How to decompose planar graphs?

Gwenaël Joret

Université libre de Bruxelles

Algorithms group at ULB

Jean Cardinal: Discrete and computational geometry, graph theory, information theory

Samuel Fiorini: Combinatorial optimization, polyhedral combinatorics, graph theory

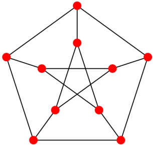
John Iacono: Data structures, computational geometry

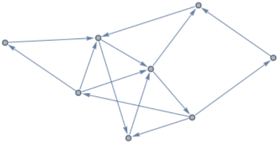
Gwenaël Joret: Graph theory, combinatorial optimization, partial orders

Stefan Langerman: Computational geometry, data structures

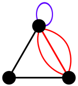
Graphs?

Graphs can be *undirected* or *directed*

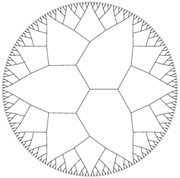




They can be *simple* or have *loops* and *parallel edges*



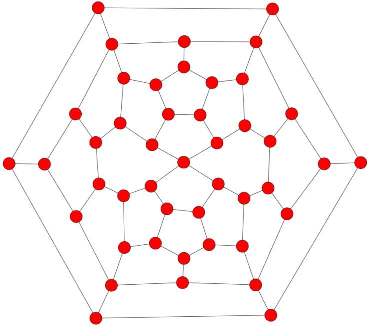
They can be *finite* or *infinite*

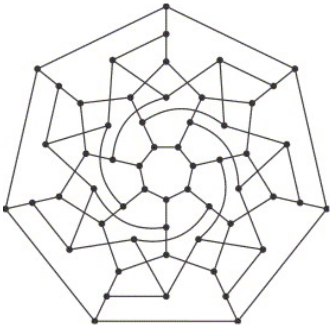


In this talk: "graph" = undirected finite simple graph

Planar graphs

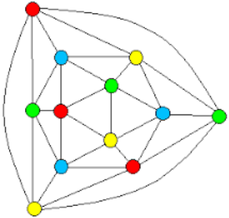
A graph is planar if it can be drawn in the plane without edge crossings





Classic results about planar graphs

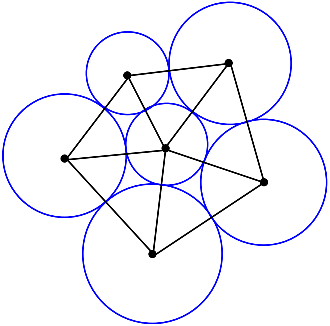
Four Color Theorem
Every planar graph can be colored using four colors



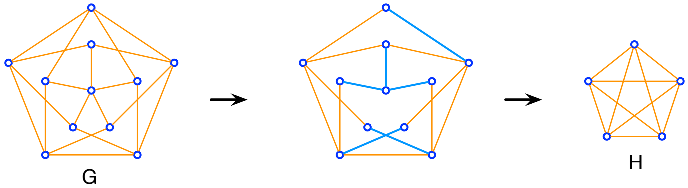
- Conjectured in 1852 by Francis Guthrie
- First computer-assisted proof by Appel and Haken (1970s)
- Second computer-assisted proof by Roberston, Sander, Seymour, Thomas (1996)
- Formal proof using Coq by Gonthier (2008)

Circle Packing Theorem

Every planar graph admits a "kissing coins" representation



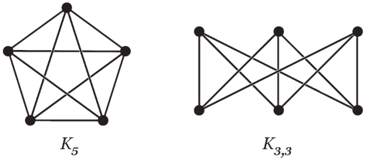
- First proved by Koebe (1936)
- Rediscovered and generalized by Thurston (1980s)



H minor of G if H can be obtained from a subgraph of G by contracting edges

Kuratowski - Wagner 1930s

A graph G is planar $\Leftrightarrow$ G contains neither K_5 nor $K_{3,3}$ as minor



Graph Minor Theorem

Graph property $\mathcal{P}$

$\mathcal{P}$ is minor-closed if G has property $\mathcal{P} \Rightarrow$ all minors of G have property $\mathcal{P}$

Graph Minor Theorem

Graph property $\mathcal{P}$

$\mathcal{P}$ is minor-closed if G has property $\mathcal{P} \Rightarrow$ all minors of G have property $\mathcal{P}$

Robertson & Seymour 1970s–2004
For every graph property $\mathcal{P}$ there exists a finite set $F_{\mathcal{P}}$ of graphs s.t. for every graph G: G has property $\mathcal{P} \Leftrightarrow G$ has no minor in $F_{\mathcal{P}}$

Graph Minor Theorem

Graph property $\mathcal{P}$

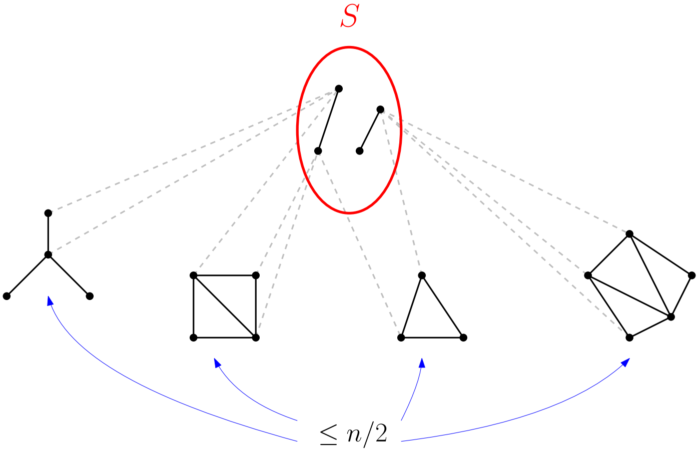
$\mathcal{P}$ is minor-closed if G has property $\mathcal{P} \Rightarrow$ all minors of G have property $\mathcal{P}$

Robertson & Seymour 1970s–2004
For every graph property $\mathcal{P}$ there exists a finite set $F_{\mathcal{P}}$ of graphs s.t. for every graph G: G has property $\mathcal{P} \Leftrightarrow G$ has no minor in $F_{\mathcal{P}}$

Proof over 600 pages

Results in existence of a polynomial-time algorithm for testing whether G has property $\mathcal{P}$

Separators

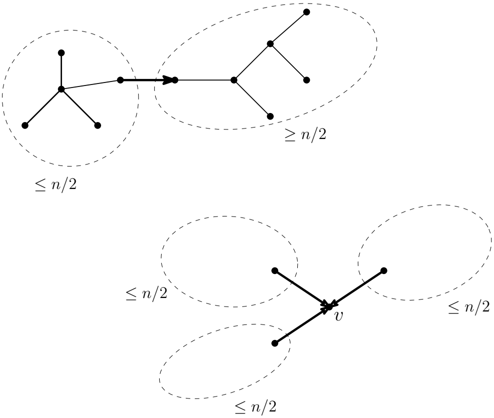


A separator of an n-vertex graph G is a vertex subset S such that every connected component of $G - S$ has $\leqslant n/2$ vertices

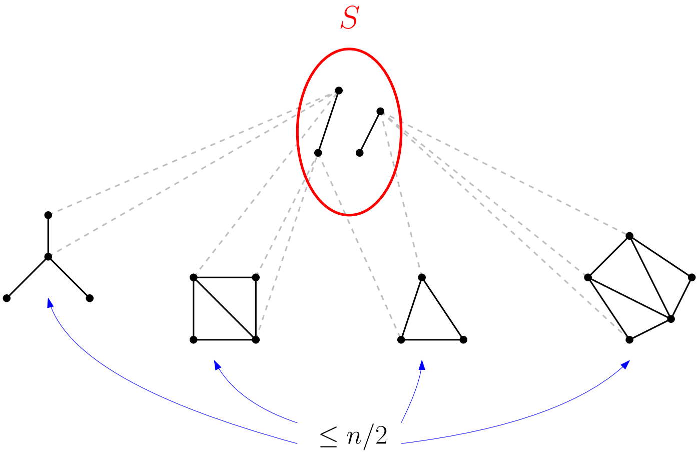
Separators in trees

T n-vertex tree

Fact: $\exists$ vertex v which is a separator of T



Separators in planar graphs

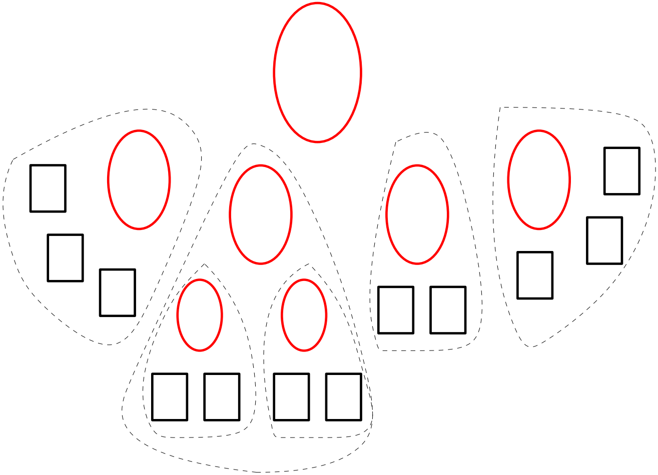


Lipton & Tarjan 1979

Every n-vertex planar graph has a separator of size $O(\sqrt{n})$

Using Lipton-Tarjan separators

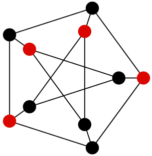
Keep decomposing until each piece has size $\leqslant k$



$$\left|\bigcup \text{separators}\right| = O\left(\frac{n}{\sqrt{k}}\right)$$

Example: Maximum Independent Set problem

Independent set: Set of vertices, no two of which are adjacent

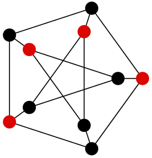


Maximum Independent Set problem on n-vertex graph G:

- NP-hard

Example: Maximum Independent Set problem

Independent set: Set of vertices, no two of which are adjacent

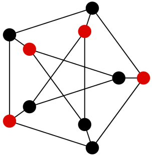


Maximum Independent Set problem on n-vertex graph G:

- NP-hard
- NP-hard to find a solution of size $\geqslant \varepsilon \cdot OPT \quad \forall \varepsilon > 0$

Example: Maximum Independent Set problem

Independent set: Set of vertices, no two of which are adjacent

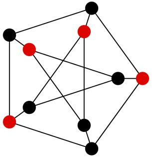


Maximum Independent Set problem on n-vertex graph G:

- NP-hard
- NP-hard to find a solution of size $\geqslant \varepsilon \cdot OPT \quad \forall \varepsilon > 0$
- NP-hard to find a solution of size $\geqslant \frac{1}{n^{1-\varepsilon}} \cdot OPT \quad \forall \varepsilon > 0$ (Zuckerman, 2007)

Example: Maximum Independent Set problem

Independent set: Set of vertices, no two of which are adjacent

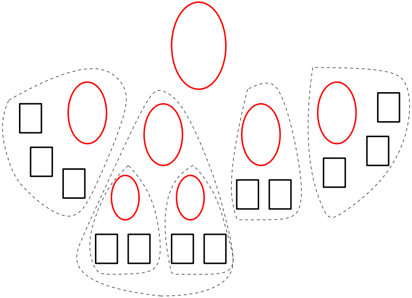


Maximum Independent Set problem on n-vertex graph G:

- NP-hard
- NP-hard to find a solution of size $\geqslant \varepsilon \cdot OPT \quad \forall \varepsilon > 0$
- NP-hard to find a solution of size $\geqslant \frac{1}{n^{1-\varepsilon}} \cdot OPT \quad \forall \varepsilon > 0$ (Zuckerman, 2007)

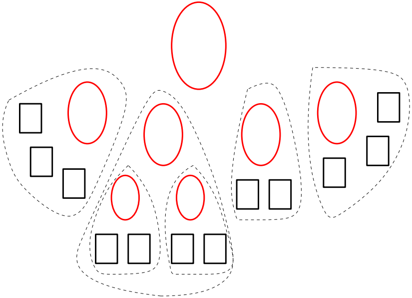
If G is planar: Problem is still NP-hard but ...

Keep decomposing until each piece has size $\leqslant k$



$$\left|\bigcup \text{separators}\right| = O\left(\frac{n}{\sqrt{k}}\right)$$

Keep decomposing until each piece has size $\leqslant k$



$$\left|\bigcup \text{separators}\right| = O\left(\frac{n}{\sqrt{k}}\right)$$

Maximum Independent Set problem: $OPT \geqslant n/4$

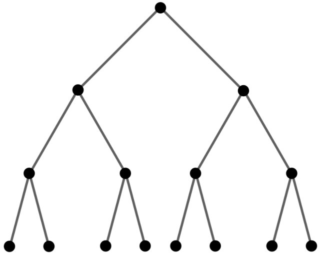
take $k = \log n$, solve problem exactly on each piece

discard $\bigcup \text{separators}$, of size $O\left(\frac{n}{\sqrt{\log n}}\right)$

$\Rightarrow$ solution has size $\geqslant OPT - O\left(\frac{n}{\sqrt{\log n}}\right) \geqslant \left(1 - \frac{c}{\sqrt{\log n}}\right) OPT$

for some $c > 0$

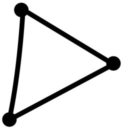
Independent sets in trees



Maximum Independent Set can be solved in polynomial time on trees using dynamic programming

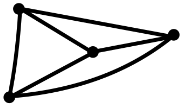
k-Trees

Inductive definition of k-trees (illustrated for $k = 3$):



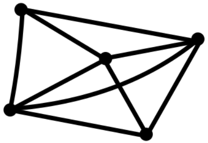
k-Trees

Inductive definition of k-trees (illustrated for $k = 3$):



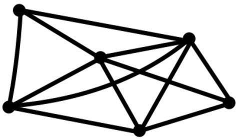
k-Trees

Inductive definition of k-trees (illustrated for $k = 3$):



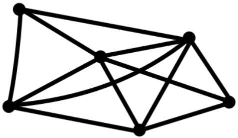
k-Trees

Inductive definition of k-trees (illustrated for $k = 3$):



k-Trees

Inductive definition of k-trees (illustrated for $k = 3$):



Treewidth of G: Smallest k s.t. G subgraph of a k-tree

Treewidth is a measure of similarity with a tree (the lower the better)

Most algorithmic problems can be solved on polynomial time on graphs with bounded treewidth using dynamic programming (Maximum Independent Set, Minimum Coloring, Traveling Salesman Problem, ...)

Treewidth

Lemma: If G has treewidth $\leqslant k$ then G has separator S of size $\leqslant k$

Treewidth

Lemma: If G has treewidth $\leqslant k$ then G has separator S of size $\leqslant k$

Dvořák, Norin 2019
If all subgraphs of G have separators of size $\leqslant k$ then G has treewidth $\leqslant 15k$

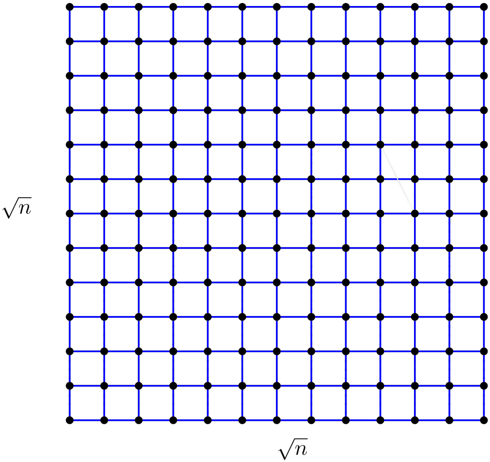
Treewidth

Lemma: If G has treewidth $\leqslant k$ then G has separator S of size $\leqslant k$

Dvořák, Norin 2019
If all subgraphs of G have separators of size $\leqslant k$ then G has treewidth $\leqslant 15k$

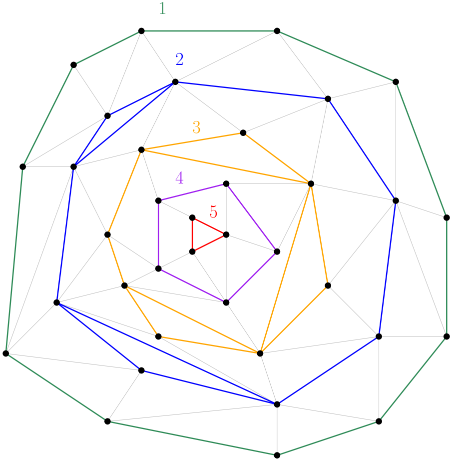
small treewidth $\Leftrightarrow$ **small separators**

Planar graphs can have large treewidth



treewidth $\sqrt{n}$

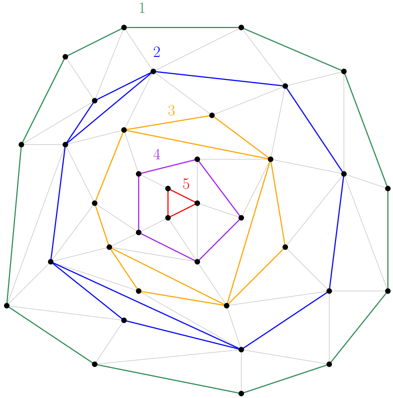
Baker's technique (1994)



Baker + Eppstein 1990s

Union of ℓ consecutive layers has treewidth $\leqslant 3\ell$

Baker's technique (1994)

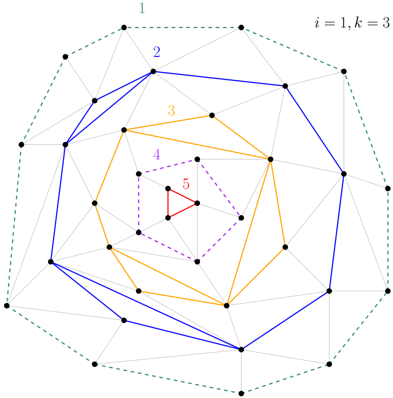


Remove all layers numbered $i \bmod k$

Choose i so that $\leqslant n/k$ vertices are removed

Solve problem on remaining graph, which has treewidth $O(k)$

Baker's technique (1994)

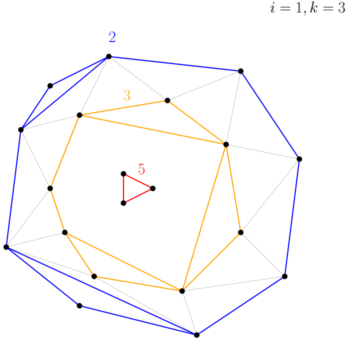


Remove all layers numbered i **mod** k

Choose i so that $\leqslant n/k$ vertices are removed

Solve problem on remaining graph, which has treewidth $O(k)$

Baker's technique (1994)

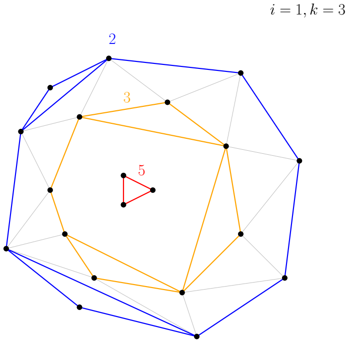


Remove all layers numbered i mod k

Choose i so that $\leqslant n/k$ vertices are removed

Solve problem on remaining graph, which has treewidth $O(k)$

Baker's technique (1994)



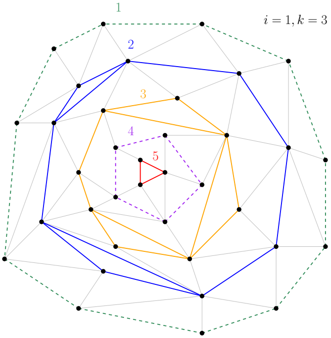
Remove all layers numbered $i \bmod k$

Choose i so that $\leqslant n/k$ vertices are removed

Solve problem on remaining graph, which has treewidth $O(k)$

With $k = \log n$, this gives a solution of size $\geqslant \left(1 - \frac{c}{\log n}\right) OPT$ for Maximum Independent Set

Baker $\Rightarrow$ Lipton-Tarjan



Take $k = \sqrt{n}$, choose i so that $\leqslant n/k = \sqrt{n}$ vertices are removed

Remaining graph has treewidth $O(k) = O(\sqrt{n})$

Take a separator S' of size $O(\sqrt{n})$ in remaining graph

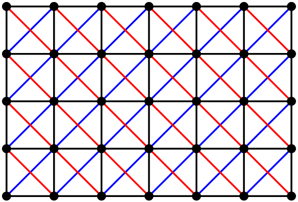
Union of vertices removed and S' is a separator of size $O(\sqrt{n})$

A new way of decomposing planar graphs

Dujmović, J., Micek, Morin, Ueckerdt, Wood 2019
Every planar graph is a subgraph of $H \boxtimes P$ for some graph H with treewidth $\leqslant 8$ and some path P

$\boxtimes$

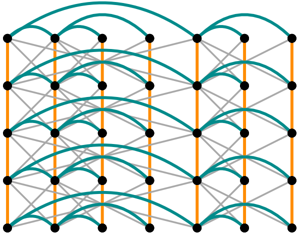
$=$

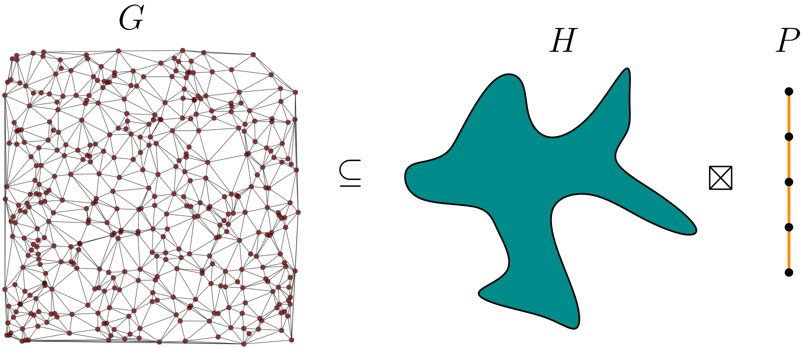


A new way of decomposing planar graphs

Dujmović, J., Micek, Morin, Ueckerdt, Wood 2019
Every planar graph is a subgraph of $H \boxtimes P$ for some graph H with treewidth $\leqslant 8$ and some path P

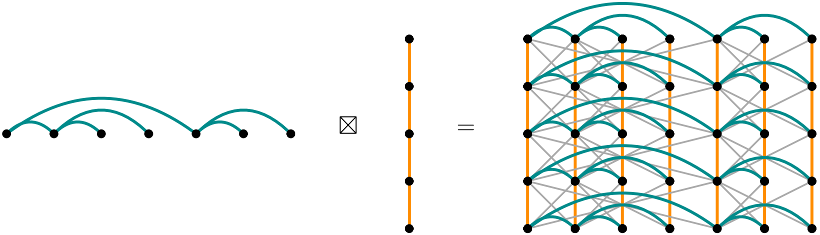

$\boxtimes$

$=$




G
$\subseteq$
H
$\boxtimes$
P

Product structure $\Rightarrow$ Baker



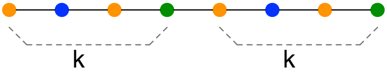
Union of ℓ consecutive layers has treewidth $\leqslant 9\ell$

Applications of product structure

- Queue-numbers
- Nonrepetitive coloring
- p-centered coloring
- Subgraph isomorphism
- Extensions to other graph classes (bounded genus, k-planar, $\cdots$)
- Adjacency labeling schemes
- $\cdots$

New research direction, lots to explore

Application: Nonrepetitive colorings

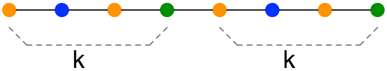


repetitively colored path

Vertex coloring nonrepetitive if $\nexists$ repetitively colored paths

Conjecture (Alon, Grytczuk, Hałuszczak, Riordan 2002)
Planar graphs have bounded nonrepetitive chromatic number

Application: Nonrepetitive colorings



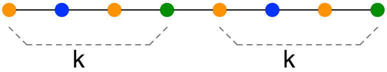
repetitively colored path

Vertex coloring nonrepetitive if $\nexists$ repetitively colored paths

Conjecture (Alon, Grytczuk, Hałuszczak, Riordan 2002)
Planar graphs have bounded nonrepetitive chromatic number

Kündgen & Pelsmayer 2008
If G has treewidth $\leqslant k$ then G has nonrepetitive coloring with 4^k colors

Application: Nonrepetitive colorings



repetitively colored path

Vertex coloring nonrepetitive if $\nexists$ repetitively colored paths

Conjecture (Alon, Grytczuk, Hałuszczak, Riordan 2002)
Planar graphs have bounded nonrepetitive chromatic number

Kündgen & Pelsmayer 2008
If G has treewidth $\leqslant k$ then G has nonrepetitive coloring with 4^k colors

Dujmović, Esperet, J., Walczak, Wood 2019
Planar graphs have nonrepetitive chromatic number $\leqslant 768$

Application: Adjacency labelings

Each vertex receives a unique label (bitstring) s.t. one can decide whether v and w are adjacent just based on their labels

Conjecture (Kannan, Naor, and Rudich 1988)
Can do labels with $\log_2 n + o(\log n)$ bits for n-vertex planar graphs

Application: Adjacency labelings

Each vertex receives a unique label (bitstring) s.t. one can decide whether v and w are adjacent just based on their labels

Conjecture (Kannan, Naor, and Rudich 1988)
Can do labels with $\log_2 n + o(\log n)$ bits for n-vertex planar graphs

Known upper bounds on label sizes:

- $4 \log_2 n$ (Kannan, Naor, and Rudich, 1988)
- $3 \log_2 n + o(\log n)$ (Chung, 1990)
- $2 \log_2 n + o(\log n)$ (Gavoille & Labourel, 2007)
- $\frac{4}{3} \log_2 n + o(\log n)$ (Bonamy, Gavoille, Pilipczuk, 2020)

Application: Adjacency labelings

Each vertex receives a unique label (bitstring) s.t. one can decide whether v and w are adjacent just based on their labels

Conjecture (Kannan, Naor, and Rudich 1988)
Can do labels with $\log_2 n + o(\log n)$ bits for n-vertex planar graphs

Known upper bounds on label sizes:

- $4 \log_2 n$ (Kannan, Naor, and Rudich, 1988)
- $3 \log_2 n + o(\log n)$ (Chung, 1990)
- $2 \log_2 n + o(\log n)$ (Gavoille & Labourel, 2007)
- $\frac{4}{3} \log_2 n + o(\log n)$ (Bonamy, Gavoille, Pilipczuk, 2020)

Application: Adjacency labelings

Each vertex receives a unique label (bitstring) s.t. one can decide whether v and w are adjacent just based on their labels

Conjecture (Kannan, Naor, and Rudich 1988)
Can do labels with $\log_2 n + o(\log n)$ bits for n-vertex planar graphs

Known upper bounds on label sizes:

- $4 \log_2 n$ (Kannan, Naor, and Rudich, 1988)
- $3 \log_2 n + o(\log n)$ (Chung, 1990)
- $2 \log_2 n + o(\log n)$ (Gavoille & Labourel, 2007)
- $\frac{4}{3} \log_2 n + o(\log n)$ (Bonamy, Gavoille, Pilipczuk, 2020)

Dujmović, Esperet, Gavoille, J., Micek, Morin, Wood 2021
Can do labels with $\log_2 n + o(\log n)$ bits for n-vertex planar graphs

Thank you!