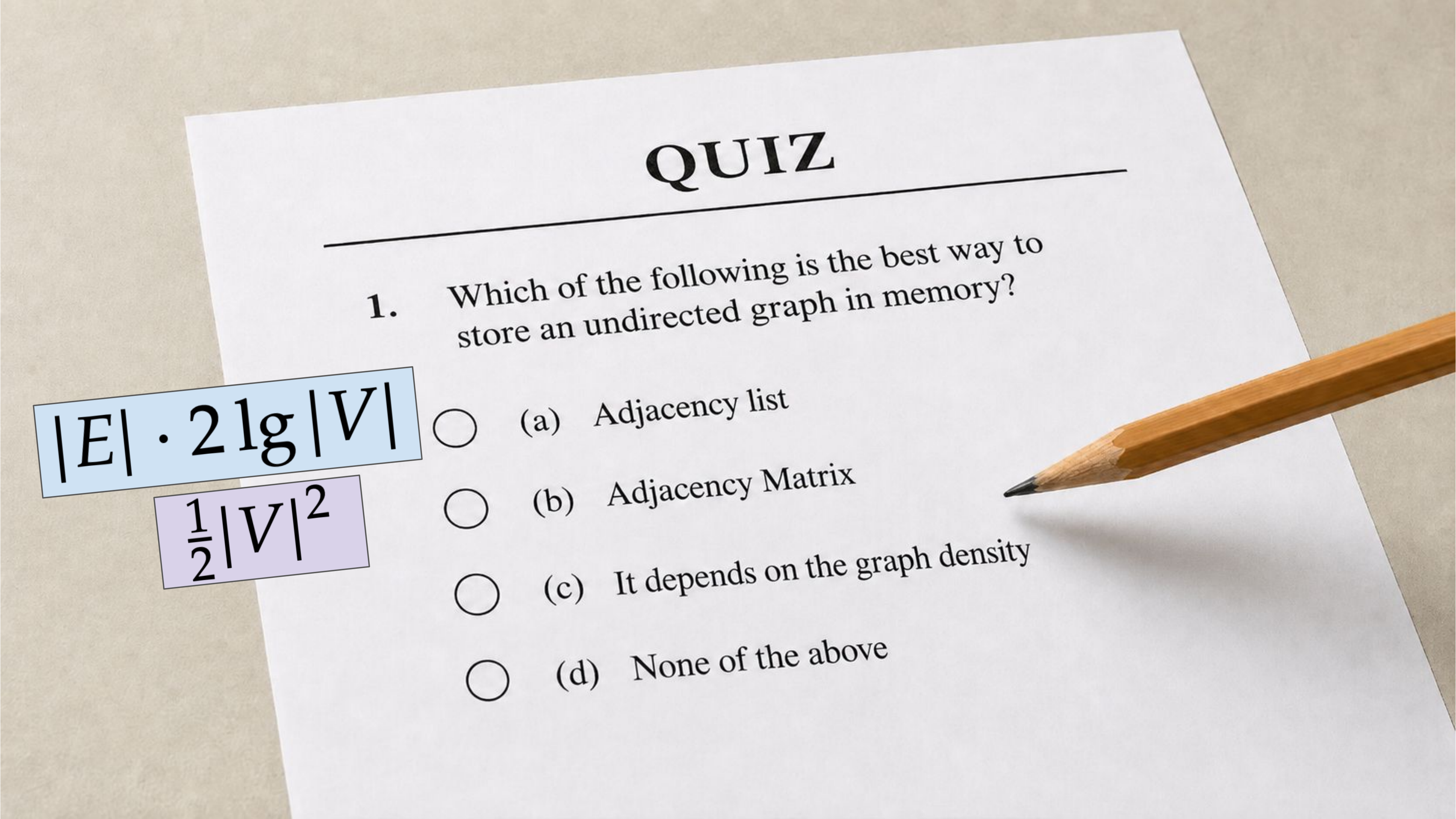
Breaking down Graphs and Hypergraphs into Structured Pieces

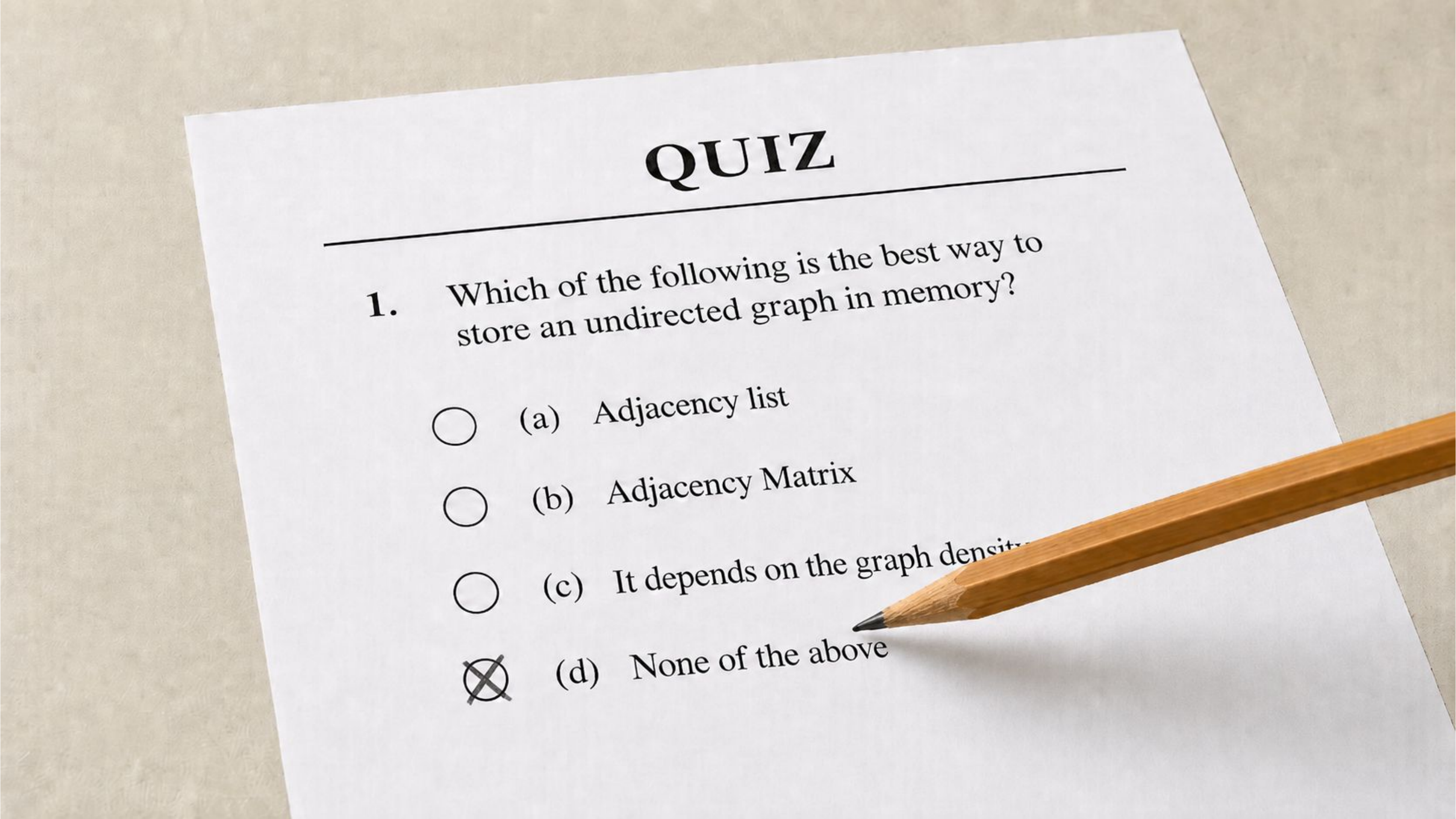
Joint work with Benjamin Przybocki, Andrew Krapivin, and Nicolás Sanhueza-Matamala

Bernardo Subercaseaux (CMU)



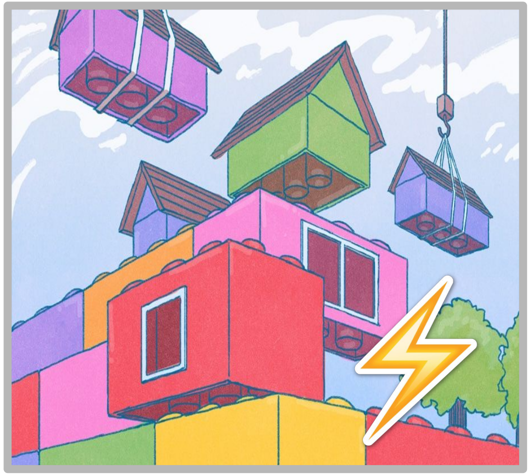
$$|E| \cdot 2 \lg |V|$$

$$\frac{1}{2}|V|^2$$


QUIZ
1. Which of the following is the best way to store an undirected graph in memory?
(a) Adjacency list
(b) Adjacency Matrix
(c) It depends on the graph
(d) None of the above

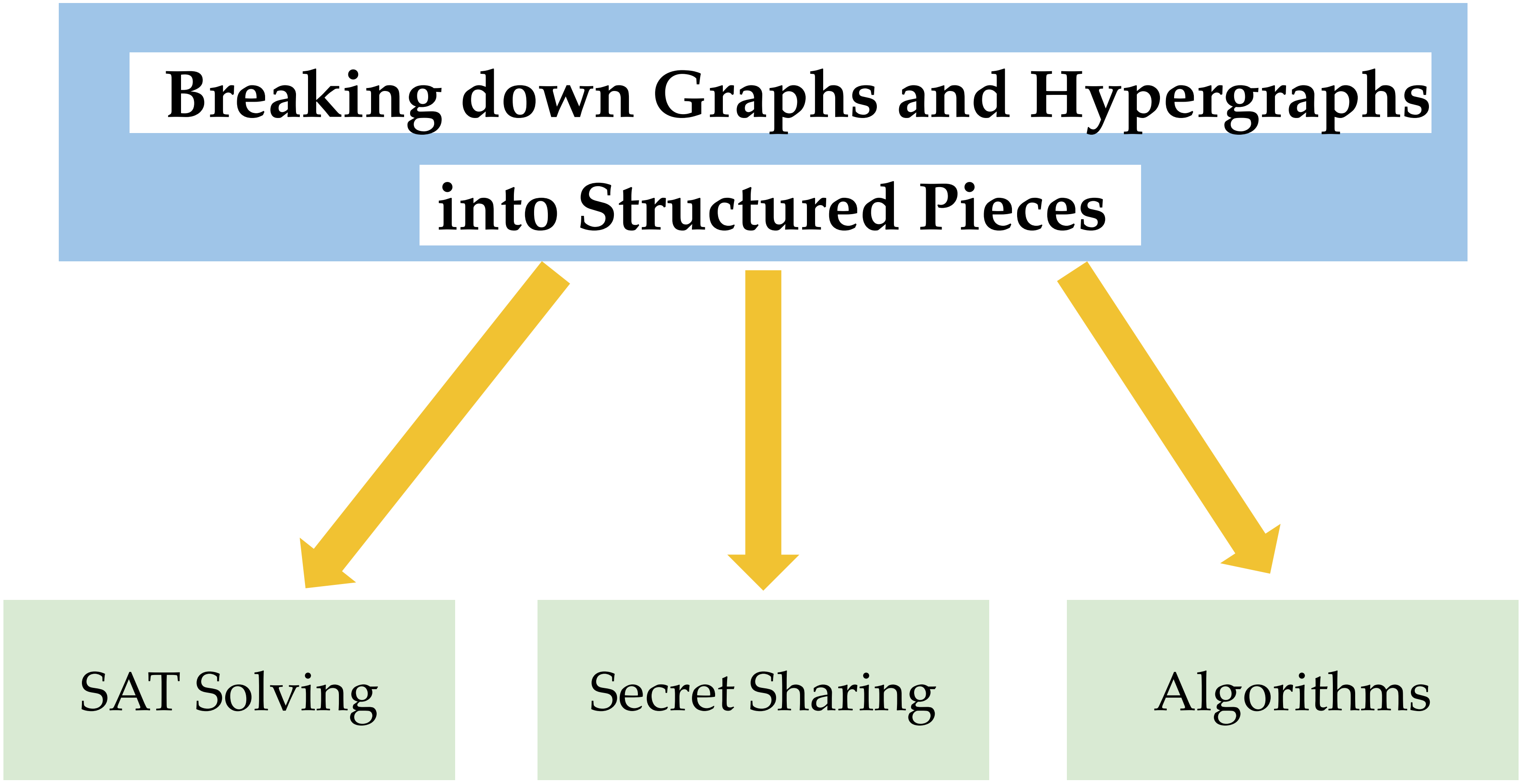
Goals

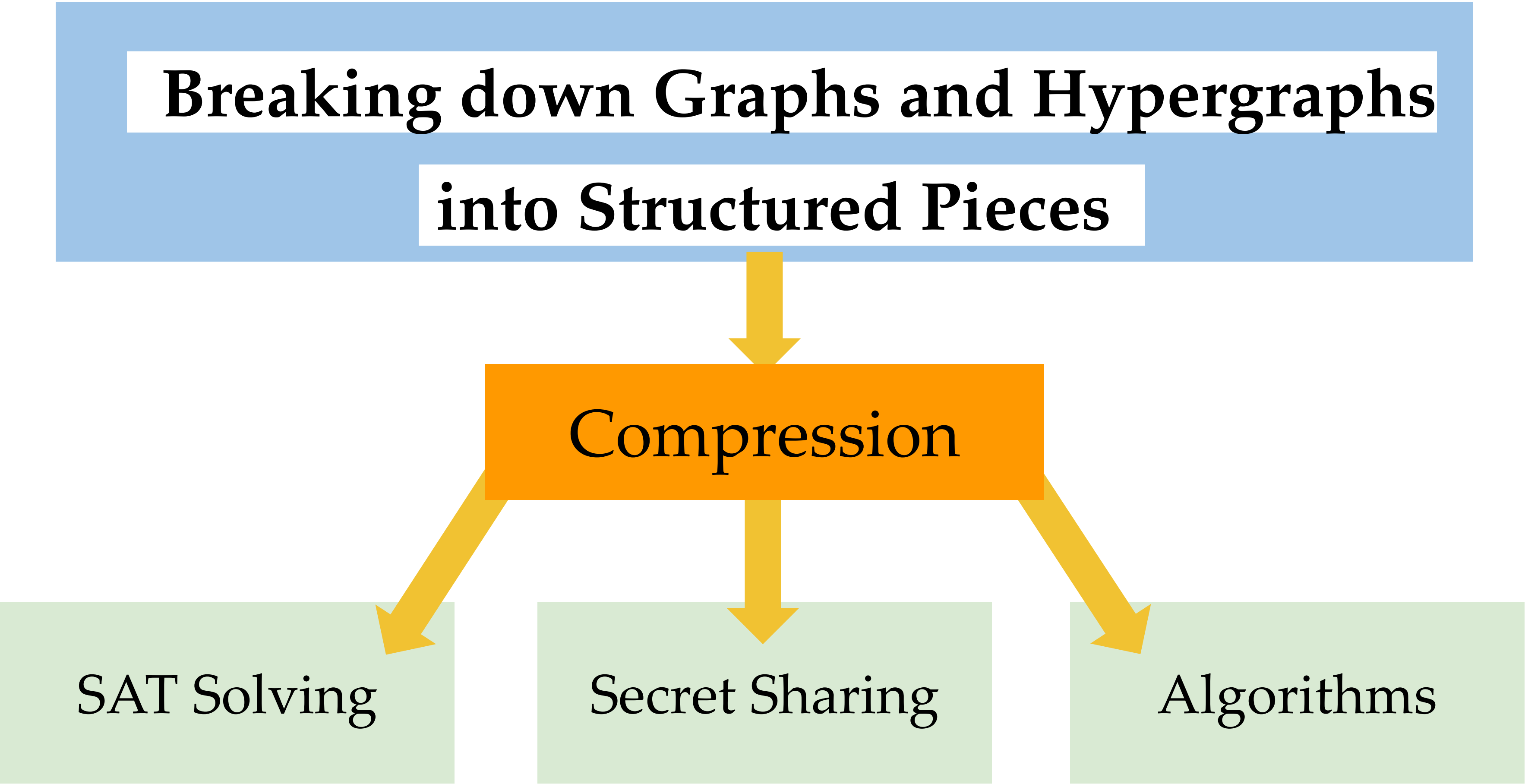
Space efficient



Fast build

Fast algorithms


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

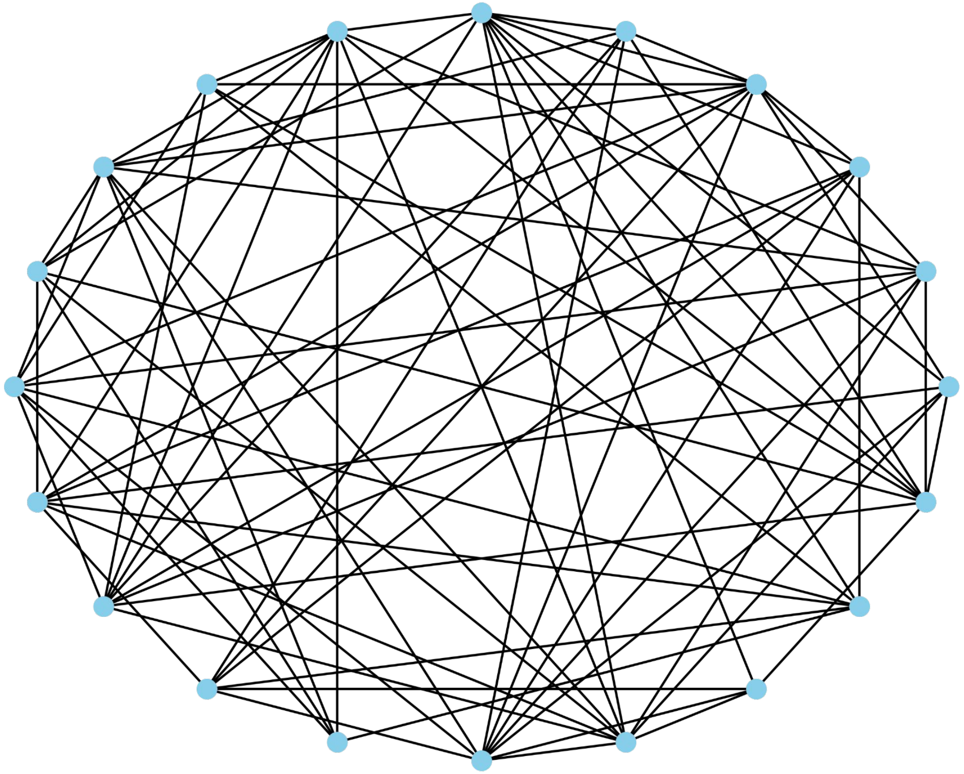
My naive view of *compression*

pattern

100010101010111111111111111
0101101101010001100100

compression

100010101010{**15x1**}
0101101101010001100100



Say we want to *compress* this graph

What could be a pattern?

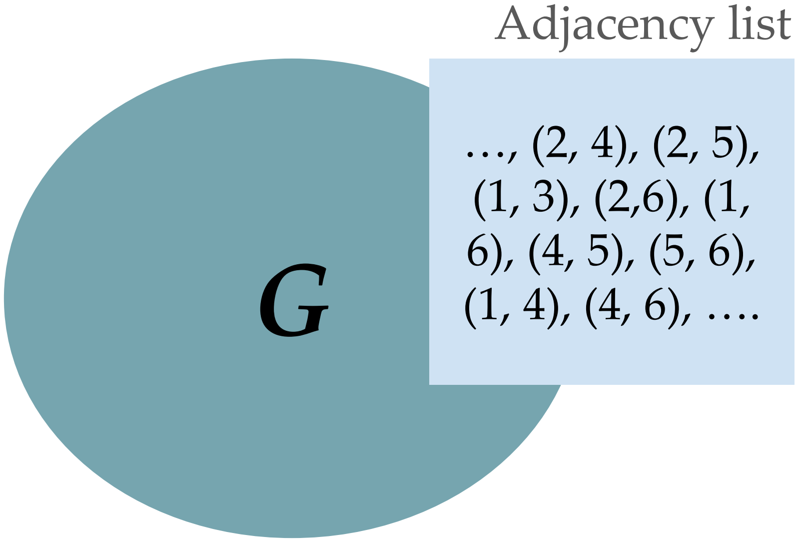
Could there be none?

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

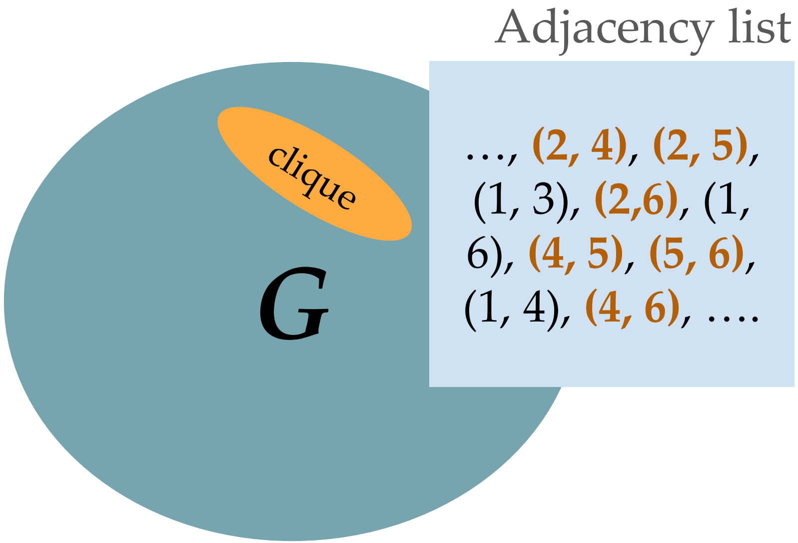
Frank Ramsey

Paul Erdős and George Szekeres

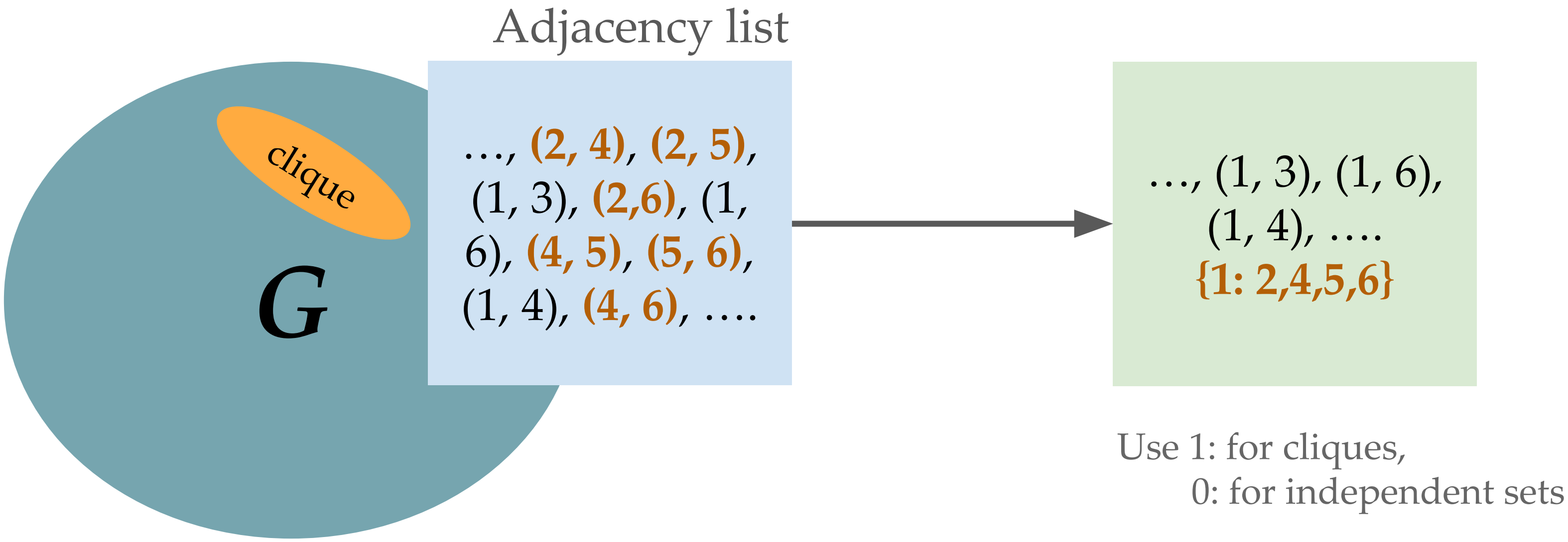
Theorem (Ramsey, Erdős, Szekeres). *Every graph on n vertices contains either a clique of size $\Omega(\lg n)$ or an independent set of size $\Omega(\lg n)$.*



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

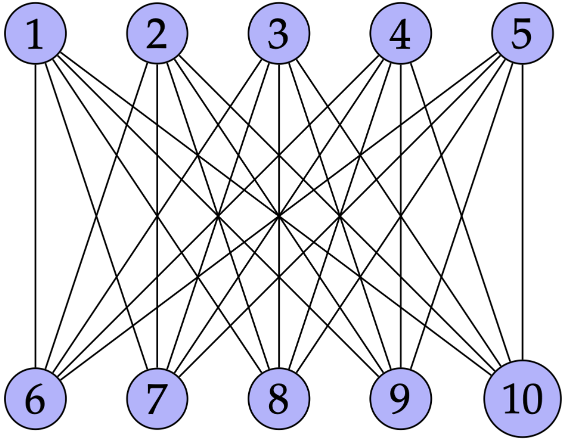


Theorem (Ramsey, Erdős, Szekeres). *Every graph on n vertices contains either a clique of size $\Omega(\lg n)$ or an independent set of size $\Omega(\lg n)$.*



Idea: write our graph **G** as a union of sets $S_1, S_2, \ldots, S_k$ each corresponding to a clique, using $|S_i| \lg n$ bits for the vertices

Idea: write our graph **G** as a union of sets $S_1, S_2, \ldots, S_k$ each corresponding to a clique, using $|S_i|$lg n bits for the vertices

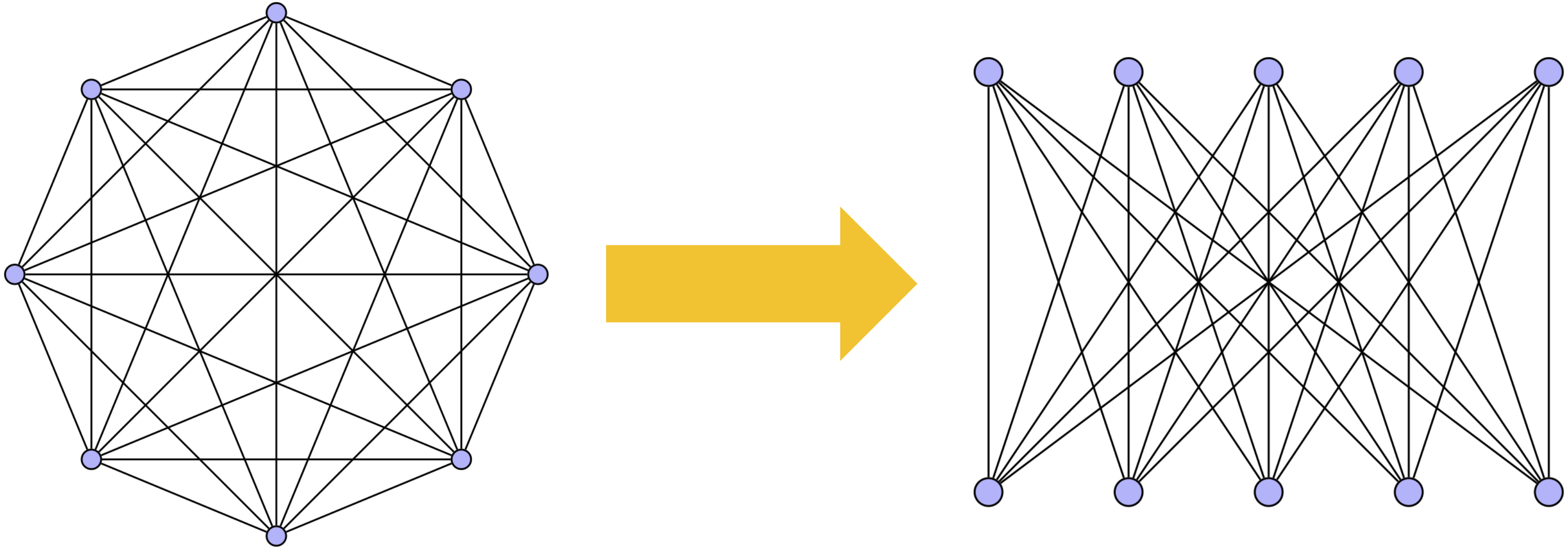


1: 1,6
1: 1,7
…
1: 5,6
…
1: 5,10

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

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

The problem is the pattern!

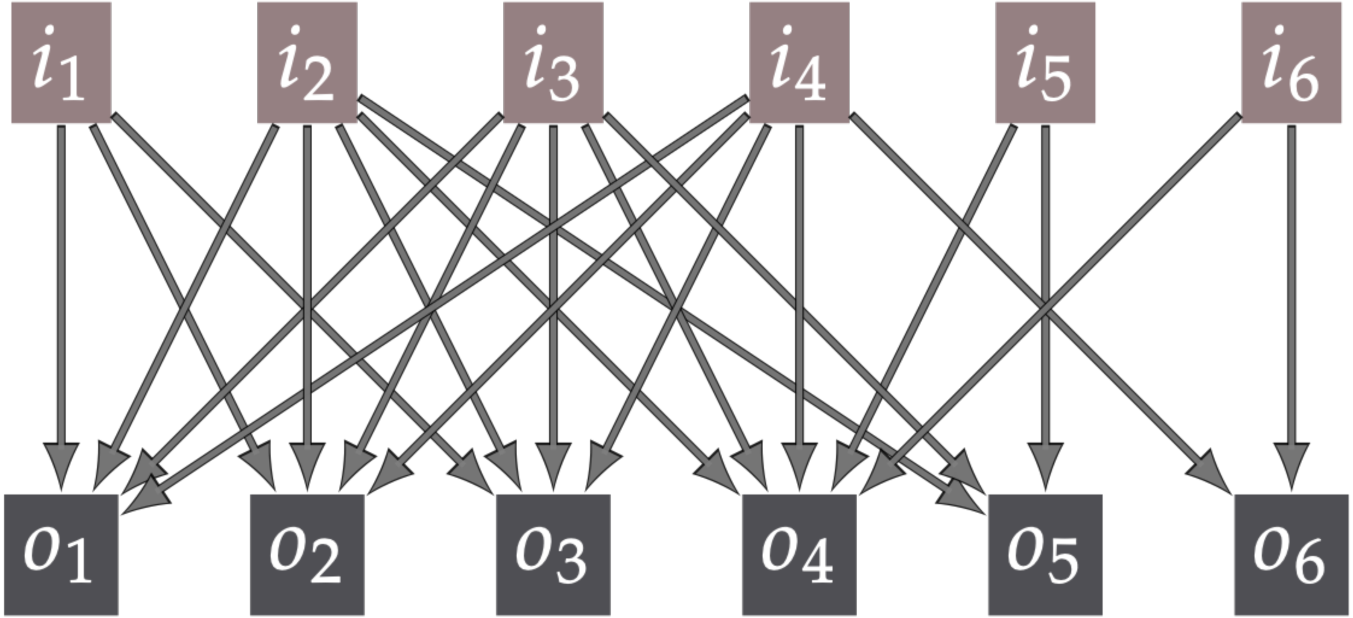


cliques

bicliques

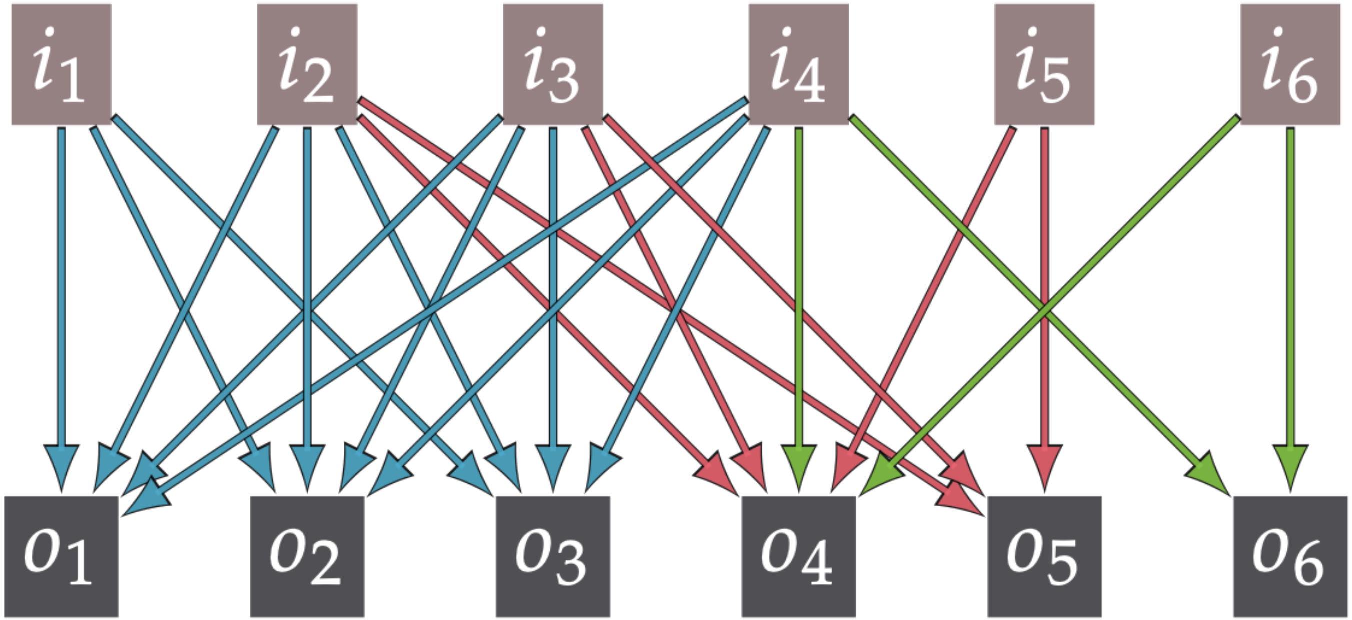
A brief history of **biclique partitions**

Suppose we have a dense electric circuit



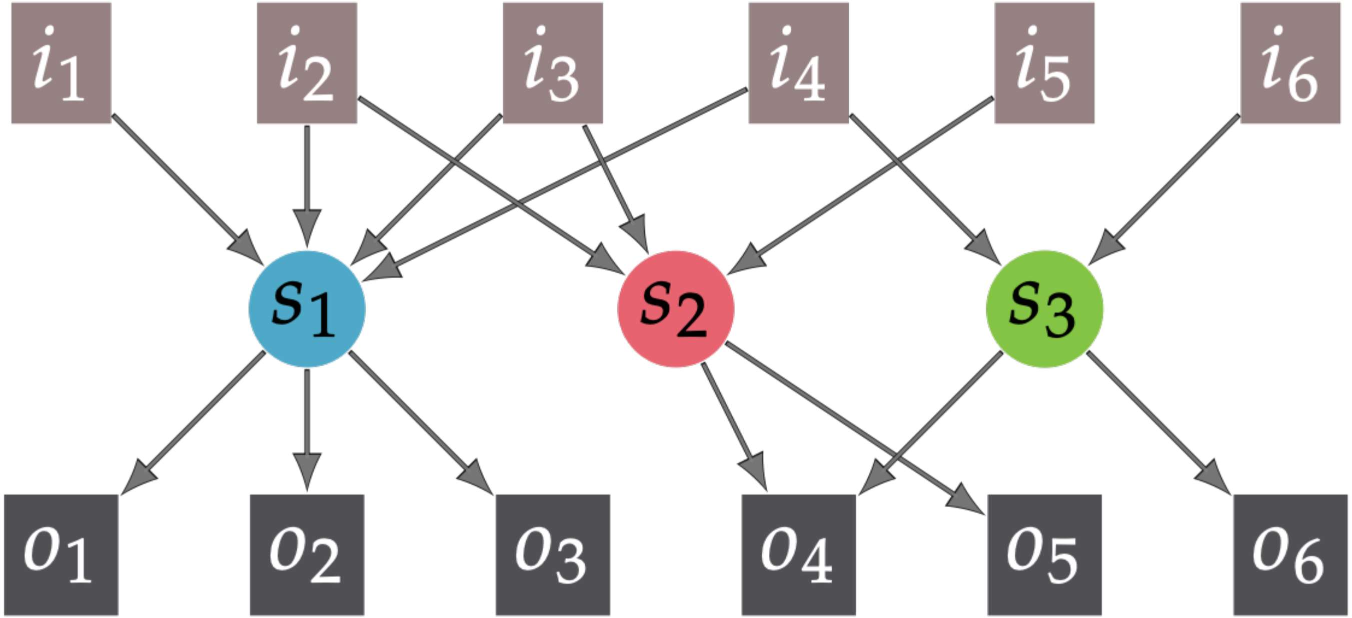
And we want to **reduce the number of wires**…

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



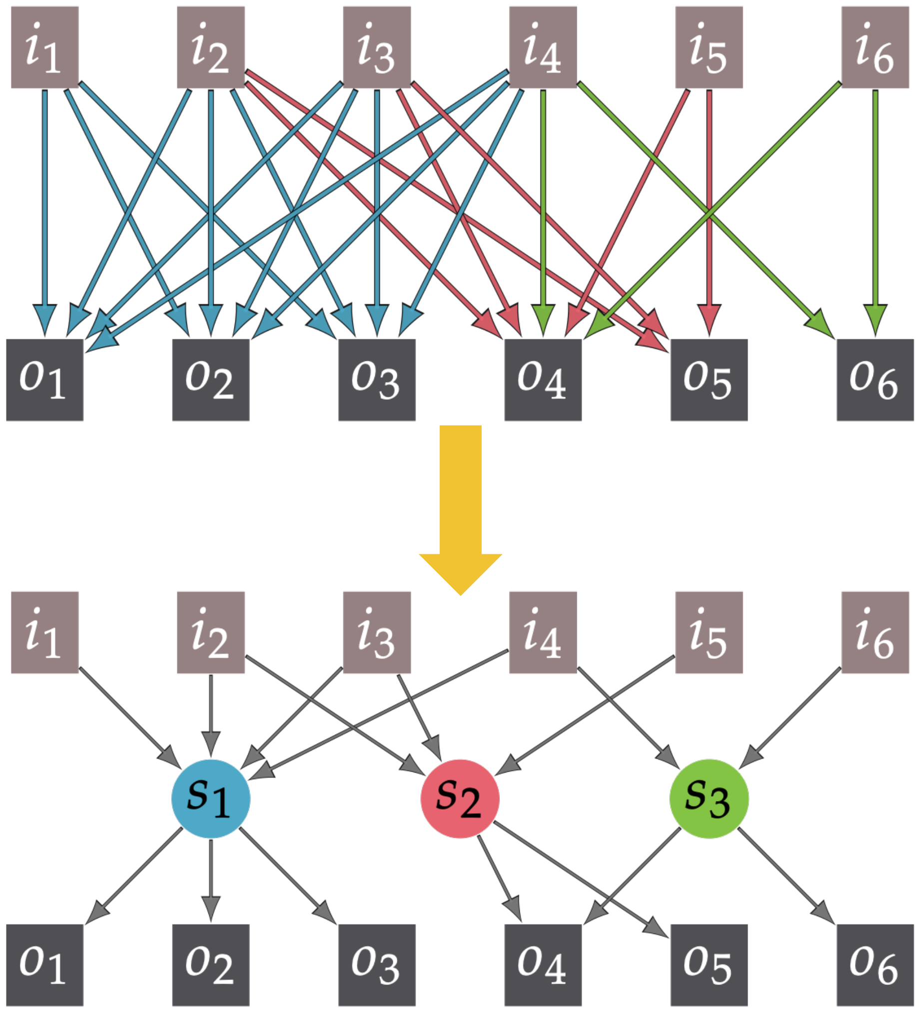
we can easily reduce each of them!

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

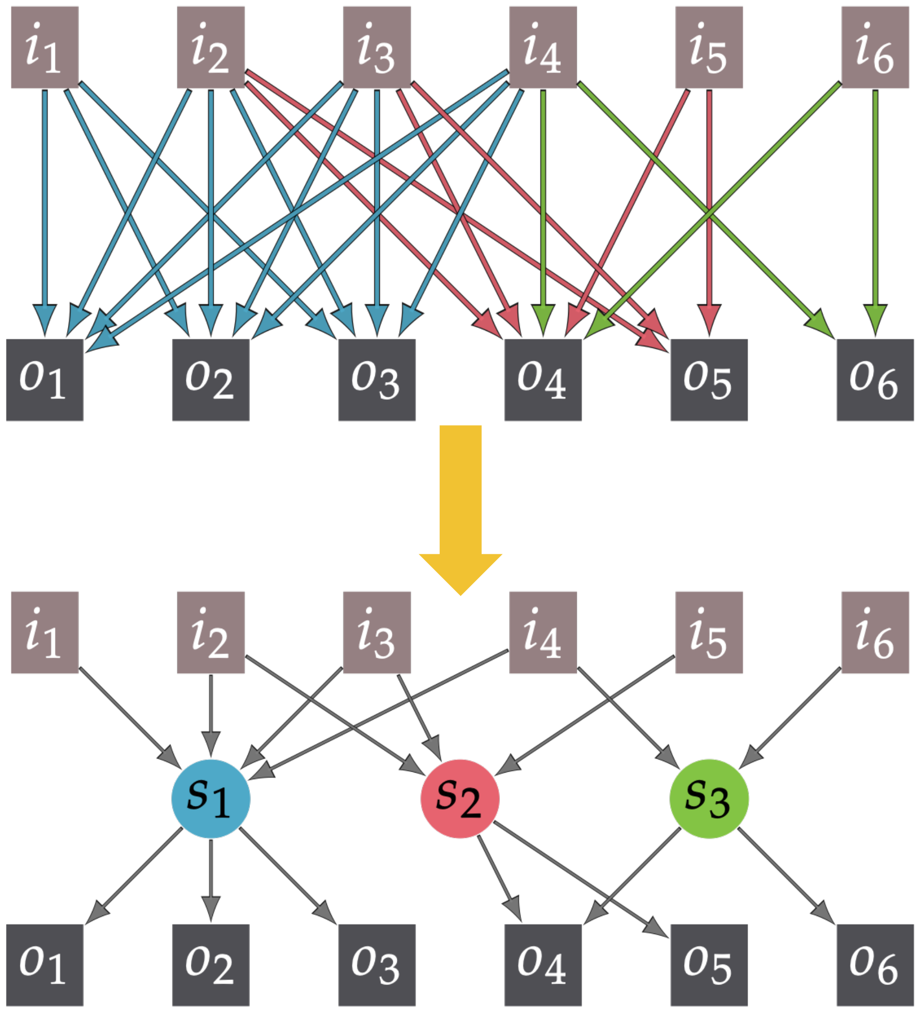


we can easily reduce each of them!

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



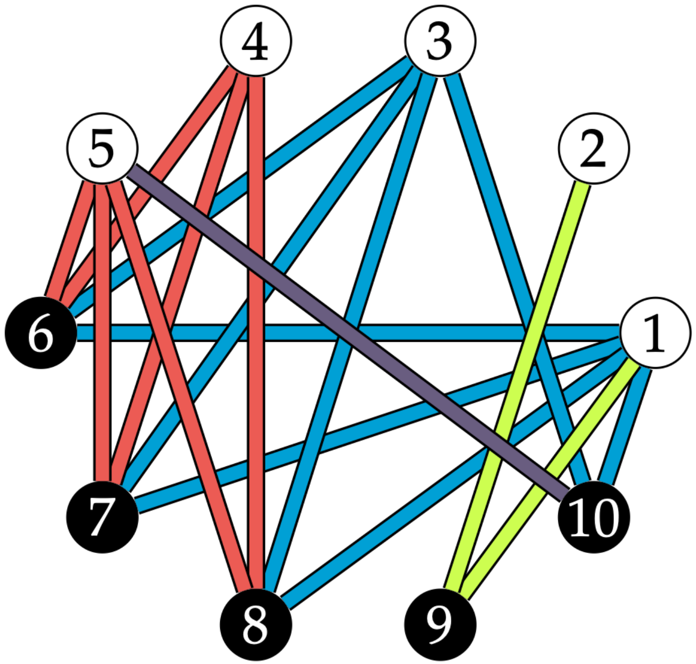
Introducing a ***switch*** we can reduce ***ab*** wires to ***a* + *b*** if we identify a $K_{a,b}$



Reduction factor

$ab/(a+b)$

Biclique Partitions



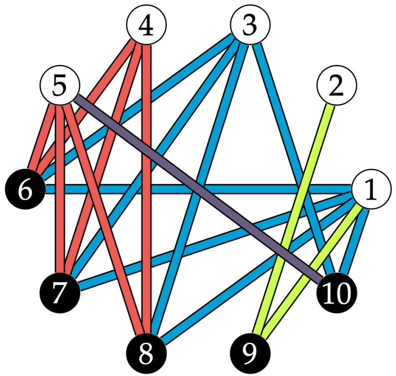
Biclique	Edges	Vertices
$K_{1,1}$	1	2
$K_{1,2}$	2	3
$K_{2,3}$	6	5
$K_{2,5}$	10	7
Total	19	**17**

Weight of the partition

Notation for Biclique Partitions

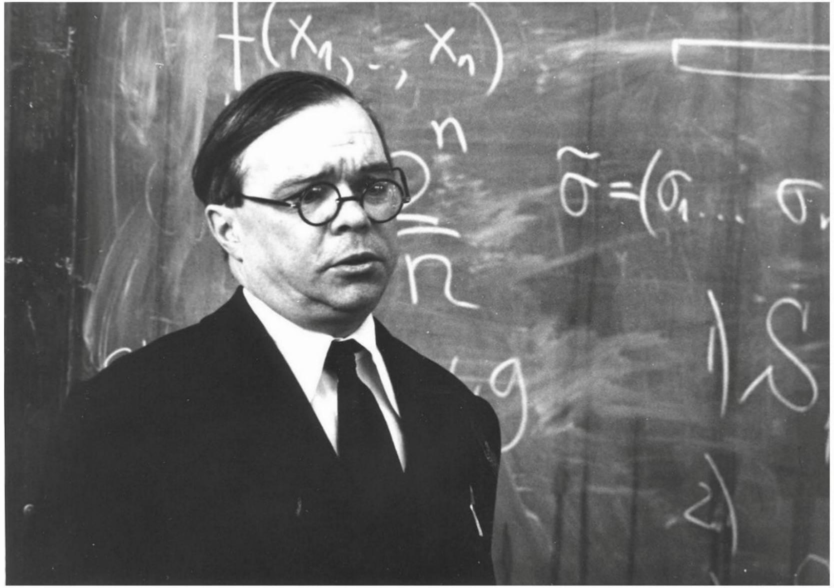
BP(G) := minimum weight of a biclique partition for G

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



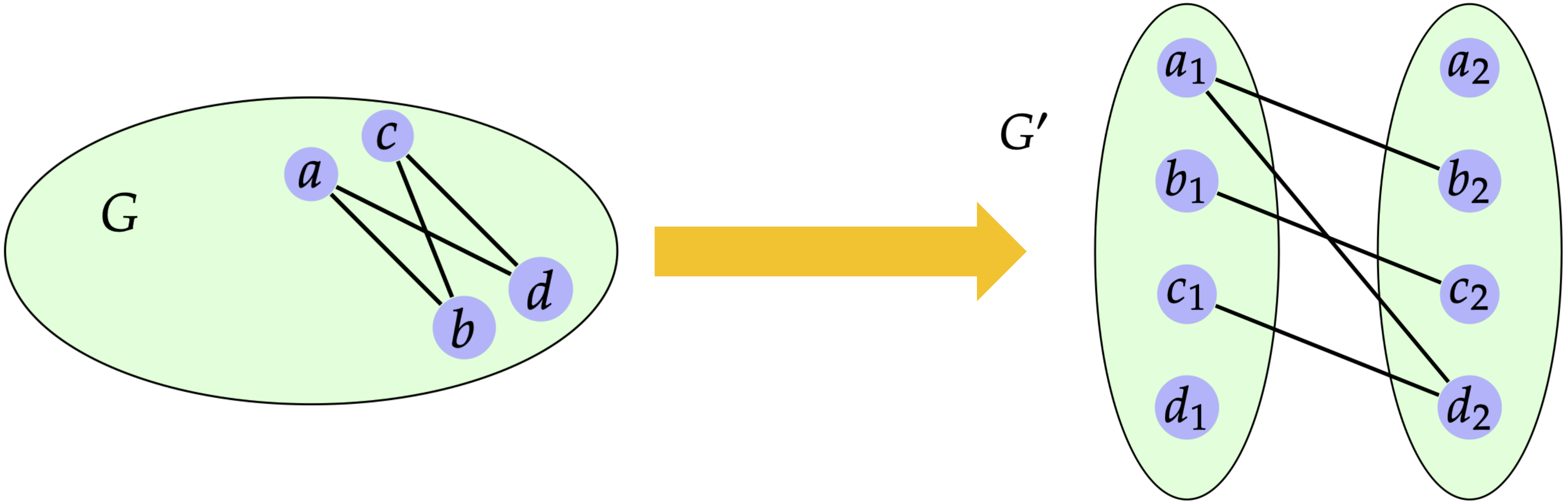
E.g., $BP(G) \leq 17$, $BP(n) \leq n^2$

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* $BP(G) \geq (1-o(1))\frac{n^2}{\lg n}$.

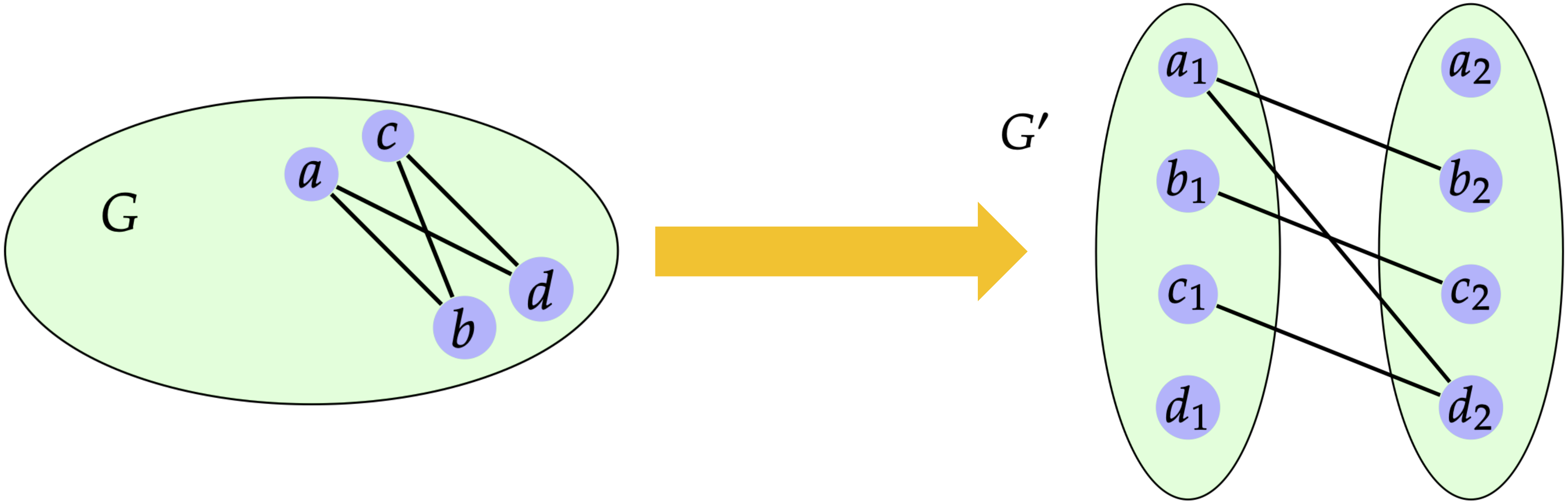


Oleg Borisovich Lupanov (1932-2006)

Lupanov for general graphs

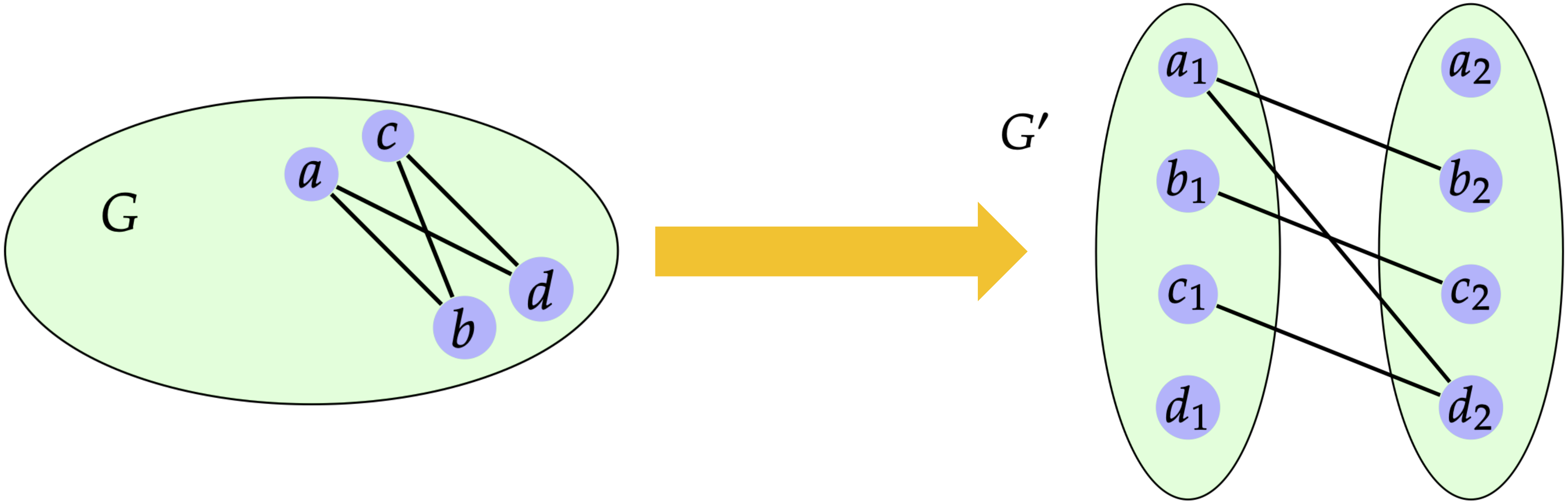


Lupanov for general graphs



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)

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

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

Upper bound: iteratively peel good bicliques

Chung–Erdős–Spencer (1983)

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

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

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

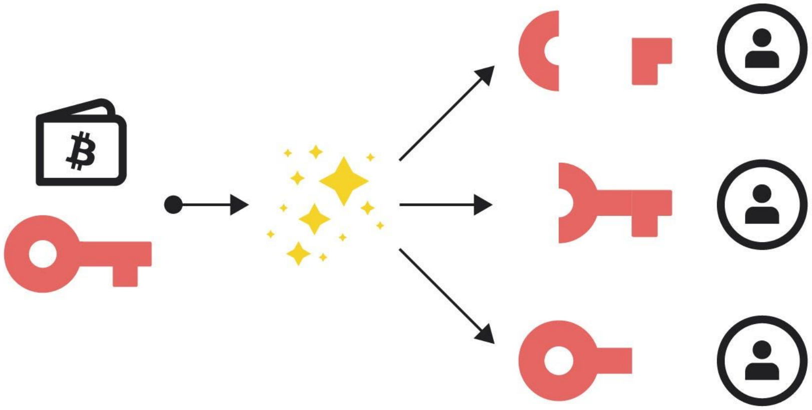
obviously.

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)

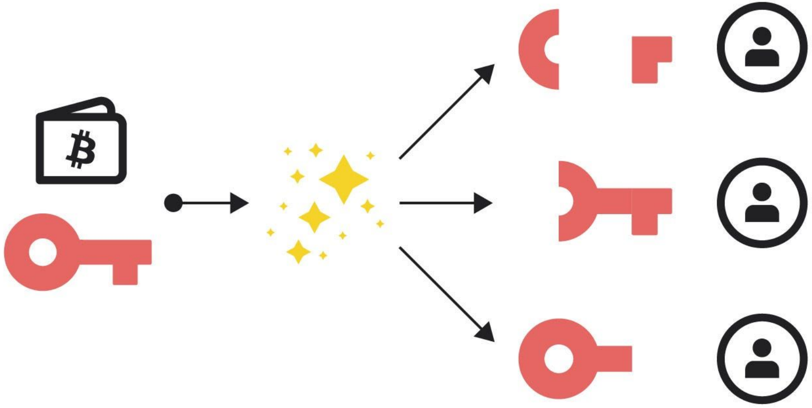
Theorem (Erdős–Pyber). *Every graph G on n vertices admits a biclique partition in which every vertex v belongs to $O(n/\lg n)$ bicliques.*

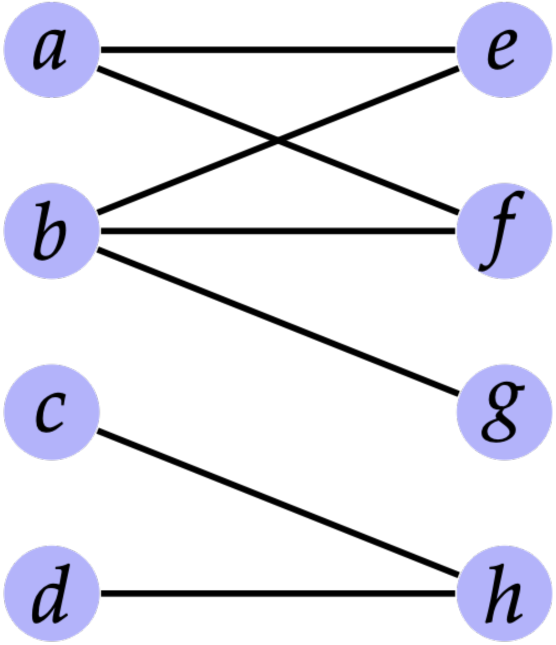


Important for secret sharing!

Erdős–Pyber (1997)

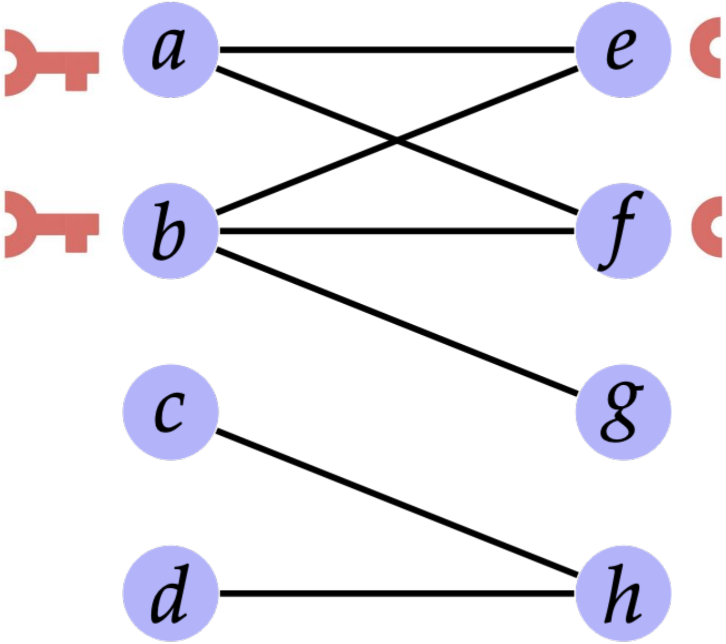
Theorem (Erdős–Pyber). *Every graph G on n vertices admits a biclique partition in which every vertex v belongs to $O(n/\lg n)$ bicliques.*





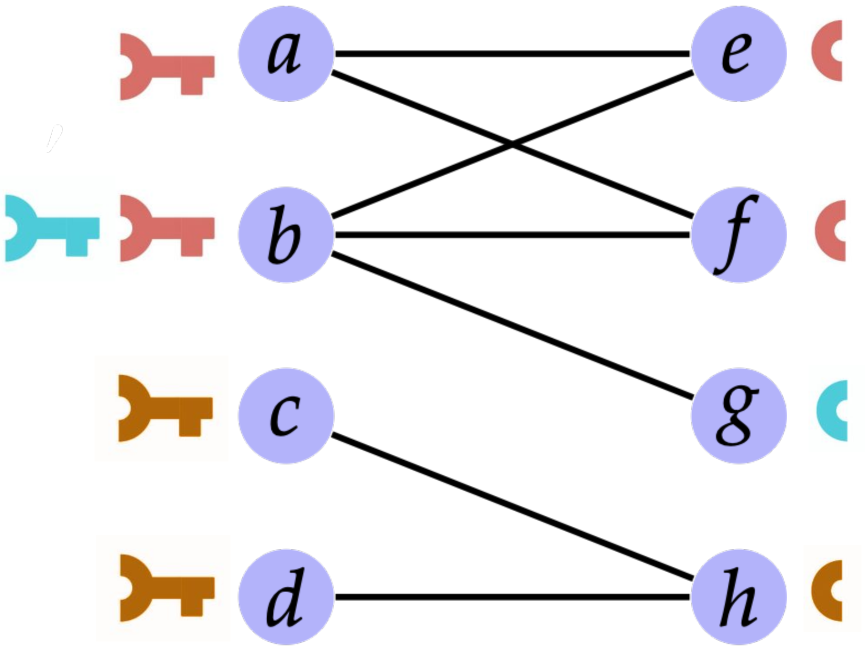
Erdős–Pyber (1997)

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 $\boldsymbol{BP(n) \leq n \cdot LBP(n)}$

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)).

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

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 · LBP(n)***

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

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

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

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

Our main result

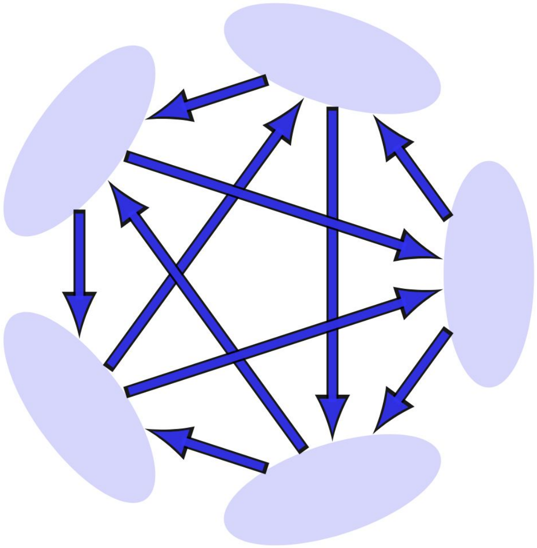
Theorem 1. *For d-uniform hypergraphs we have*

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

Graph Upper bound

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

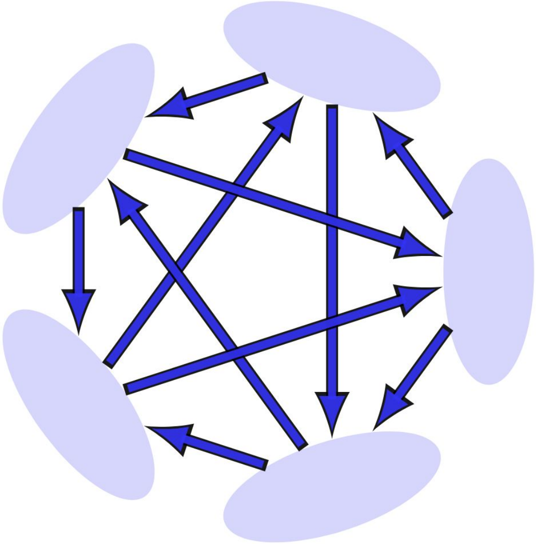
and set up an almost-regular tournament between parts



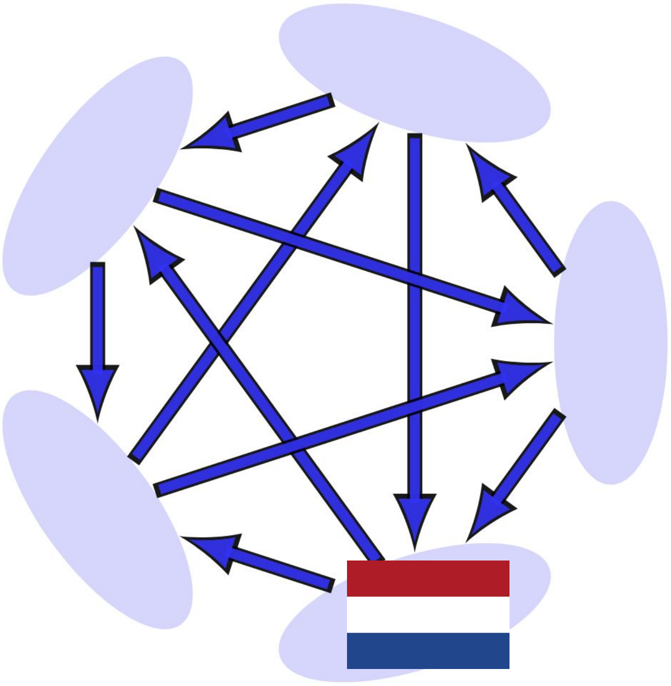
Graph Upper bound

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

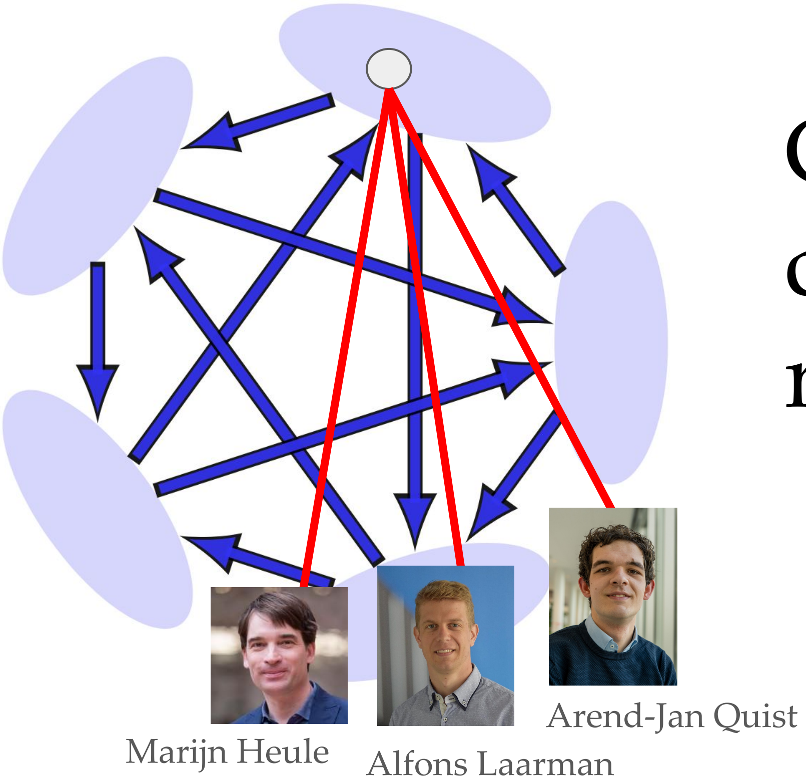
and set up an almost-regular tournament between parts



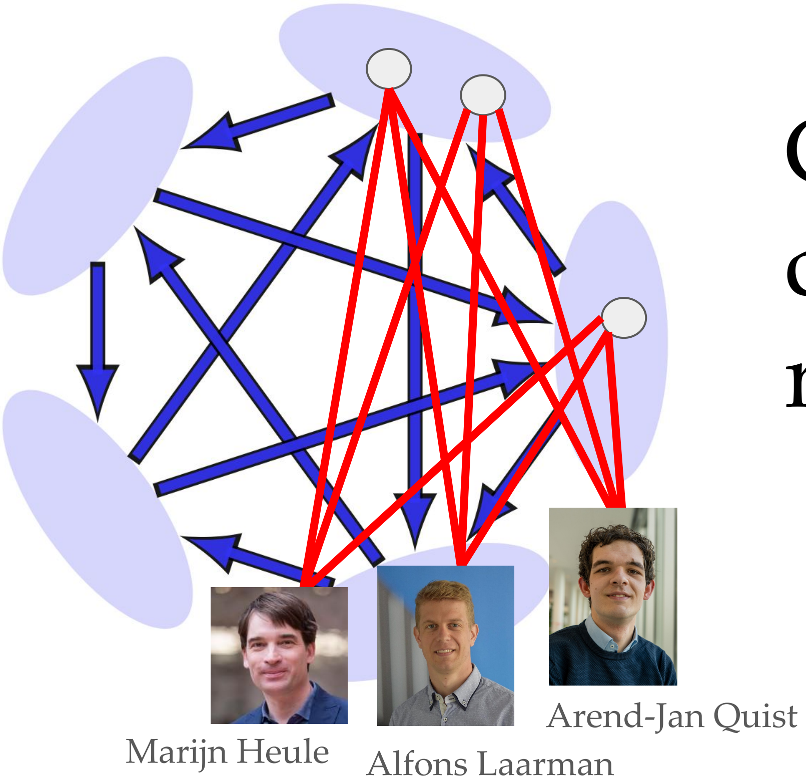
Step 2: Take each edge within a part as its own biclique.
Each vertex participates in at most r - $1 = o(n/ \lg n)$ of these.



Consider the graph of computer science researchers



Consider the graph of computer science researchers

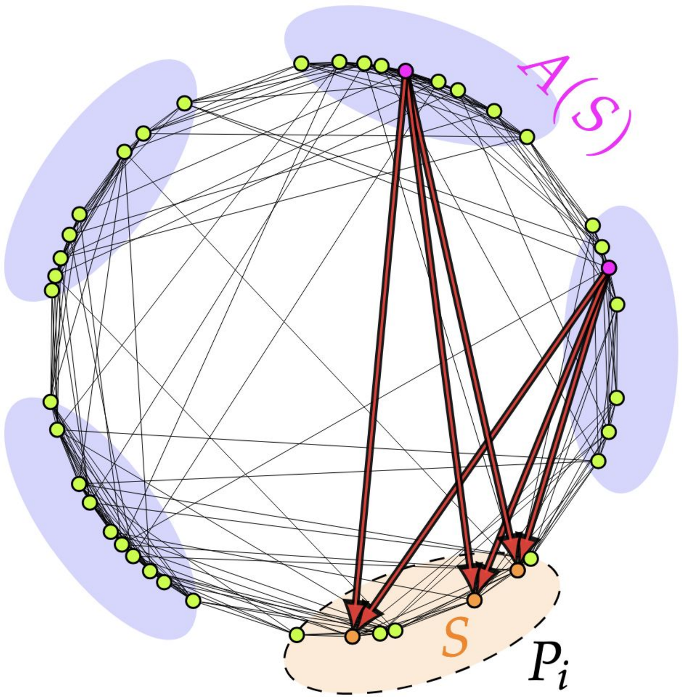


Consider the graph of computer science researchers

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*.

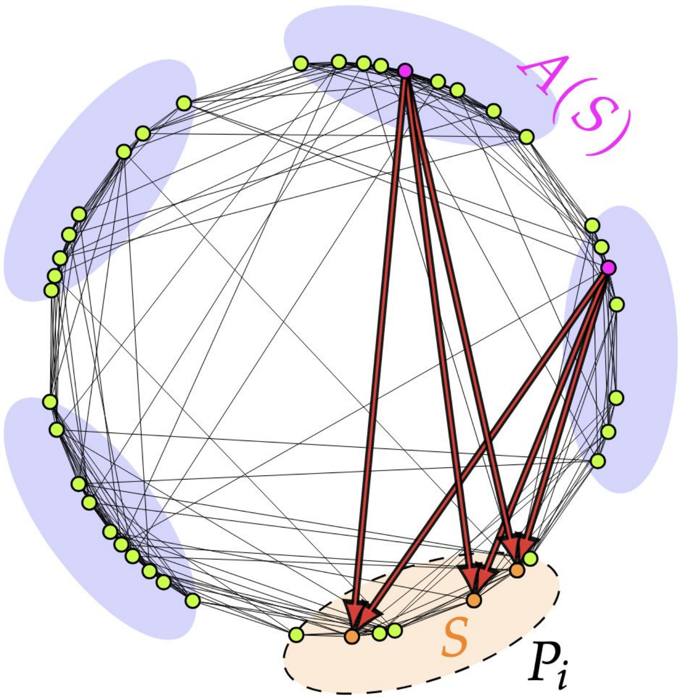


Graph Upper bound

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

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

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

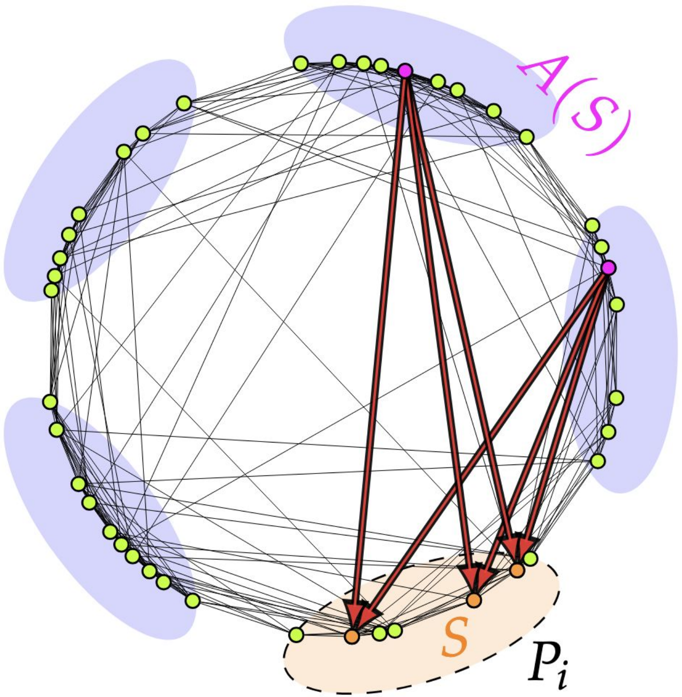
Q: In how many such bicliques does each vertex v participate?

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:

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

and thus:

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

Tight information-theoretic lower bound!

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 $(i < j$ and $d_n(i, j) \equiv 0 \pmod 2)$ or $(i > j$ 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 $(\{u\}, \{v\})$

Density-aware partitions

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

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

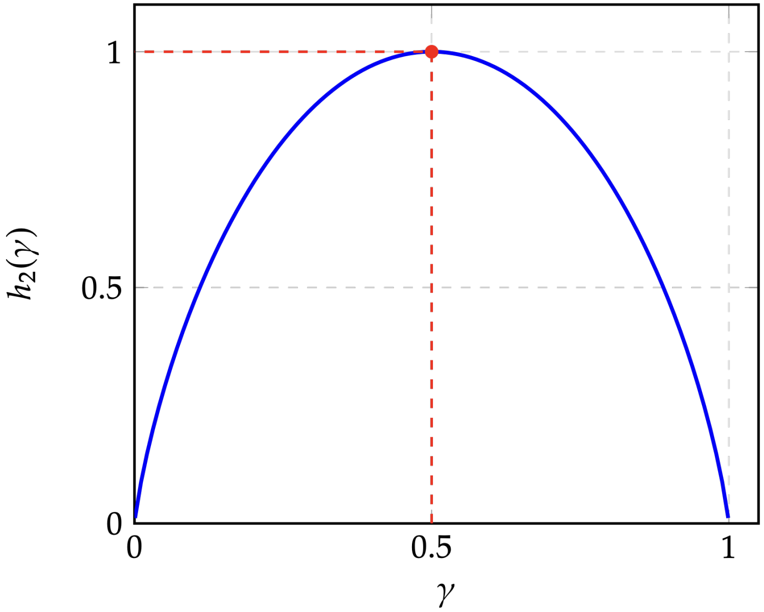
where $h_2(x) := -x\lg(x) - (1-x)\lg(1-x)$ *is the binary entropy function.*

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

Theorem *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 (*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

001110101000001000000011111111000001000010000100000001

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

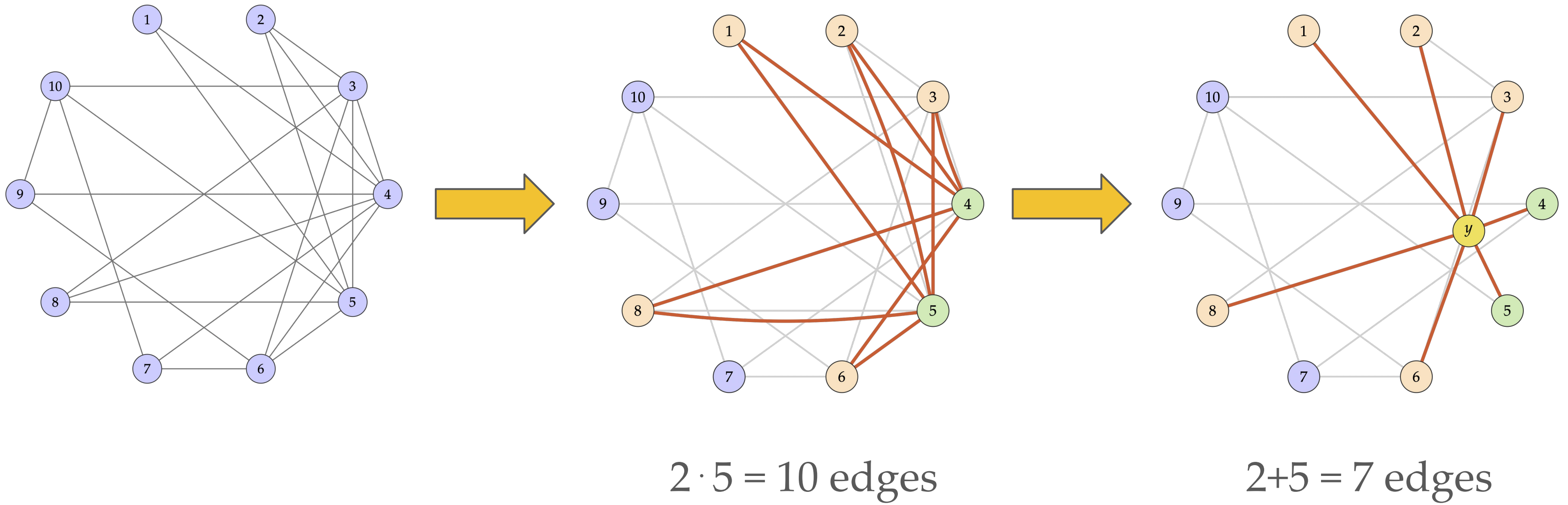
On top of splitting into equal-sized parts (now bigger), and setting up the almost-regular tournament, we view the adjacencies of vertex v into part P_i as a binary string like

001110101000001000000011111111000001000010000100000001

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

“Clique Partitions, Graph Compression and Speeding-Up Algorithms”

Feder & Motwani (STOC’ 91)



Graph representation result

	Runtime	Space
FM'91	$O(mn^{\delta} \lg n)$	$O\left(m \cdot \frac{\lg(1/\gamma)}{\delta}\right)$
Ours	$O(m)$	$m \cdot \frac{h_2(\gamma)}{\gamma}$

Optimal to a (1 + o(1)) factor!

Algorithmic applications

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

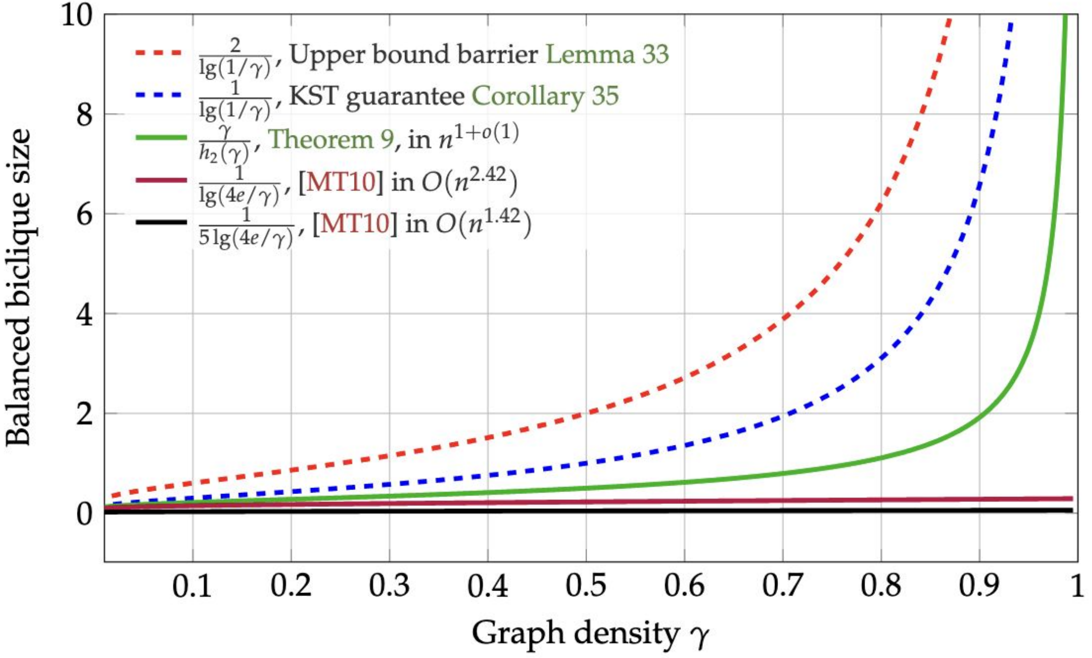
(*and* ***compact*** if we also store the vertex-biclique incidence list)

Theorem *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 $O(n^2/\lg \alpha)$.

Faster algorithm for finding density-guaranteed balanced bicliques



Prime example

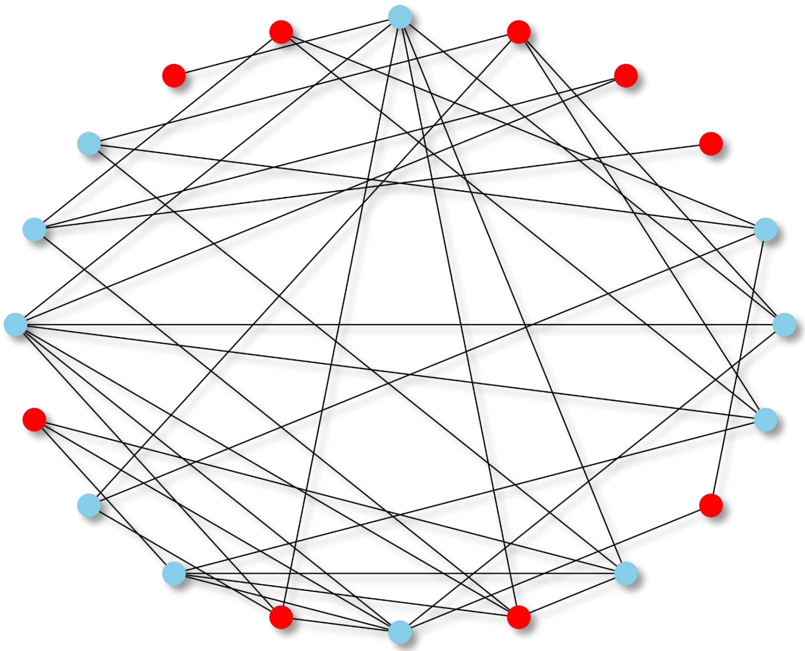
A = {161, 881, 8819, 19235, 21671, 27089, 31325, 50705, 133811, 261449, 269849, 411005, 531845}

B = {2, 188, 342, 558, 678, 4512, 33438, 45618, 48288, 94488, 140102, 285612, 366168}

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

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

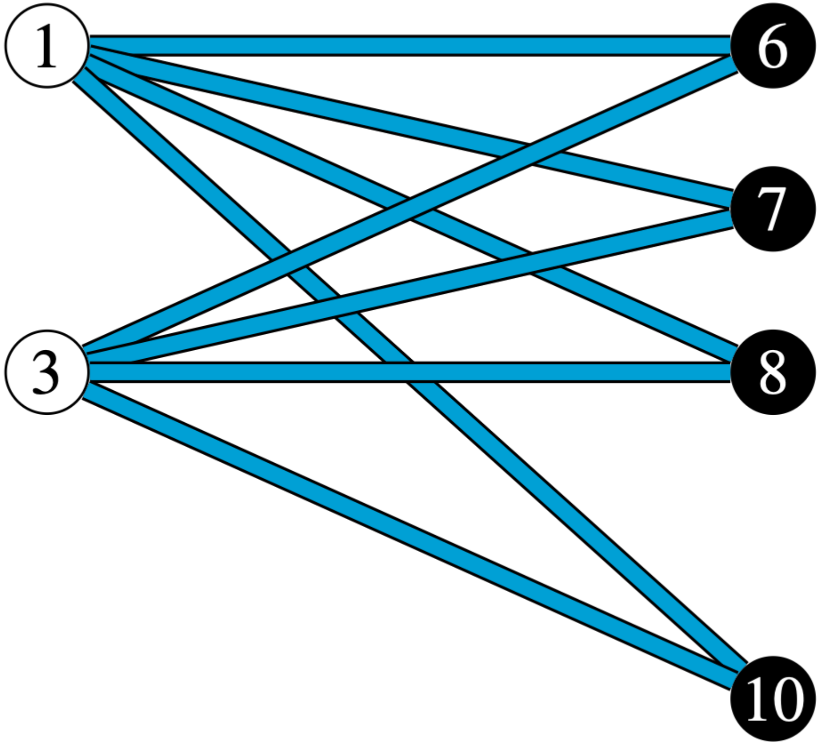
Graph coloring

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

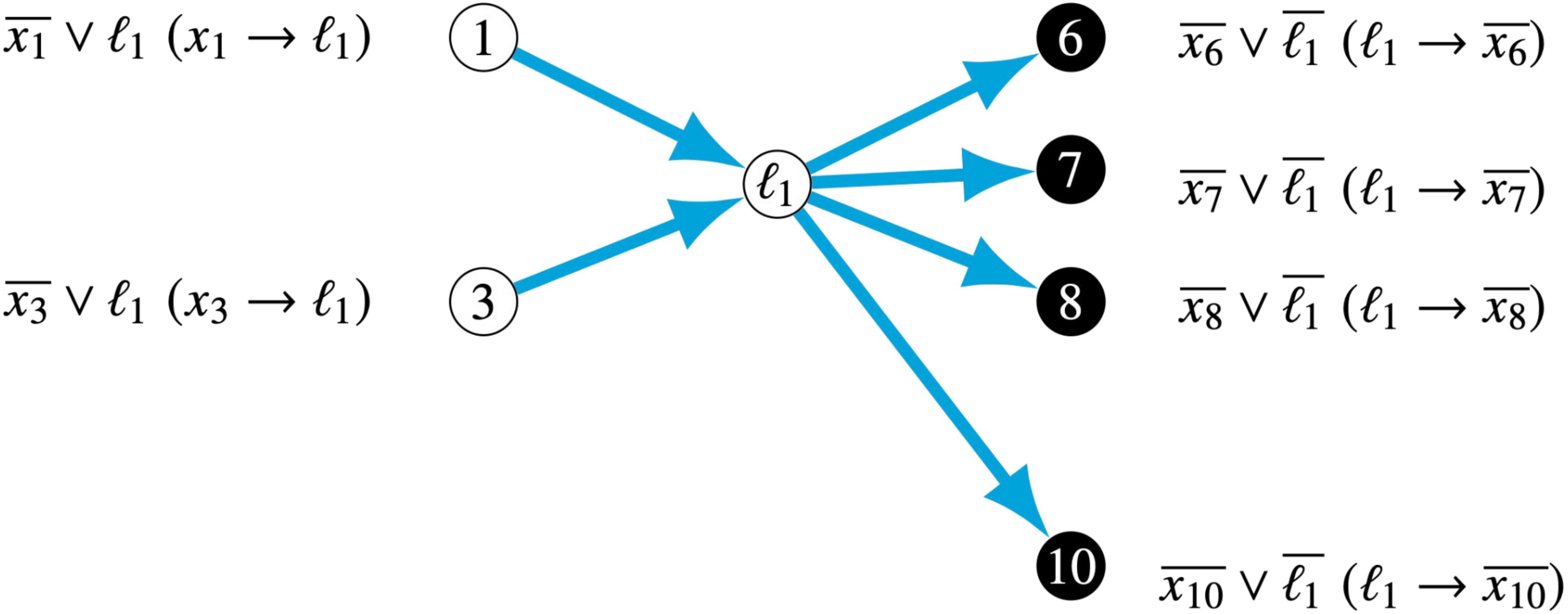
Suppose your graph is a biclique

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

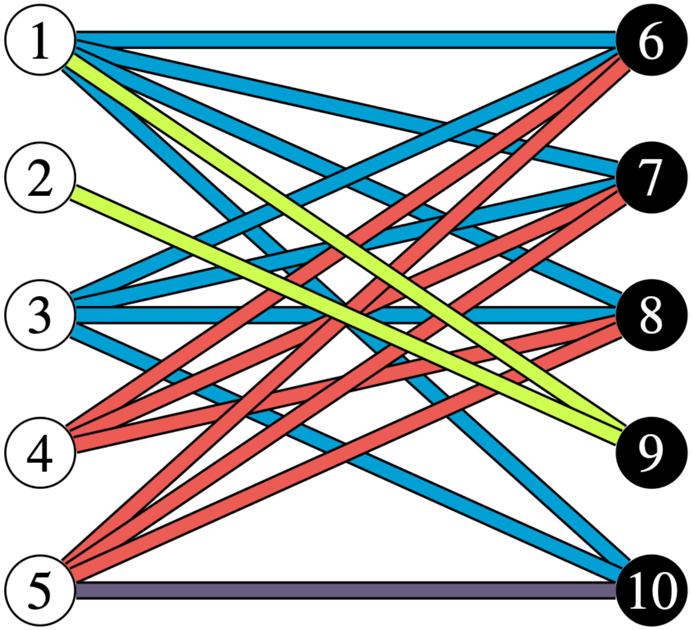
$\overline{x_3} \vee \overline{x_6}$
$\overline{x_3} \vee \overline{x_7}$
$\overline{x_3} \vee \overline{x_8}$
$\overline{x_3} \vee \overline{x_{10}}$



Suppose your graph is a biclique



Clauses ≤ BP(G)



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

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

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

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

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

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

Breaking down Graphs and Hypergraphs into Structured Pieces

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

Bernardo Subercaseaux (CMU)