Local structure for vertex-minors

Rose McCarty

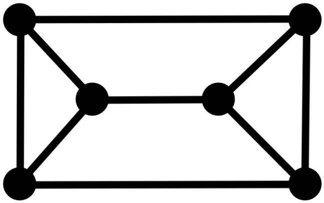
PRINCETON UNIVERSITY

October 17th, 2022

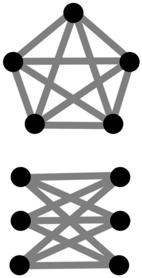
Joint work with Jim Geelen and Paul Wollan.

Kuratowski's Theorem

A graph is planar iff it has no K_5 or $K_{3,3}$ minor.



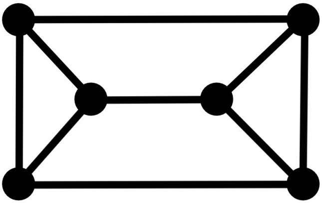
planar graphs



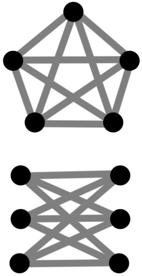
forbidden minors

Kuratowski's Theorem

A graph is planar iff it has no K_5 or $K_{3,3}$ minor.



planar graphs



forbidden minors

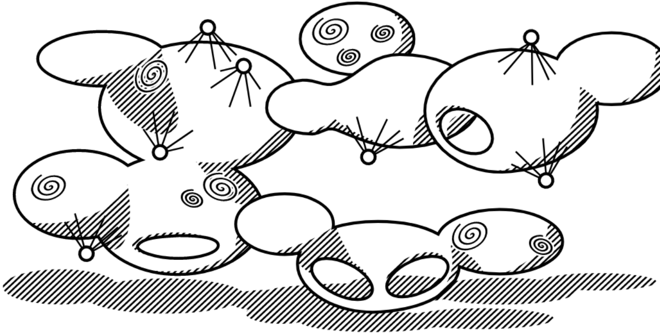
Graph Minors Theorem (Robertson & Seymour 2004)

Every minor-closed class has finitely many forbidden minors.

Theorem (Robertson & Seymour 2003)

The graphs in any proper minor-closed class "decompose" into parts that "almost embed" in a surface of bounded genus.

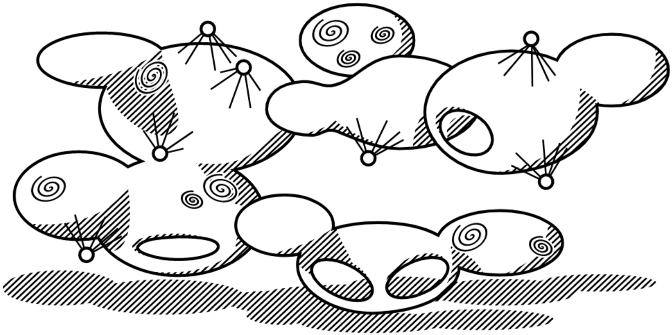
Figure by Felix Reidl



Theorem (Robertson & Seymour 2003)

The graphs in any proper minor-closed class "decompose" into parts that "almost embed" in a surface of bounded genus.

Figure by Felix Reidl

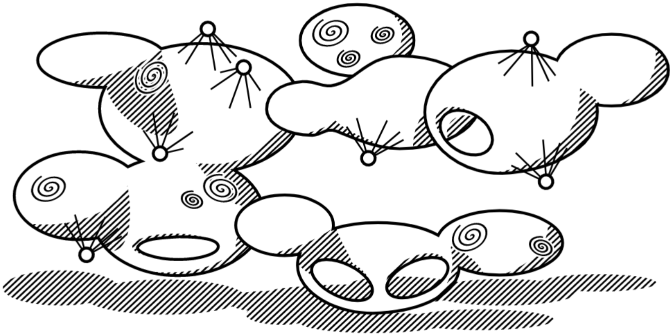


minors $\longrightarrow$ **vertex-minors**

Theorem (Robertson & Seymour 2003)

The graphs in any proper minor-closed class "decompose" into parts that "almost embed" in a surface of bounded genus.

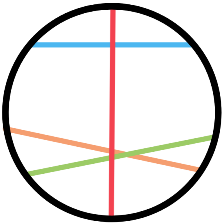
Figure by Felix Reidl



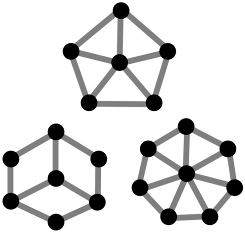
planar graphs $\longrightarrow$ **circle graphs**

Bouchet's Theorem

A graph is a **circle graph** *iff it has no* W_5, $\hat{W}_6$, *or* W_7 **vertex-minor**.



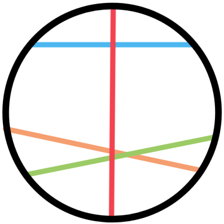
circle graphs



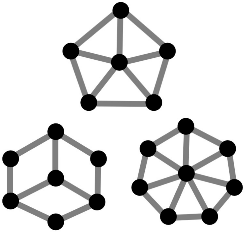
forbidden vertex-minors

Bouchet's Theorem

A graph is a **circle graph** *iff it has no W_5, $\hat{W}_6$, or W_7* **vertex-minor**.



circle graphs



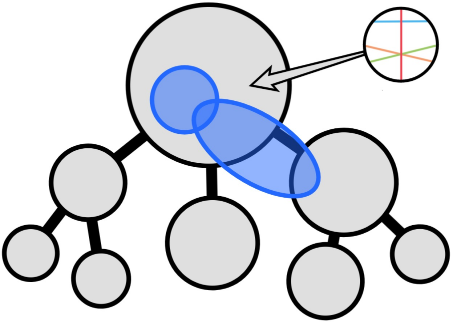
forbidden vertex-minors

Conjecture (Oum 2017)

Every **vertex-minor***-closed class has finitely many forbidden vertex-minors.*

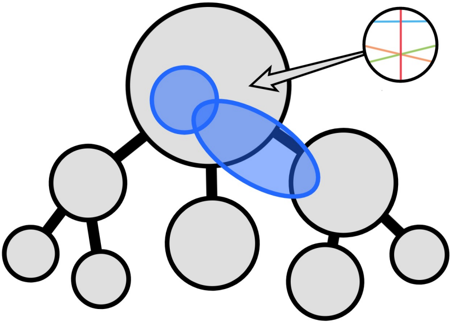
Conjecture (Geelen)

The graphs in any proper **vertex-minor***-closed class "decompose" into parts that are "almost"* **circle graphs***.*



Conjecture (Geelen)

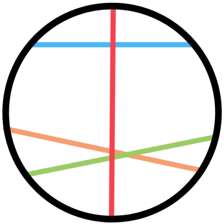
The graphs in any proper **vertex-minor***-closed class "decompose" into parts that are "almost"* **circle graphs***.*



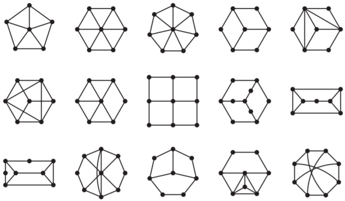
Ongoing project with Jim Geelen & Paul Wollan aiming to prove the conjecture.

Geelen and Oum's Theorem

A graph is a **circle graph** *iff it has no* $W_5, W_6, \ldots$ **pivot-minor***.*



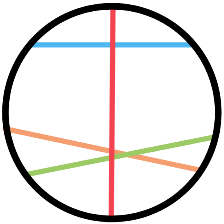
circle graphs



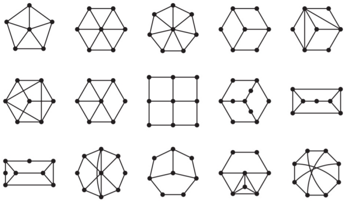
forbidden pivot-minors

Geelen and Oum's Theorem

A graph is a **circle graph** *iff it has no* $W_5, W_6, \ldots$ **pivot-minor**.



circle graphs



forbidden pivot-minors

Conjecture (Oum 2017)

Every **pivot-minor**-*closed class has finitely many forbidden pivot-minors.*

Geelen and Oum's Theorem

A graph is a **circle graph** *iff it has no* $W_5, W_6, \ldots$ **pivot-minor**.

Common generalization!

(Bouchet 1988; de Fraysseix 1981)

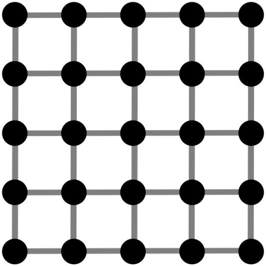
Conjecture (Oum 2017)

Every **pivot-minor***-closed class has finitely many forbidden pivot-minors.*

Grid Theorem (Robertson & Seymour 1986)

For any planar graph H, every graph with tree-width $\geq f(H)$ has a minor isomorphic to H.

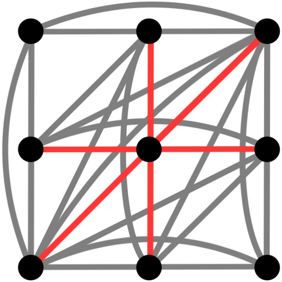
If you cannot "decompose away the whole graph", then there is a big grid as a minor.



Theorem (Geelen, Kwon, McCarty, & Wollan 2020)

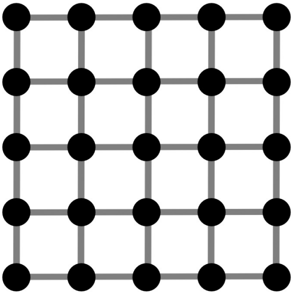
For any **circle graph** H*, every graph with* **rank-width** $\geq f(H)$ *has a* **vertex-minor** *isomorphic to* H.

If you cannot "rank-decompose away the whole graph", then there is a big **comparability grid** as a vertex-minor.



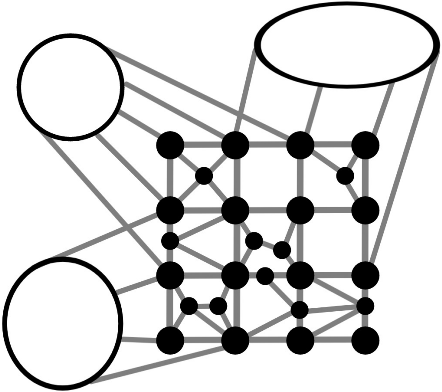
Flat Wall Theorem (Robertson & Seymour 1995)

For any proper minor-closed class $\mathcal{F}$ and any $G \in \mathcal{F}$ with a large grid minor, there is a planar subgraph containing a lot of the grid so that the rest of G "almost attaches" onto just the outer face.



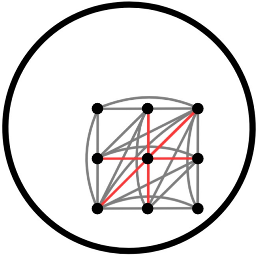
Flat Wall Theorem (Robertson & Seymour 1995)

For any proper minor-closed class $\mathcal{F}$ and any $G \in \mathcal{F}$ with a large grid minor, there is a planar subgraph containing a lot of the grid so that the rest of G "almost attaches" onto just the outer face.



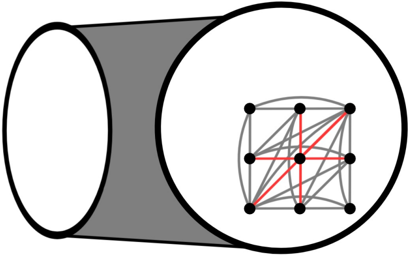
Local Structure Theorem (Geelen, McCarty, & Wollan)

For any proper **vertex-minor**-*closed class* $\mathcal{F}$ *and any* $G \in \mathcal{F}$ *with a prime* **circle graph** *containing a* **comparability grid**, *the rest of* G *"almost attaches" in a way that is "mostly compatible".*



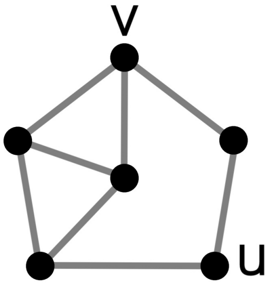
Local Structure Theorem (Geelen, McCarty, & Wollan)

For any proper **vertex-minor***-closed class* $\mathcal{F}$ *and any* $G \in \mathcal{F}$ *with a prime* **circle graph** *containing a* **comparability grid***, the rest of* G *"almost attaches" in a way that is "mostly compatible".*



The **vertex-minors** of a graph G are the graphs that can be obtained from G by

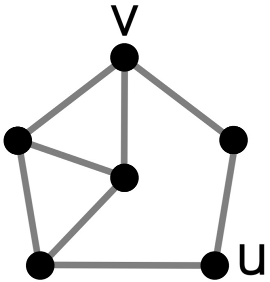
1) vertex deletion and
2) **local complementation**



G

The **vertex-minors** of a graph G are the graphs that can be obtained from G by

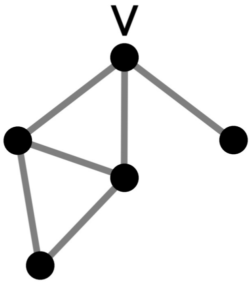
1) vertex deletion and
2) **local complementation**



G

The **vertex-minors** of a graph G are the graphs that can be obtained from G by

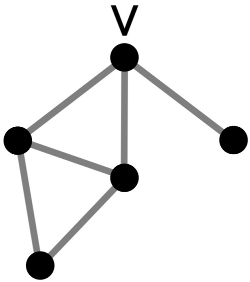
1) vertex deletion and
2) **local complementation**



$$G - u$$

The **vertex-minors** of a graph G are the graphs that can be obtained from G by

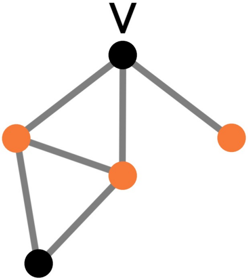
1) vertex deletion and
2) **local complementation**



$G - u$

The **vertex-minors** of a graph G are the graphs that can be obtained from G by

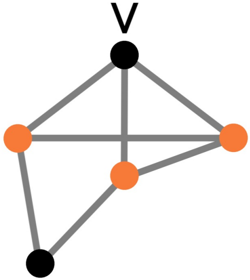
1) vertex deletion and
2) **local complementation**: select a vertex v and replace the induced subgraph on neighborhood(v) by its complement.



$G - u$

The **vertex-minors** of a graph G are the graphs that can be obtained from G by

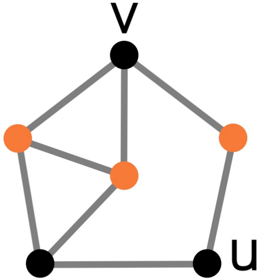
1) vertex deletion and
2) **local complementation**: select a vertex v and replace the induced subgraph on neighborhood(v) by its complement.



$$(G - u) * v$$

The **vertex-minors** of a graph G are the graphs that can be obtained from G by

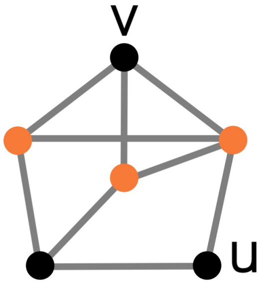
1) vertex deletion and
2) **local complementation**: select a vertex v and replace the induced subgraph on neighborhood(v) by its complement.



G

The **vertex-minors** of a graph G are the graphs that can be obtained from G by

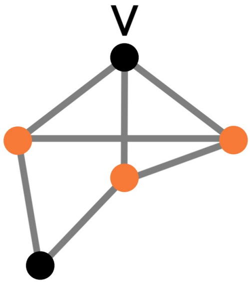
1) vertex deletion and
2) **local complementation**: select a vertex v and replace the induced subgraph on neighborhood(v) by its complement.



$G * v$

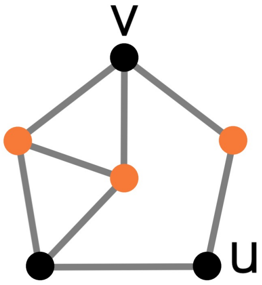
The **vertex-minors** of a graph G are the graphs that can be obtained from G by

1) vertex deletion and
2) **local complementation**: select a vertex v and replace the induced subgraph on neighborhood(v) by its complement.



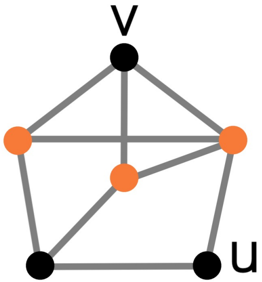
$$G * v - u$$

The **vertex-minors** of a graph G are the induced subgraphs of graphs that are **locally equivalent** to G (that is, can be obtained from G by a sequence of local complementations).



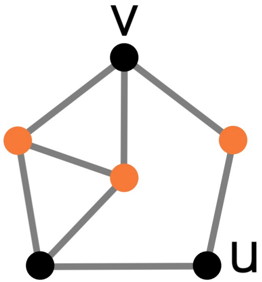
G

The **vertex-minors** of a graph G are the induced subgraphs of graphs that are **locally equivalent** to G (that is, can be obtained from G by a sequence of local complementations).



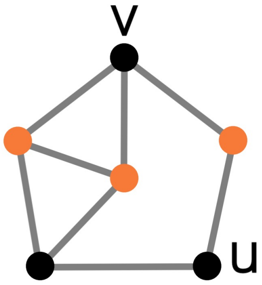
$G * v$

The **vertex-minors** of a graph G are the induced subgraphs of graphs that are **locally equivalent** to G (that is, can be obtained from G by a sequence of local complementations).



$G * v * v$

The **vertex-minors** of a graph G are the induced subgraphs of graphs that are **locally equivalent** to G (that is, can be obtained from G by a sequence of local complementations).



$$G * v * v = G$$

Why **local equivalence** classes?

- nice interpretation for graph states in quantum computing

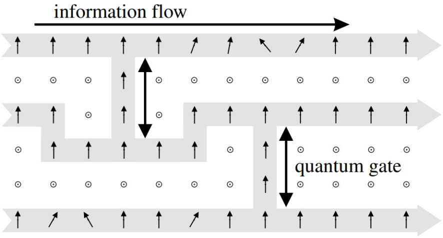


FIG. 1. Quantum computation by measuring two-state parti-

(Raussendorf-Briegel, Van den Nest-Dehaene-De Moor)

Why **local equivalence** classes?

- nice interpretation for graph states in quantum computing

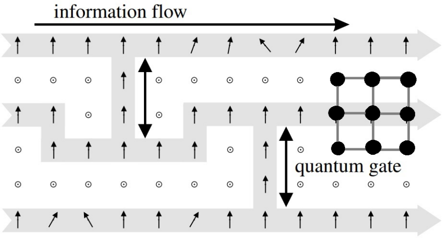


FIG. 1. Quantum computation by measuring two-state parti-

(Raussendorf-Briegel, Van den Nest-Dehaene-De Moor)

Why **local equivalence** classes?

- nice interpretation for graph states in quantum computing

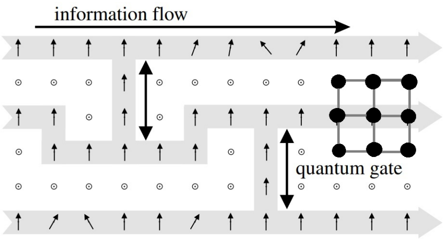


FIG. 1. Quantum computation by measuring two-state parti-

(Raussendorf-Briegel, Van den Nest-Dehaene-De Moor)

Conjecture (Geelen)

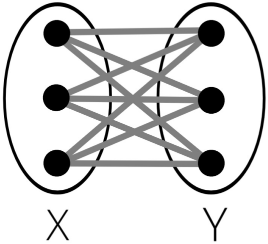
If the graph states that can be prepared come from a proper **vertex-minor-closed** *class* $\mathcal{F}$*, then* $BQP_{\mathcal{F}} = BPP$.

Why **local equivalence** classes?

- nice interpretation for graph states in quantum computing
- locally equivalent graphs have the same **cut-rank** function

$$
\begin{array}{c} \\ X \\ \\ \\ Y \\ \\ \end{array}
\overset{\qquad X \qquad\qquad\qquad Y}{\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}}
$$

adjacency matrix

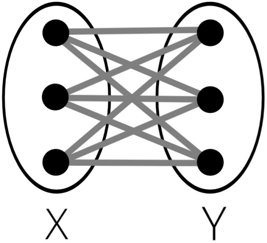


Why **local equivalence** classes?

- nice interpretation for graph states in quantum computing
- locally equivalent graphs have the same **cut-rank** function

$$
\begin{array}{c}
 \\ X \\ \\ \\ Y \\ \\
\end{array}
\overset{\displaystyle \quad X \qquad\qquad\qquad Y}{\begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0
\end{bmatrix}}
$$

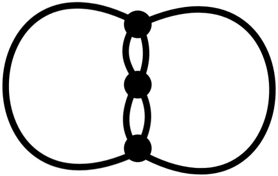
adjacency matrix

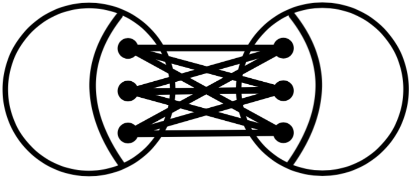


The **cut-rank** of $X \subseteq V(G)$ is the rank of **adj**$[X, \overline{X}]$ over GF_2.

Why **local equivalence** classes?

- nice interpretation for graph states in quantum computing
- locally equivalent graphs have the same **cut-rank** function

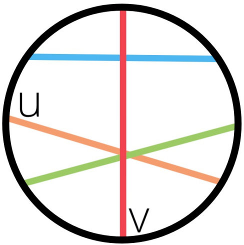




separators $\longrightarrow$ **cut-rank**

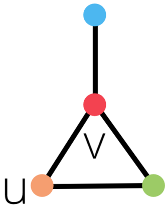
Why **local equivalence** classes?

- nice interpretation for graph states in quantum computing
- locally equivalent graphs have the same **cut-rank** function
- locally equivalent **circle graphs** can be efficiently represented

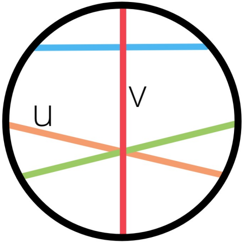


chord diagram

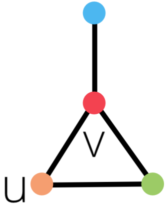
$\longrightarrow$



circle graph

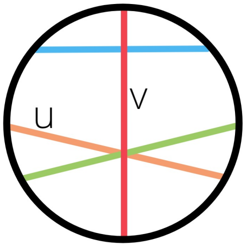


chord diagram

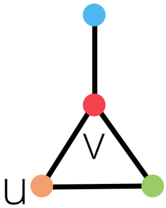


circle graph

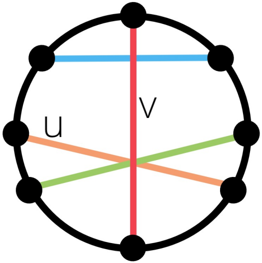
tour graph



chord diagram

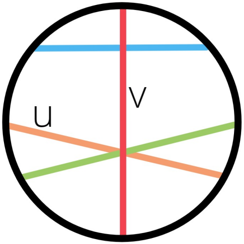


circle graph

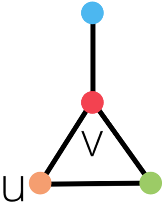


tour graph

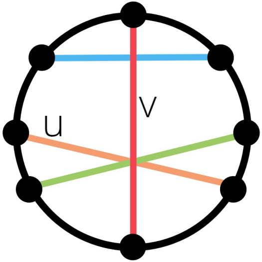
View the **chord diagram** as a 3-regular graph...



chord diagram

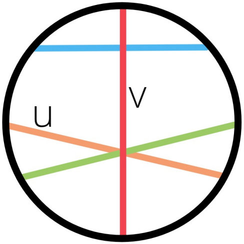


circle graph

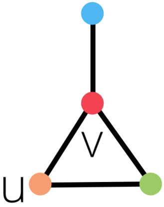


tour graph

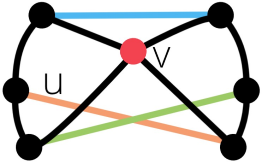
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**.



chord diagram

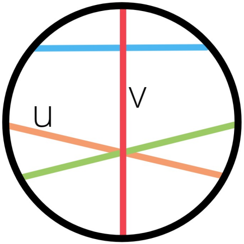


circle graph

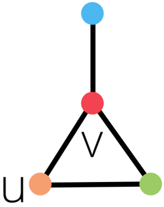


tour graph

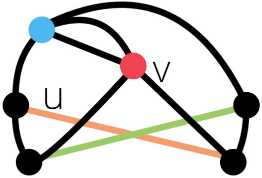
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**.



chord diagram

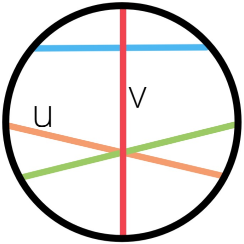


circle graph

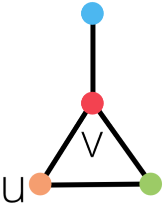


tour graph

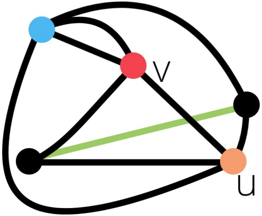
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**.



chord diagram

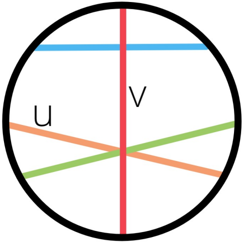


circle graph

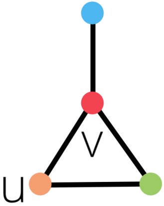


tour graph

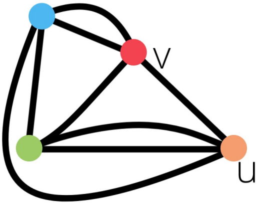
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**.



chord diagram

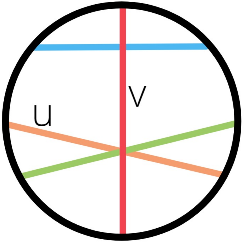


circle graph

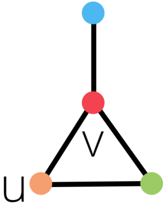


tour graph

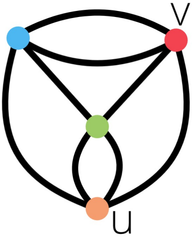
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**.



chord diagram

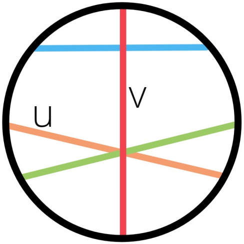


circle graph

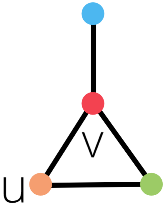


tour graph

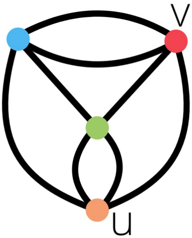
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**.



chord diagram

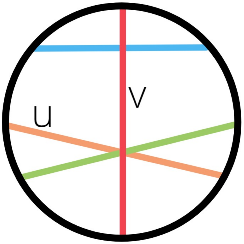


circle graph

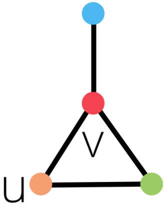


tour graph

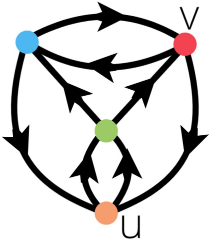
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit.



chord diagram

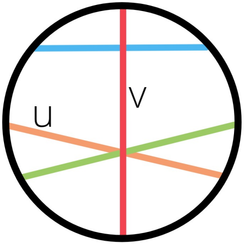


circle graph

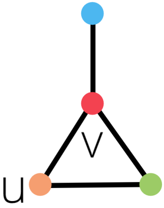


tour graph

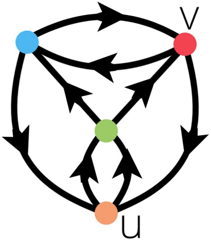
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit.



chord diagram

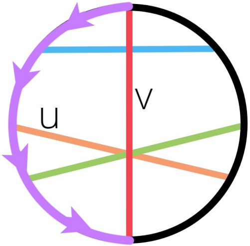


circle graph

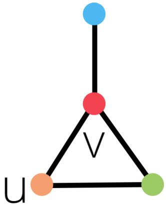


tour graph

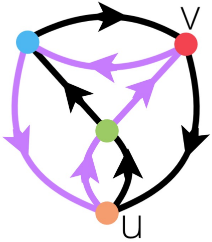
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v.



chord diagram

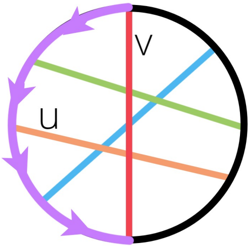


circle graph

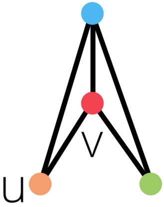


tour graph

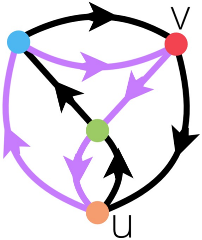
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v.



chord diagram

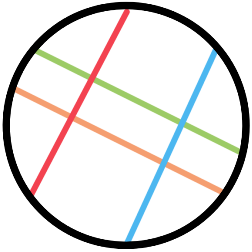


circle graph

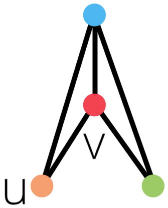


tour graph

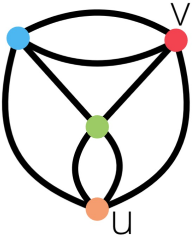
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v.



chord diagram

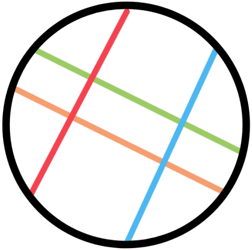


circle graph

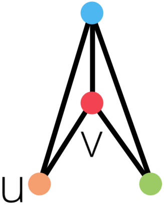


tour graph

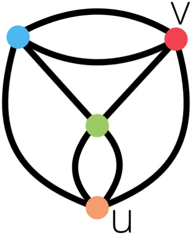
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v.



chord diagram

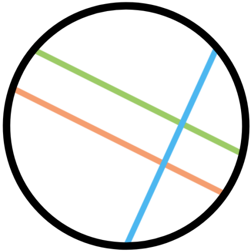


circle graph

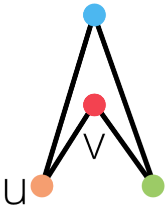


tour graph

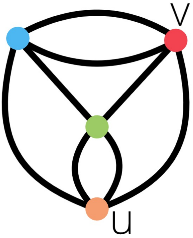
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v then u.



chord diagram

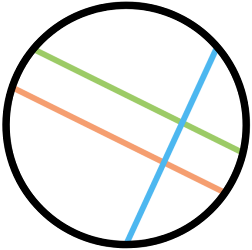


circle graph

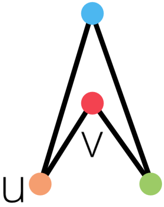


tour graph

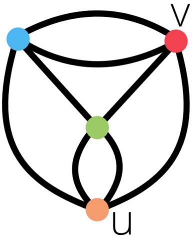
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v then u.



chord diagram

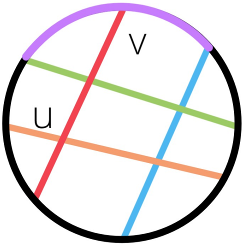


circle graph

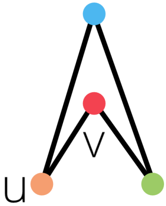


tour graph

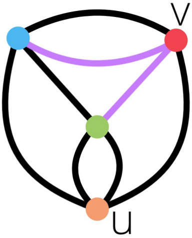
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v then u. To delete v...



chord diagram

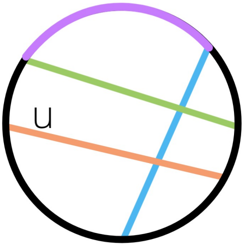


circle graph

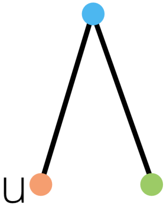


tour graph

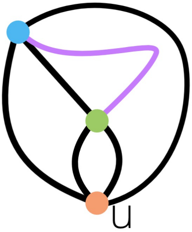
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v then u. To delete v...



chord diagram

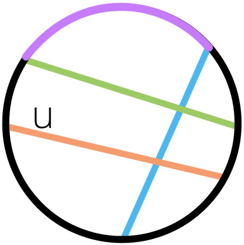


circle graph

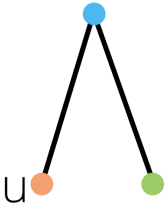


tour graph

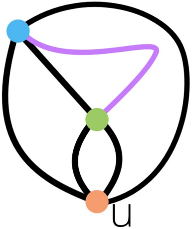
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v then u. To delete v...



chord diagram

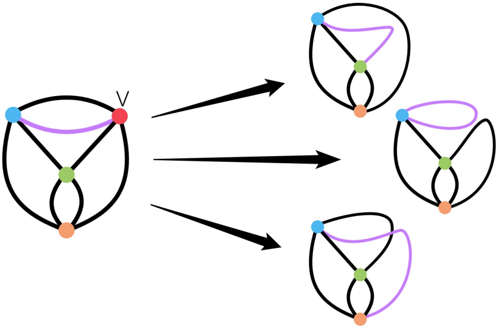


circle graph

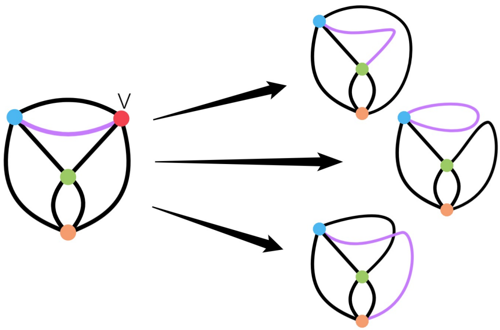


tour graph

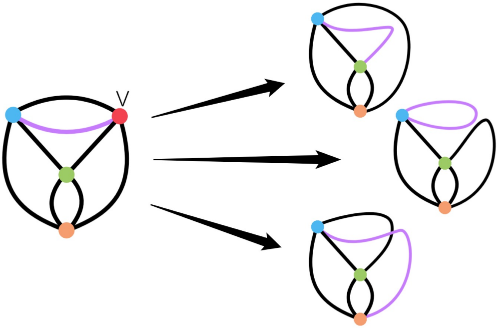
View the **chord diagram** as a 3-regular graph and contract each of the chords to get the **tour graph**. It has a specified Eulerian circuit. Consider locally complementing at v then u. To delete v, **split it off** in the **tour graph**.



In a 4-regular graph, there are 3 ways to **split off** v.



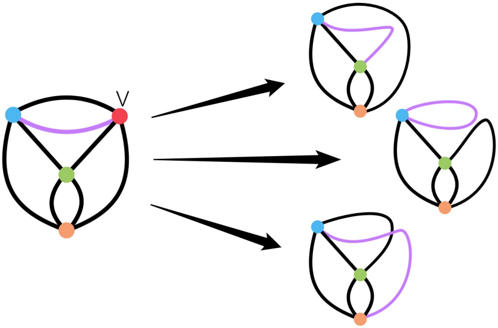
In a 4-regular graph, there are 3 ways to **split off** v. We say that a graph H **completely immerses** into G if H can be obtained from G by splitting off vertices.



In a 4-regular graph, there are 3 ways to **split off** v. We say that a graph H **completely immerses** into G if H can be obtained from G by splitting off vertices.

Theorem (Kotzig, Bouchet)

If H and G are **prime** *circle graphs, then H is a vertex-minor of G $\iff$ tour(H)* **completely immerses** *into tour(G).*

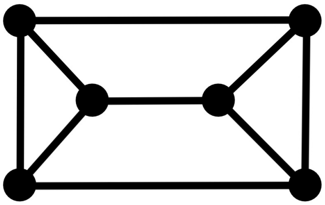


Lemma (Bouchet)

If H is a vertex-minor of G and $v \in V(G) \setminus V(H)$, then H is also a **vertex-minor** *of either*

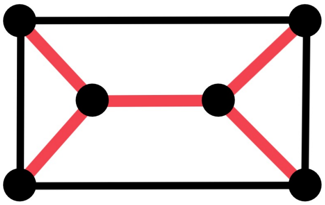
1) $G - v$,
2) $G * v - v$, *or*
3) $G \times uv - v$ *for each neighbour u of v.*

Consider a planar graph with a spanning tree **T**. Draw a curve closely around **T**. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.



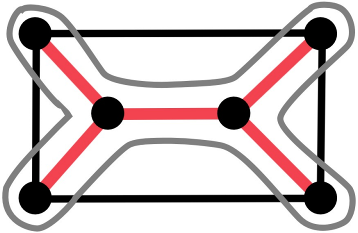
planar graph

Consider a planar graph with a spanning tree **T**. Draw a curve closely around **T**. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.



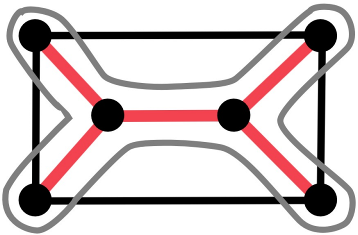
planar graph

Consider a planar graph with a spanning tree **T**. Draw a curve closely around **T**. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.

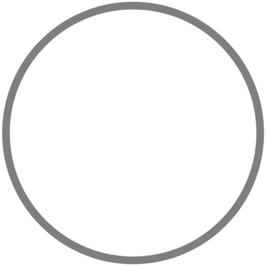


planar graph

Consider a planar graph with a spanning tree **T**. Draw a curve closely around **T**. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.

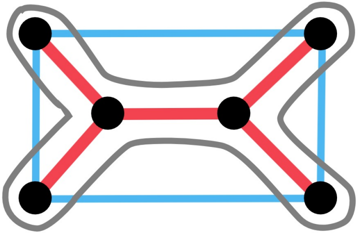


planar graph

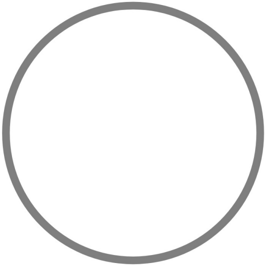


chord diagram

Consider a planar graph with a spanning tree $\mathbf{T}$. Draw a curve closely around $\mathbf{T}$. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.

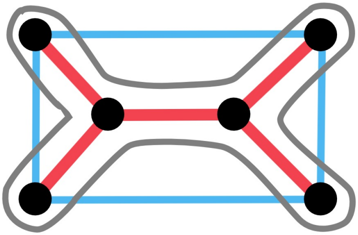


planar graph

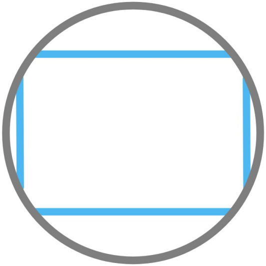


chord diagram

Consider a planar graph with a spanning tree **T**. Draw a curve closely around **T**. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.

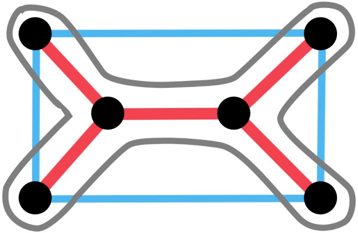


planar graph

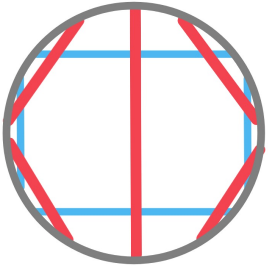


chord diagram

Consider a planar graph with a spanning tree **T**. Draw a curve closely around **T**. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.

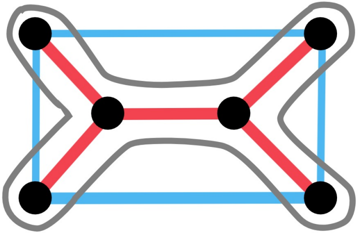


planar graph

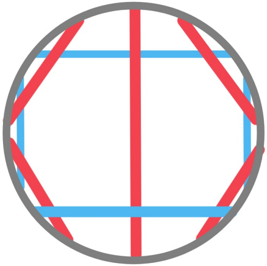


chord diagram

Consider a planar graph with a spanning tree T. Draw a curve closely around T. So $E(G) \setminus E(\mathsf{T})$ yields one set of non-crossing chords and $E(\mathsf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathsf{T})$. Taking the dual just exchanges sides.

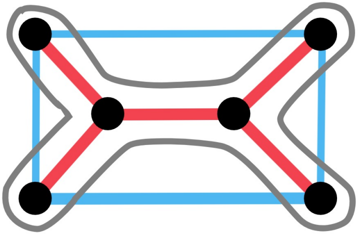


planar graph

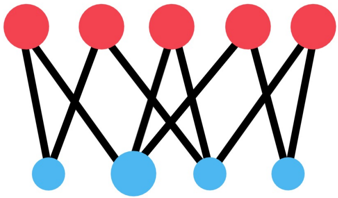


chord diagram

Consider a planar graph with a spanning tree $\mathbf{T}$. Draw a curve closely around $\mathbf{T}$. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.

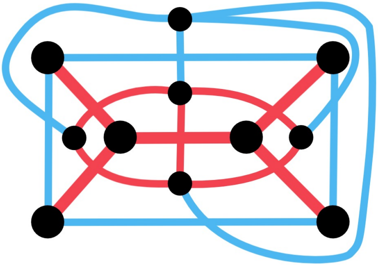


planar graph

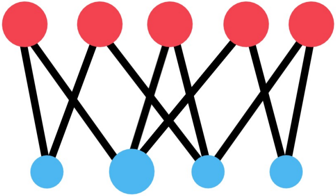


$\mathrm{fund}(\mathbf{T})$

Consider a planar graph with a spanning tree **T**. Draw a curve closely around **T**. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.

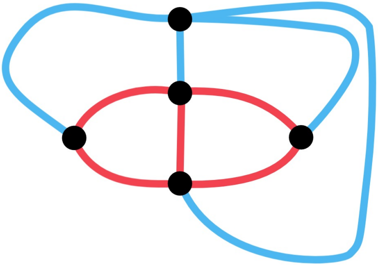


planar graph

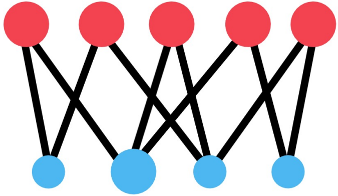


$\mathrm{fund}(\mathbf{T})$

Consider a planar graph with a spanning tree **T**. Draw a curve closely around **T**. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.



planar graph



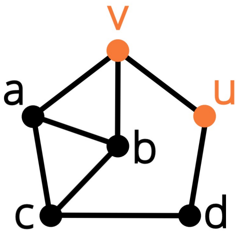
$\mathrm{fund}(\mathbf{T}^*)$

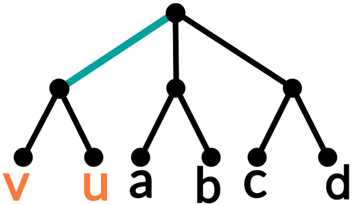
Consider a planar graph with a spanning tree $\mathbf{T}$. Draw a curve closely around $\mathbf{T}$. So $E(G) \setminus E(\mathbf{T})$ yields one set of non-crossing chords and $E(\mathbf{T})$ yields another. The circle graph is $\mathrm{fund}(\mathbf{T})$. Taking the dual just exchanges sides.

Theorem (Bouchet)

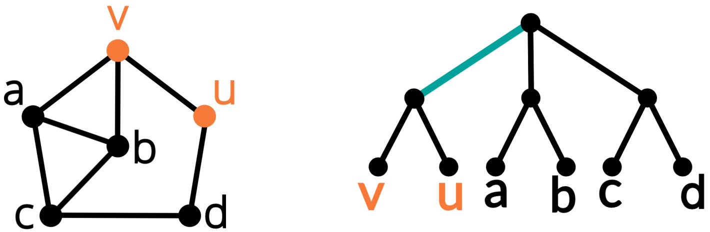
The fundamental graphs of two distinct, connected **binary matroids** *are pivot equivalent iff the matroids are* **dual**.

Rank-width(G) is the minimum **width** of a subcubic tree T with leafs $V(G)$.





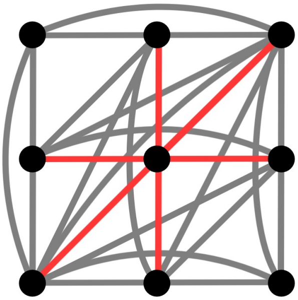
Rank-width(G) is the minimum **width** of a subcubic tree T with leafs $V(G)$.



$$\textbf{width}(T) = \max_{e \in E(T)} \text{cut-rank}(X_e)$$

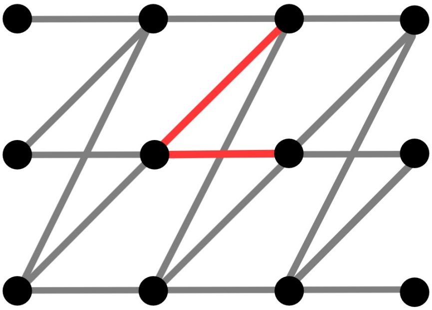
Theorem (Geelen, Kwon, McCarty, & Wollan 2020)

For any **circle graph** *H, every graph with* **rank-width** $\geq f(H)$ *has a* **vertex-minor** *isomorphic to H.*



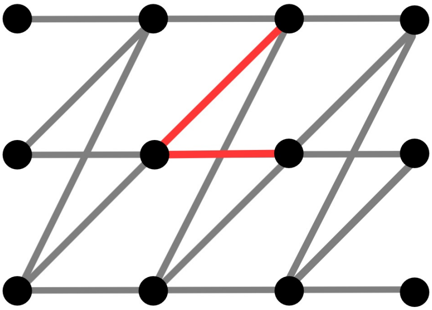
Conjecture (Oum 2009)

A class of graphs has bounded rank-width if and only if it does not contain all ***bipartite*** *circle graph as* ***pivot-minors****.*



Conjecture (Oum 2009)

A class of graphs has bounded rank-width if and only if it does not contain all **bipartite** *circle graph as* **pivot-minors**.



Would be a common generalization!

Thank you!