Local certification of graph classes

Laurent Feuilloley

Université Lyon 1

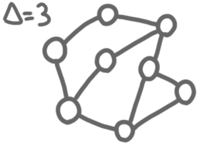
ADGA workshop · DISC 2021

Note: bibliographic pointers at the end of the talk.

Classic network assumptions

Classic LOCAL/CONGEST algorithms are designed for:

(a) General networks



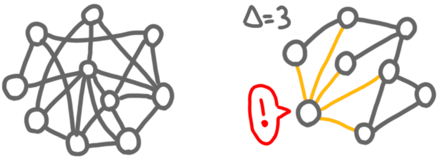
(b) Bounded degree

Two reasons for that:

- Fundamental settings.
- Locally checkable structures (we can raise alarm if needed).

Classic network assumptions

Classic LOCAL/CONGEST algorithms are designed for:



(a) General networks (b) Bounded degree

Two reasons for that:

- Fundamental settings.
- Locally checkable structures (we can raise alarm if needed).

Limitations of these two classes

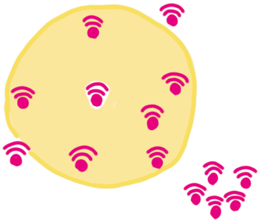
These classes do not capture the properties of real-world networks.

Example: unit-disks are a model for wireless networks.

Limitations of these two classes

These classes do not capture the properties of real-world networks.

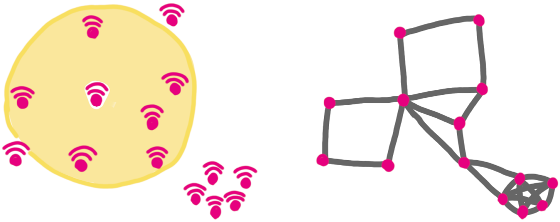
Example: unit-disks are a model for wireless networks.



Limitations of these two classes

These classes do not capture the properties of real-world networks.

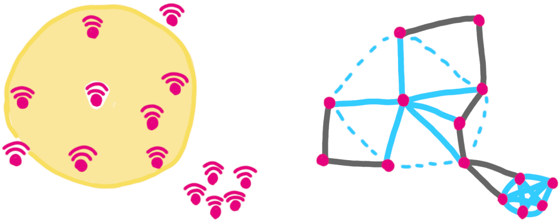
Example: unit-disks are a model for wireless networks.



Limitations of these two classes

These classes do not capture the properties of real-world networks.

Example: unit-disks are a model for wireless networks.



- The degree is unbounded.
- The independence number is low

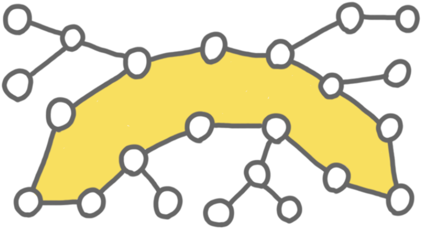
→ Classic algorithms do not give a good estimate.

Most network structures cannot be checked

There exists algorithms for unit-disks, planar, small-diameter etc.

But these network structures cannot be checked locally.

Example: Trees cannot be checked locally.



Most network structures cannot be checked

There exists algorithms for unit-disks, planar, small-diameter etc.

But these network structures cannot be checked locally.

Example: Trees cannot be checked locally.

New notion: Local certification

Local certification is a mechanism to allow local checking.

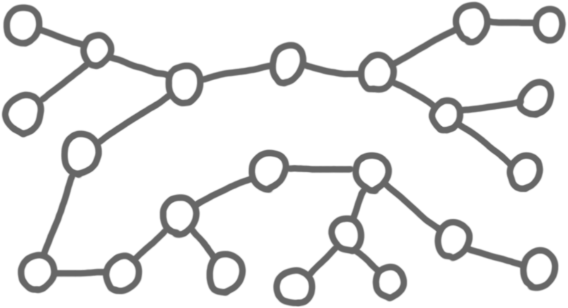
Idea: A labeling of the nodes that certifies that the network structure, and that can be checked locally. Coming from another algorithm, or from the network designer.

Requirements: There exists a local verification algorithm s.t.:

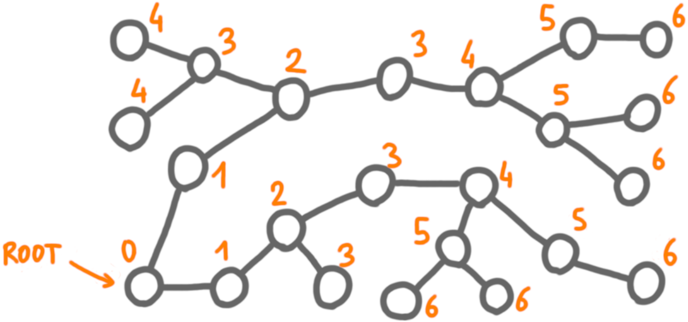
- For every graph in the class, there exists a labeling such that the algorithm accepts.
- For every graph *not* in the class, for every labeling, the algorithm rejects on at least one node.

About the locality: 1 round or $O(1)$ rounds.

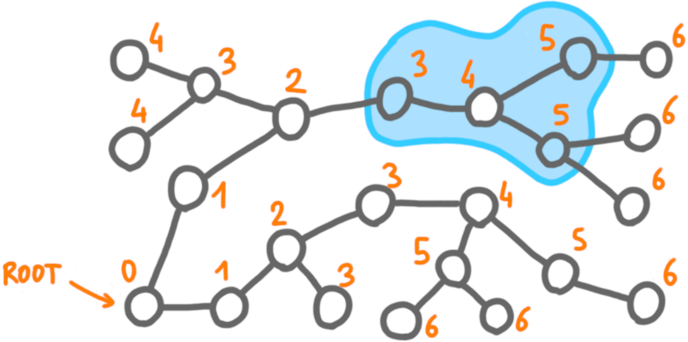
Example: local certification of trees



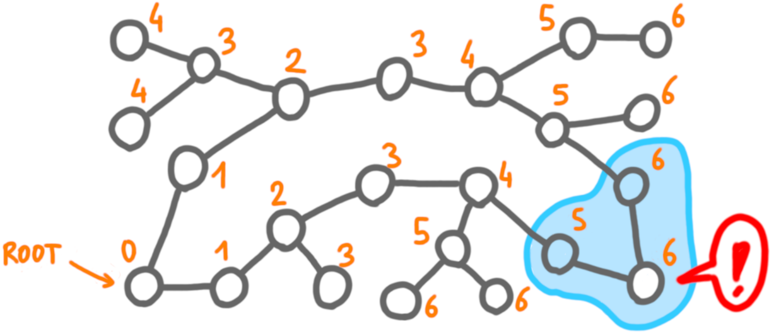
Example: local certification of trees



Example: local certification of trees



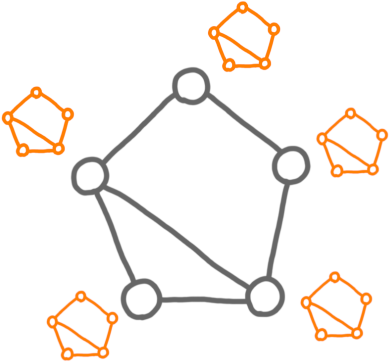
Example: local certification of trees



Can we certify any graph class?

Question: Take a graph class $\mathcal{C}$ (*i.e.* an infinite set of graphs). Can we design a labeling and a local verification algorithm that fulfill the requirements?

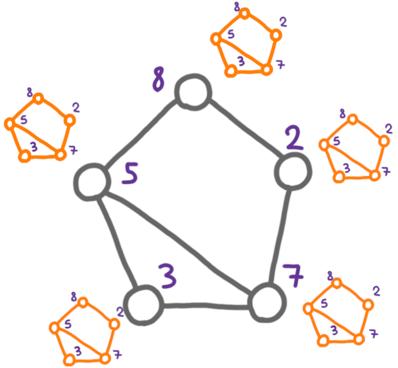
Answer: Yes! (By abusing the assumptions and having large certificates.)



Can we certify any graph class?

Question: Take a graph class $\mathcal{C}$ (*i.e.* an infinite set of graphs). Can we design a labeling and a local verification algorithm that fulfill the requirements?

Answer: Yes! (By abusing the assumptions and having large certificates.)



Optimal certificate size

Measure of quality: the size of the certificates.

Example: for trees,

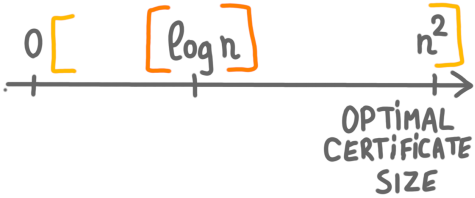
- the optimal certificate size for trees is > 0, and $\leq O(n^2)$,
- the distance labeling gives $O(\log n)$
- $O(\log n)$ is actually optimal

Optimal certificate size

Measure of quality: the size of the certificates.

Example: for trees,

- the optimal certificate size for trees is > 0, and $\leq O(n^2)$,
- the distance labeling gives $O(\log n)$
- $O(\log n)$ is actually optimal



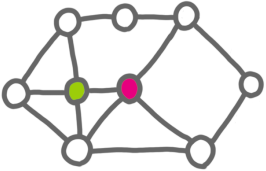
Research program

$\rightarrow$ **Design optimal certification for relevant graph classes.**

List of relevant classes:

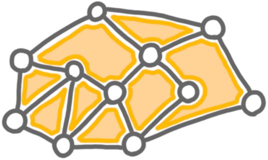
- Classes used in DC: trees, grids, planar, unit-disk, cliques, small-diameter
- Classic classes: planar, chordal, interval, cographs, bipartite
- Families of classes: H-free, H-minor-free.

A case study: 4 approaches to planar graphs

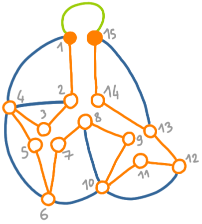


Embedding

Minors



Faces



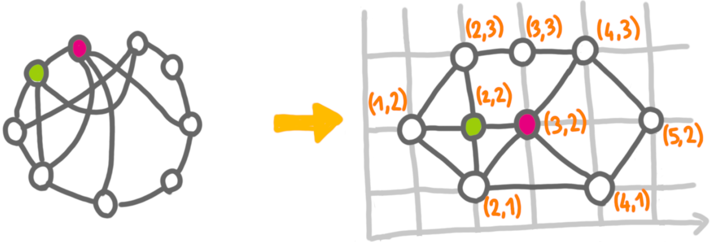
Spanning tree

Planar graphs: via the embedding

Embedding characterization: Planar graphs are the graphs that can be embedded in the plane without edge crossings.

Planar graphs: via the embedding

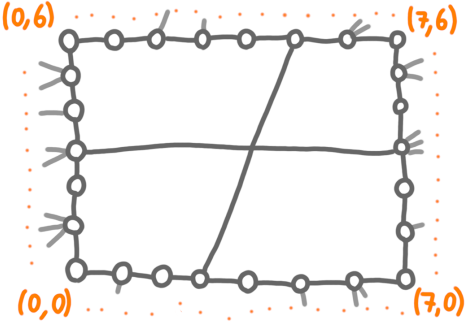
Embedding characterization: Planar graphs are the graphs that can be embedded in the plane without edge crossings.



Certification idea: Give the coordinates to the nodes.

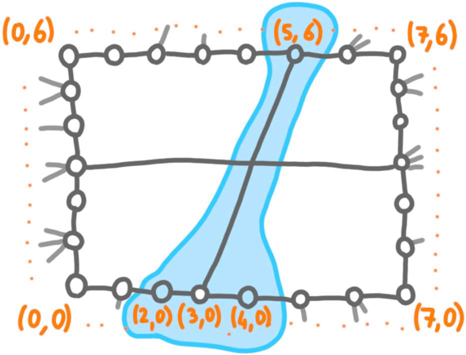
Planar graphs: via the embedding

Problem: The nodes can be fooled by the coordinates.



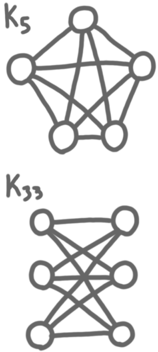
Planar graphs: via the embedding

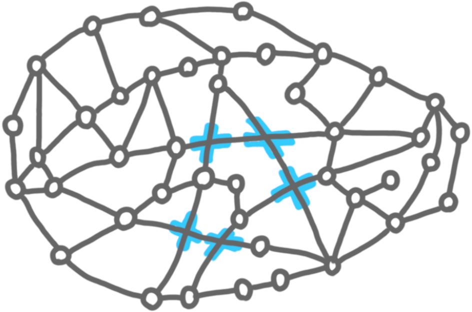
Problem: The nodes can be fooled by the coordinates.



Planar graphs: via minors

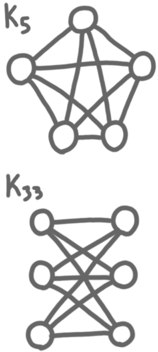
Minor characterization: Planar graphs are the graphs with no K_5 or $K_{3,3}$ minor.





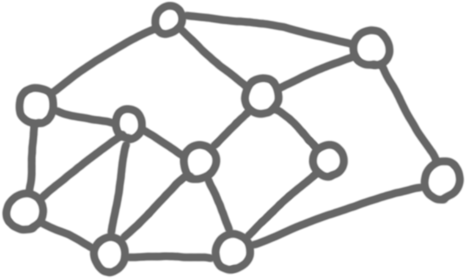
Planar graphs: via minors

Minor characterization: Planar graphs are the graphs with no K_5 or $K_{3,3}$ minor.



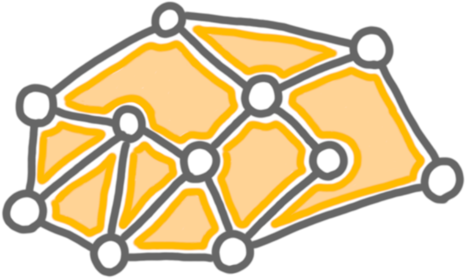
Planar graphs: via faces

- Given a planar embedding, we can define faces.
- But this is not enough the surface can be more complicated.
- Euler formula: $|V| - |F| + |E| = 2$, only in planar graphs.



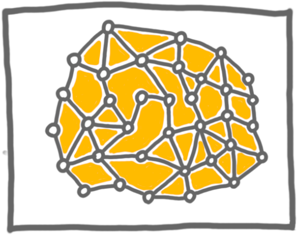
Planar graphs: via faces

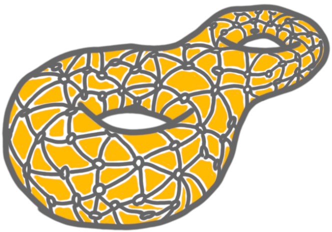
- Given a planar embedding, we can define faces.
- But this is not enough the surface can be more complicated.
- Euler formula: $|V| - |F| + |E| = 2$, only in planar graphs.



Planar graphs: via faces

- Given a planar embedding, we can define faces.
- But this is not enough the surface can be more complicated.
- Euler formula: $|V| - |F| + |E| = 2$, only in planar graphs.





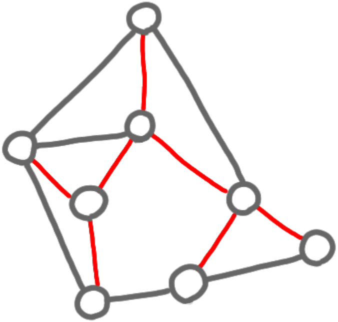
Planar graphs: via faces

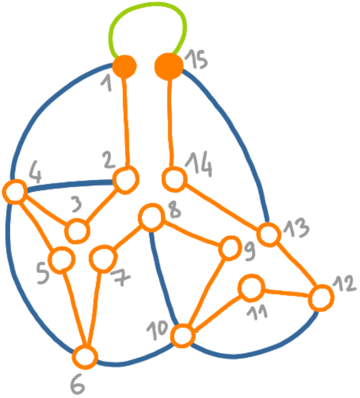
Certification idea:

- Use rotation systems to encode faces.
- Use a spanning tree to gather $|V|$, $|F|$ and $|E|$ at one node.

Planar graphs: via a spanning tree

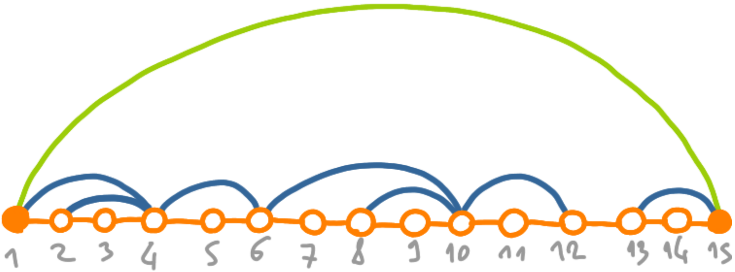
Spanning tree characterization: For any spanning tree, there is no crossing of the outer edges.



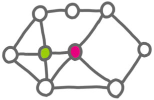


Planar graphs: via a spanning tree

Spanning tree characterization: For any spanning tree, there is no crossing of the outer edges.

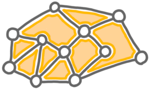


Planar graphs: summary and theorem

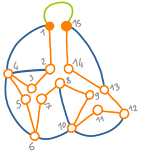


Embedding

Minors



Faces



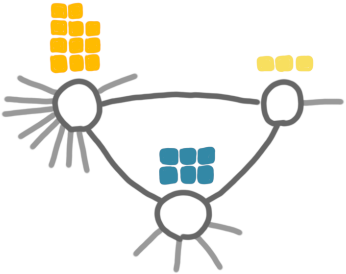
Spanning tree

Theorem: Planar graphs can be certified with $O(\log n)$ bits.*

*: Taming high degrees

Problem: In the certifications given, the certificate size can be of size $\delta \log n$, where δ is the vertex degree.

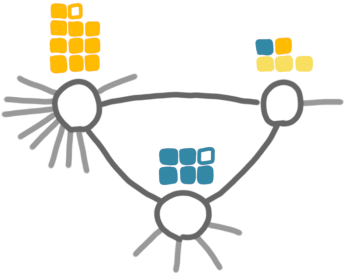
Solution: In every planar graph there exists a vertex of degree at most 6. $\rightarrow$ Degeneracy ordering $\rightarrow$ Certificate load balancing.



*: Taming high degrees

Problem: In the certifications given, the certificate size can be of size $\delta \log n$, where δ is the vertex degree.

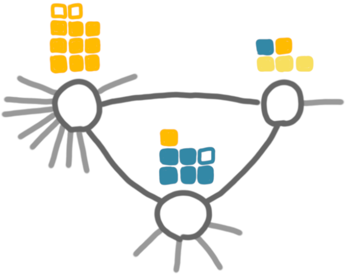
Solution: In every planar graph there exists a vertex of degree at most 6. $\rightarrow$ Degeneracy ordering $\rightarrow$ Certificate load balancing.



*: Taming high degrees

Problem: In the certifications given, the certificate size can be of size $\delta \log n$, where δ is the vertex degree.

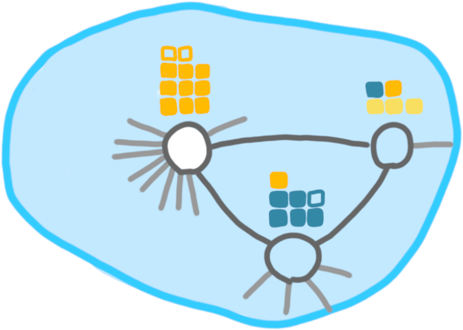
Solution: In every planar graph there exists a vertex of degree at most 6. $\rightarrow$ Degeneracy ordering $\rightarrow$ Certificate load balancing.



*: Taming high degrees

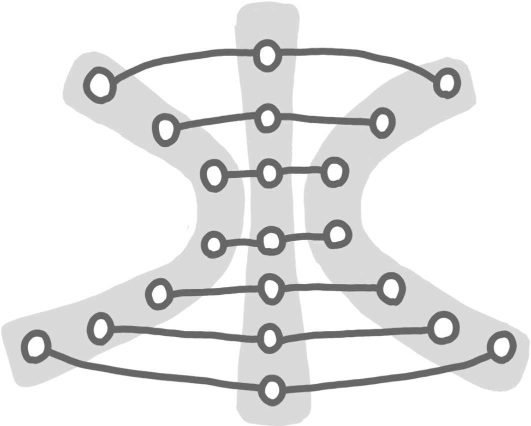
Problem: In the certifications given, the certificate size can be of size $\delta \log n$, where δ is the vertex degree.

Solution: In every planar graph there exists a vertex of degree at most 6. $\rightarrow$ Degeneracy ordering $\rightarrow$ Certificate load balancing.



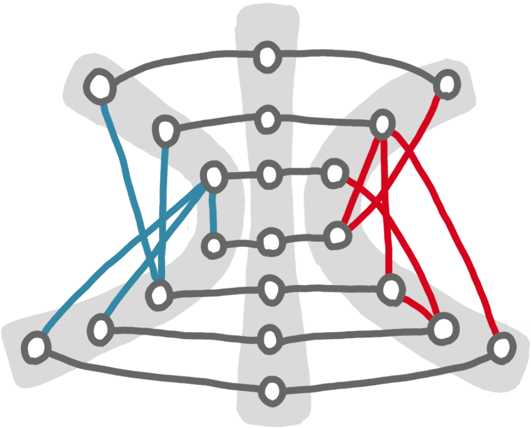
A class that requires large certificates

Class: Graph of diameter at most 3.
Model: Look at distance 1.
Theorem: Optimal certificate size in $\tilde{\Omega}(n)$.



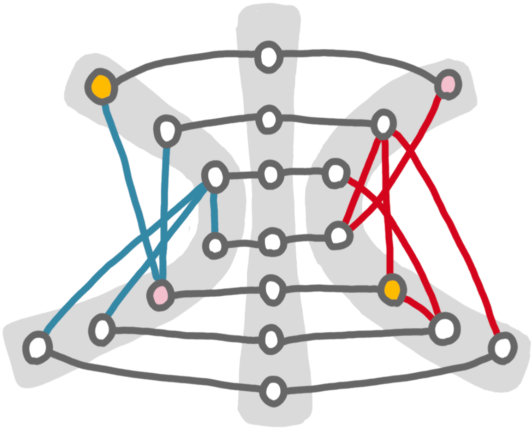
A class that requires large certificates

Class: Graph of diameter at most 3.
Model: Look at distance 1.
Theorem: Optimal certificate size in $\tilde{\Omega}(n)$.



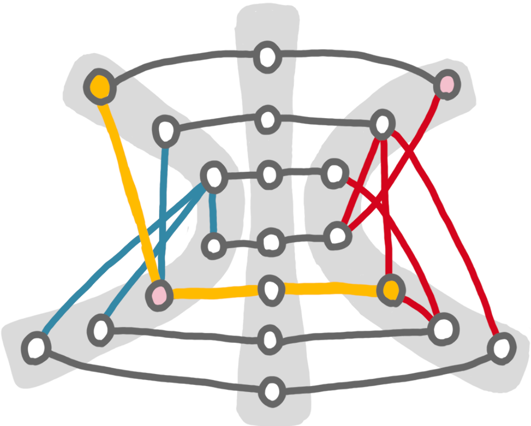
A class that requires large certificates

Class: Graph of diameter at most 3.
Model: Look at distance 1.
Theorem: Optimal certificate size in $\tilde{\Omega}(n)$.



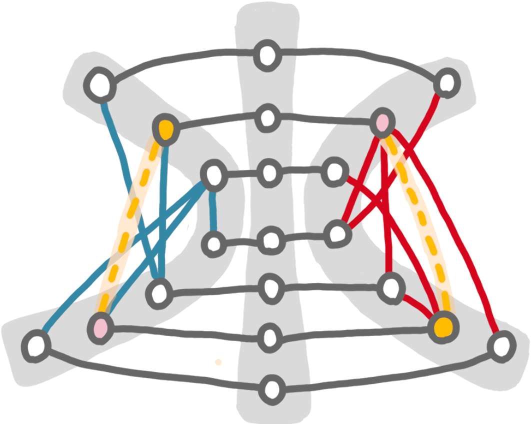
A class that requires large certificates

Class: Graph of diameter at most 3.
Model: Look at distance 1.
Theorem: Optimal certificate size in $\tilde{\Omega}(n)$.



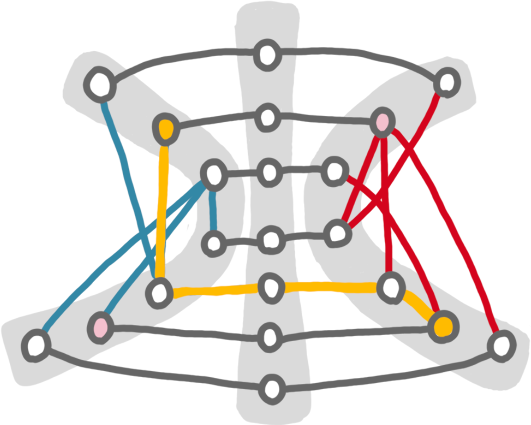
A class that requires large certificates

Class: Graph of diameter at most 3.
Model: Look at distance 1.
Theorem: Optimal certificate size in $\tilde{\Omega}(n)$.



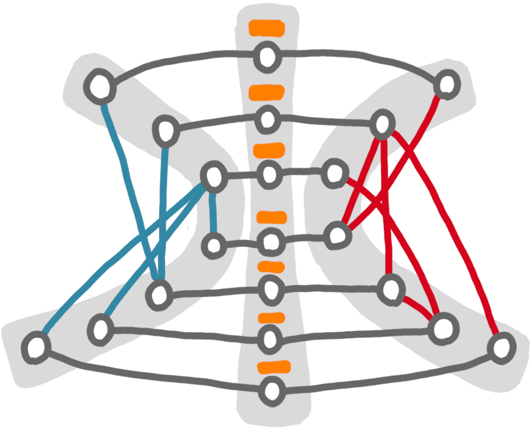
A class that requires large certificates

Class: Graph of diameter at most 3.
Model: Look at distance 1.
Theorem: Optimal certificate size in $\tilde{\Omega}(n)$.

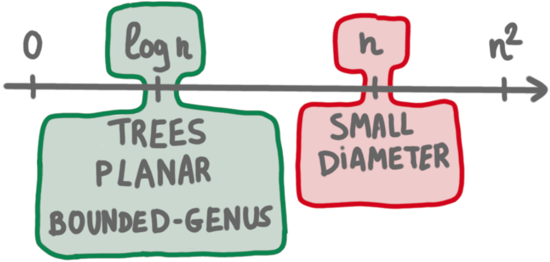


A class that requires large certificates

Class: Graph of diameter at most 3.
Model: Look at distance 1.
Theorem: Optimal certificate size in $\tilde{\Omega}(n)$.



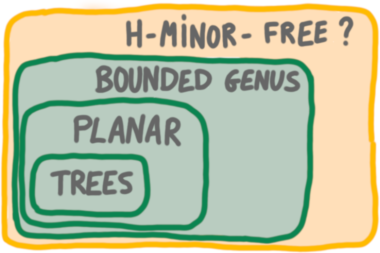
Status and research directions



Direction 1: Aim for a generalization of the log n region.
Direction 2: Target other relevant graph classes and parameters.

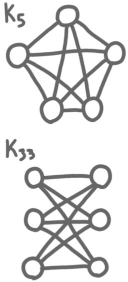
Generalization to H-minor-free

- H-minor-free is a natural generalization of the "good classes".
- They are hereditary, which is good for compact certification.
- But we don't know how to certify that something is not there.



Generalization to H-minor-free

- H-minor-free is a natural generalization of the "good classes".
- They are hereditary, which is good for compact certification.
- But we don't know how to certify that something is not there.

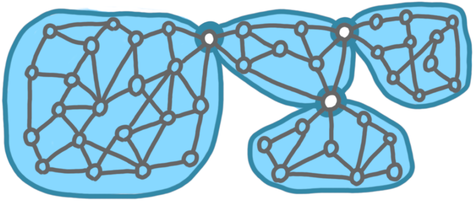


Generalization to H-minor-free

Open question: Does every graph class characterized by forbidden minors have a compact local certification?

Theorem: When the minors are small ($|H| \leq 4$, or $|H| = 5$ with a special structure), the answer is positive.

A key tool: Certification of 2(and 3)-connectivity.



Other classes and parameters

List of relevant classes:

- Classes used in DC: trees, grids, planar, unit-disk, cliques, small-diameter.
- Classic classes: planar, chordal, interval, cographs, bipartite.

Open questions:

- Do unit-disk graphs have a compact certification?
- Can we certify treewidth k efficiently?
- What about k-connectivity?

Bibliographic pointers

Local certification papers mentioned:

- Proof-labeling schemes (Korman, Kutten, Peleg - 2010). doi:10.1007/s00446-010-0095-3
- Memory-efficient self stabilizing protocols for general networks (Afek, Kutten, Young - 1990). doi:10.1007/3-540-54099-7_2
- Locally checkable proofs in distributed computing (Göös, Suomela - 2016). doi:10.4086/toc.2016.v012a019

Tutorial on local certification

- Introduction to local certification (Feuilloley - 2021). doi:10.46298/dmtcs.6280 + Gem talk at PODC (on youtube).

Bibliographic pointers

Certification of planar and bounded-genus graphs

- Compact distributed certification of planar graphs (Feuilloley, Fraigniaud, Montealegre, Rapaport, Rémila, Todinca, 2021) doi:10.1007/s00453-021-00823-w + Talks at PODC by Montealegre
- Local Certification of Graphs with Bounded Genus (Same as above.) arxiv:2007.08084
- Local certification of graphs on surfaces (Esperet, Leveque - 2021) arxiv:2102.04133

Small diameter lower bound

- Approximate proof-labeling schemes (Censor-Hillel, Paz, Perry - 2020) doi:10.1016/j.tcs.2018.08.020

Bibliographic pointers

Certification of H-minor-free graphs

- Local certification of graph decompositions and applications to minor-free classes (Bousquet, Feuilloley, Pierron - 2021) arxiv:2108.00059 + BA at DISC.

Other specific classes

- Compact Distributed Interactive Proofs for the Recognition of Cographs and Distance-Hereditary Graphs (Montealegre, Ramírez-Romero, and Rapaport - 2021) arxiv:2012.03185 (+ personal communication)