Ortho-polygon Visibility Representations of Embedded Graphs

Emilio Di Giacomo[1], Walter Didimo[1],
William S. Evans[2], Giuseppe Liotta[1], Henk Meijer[3],
Fabrizio Montecchiani[1], and Stephen K. Wismath[4]

[1]University of Perugia, Italy
[2]University of British Columbia, Canada
[3]University College Roosevelt, the Netherlands
[4]University of Lethbridge, Canada

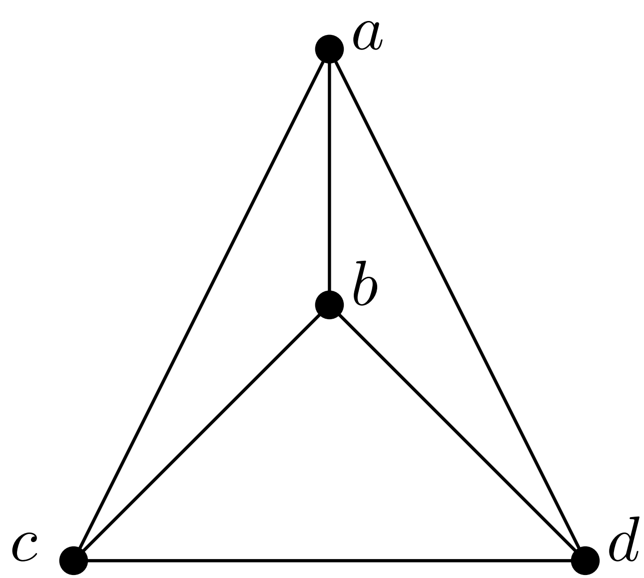
GD 2016, September 19-21, 2016, Athens

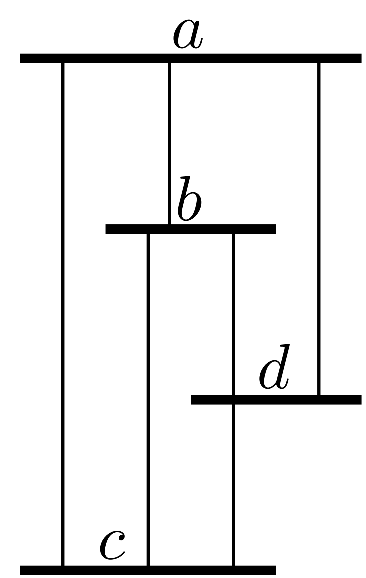
Visibility Representations: State of the Art

Bar Visibility representation (BVR) of a planar graph G:

Vertices $\rightarrow$ Horizontal bars

Edges $\rightarrow$ Vertical visibilities



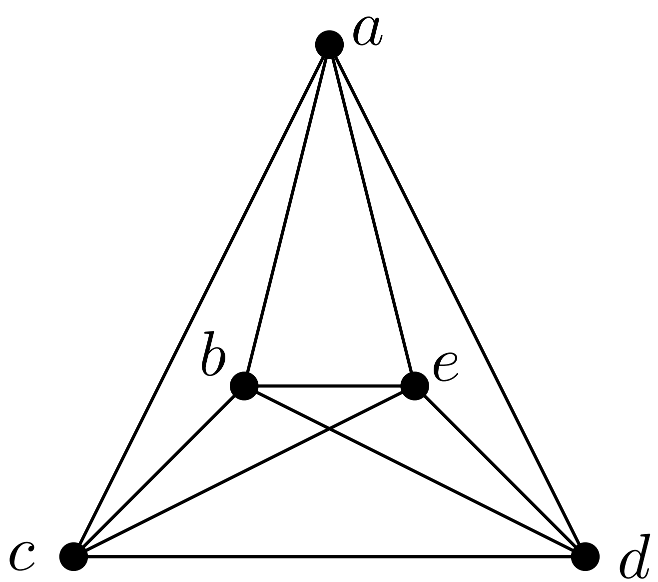


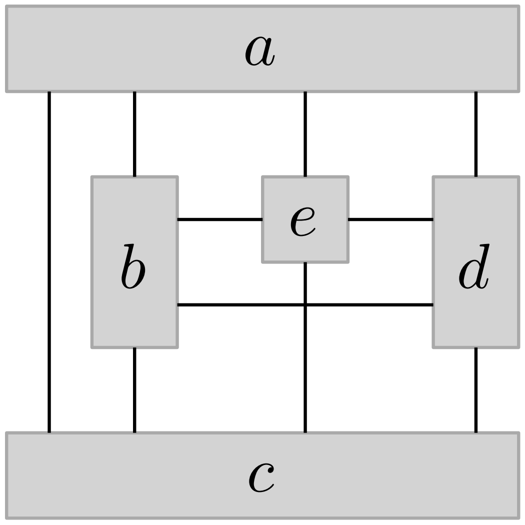
Every planar graph admits a BVR

[Duchet et al. 1983, Thomassen 1984, Wismath 1985, Rosenthiel & Tarjan 1986, Tamassia & Tollis 1986]

Visibility Representations: State of the Art

Rectangle Visibility representation (RVR) of a graph G:
Vertices $\rightarrow$ Axis-aligned rectangles
Edges $\rightarrow$ Horizontal/Vertical visibilities



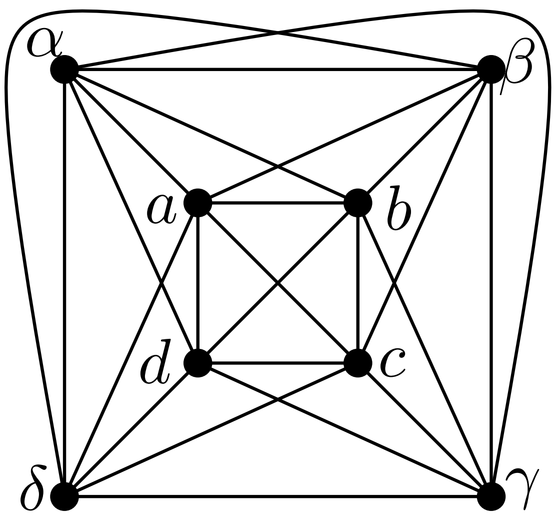


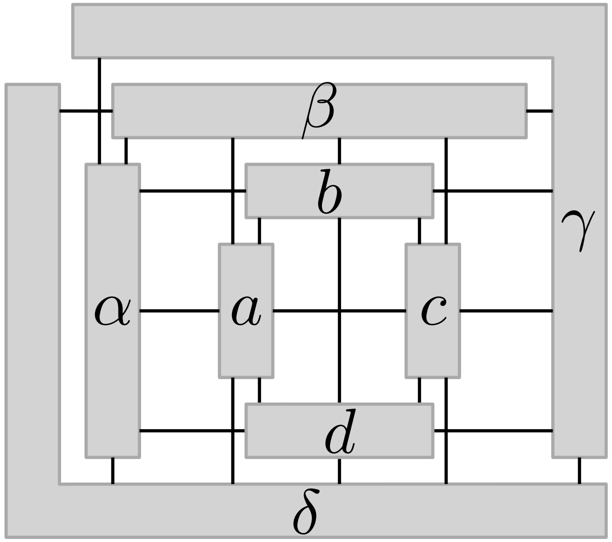
Recognition is NP-complete in general [Shermer 1996] and polynomial if the embedding is fixed and must be preserved [Biedl, Liotta, M. 2016]

Not all 1-planar graphs admit an RVR [Biedl, Liotta, M. 2016]

A new visibility model

Ortho-polygon Visibility representation (OPVR) of a graph G:
Vertices $\rightarrow$ Orthogonal polygons
Edges $\rightarrow$ Horizontal/Vertical visibilities

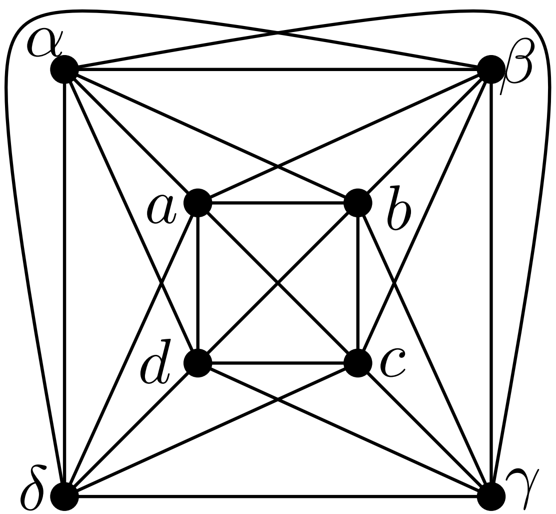


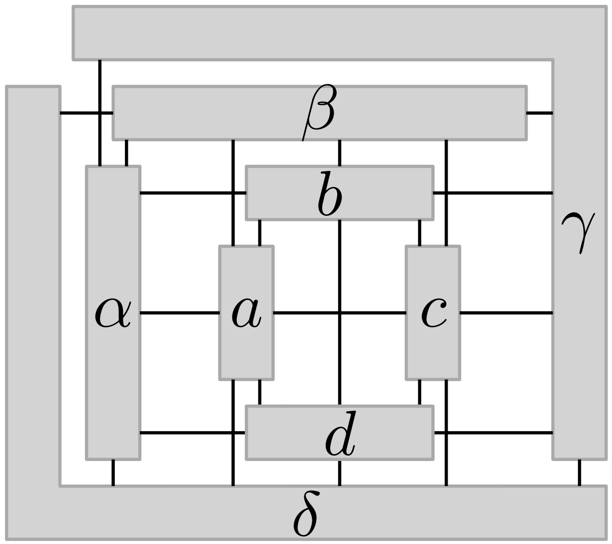


What embedded graphs can we draw as OPVRs?
Can we realize all 1-plane graphs?

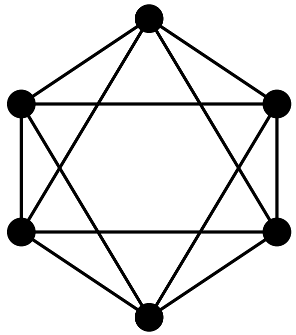
A new visibility model

Ortho-polygon Visibility representation (OPVR) of a graph G:
Vertices $\rightarrow$ Orthogonal polygons
Edges $\rightarrow$ Horizontal/Vertical visibilities





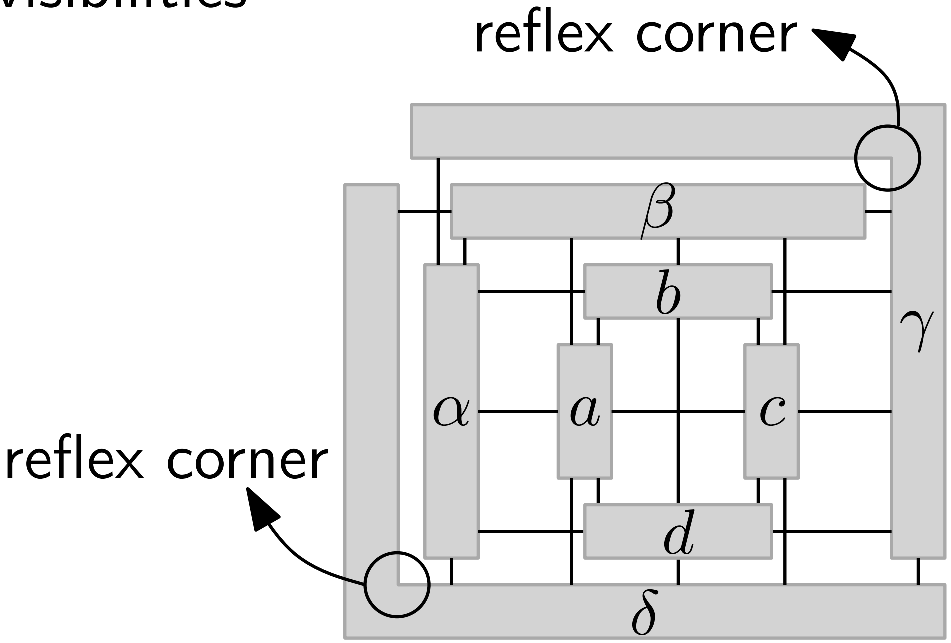
What embedded graphs can we draw as OPVRs?
Can we realize all 1-plane graphs?



graph with no OPVR
(although it has thickness two)

A new visibility model

Ortho-polygon Visibility representation (OPVR) of a graph G:
Vertices $\rightarrow$ Orthogonal polygons
Edges $\rightarrow$ Horizontal/Vertical visibilities



The vertex complexity of an OPVR is the smallest k such that any polygon representing a vertex has at most k reflex corners.

Minimizing the vertex complexity is NP-hard in general, what if the embedding is fixed?

Our contribution

1. Quadratic-time algorithm to test if an embedded graph admits an embedding-preserving OPVR.
Every 1-plane graph has an embedding-preserving OPVR.

Our contribution

1. Quadratic-time algorithm to test if an embedded graph admits an embedding-preserving OPVR.
Every 1-plane graph has an embedding-preserving OPVR.

2. Drawing algorithm to compute embedding-preserving OPVR with minimum v. c. in $O(n^{\frac{5}{2}} \log^{\frac{3}{2}} n)$ time (if one exists).

Our contribution

1. Quadratic-time algorithm to test if an embedded graph admits an embedding-preserving OPVR.
Every 1-plane graph has an embedding-preserving OPVR.

2. Drawing algorithm to compute embedding-preserving OPVR with minimum v. c. in $O(n^{\frac{5}{2}} \log^{\frac{3}{2}} n)$ time (if one exists).

3. Every 3-connected 1-plane graph has an embedding-preserving OPVR with vertex complexity at most 12 (a lower bound of 2 can be proved).
An OPVR with minimum v. c. can be computed in $O(n^{\frac{7}{4}} \log^{\frac{1}{2}} n)$ time.

Our contribution

1. Quadratic-time algorithm to test if an embedded graph admits an embedding-preserving OPVR.
Every 1-plane graph has an embedding-preserving OPVR.

2. Drawing algorithm to compute embedding-preserving OPVR with minimum v. c. in $O(n^{\frac{5}{2}} \log^{\frac{3}{2}} n)$ time (if one exists).

3. Every 3-connected 1-plane graph has an embedding-preserving OPVR with vertex complexity at most 12 (a lower bound of 2 can be proved).
An OPVR with minimum v. c. can be computed in $O(n^{\frac{7}{4}} \log^{\frac{1}{2}} n)$ time.

4. $\Omega(n)$ lower bound for the v.c. of 2-connected 1-planar graphs. But the absence of a particular subgraph guarantees v.c. at most 22.

Our contribution

1. Quadratic-time algorithm to test if an embedded graph admits an embedding-preserving OPVR.
Every 1-plane graph has an embedding-preserving OPVR.

2. Drawing algorithm to compute embedding-preserving OPVR with minimum v. c. in $O(n^{\frac{5}{2}} \log^{\frac{3}{2}} n)$ time (if one exists).

3. Every 3-connected 1-plane graph has an embedding-preserving OPVR with vertex complexity at most 12 (a lower bound of 2 can be proved).
An OPVR with minimum v. c. can be computed in $O(n^{\frac{7}{4}} \log^{\frac{1}{2}} n)$ time.

4. $\Omega(n)$ lower bound for the v.c. of 2-connected 1-planar graphs. But the absence of a particular subgraph guarantees v.c. at most 22.

5. Experiments on 1-plane graphs to estimate both the v.c. in practice and the percentage of vertices that are not rectangles.

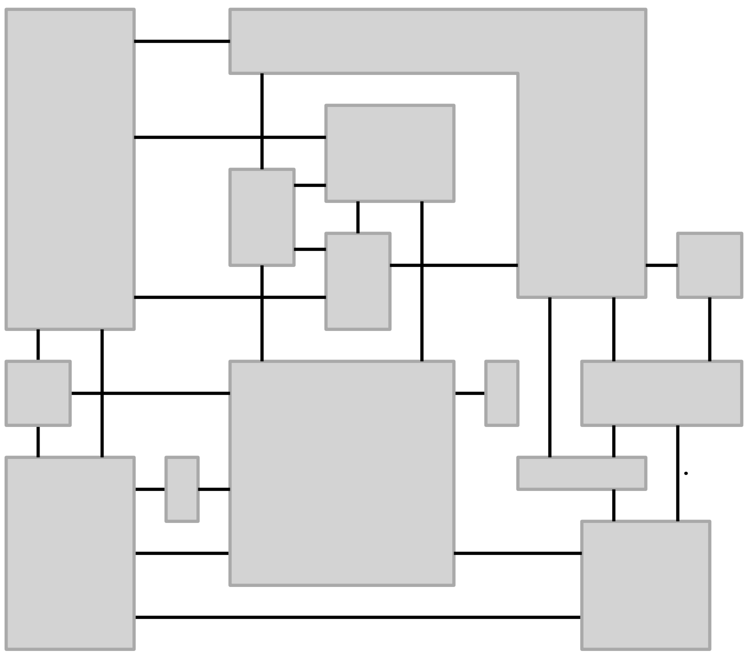
Testing & Minimization for General Embedded Graphs

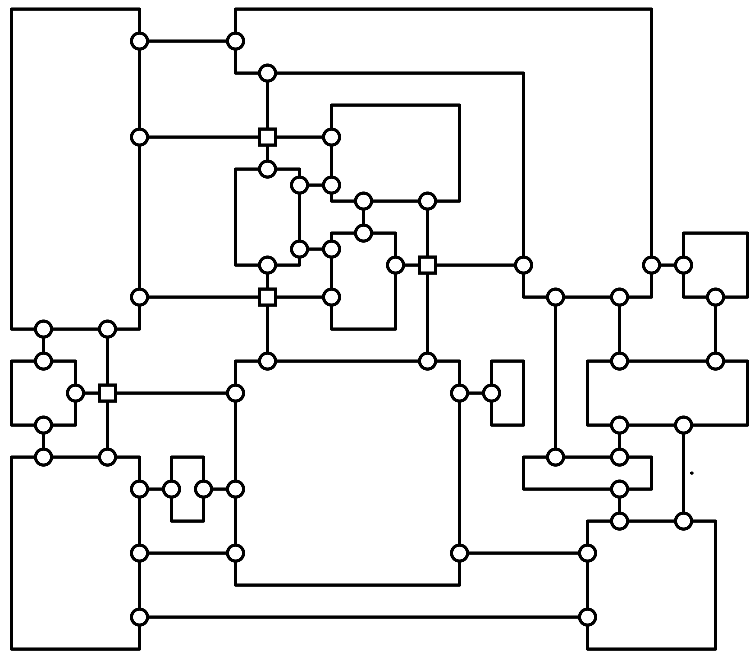
OPVRs and Orthogonal Drawings

Observation 1 *An OPVR with vertex complexity k of an embedded graph G can be regarded as an orthogonal drawing such that:*

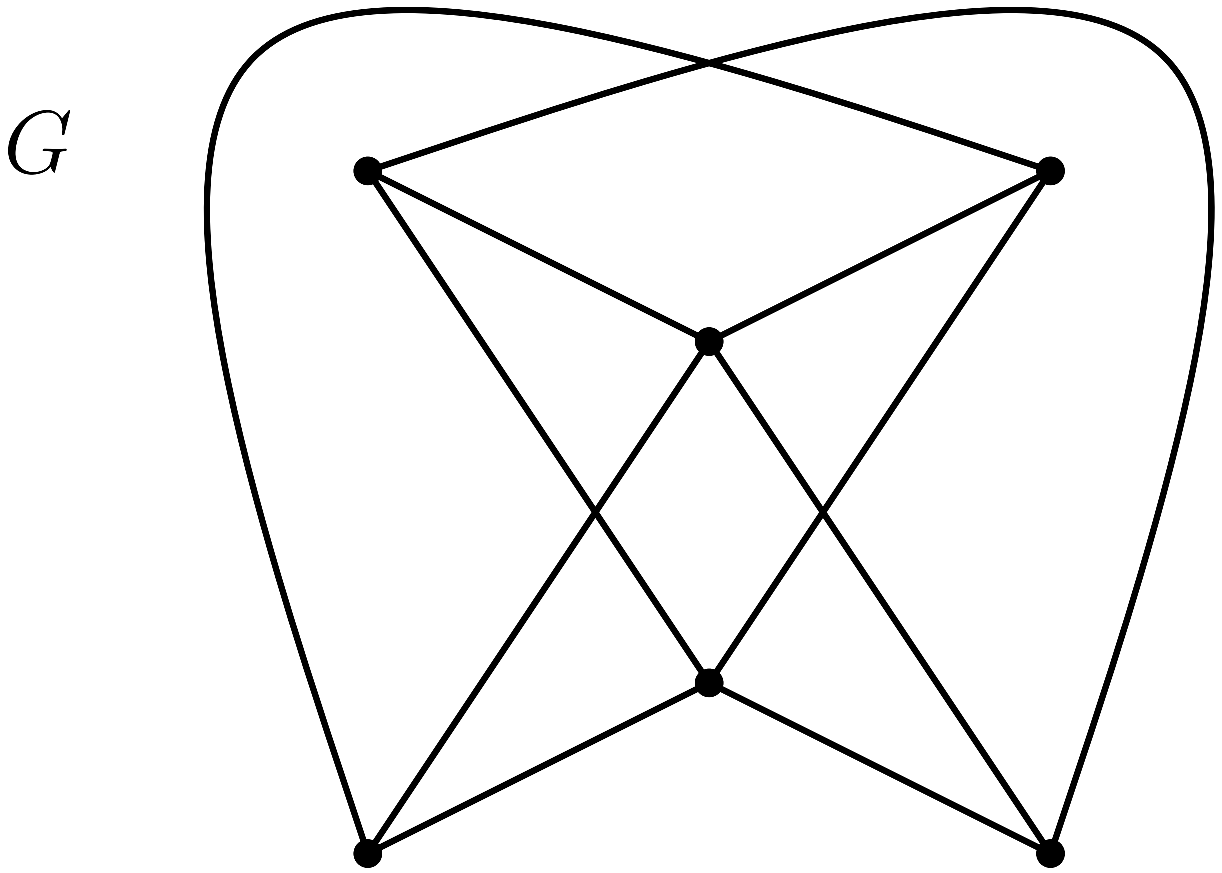
- *ortho-polygons are cycles with at most $2k + 4$ bends*
- *visibilities are straight-line segments (no bends)*

The idea is to test whether G admits a suitable orthogonal drawing, through the topology-shape-metrics framework [Tamassia, 1987]





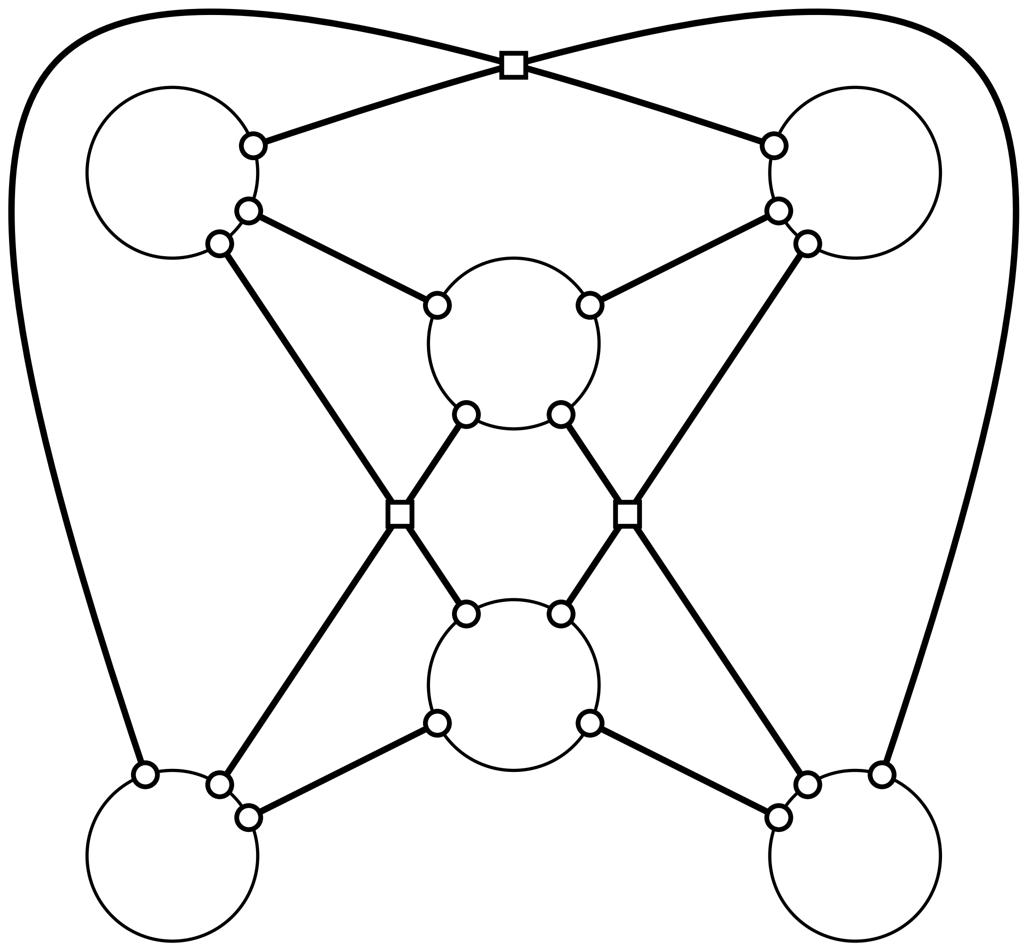
Testing



Input: an embedded graph G

Testing

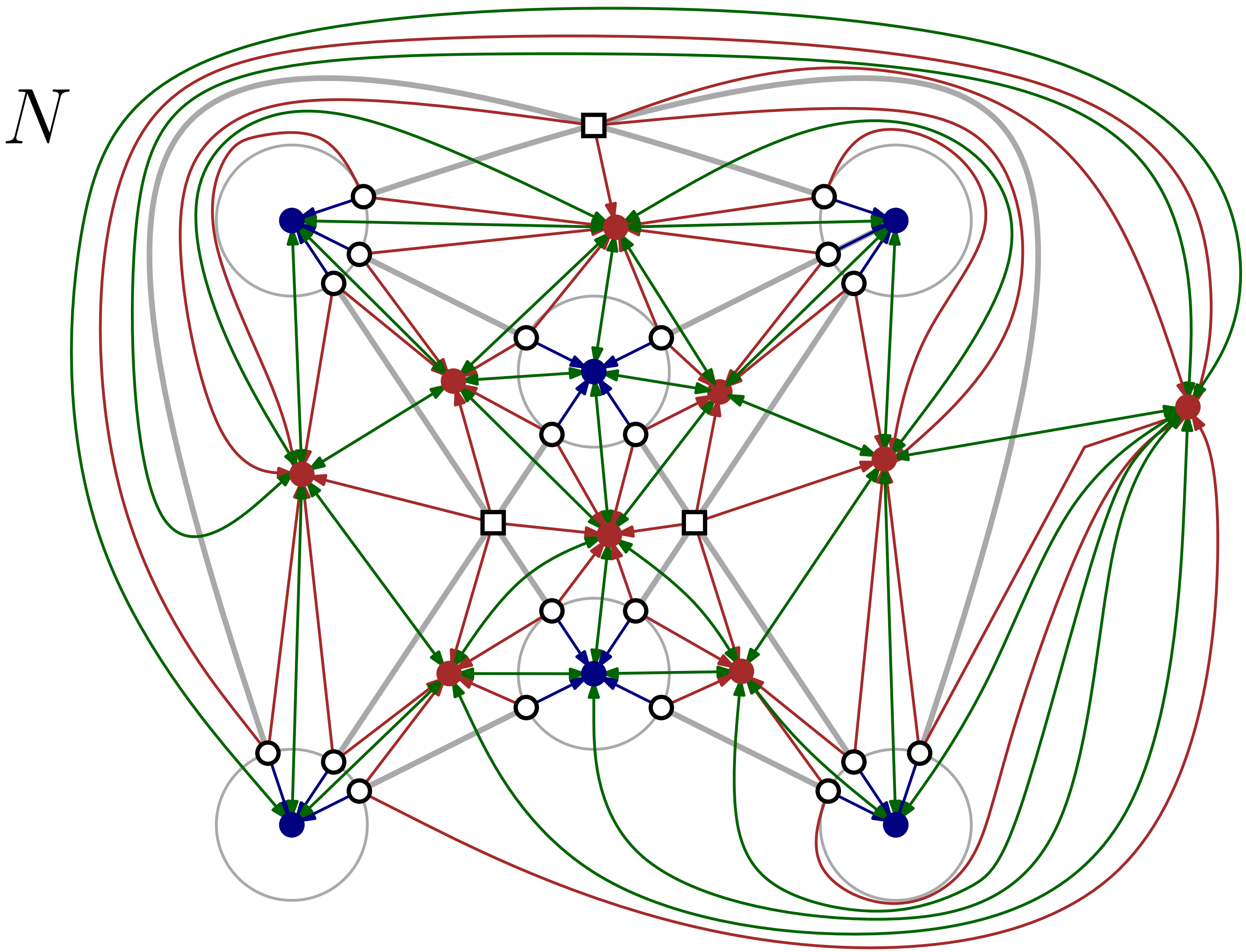
$\overline{G}$



$\overline{G}$ = replace each vertex of G with an expansion cycle and each crossing with a dummy vertex

G admits an OPVR $\iff$ $\overline{G}$ admits an orthogonal drawing such that the edges of G (bold) are bendless

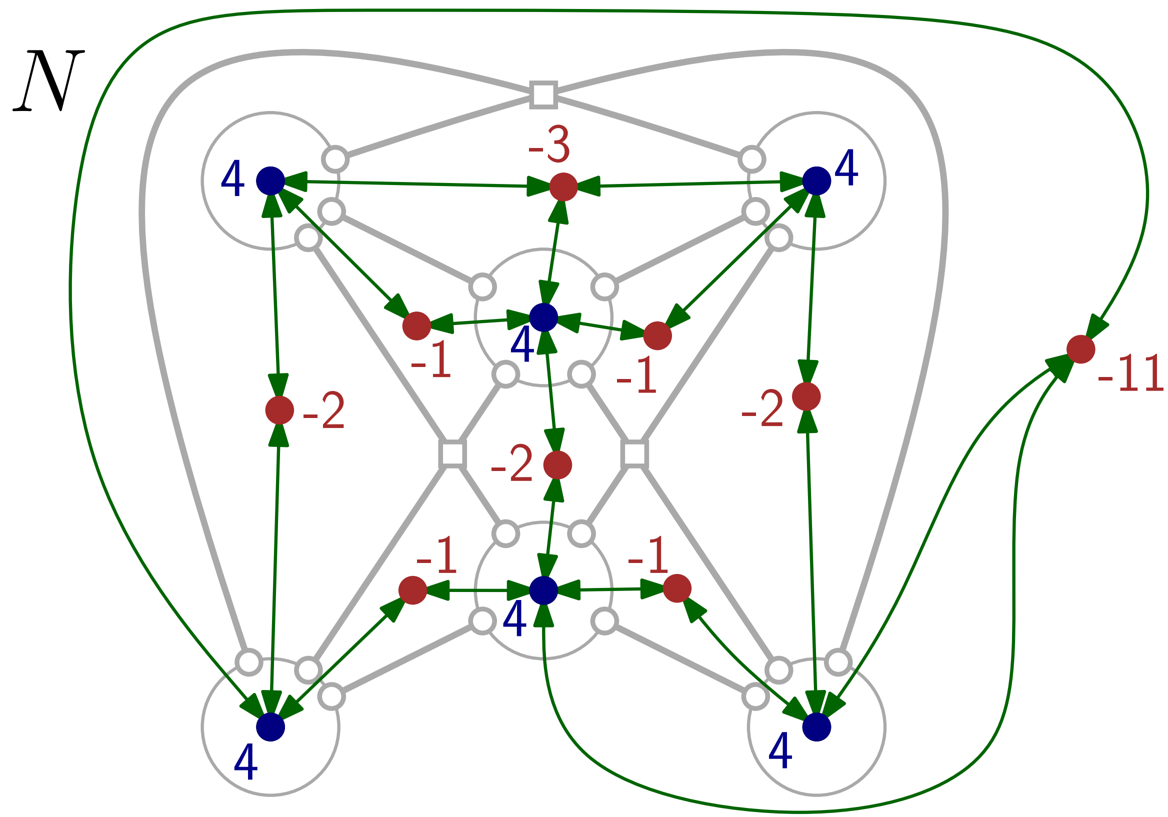
Testing



Construct a flow network N based on the TSM framework

$\overline{G}$ has the desired orthogonal drawing $\iff$ N has a feasible flow.

Testing

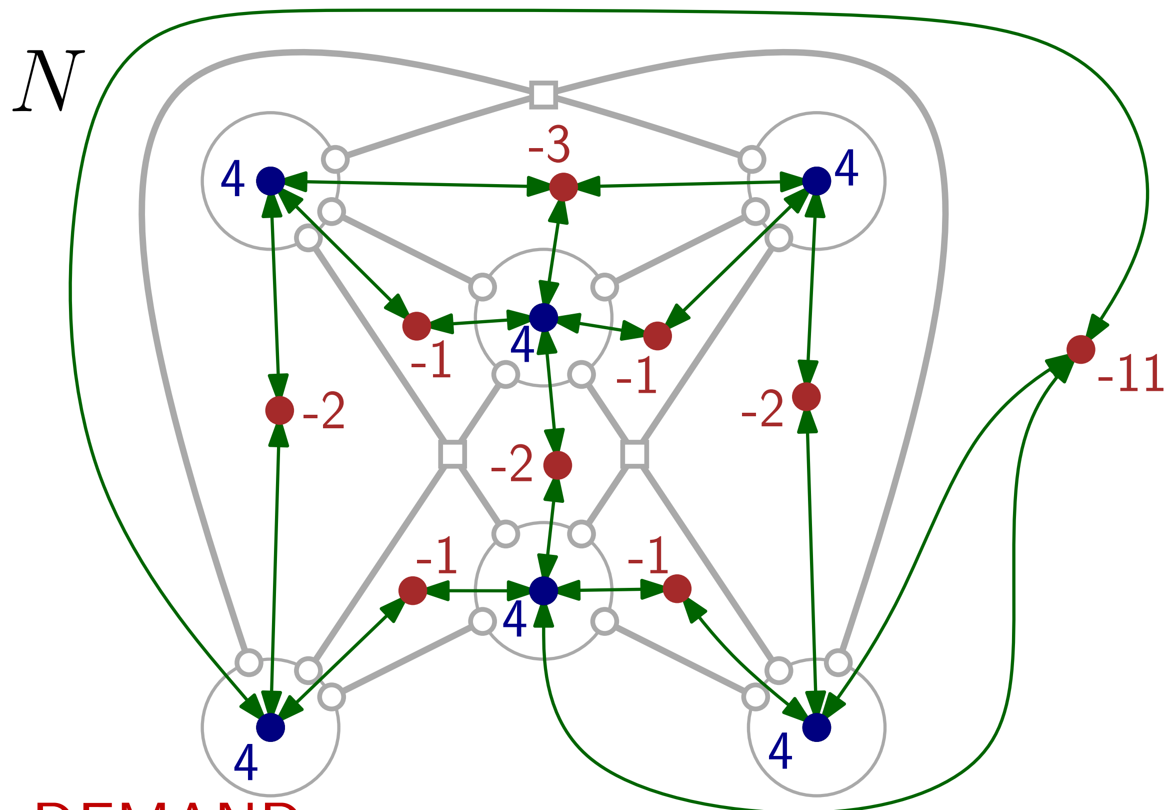


Simplifcations:

1. The original edges of G cannot bend $\rightarrow$ we can remove the corresponding dual edges from the flow network.

2. Each “attaching” vertex supplies 2 units of flow towards its expansion cycle and 1 unit of flow towards each other face. Since the demand of each face is known $\rightarrow$ we can remove vertex-face edges and update the demand of each face

Testing



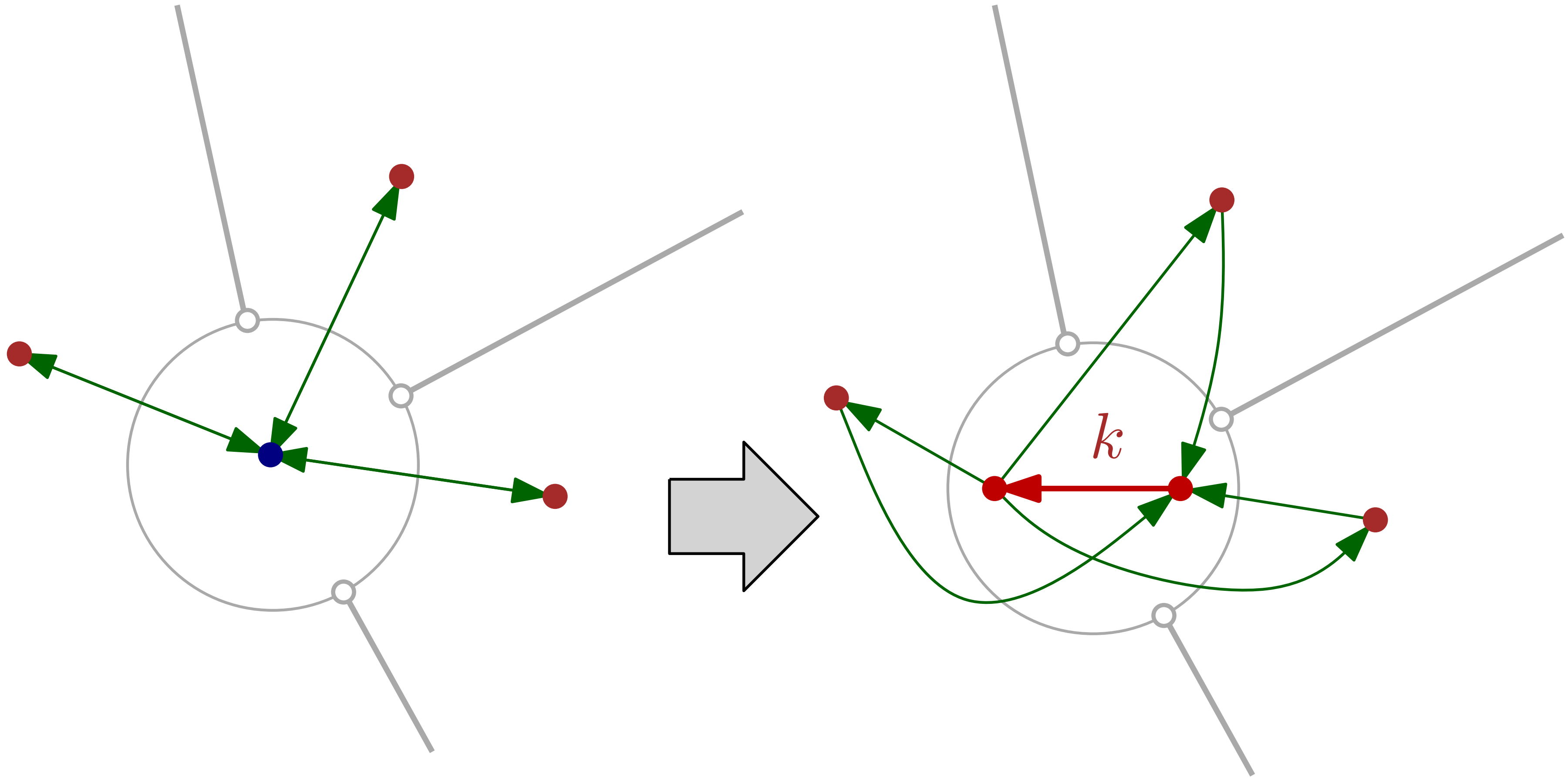
4+4+4+4+4+4-11-3-2-2-2-1-1-1-1=0

The flow network N is uncapacited and undirected. Hence, it has a solution if and only if, for each connected component, the supply is equal to the demand.

→ This condition always holds for 1-plane graphs.

→ We can test if an OPVR of G exists in $O(n^2)$ time.

Minimization



We can use “bottleneck” gadgets to control the amount of flow passing through each expansion cycles.

The bold edge has capacity k to ensure that there will be at most k reflex corners in the corresponding ortho-polygon

Testing & Minimization

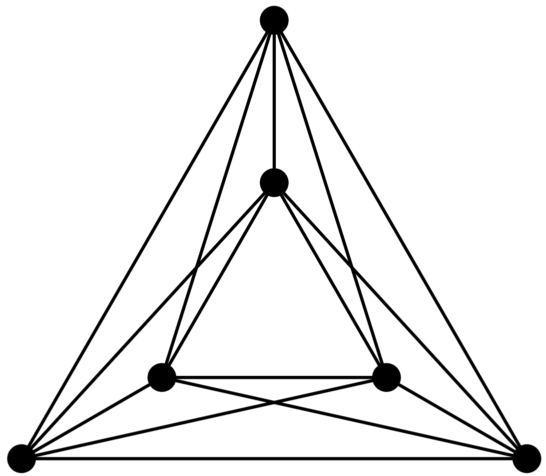
For a fixed k, we apply a min-cost flow algorithm to test if G has an OPVR with v.c. at most k in $O(n^{\frac{5}{2}} \log^{\frac{1}{2}} n)$ [Garg & Tamassia, 1996].

Binary search in the range $[0, 4n]$ to find the minimum value of k s.t. an OPVR of G with v.c. k exists.

Theorem 1 *Let G be an n-vertex embedded graph. There exists an $O(n^2)$-time algorithm that tests if G admits an embedding preserving OPVR and, if so, it computes an embedding preserving OPVR with minimum vertex complexity γ in $O(n^{\frac{5}{2}} \log^{\frac{3}{2}} n)$ time.*

Bounds & Minimization for 1-plane Graphs

1-plane Graphs



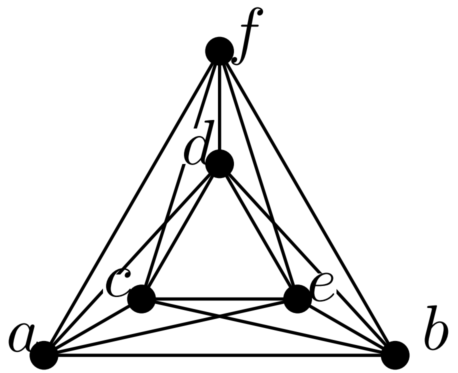
An embedded graph is 1-*plane* if it has at most one crossing per edge.

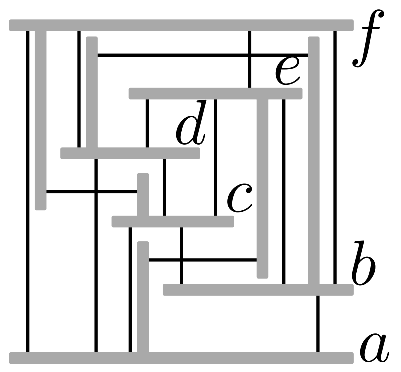
Not all 1-plane graphs admit an embedding-preserving RVR (i.e., an OPVR with vertex complexity 0) [Biedl, Liotta, M., 2016]

Every 1-plane graph has an embedding preserving OPVR.
Can we obtain small vertex complexity?

OPVRs of 1-plane Graphs with Small V.C.

Input: A 3-connected 1-plane graph G
Output: An embedding-preserving OPVR of G with small vertex complexity



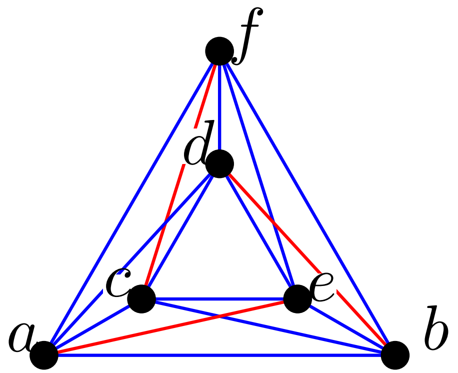


OPVRs of 1-plane Graphs with Small V.C.

Input: A 3-connected 1-plane graph G

Output: An embedding-preserving OPVR of G with small vertex complexity

- Color blue and red the edges of G such that both blue and red edges induce a plane graph, and the red graph has small degree

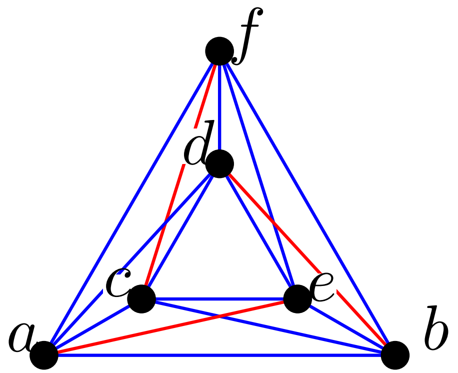


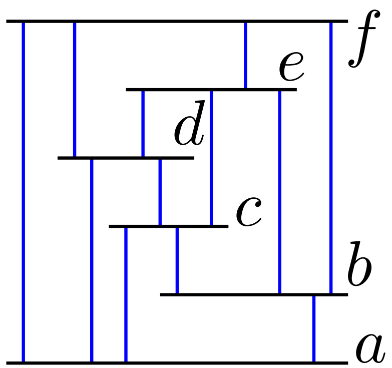
OPVRs of 1-plane Graphs with Small V.C.

Input: A 3-connected 1-plane graph G

Output: An embedding-preserving OPVR of G with small vertex complexity

- Color blue and red the edges of G such that both blue and red edges induce a plane graph, and the red graph has small degree
- Compute a BVR of the blue graph

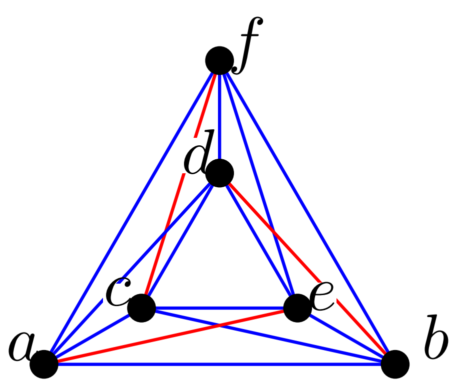


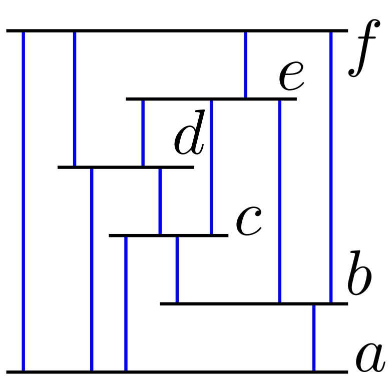


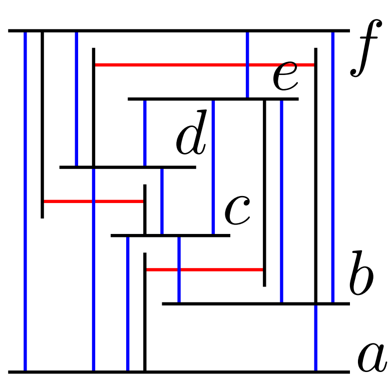
OPVRs of 1-plane Graphs with Small V.C.

Input: A 3-connected 1-plane graph G
Output: An embedding-preserving OPVR of G with small vertex complexity

- Color blue and red the edges of G such that both blue and red edges induce a plane graph, and the red graph has small degree
- Compute a BVR of the blue graph
- Reinsert the red edges by adding (few) vertical bars to each horizontal bar





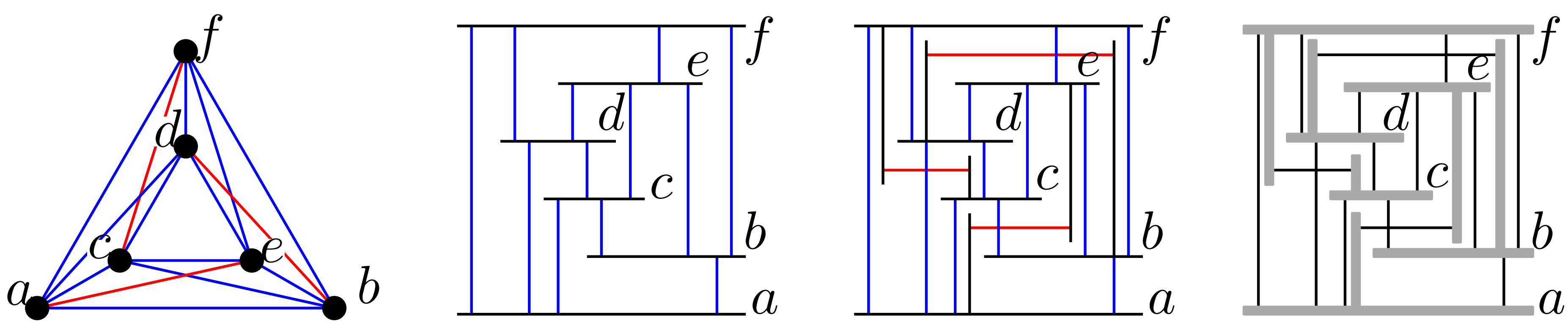


OPVRs of 1-plane Graphs with Small V.C.

Input: A 3-connected 1-plane graph G

Output: An embedding-preserving OPVR of G with small vertex complexity

- Color blue and red the edges of G such that both blue and red edges induce a plane graph, and the red graph has small degree
- Compute a BVR of the blue graph
- Reinsert the red edges by adding (few) vertical bars to each horizontal bar
- Each "rake"-shaped object can be used as the skeleton of an ortho-polygon that has two reflex corners per vertical bar

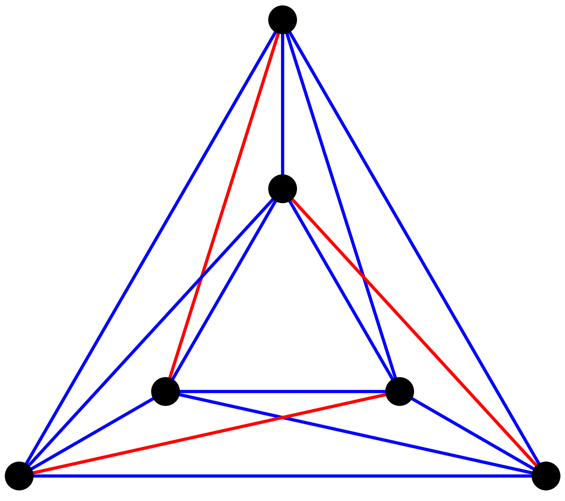


Edge Partitions of 1-plane Graphs

An edge partition of a 1-plane graph G is a coloring of its edges with one of two colors, red and blue, such that both the red graph G_R induced by the red edges and the blue graph G_B induced by the blue edges are plane.

Every 1-plane graph has an edge partition such that G_R is a forest. [Ackerman, 2013]

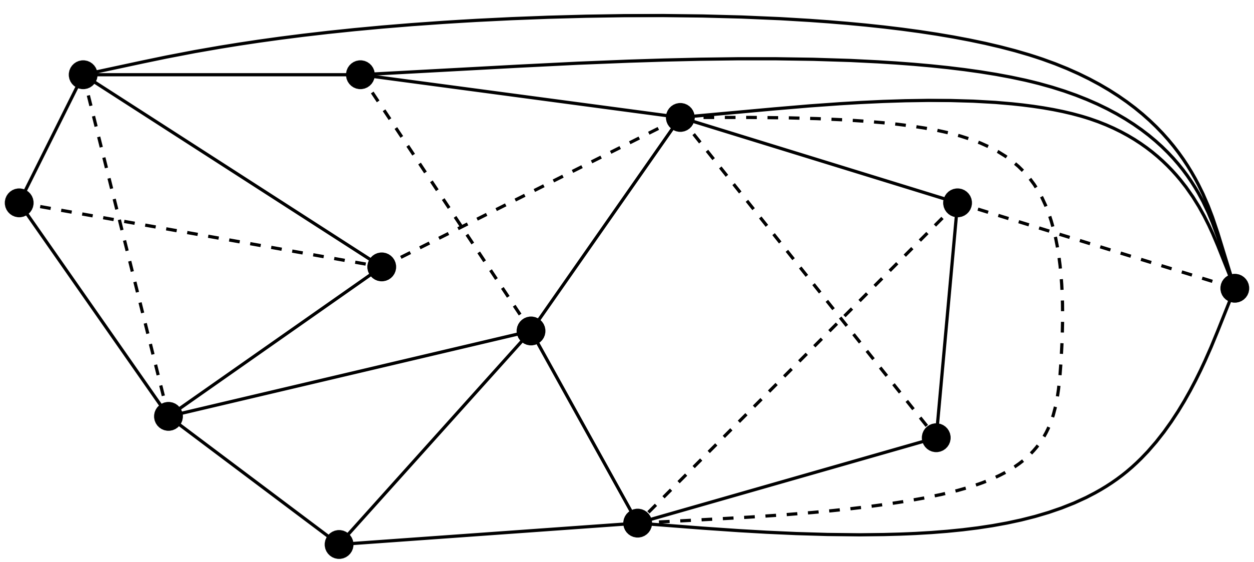
Every optimal (i.e., with $4n - 8$ edges) 1-plane graph has an edge partition such that G_R has maximum vertex degree 4 (worst-case optimal). [Lenhart, Liotta, M., 2015]



Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that* G_R *has maximum vertex degree 6 (worst-case optimal), which can be computed in* $O(n)$ *time.*

Proof sketch:

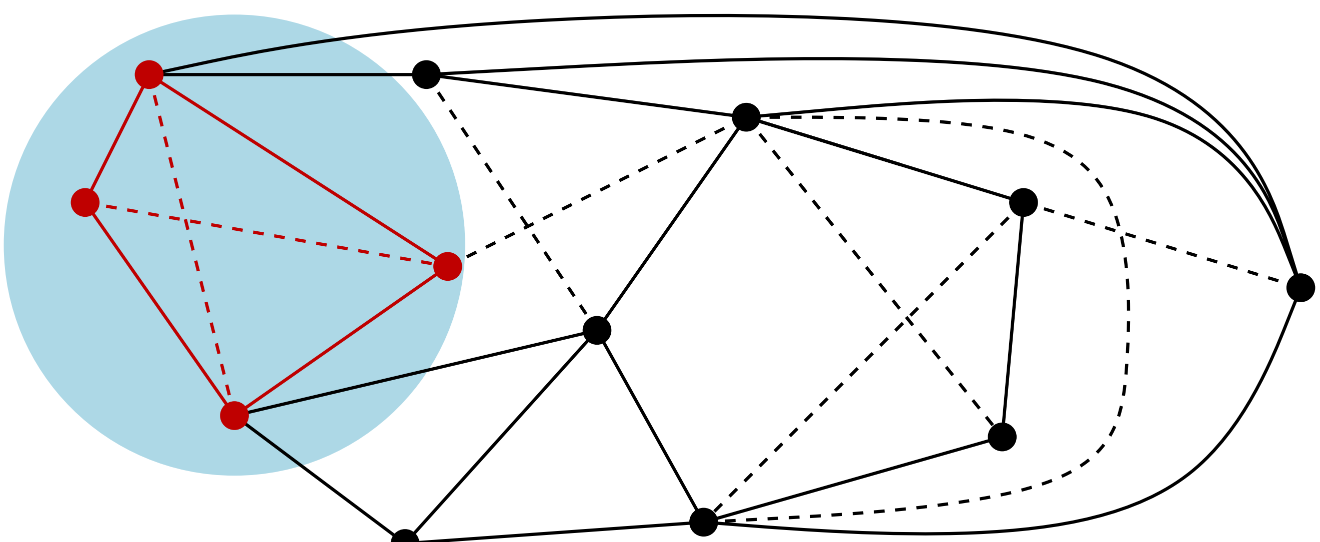


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that G_R has maximum vertex degree 6 (worst-case optimal), which can be computed in $O(n)$ time.*

Proof sketch:

- Augment each crossing to become a K_4

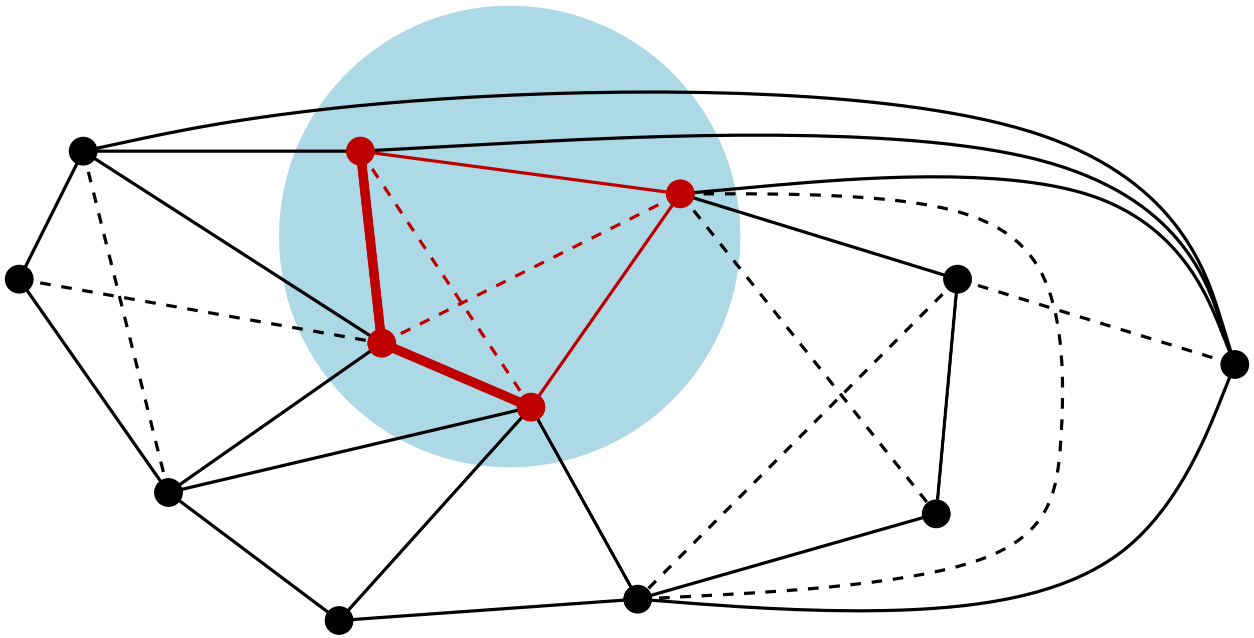


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that G_R has maximum vertex degree 6 (worst-case optimal), which can be computed in $O(n)$ time.*

Proof sketch:

- Augment each crossing to become a K_4

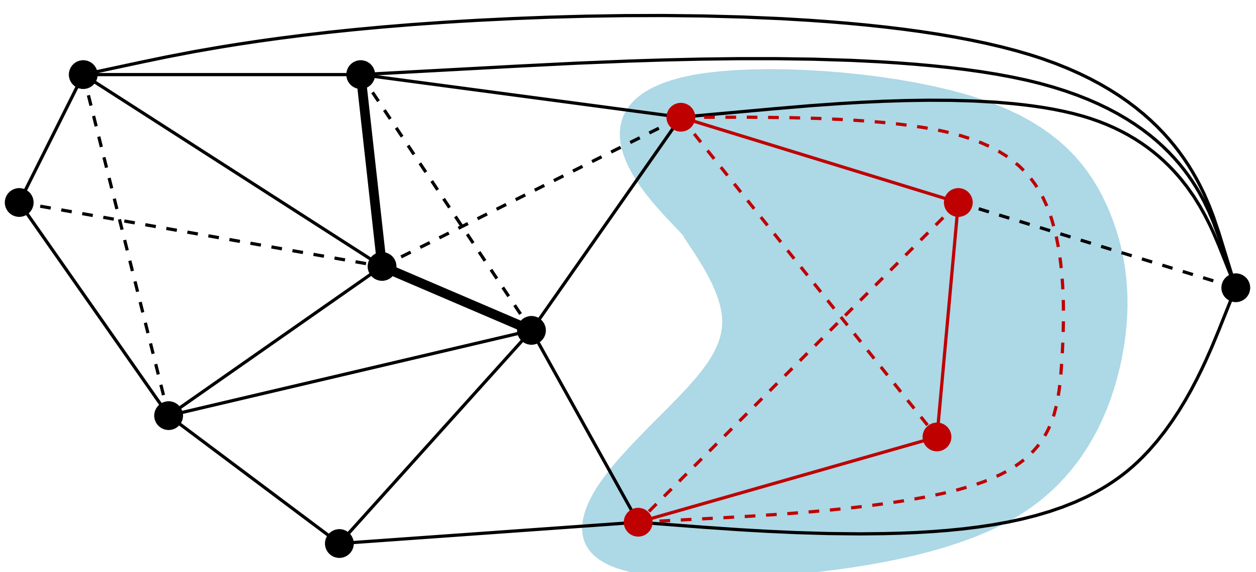


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that G_R has maximum vertex degree 6 (worst-case optimal), which can be computed in $O(n)$ time.*

Proof sketch:

- Augment each crossing to become a K_4

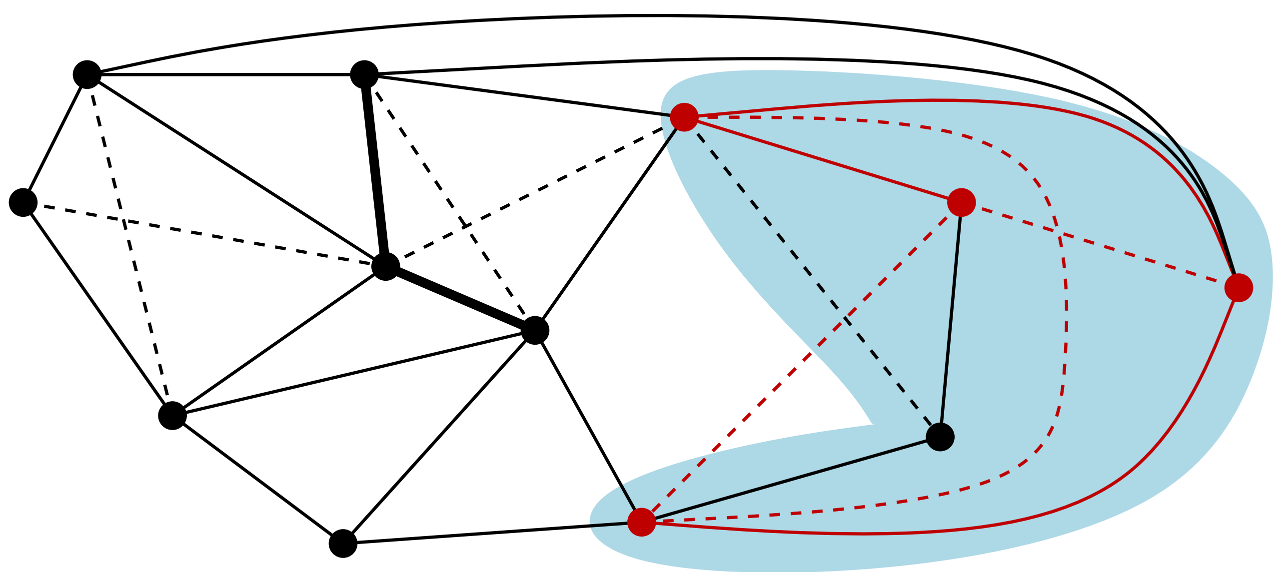


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that G_R has maximum vertex degree 6 (worst-case optimal), which can be computed in $O(n)$ time.*

Proof sketch:

- Augment each crossing to become a K_4

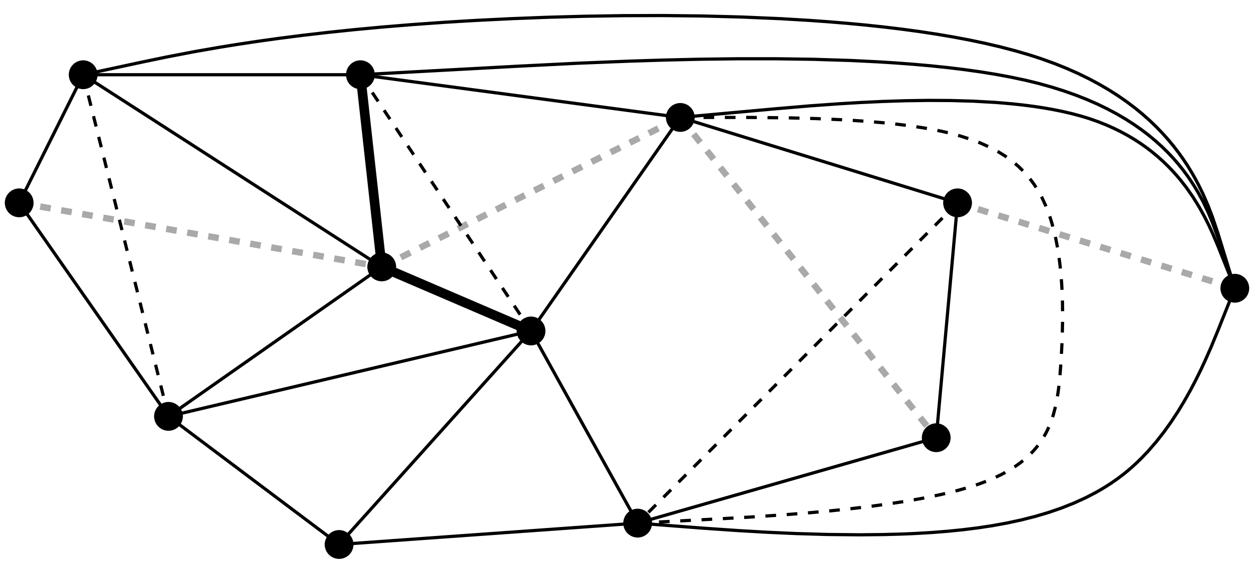


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that G_R has maximum vertex degree 6 (worst-case optimal), which can be computed in $O(n)$ time.*

Proof sketch:

- Augment each crossing to become a K_4
- Remove one special edge from each crossing (the one that does not belong to a different K_4)



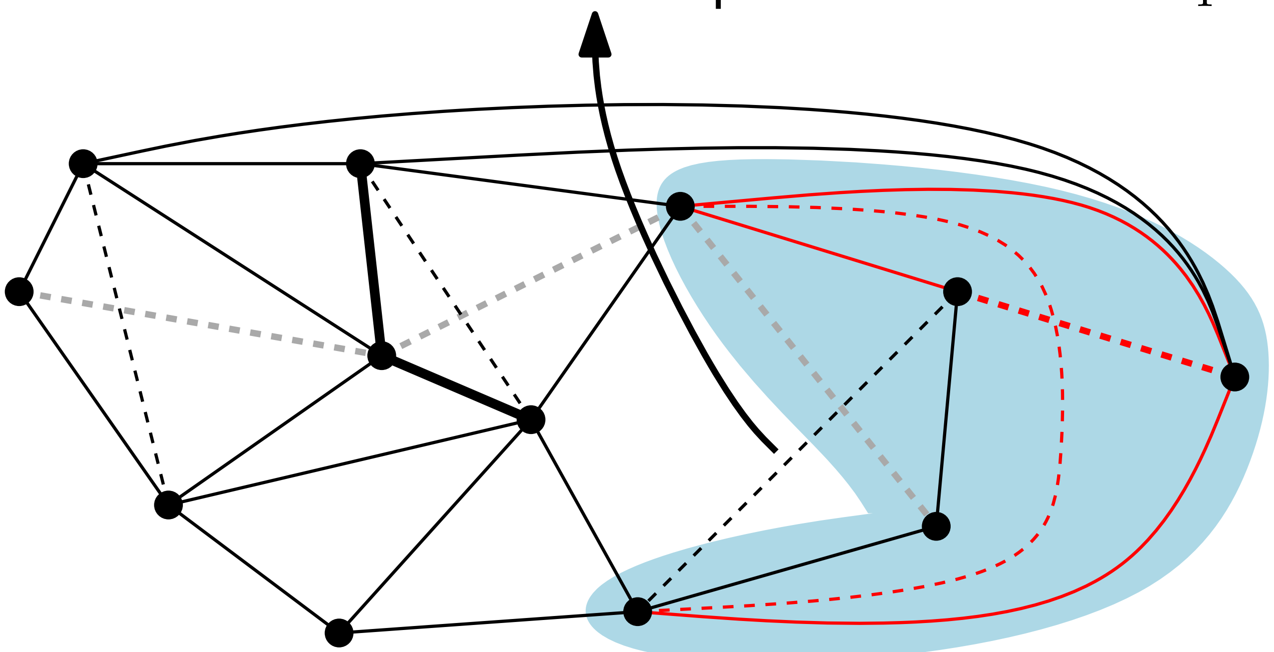
Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that G_R has maximum vertex degree 6 (worst-case optimal), which can be computed in $O(n)$ time.*

Proof sketch:

- Augment each crossing to become a K_4
- Remove one special edge from each crossing (the one that does not belong to a different K_4)

This edge cannot be removed because it is also part of another K_4

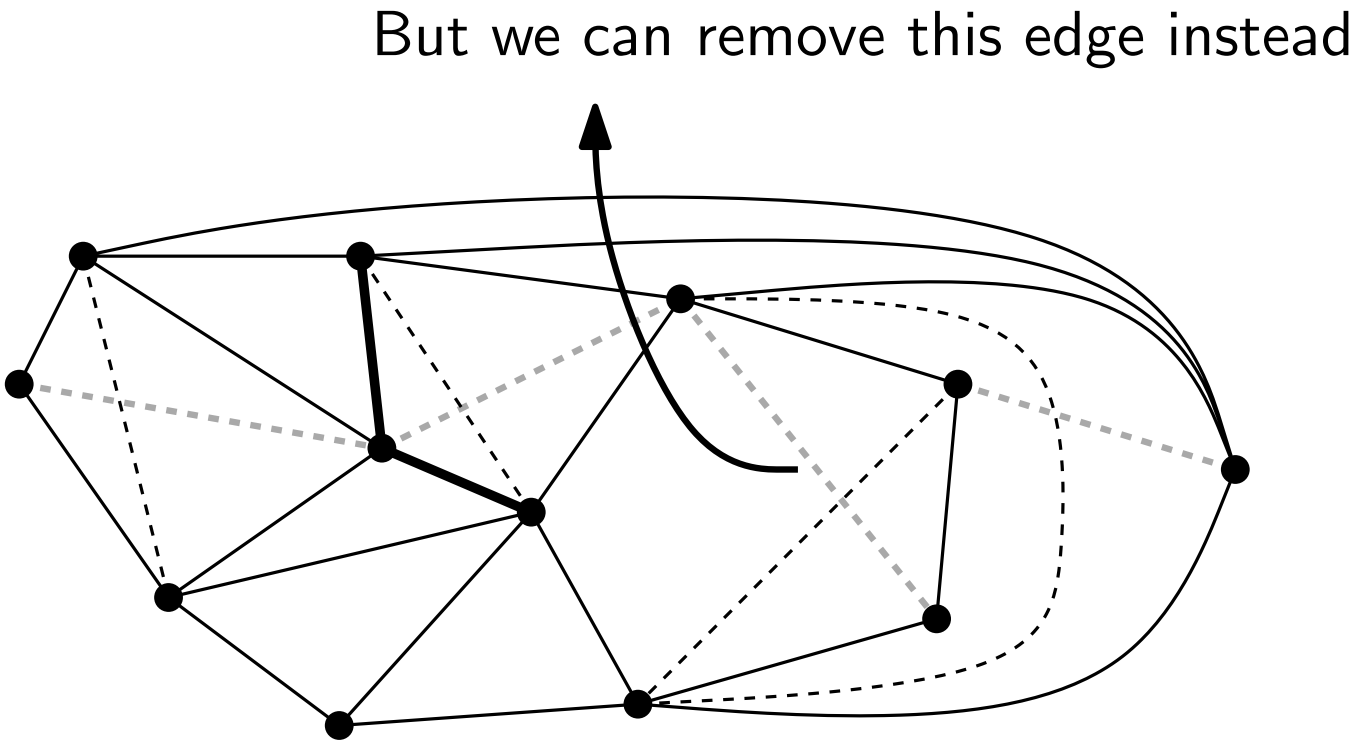


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that G_R has maximum vertex degree 6 (worst-case optimal), which can be computed in $O(n)$ time.*

Proof sketch:

- Augment each crossing to become a K_4
- Remove one special edge from each crossing (the one that does not belong to a different K_4)

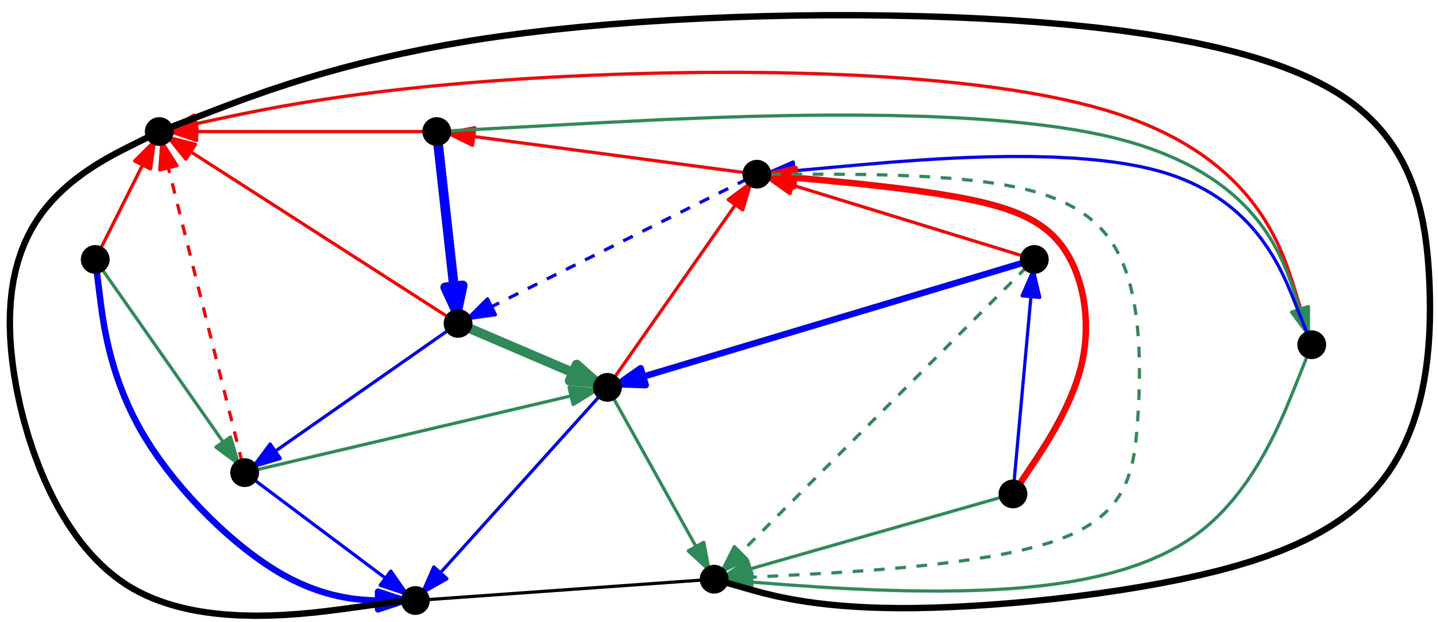


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that* G_R *has maximum vertex degree 6 (worst-case optimal), which can be computed in* $O(n)$ *time.*

Proof sketch:

- Augment each crossing to become a K_4
- Remove one special edge from each crossing (the one that does not belong to a different K_4)
- Triangulate the graph and compute Schnyder trees

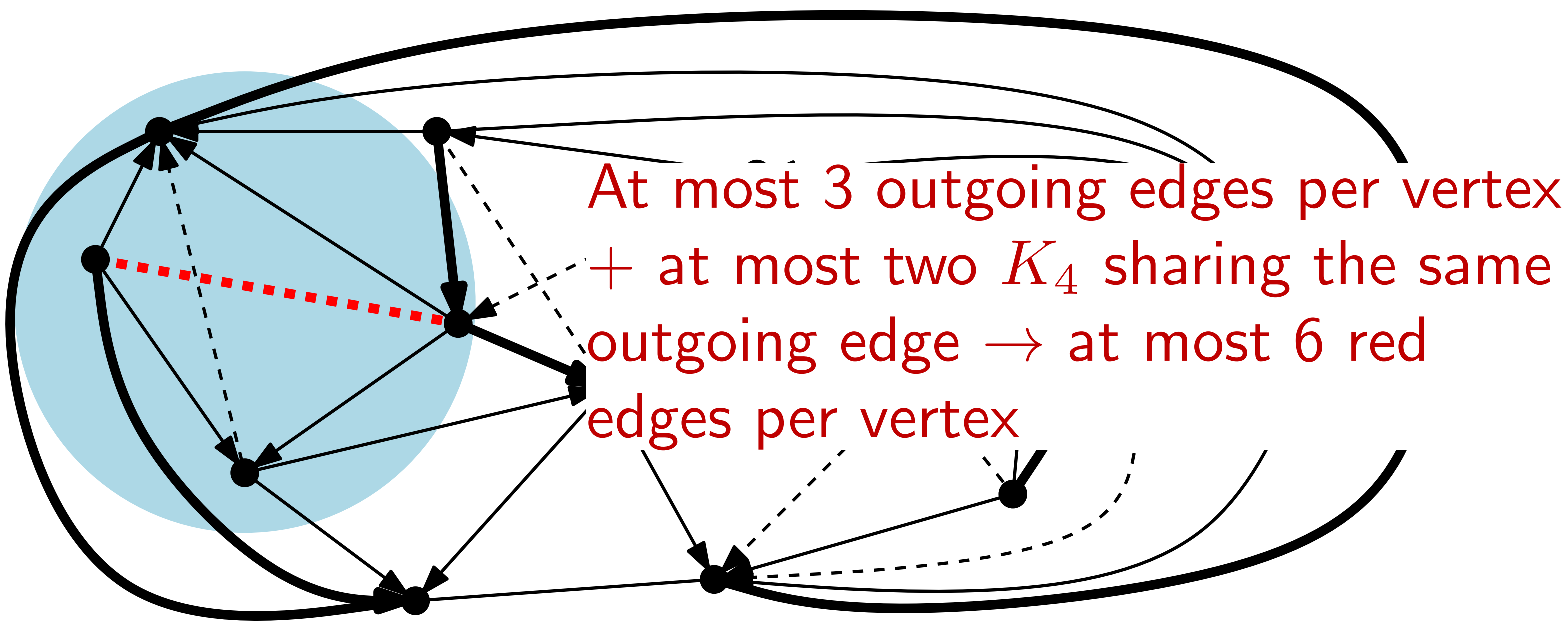


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that G_R has maximum vertex degree 6 (worst-case optimal), which can be computed in $O(n)$ time.*

Proof sketch:

- Augment each crossing to become a K_4
- Remove one special edge from each crossing (the one that does not belong to a different K_4)
- Triangulate the graph and compute Schnyder trees
- Between two crossing edges, color red the one whose endpoints are both incident to an outgoing edge

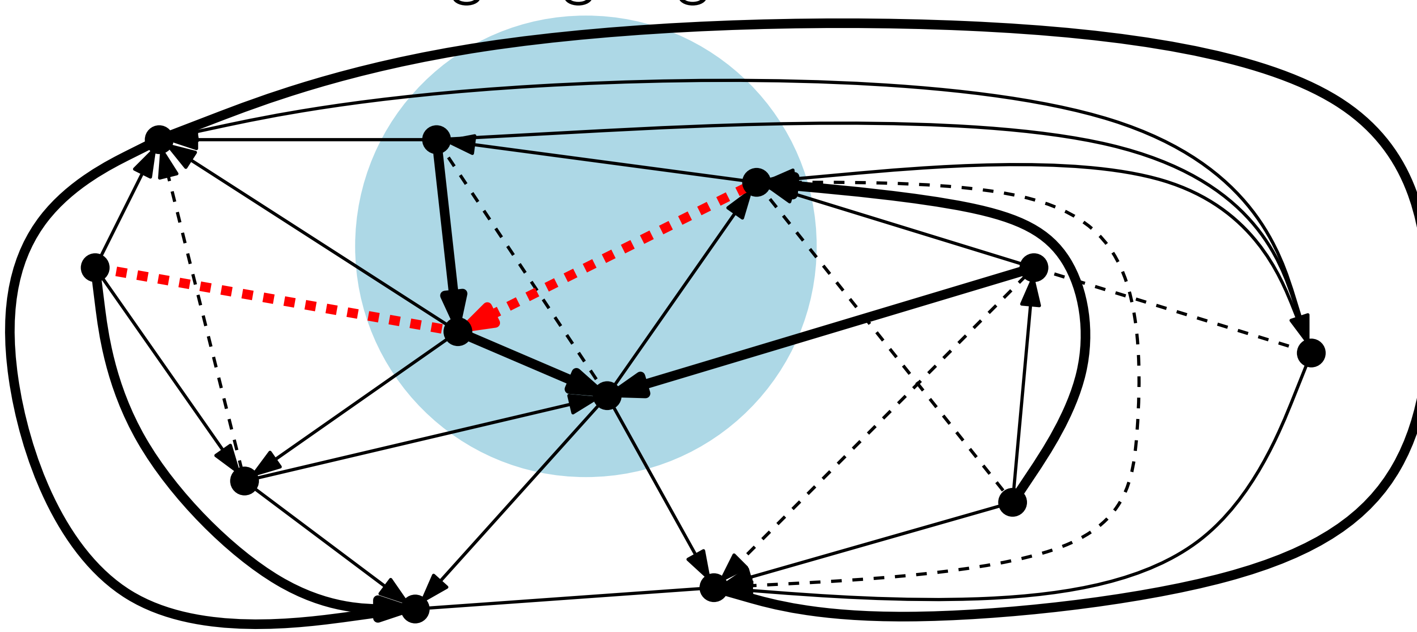


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that G_R has maximum vertex degree 6 (worst-case optimal), which can be computed in $O(n)$ time.*

Proof sketch:

- Augment each crossing to become a K_4
- Remove one special edge from each crossing (the one that does not belong to a different K_4)
- Triangulate the graph and compute Schnyder trees
- Between two crossing edges, color red the one whose endpoints are both incident to an outgoing edge

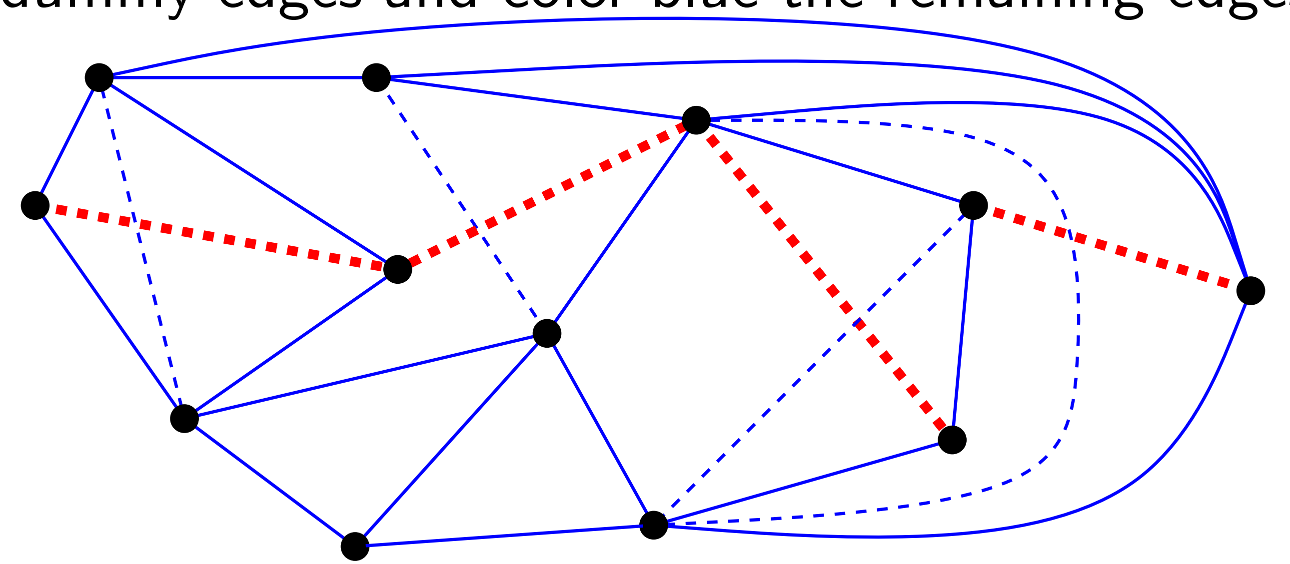


Edge Partitions of 1-plane Graphs

Lemma 1 *Every 3-connected 1-plane graph has an edge partition such that* G_R *has maximum vertex degree 6 (worst-case optimal), which can be computed in* $O(n)$ *time.*

Proof sketch:

- Augment each crossing to become a K_4
- Remove one special edge from each crossing (the one that does not belong to a different K_4)
- Triangulate the graph and compute Schnyder trees
- Between two crossing edges, color red the one whose endpoints are both incident to an outgoing edge of the kite
- Remove dummy edges and color blue the remaining edges



3-connected 1-plane Graphs

Theorem 2 *Let G be a 3-connected 1-plane graph with n vertices. There exists an $O(n)$-time algorithm that computes an an embedding-preserving OPVR of G with vertex complexity at most 12, on an integer grid of size $O(n) \times O(n)$.*

3-connected 1-plane Graphs

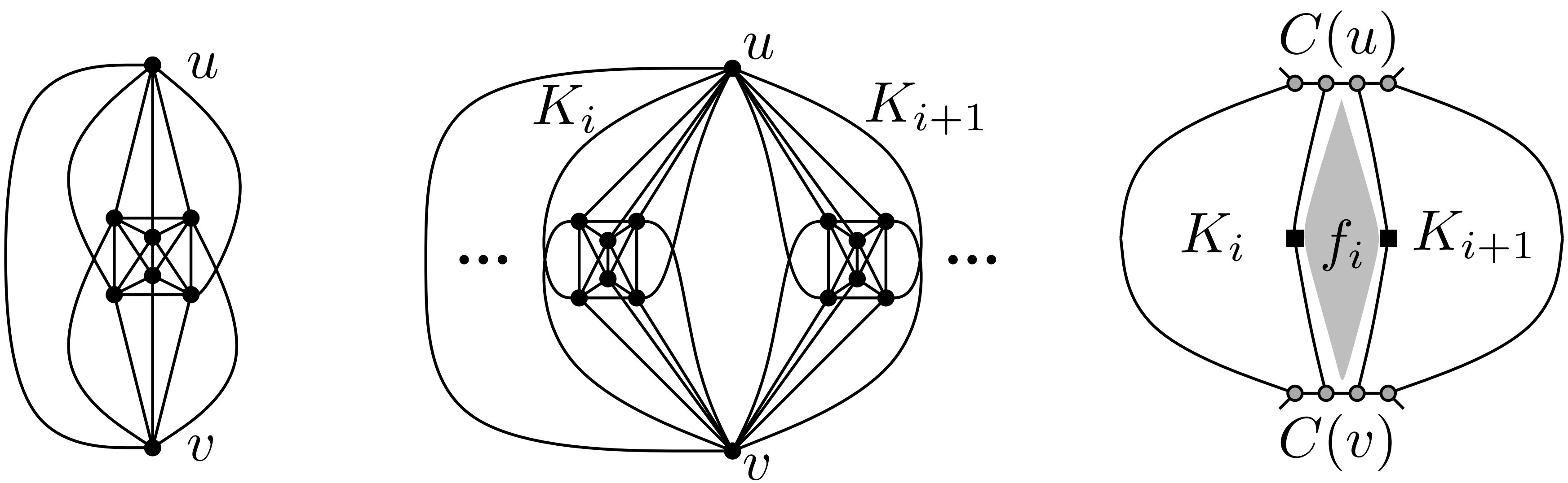
Theorem 2 *Let G be a 3-connected 1-plane graph with n vertices. There exists an $O(n)$-time algorithm that computes an an embedding-preserving OPVR of G with vertex complexity at most 12, on an integer grid of size $O(n) \times O(n)$.*

$\rightarrow$ Small range where to search for the minimum k such that G has an OPVR with vertex complexity k
$\rightarrow$ Maximum cost of the flow in the flow network is $O(n)$

Theorem 3 *Let G be a 3-connected 1-plane graph with n vertices. There exists an $O(n^{\frac{7}{4}}\sqrt{\log n})$-time algorithm that computes an embedding-preserving OPVR γ of G with minimum vertex complexity, on an integer grid of size $O(n) \times O(n)$.*

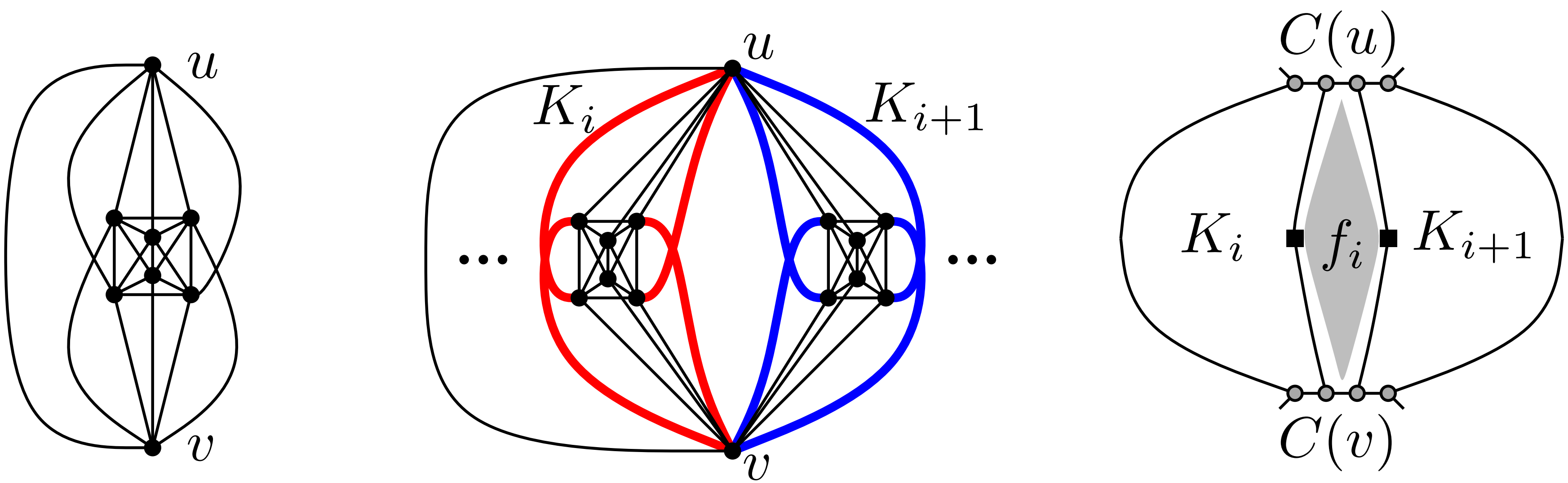
2-connected 1-plane Graphs

Theorem 4 *For every positive integer n, there exists a 2-connected 1-planar graph G with $O(n)$ vertices such that, for every 1-planar embedding of G, any embedding preserving OPVR of G has vertex complexity $\Omega(n)$.*



2-connected 1-plane Graphs

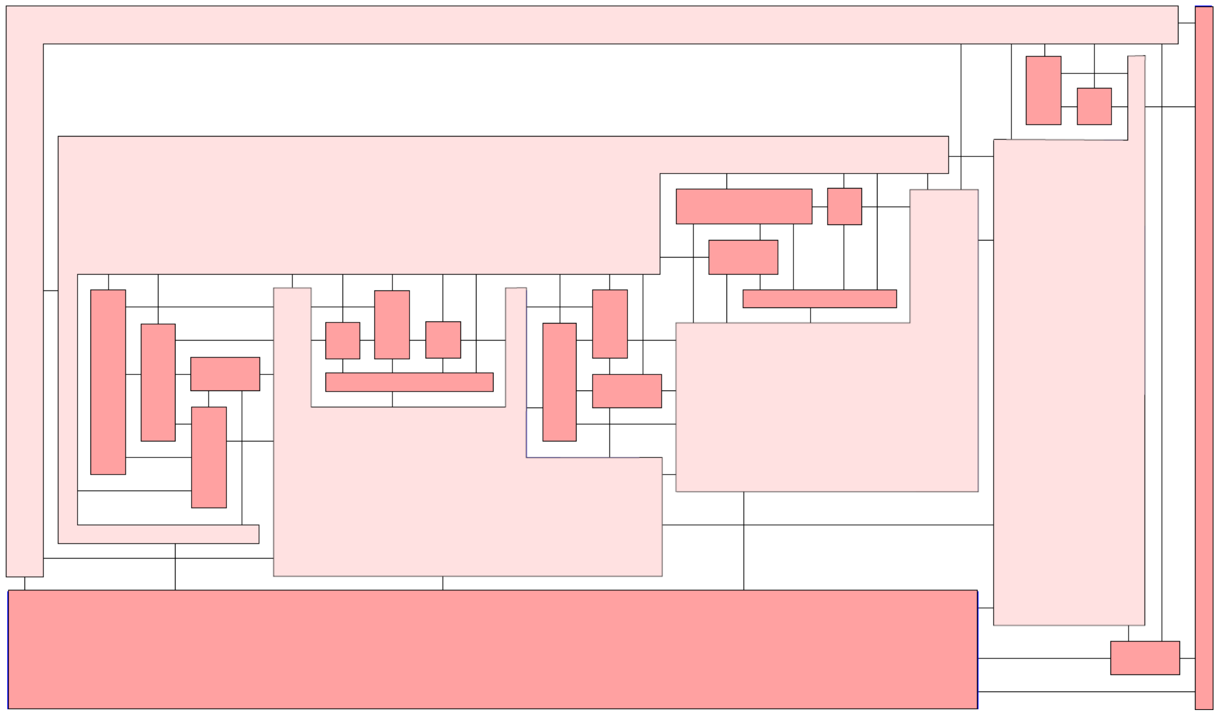
Theorem 4 *For every positive integer n, there exists a 2-connected 1-planar graph G with $O(n)$ vertices such that, for every 1-planar embedding of G, any embedding preserving OPVR of G has vertex complexity $\Omega(n)$.*



Theorem 5 *Let G be a 2-connected 1-plane graph with n vertices and no W-configurations. A 1-planar OPVR of G with vertex complexity at most 22 on an integer grid of size $O(n) \times O(n)$ can be computed in $O(n)$ time.*

Experiments & Open Problems

Experiments



We implemented the minimization algorithm in GDToolkit to compute OPVRs with min. v. c. of a large set of 1-plane graphs ($n \leq 100$).

- All OPVRs computed for 3-connected 1-plane graphs have v.c. ≤ 2 (which is the lower bound we proved).

- All OPVRs computed for 2-connected 1-plane graphs had v. c. ≤ 3.

- More than 75% of the vertices are drawn as rectangles.

Open Problems

OP1: Close the gap between the upper bound (12) and the lower bound (2) on the vertex complexity of OPVRs of 3-connected 1-plane graphs.

Open Problems

OP1: Close the gap between the upper bound (12) and the lower bound (2) on the vertex complexity of OPVRs of 3-connected 1-plane graphs.

OP2: Characterize 2-connected 1-plane graphs with bounded vertex complexity.

Open Problems

OP1: Close the gap between the upper bound (12) and the lower bound (2) on the vertex complexity of OPVRs of 3-connected 1-plane graphs.

OP2: Characterize 2-connected 1-plane graphs with bounded vertex complexity.

OP3: Study the problem of maximizing the number of vertices drawn as rectangles.

Open Problems

OP1: Close the gap between the upper bound (12) and the lower bound (2) on the vertex complexity of OPVRs of 3-connected 1-plane graphs.

OP2: Characterize 2-connected 1-plane graphs with bounded vertex complexity.

OP3: Study the problem of maximizing the number of vertices drawn as rectangles.

THANK YOU