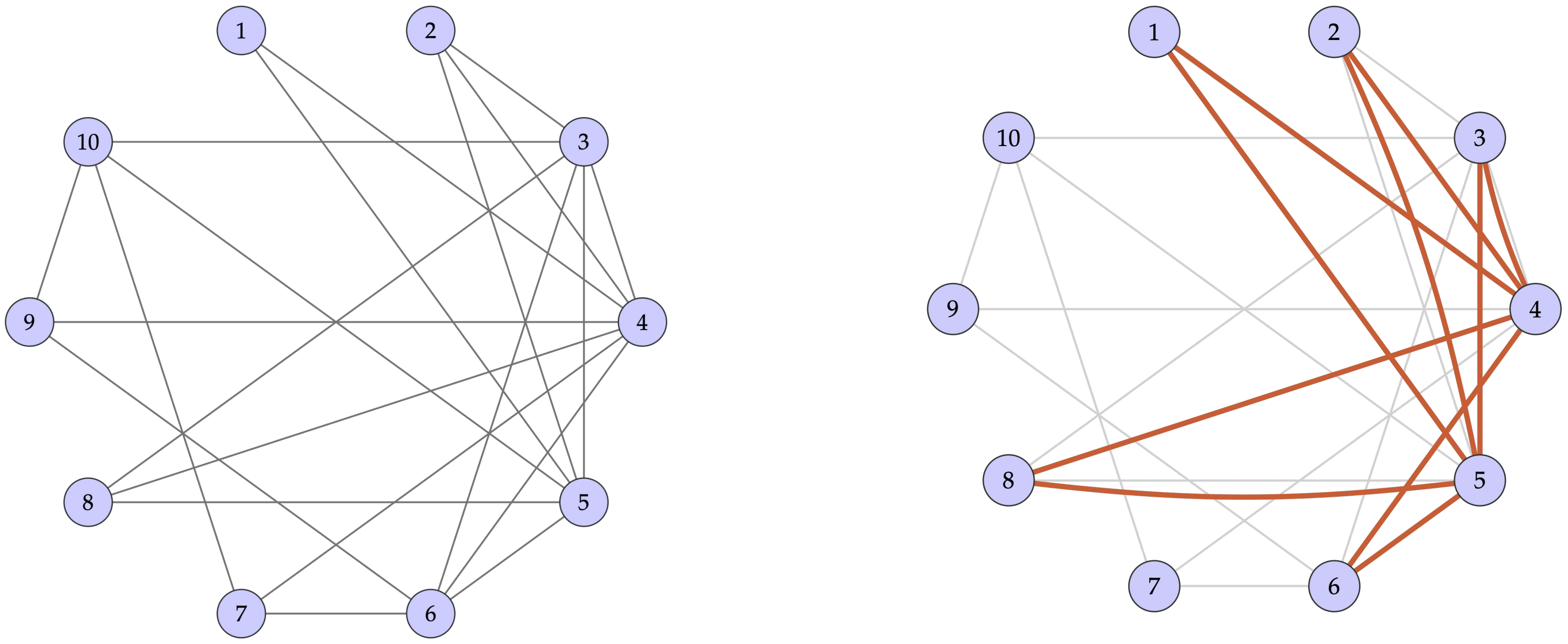
My naive view of *compression*

pattern

100010101010 111111111111111
0101101101010001100100

compression

100010101010 **{15x1}**
0101101101010001100100



Bicliques are the graph-compression pattern!

Algorithmic applications

The previous result implies that biclique partitions make for a ***Succinct data structure!***

(*and* ***compact*** if we also store the vertex-biclique incidence list)

Algorithmic applications

The previous result implies that biclique partitions make for a ***Succinct data structure!***

(*and* ***compact*** if we also store the vertex-biclique incidence list)

Theorem 1. *Using a biclique partition of weight $O(n^2/\lg n)$ as a succinct data structure we can:*

1. *Given $S \subseteq V$, determine if S is independent in $O(n^2/\lg n)$,*
2. *Given disjoint $S, T \subseteq V$, determine $e(S,T)$ in $O(n^2/\lg n)$.*

Moreover, as a compact data structure, for any $\alpha > 1$, we can compute a 2α-approximation for densest subgraph in time $O(n^2/\lg \alpha)$.

We store our graph as

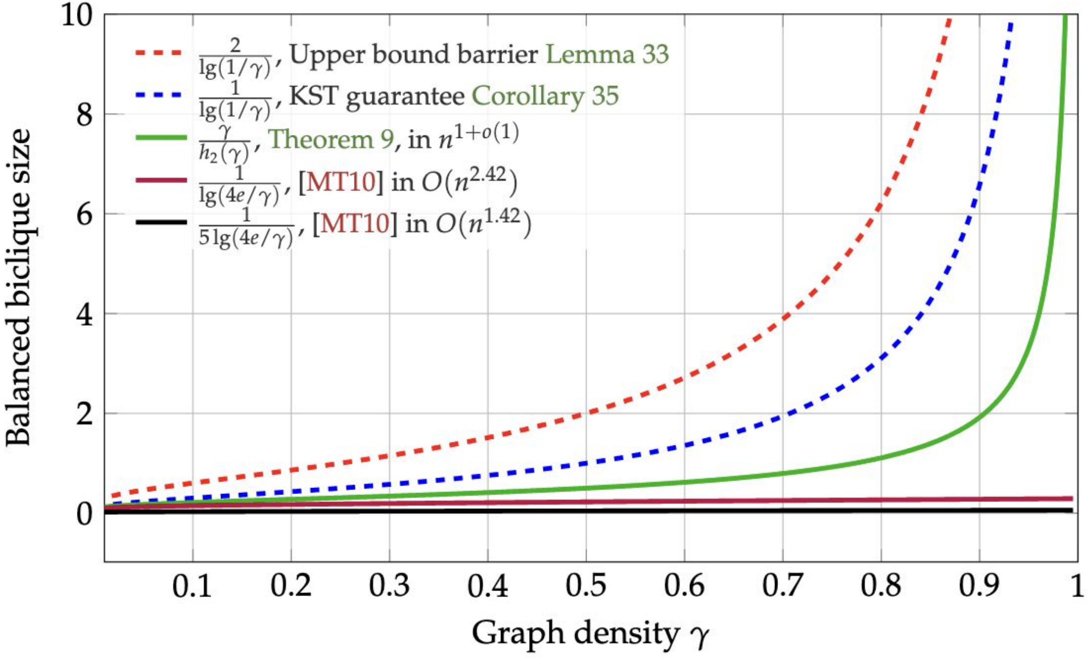
$$(L_1, R_1), (L_2, R_2), \ldots, (L_k, R_k)$$

A set S is independent iff for each i either $S \cap L_i = \varnothing$ or $S \cap R_i = \varnothing$

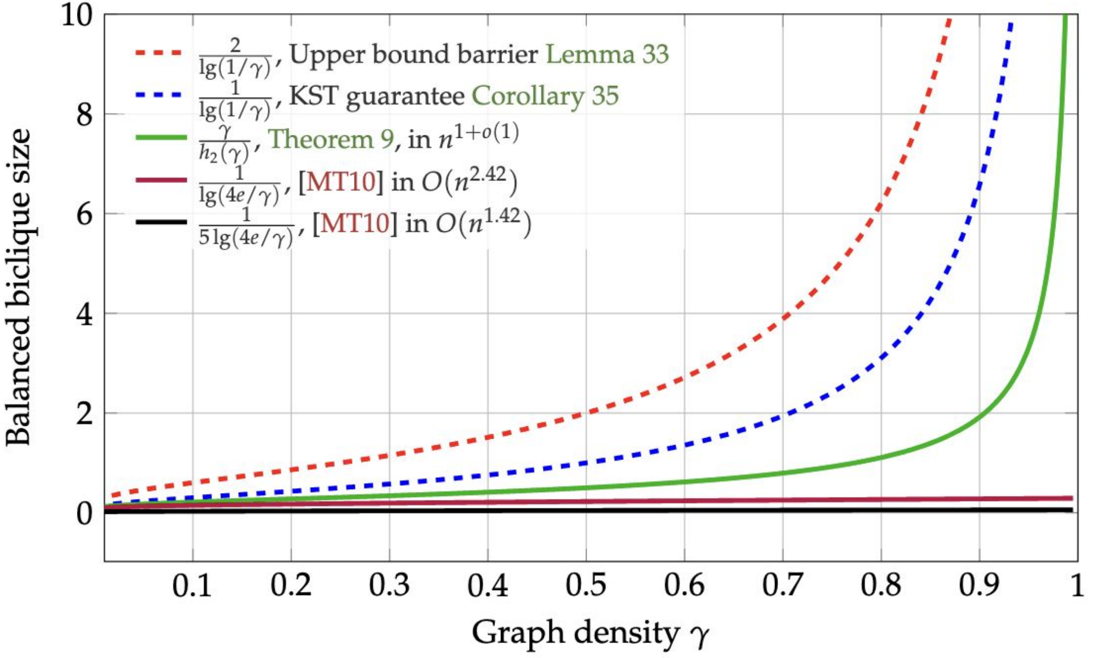
We can store S *as a bitvector* B_S *where* $B_S[j] = 1$ *iff* j *is in* S

Tben, for each (L_i, R_i) *we go through their vertices and check on* B_S, *if both* L_i *and* R_i *are a 'hit' for* B_S, *then* S *is not independent. If the test passes for every* i*, then independent.*

Faster algorithm for finding density-guaranteed balanced bicliques



Theorem 9. *Given a graph with edge density γ such that $\max\{\gamma^{-1}, (1-\gamma)^{-1}\} = n^{o(1)}$, there is a randomized $O(\frac{n \lg n}{h_2(\gamma)}) = n^{1+o(1)}$ time algorithm that returns a biclique $K_{t,t}$ with $t = (1-o(1))\frac{\gamma \lg n}{h_2(\gamma)}$ with high probability. Furthermore, if we are allowed to query the degree of a vertex in $O(1)$ time, the algorithm can be made deterministic.*



Breaking down Graphs and Hypergraphs into Structured Pieces

Joint work with Benjamin Przybocki, Andrew Krapivin, and Nicolás Sanhueza-Matamala

Bernardo Subercaseaux (CMU)

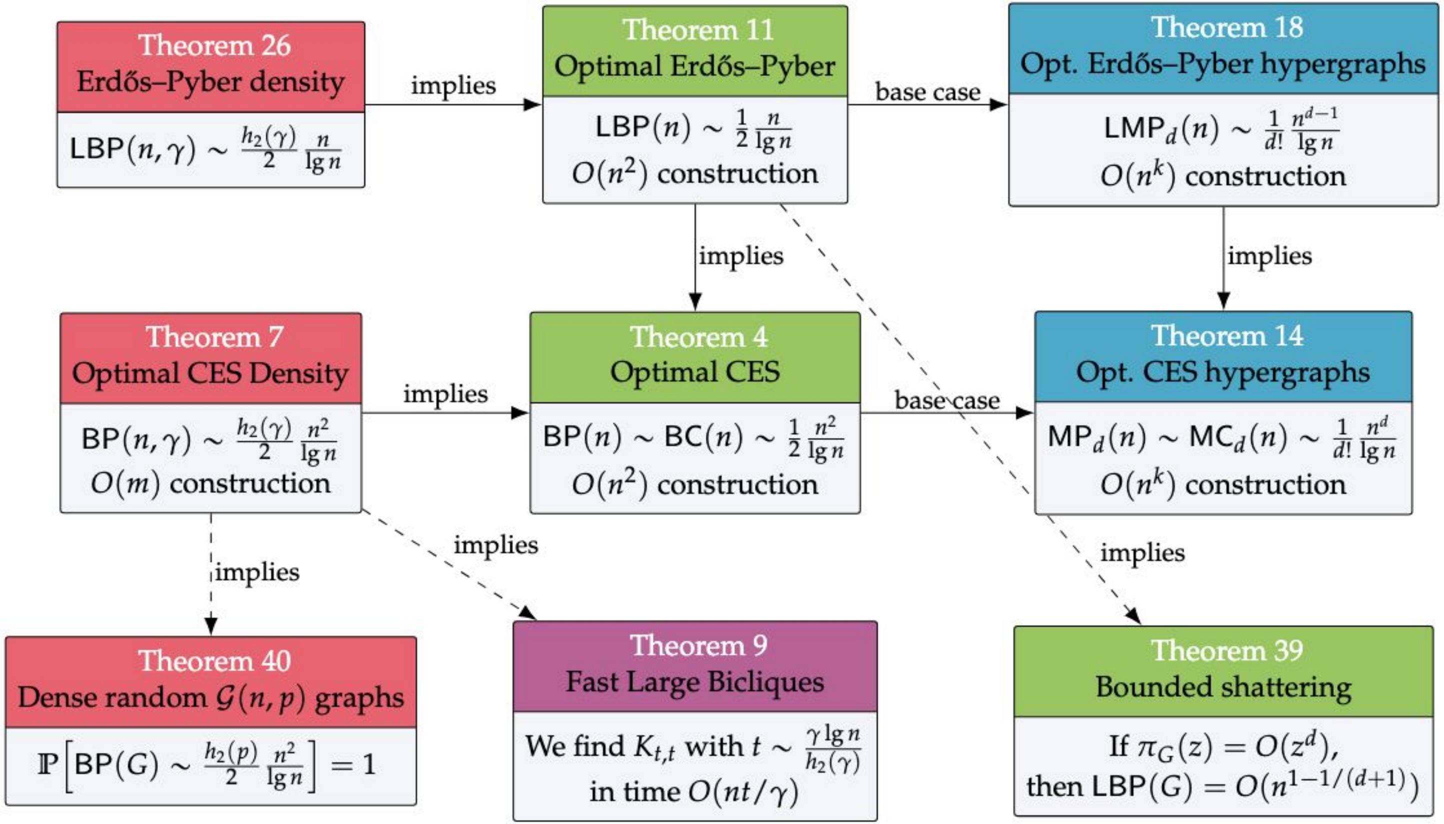


Figure 2: Summary of our results. Dashed implications represent that other tools are also required. (■ := density-aware, ■ := graph partitions, ■ := hypergraph partitions, ■ := finding bicliques)