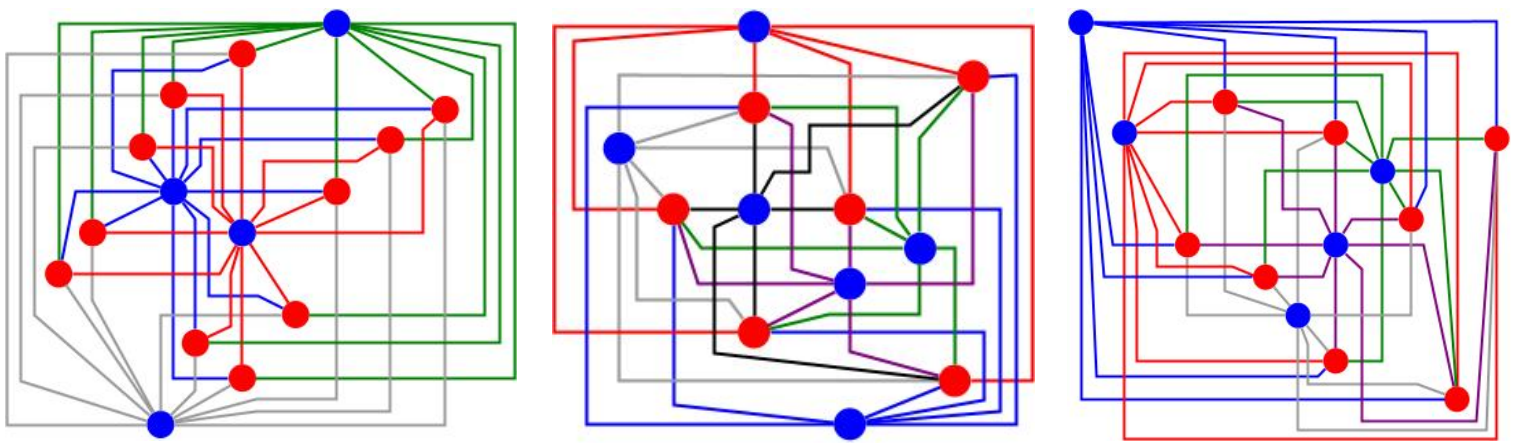
Efficient Generation of Different Topological Representations of Graphs Beyond-Planarity

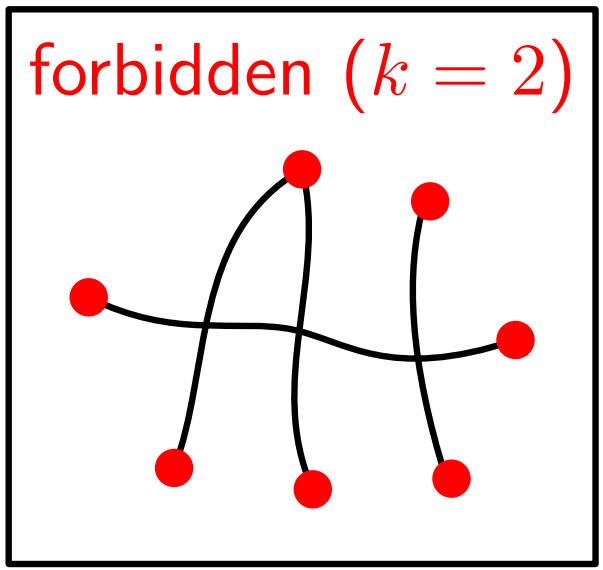
Patrizio Angelini, Michael A. Bekos, Michael Kaufmann, Thomas Schneck

Universität Tübingen



Graph Classes Beyond Planarity

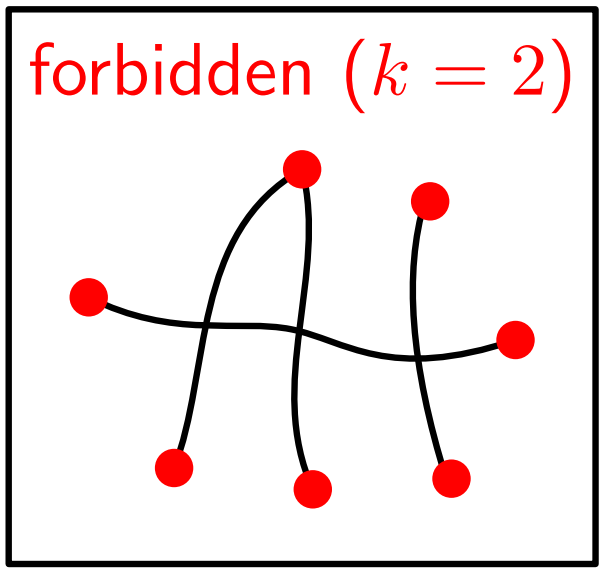
k-planar graphs:
Each edge is crossed at most k times.



$m \leq 4n - 8$ (1-pl)
$m \leq 5n - 10$ (2-pl)

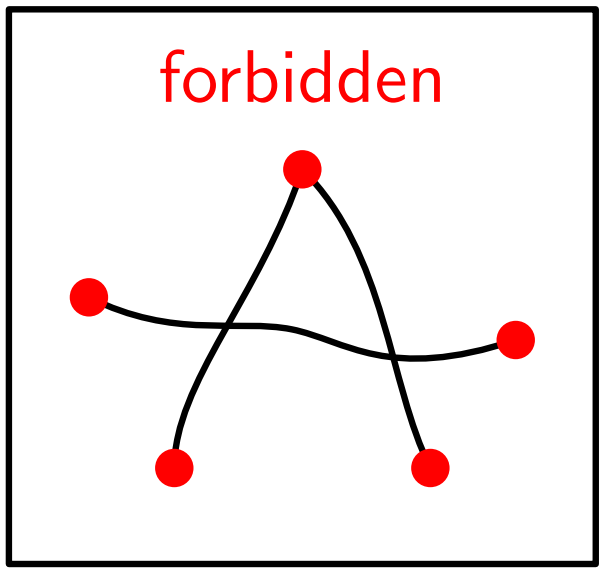
Graph Classes Beyond Planarity

k-planar graphs: Each edge is crossed at most k times.



$m \leq 4n - 8$ (1-pl)
$m \leq 5n - 10$ (2-pl)

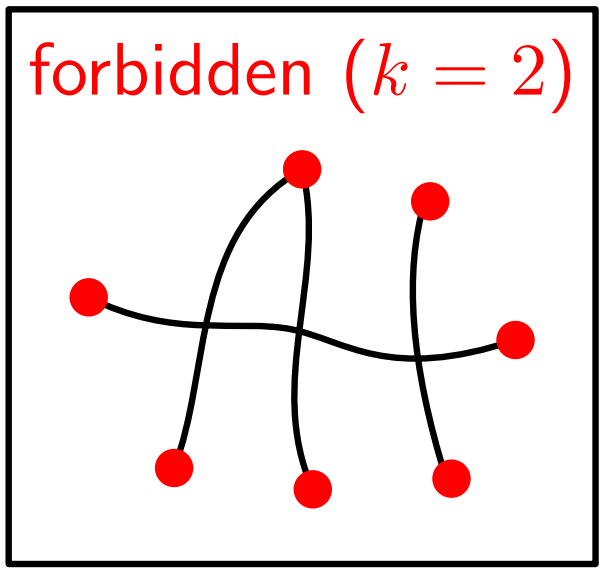
fan-crossing free graphs: An edge is not allowed to cross a fan.



$m \leq 4n - 8$

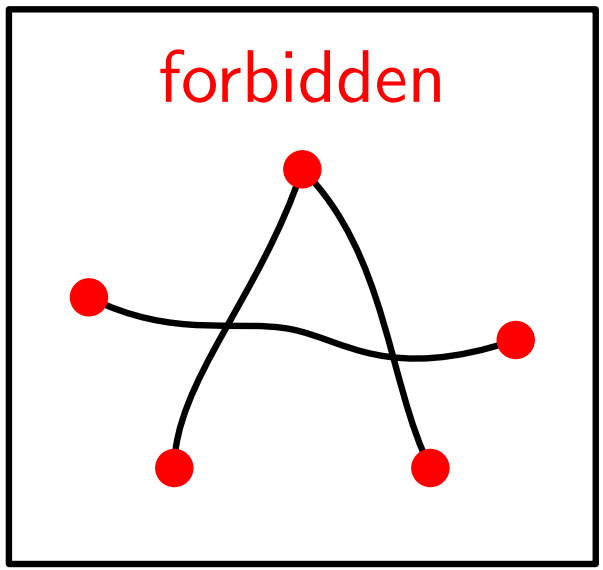
Graph Classes Beyond Planarity

k-planar graphs: Each edge is crossed at most k times.



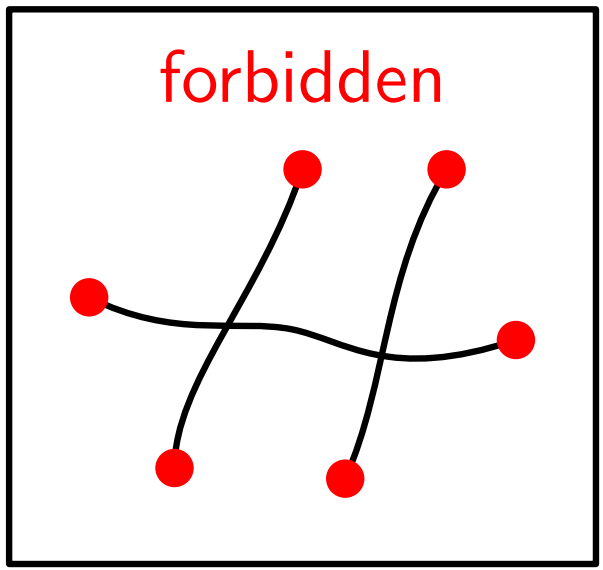
$m \leq 4n - 8$ (1-pl)
$m \leq 5n - 10$ (2-pl)

fan-crossing free graphs: An edge is not allowed to cross a fan.



$m \leq 4n - 8$

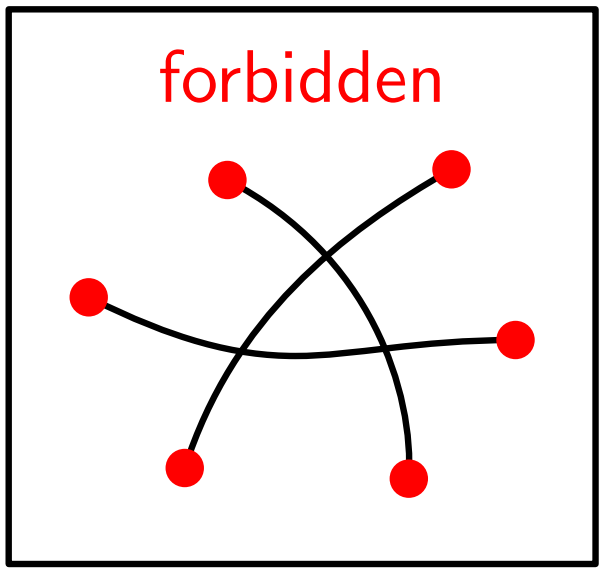
fan-planar graphs: An edge is only allowed to cross a fan.



$m \leq 5n - 10$

Graph Classes Beyond Planarity

quasiplanar graphs:
No three edges cross pairwise.

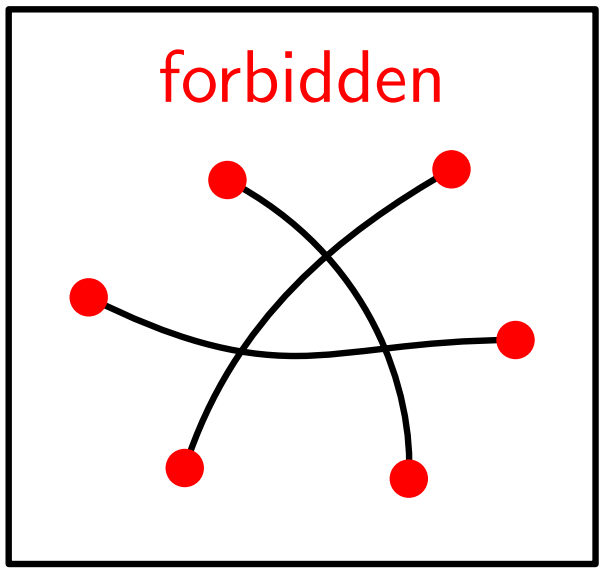


$m \leq 6.5n - 20$

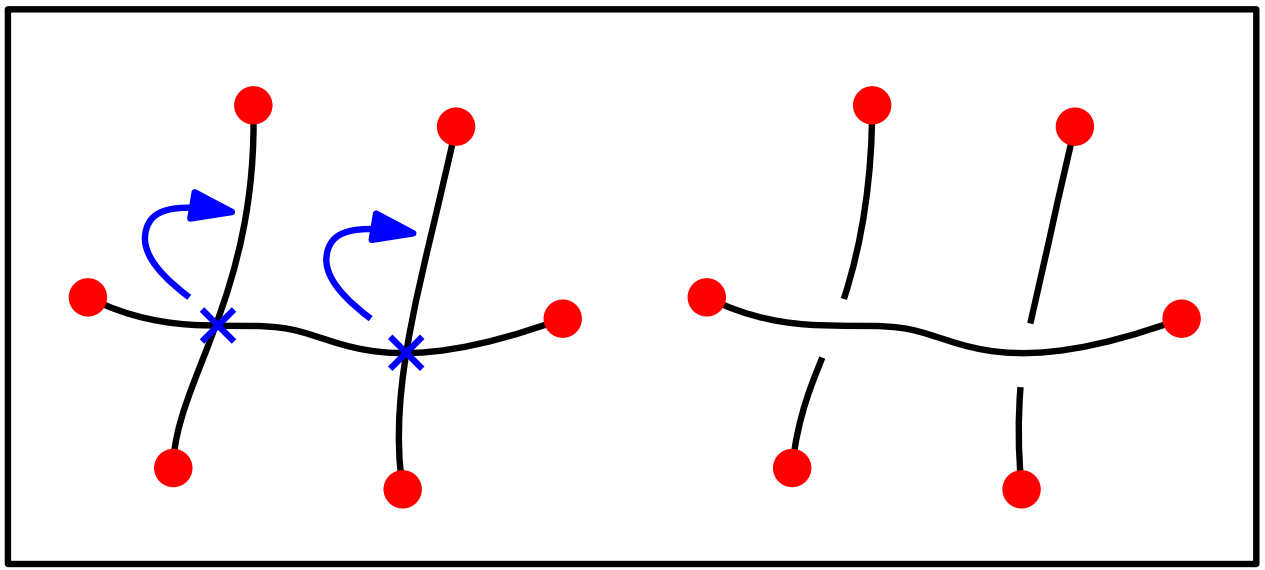
Graph Classes Beyond Planarity

quasiplanar graphs:
No three edges cross pairwise.

gap-planar graphs: Each crossing must be assigned to one of the two crossing edges, and each edge can have at most one crossing assigned to it.



$m \leq 6.5n - 20$



$m \leq 5n - 10$

Motivation

Goal: Decide which complete / complete bipartite graphs belong to a certain graph class $\mathcal{C}$ and which not.

Motivation

Goal: Decide which complete / complete bipartite graphs belong to a certain graph class $\mathcal{C}$ and which not.

- Indicator how powerful a graph class is
  (e.g. K_5 / $K_{3,3}$ not planar)

Motivation

Goal: Decide which complete / complete bipartite graphs belong to a certain graph class $\mathcal{C}$ and which not.

- Indicator how powerful a graph class is
  (e.g. K_5 / $K_{3,3}$ not planar)
- $K_n \in \mathcal{C} \Rightarrow$ chromatic number $\geq n$

Motivation

Goal: Decide which complete / complete bipartite graphs belong to a certain graph class $\mathcal{C}$ and which not.

- Indicator how powerful a graph class is
  (e.g. K_5 / $K_{3,3}$ not planar)
- $K_n \in \mathcal{C} \Rightarrow$ chromatic number $\geq n$
- $K_n \in \mathcal{C}_1$ and $K_n \notin \mathcal{C}_2 \Rightarrow \mathcal{C}_1 \not\subset \mathcal{C}_2$

Motivation

Goal: Decide which complete / complete bipartite graphs belong to a certain graph class $\mathcal{C}$ and which not.

- Indicator how powerful a graph class is
  (e.g. K_5 / $K_{3,3}$ not planar)
- $K_n \in \mathcal{C} \Rightarrow$ chromatic number $\geq n$
- $K_n \in \mathcal{C}_1$ and $K_n \notin \mathcal{C}_2 \Rightarrow \mathcal{C}_1 \not\subset \mathcal{C}_2$
- $K_{a,a} \notin \mathcal{C} \Rightarrow$ not all graphs with max-degree a in $\mathcal{C}$

Methods

- **Density:**
  - 1-planar: $m \leq 4n - 8$
  - K_n: $m = \frac{n(n-1)}{2}$

Methods

- **Density:**
  - 1-planar: $m \leq 4n - 8$
  - K_n: $m = \frac{n(n-1)}{2}$

$\Rightarrow$ K_7 not 1-planar
($m = 21$, $4n - 8 = 20$)

Methods

- **Density:**
  - 1-planar: $m \leq 4n - 8$
  - K_n: $m = \frac{n(n-1)}{2}$

  $\Rightarrow$ K_7 not 1-planar ($m = 21$, $4n - 8 = 20$)

- **Crossing number:**
  - 1-planar: $cr \leq \left\lfloor \frac{m}{2} \right\rfloor$
  - K_n: $cr = \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$

Methods

- **Density:**
  - 1-planar: $m \leq 4n - 8$
  - K_n: $m = \frac{n(n-1)}{2}$

  $\Rightarrow$ K_7 not 1-planar ($m = 21$, $4n - 8 = 20$)

- **Crossing number:**
  - 1-planar: $cr \leq \left\lfloor \frac{m}{2} \right\rfloor$
  - K_n: $cr = \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$

  $\Rightarrow$ K_8 not 1-planar

Methods

- **Density:**
  - 1-planar: $m \leq 4n - 8$
  - K_n: $m = \frac{n(n-1)}{2}$

  $\Rightarrow$ K_7 not 1-planar ($m = 21$, $4n - 8 = 20$)

- **Crossing number:**
  - 1-planar: $cr \leq \left\lfloor \frac{m}{2} \right\rfloor$
  - K_n: $cr = \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$

  $\Rightarrow$ K_8 not 1-planar

- **Limitation:** Not always tight bounds

Methods

- **Density:**
  - 1-planar: $m \leq 4n - 8$
  - K_n: $m = \frac{n(n-1)}{2}$

  $\Rightarrow$ K_7 not 1-planar ($m = 21$, $4n - 8 = 20$)

- **Crossing number:**
  - 1-planar: $cr \leq \left\lfloor \frac{m}{2} \right\rfloor$
  - K_n: $cr = \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$

  $\Rightarrow$ K_8 not 1-planar

- **Limitation:** Not always tight bounds

- **Our approach:**

Methods

- **Density:**
  - 1-planar: $m \leq 4n - 8$
  - K_n: $m = \frac{n(n-1)}{2}$

  $\Rightarrow$ K_7 not 1-planar ($m = 21$, $4n - 8 = 20$)

- **Crossing number:**
  - 1-planar: $cr \leq \left\lfloor \frac{m}{2} \right\rfloor$
  - K_n: $cr = \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$

  $\Rightarrow$ K_8 not 1-planar

- **Limitation:** Not always tight bounds

- **Our approach:**
  - Create all non-isomorphic drawings.

Methods

- **Density:**
  - 1-planar: $m \leq 4n - 8$
  - K_n: $m = \frac{n(n-1)}{2}$

  $\Rightarrow$ K_7 not 1-planar ($m = 21$, $4n - 8 = 20$)

- **Crossing number:**
  - 1-planar: $cr \leq \left\lfloor \frac{m}{2} \right\rfloor$
  - K_n: $cr = \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$

  $\Rightarrow$ K_8 not 1-planar

- **Limitation:** Not always tight bounds

- **Our approach:**
  - Create all non-isomorphic drawings.
  - Large case analysis.

Methods

- **Density:**
  - 1-planar: $m \leq 4n - 8$
  - K_n: $m = \frac{n(n-1)}{2}$

  $\Rightarrow$ K_7 not 1-planar ($m = 21$, $4n - 8 = 20$)

- **Crossing number:**
  - 1-planar: $cr \leq \left\lfloor \frac{m}{2} \right\rfloor$
  - K_n: $cr = \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$

  $\Rightarrow$ K_8 not 1-planar

- **Limitation:** Not always tight bounds

- **Our approach:**
  - Create all non-isomorphic drawings.
  - Large case analysis.
  - Computer program can do it.

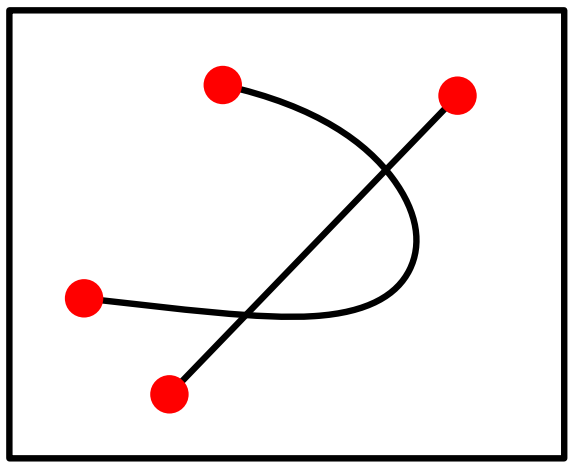
Assumption: Simplicity

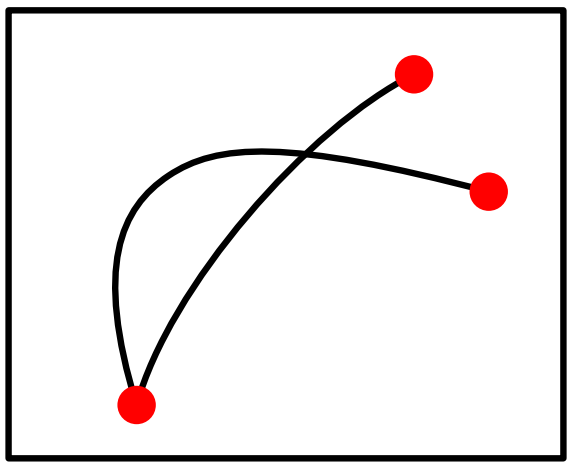
- Two different edges have at most one point in common: Either an endpoint or a crossing.

Assumption: Simplicity

- Two different edges have at most one point in common: Either an endpoint or a crossing.

Not allowed:

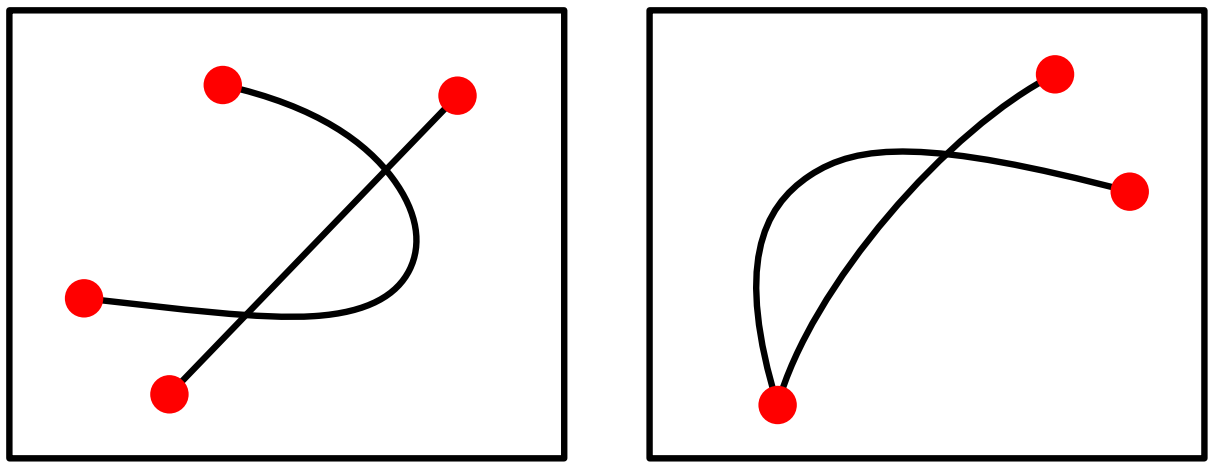




Assumption: Simplicity

- Two different edges have at most one point in common: Either an endpoint or a crossing.
- No edge crosses itself.

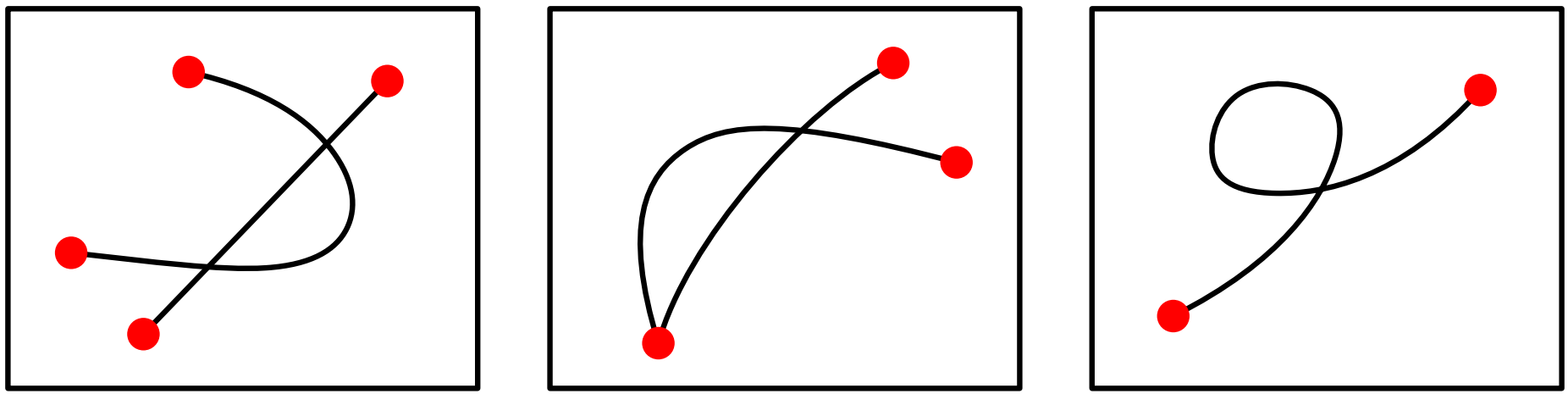
Not allowed:



Assumption: Simplicity

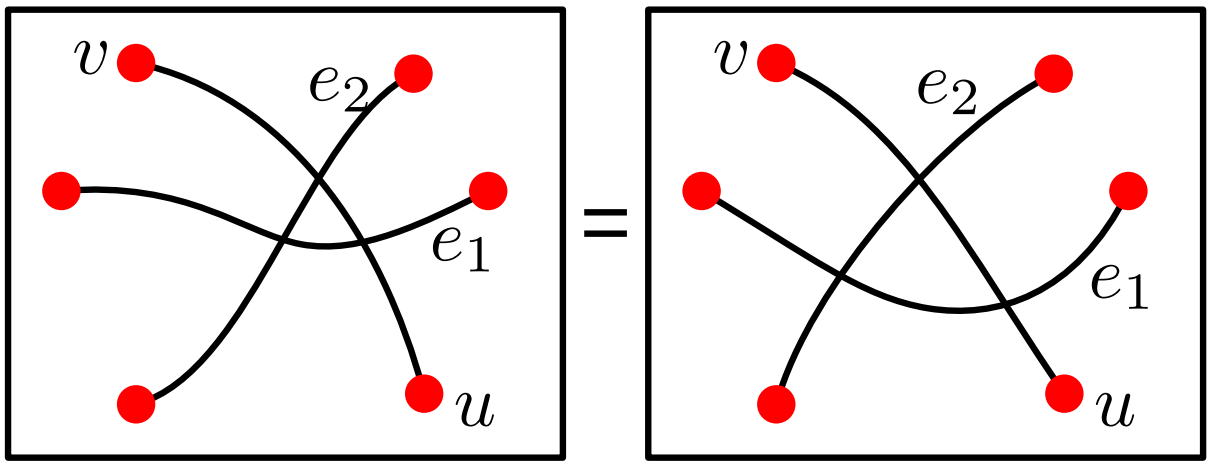
- Two different edges have at most one point in common: Either an endpoint or a crossing.
- No edge crosses itself.

Not allowed:



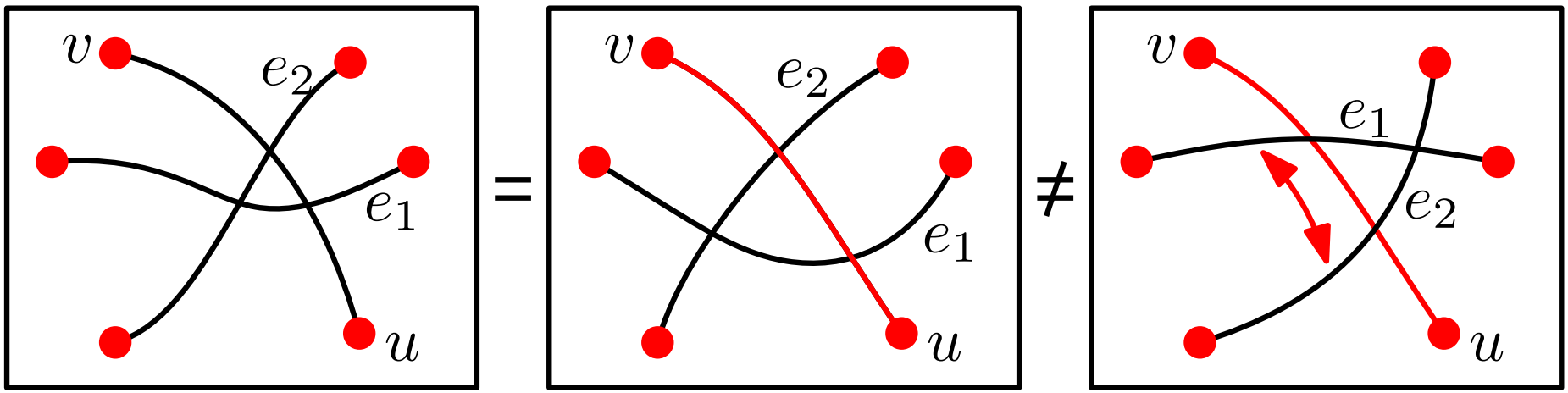
Definitions

- Two drawings are *equivalent*: for each edge the order of the crossing edges is the same



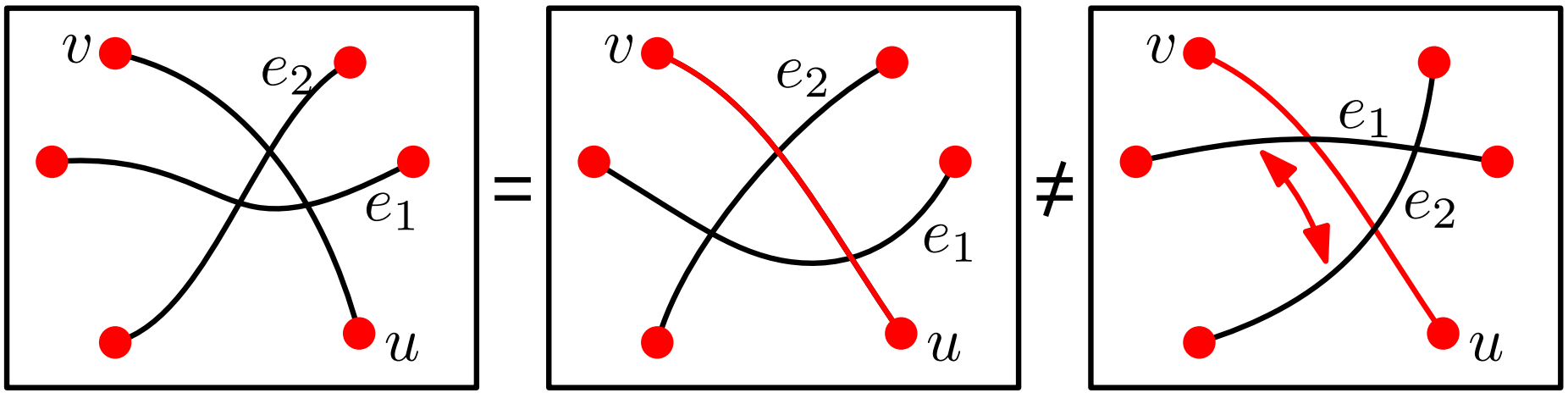
Definitions

- Two drawings are *equivalent*: for each edge the order of the crossing edges is the same

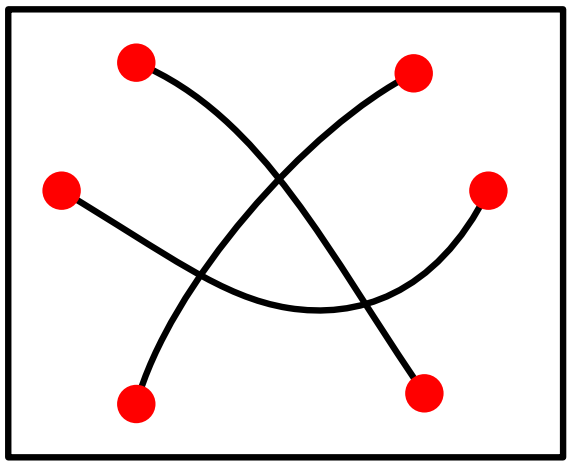


Definitions

- Two drawings are *equivalent*: for each edge the order of the crossing edges is the same

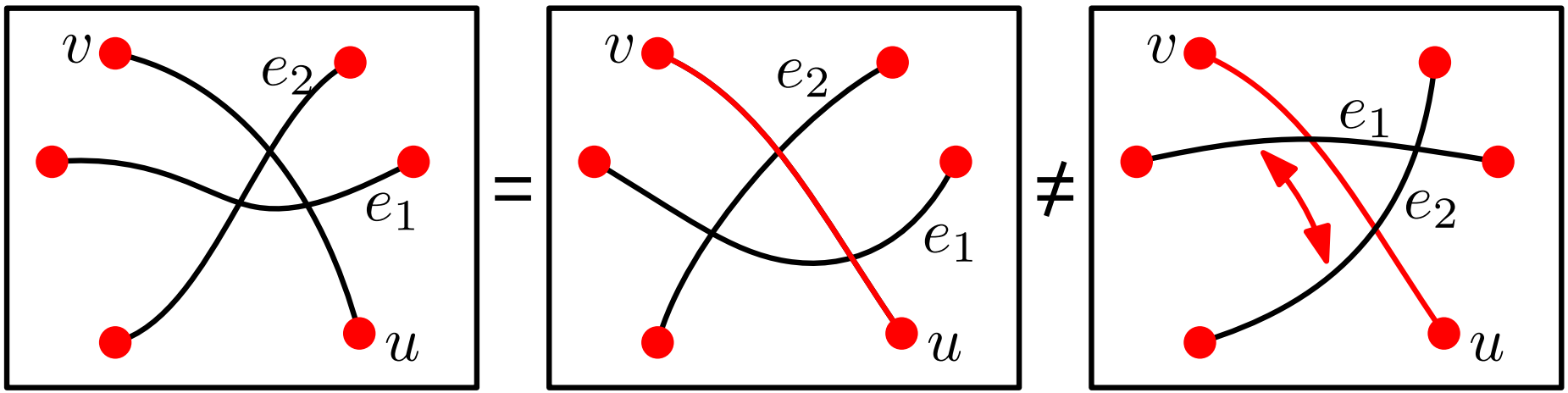


- **Planarization:**

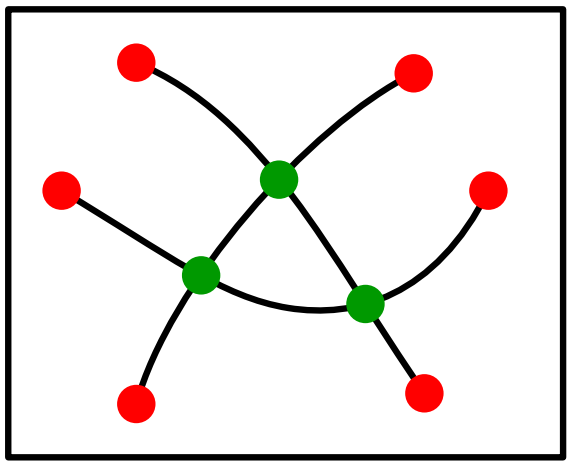


Definitions

- Two drawings are *equivalent*: for each edge the order of the crossing edges is the same



- **Planarization:**



Algorithm

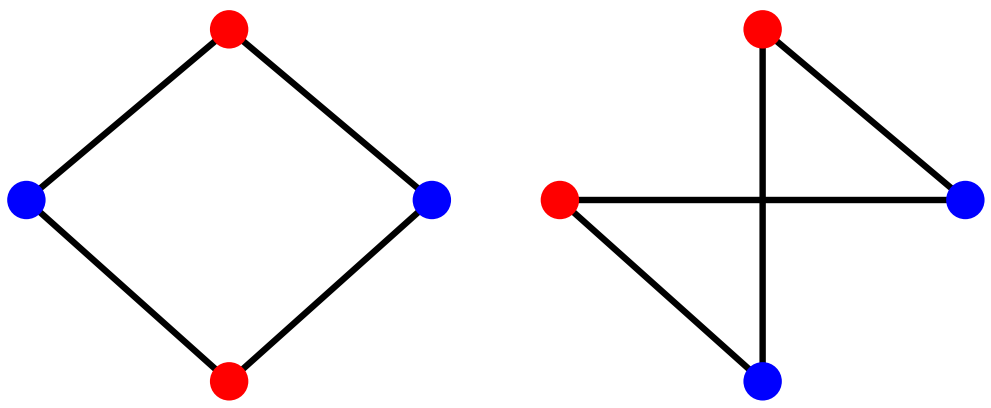
1. Insert all drawings of K_3 / $K_{2,2}$ in a list L
2. **While** $L \neq \emptyset$ **do**
3. $L' \leftarrow \emptyset$
4. **for all** $D \in L$
5. Insert 'next' vertex into D in all possible ways and save new drawings in L'
6. Delete duplicates from L'
7. Save drawings of L' (on hard disk)
8. $L \leftarrow L'$

Algorithm – Start

1. Insert all drawings of K_3 / $K_{2,2}$ in a list L

Algorithm – Start

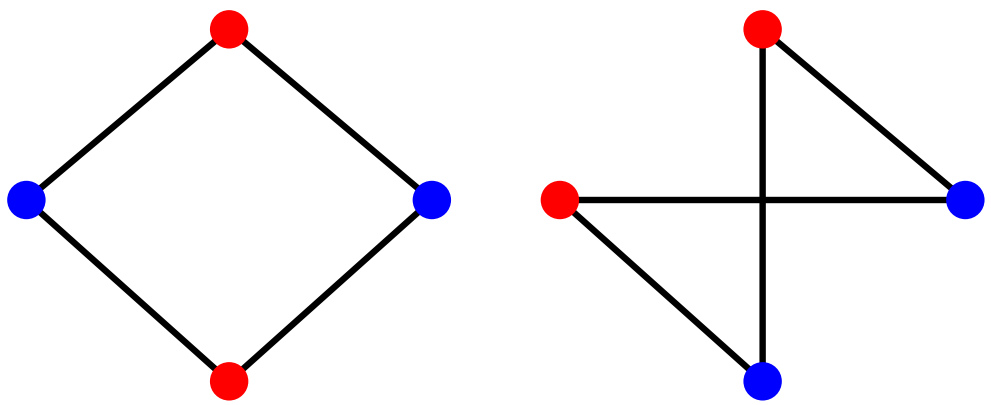
1. Insert all drawings of K_3 / $K_{2,2}$ in a list L



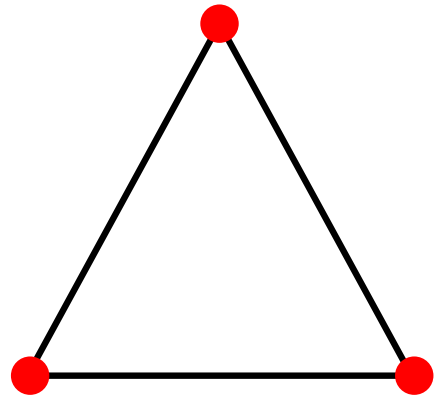
all drawings of $K_{2,2}$

Algorithm – Start

1. Insert all drawings of K_3 / $K_{2,2}$ in a list L



all drawings of $K_{2,2}$



all drawings of K_3

Algorithm – Insertion

4. **for all** $D \in L$
5. Insert 'next' vertex into D in all possible ways;

Algorithm – Insertion

4. **for all** $D \in L$
5. Insert 'next' vertex into D in all possible ways;

- List L: contains all drawings of K_n / $K_{a,b}$

Algorithm – Insertion

4. **for all** $D \in L$
5. Insert 'next' vertex into D in all possible ways;

- List L: contains all drawings of K_n / $K_{a,b}$
- K_n + one vertex $\rightarrow$ all drawings of K_{n+1}

Algorithm – Insertion

4. **for all** $D \in L$
5. Insert 'next' vertex into D in all possible ways;

- List L: contains all drawings of K_n / $K_{a,b}$
- K_n + one vertex $\rightarrow$ all drawings of K_{n+1}
- $K_{a,b}$ + one vertex $\rightarrow$ all drawings of $K_{a+1,b}$ or $K_{a,b+1}$

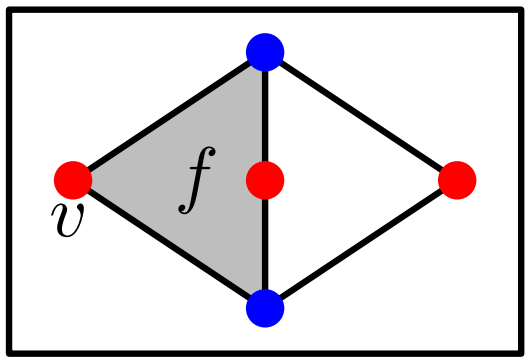
Algorithm – Insertion

4. **for all** $D \in L$
5. Insert 'next' vertex into D in all possible ways;

- List L: contains all drawings of K_n / $K_{a,b}$
- K_n + one vertex $\rightarrow$ all drawings of K_{n+1}
- $K_{a,b}$ + one vertex $\rightarrow$ all drawings of $K_{a+1,b}$ or $K_{a,b+1}$

Algorithm – Insertion

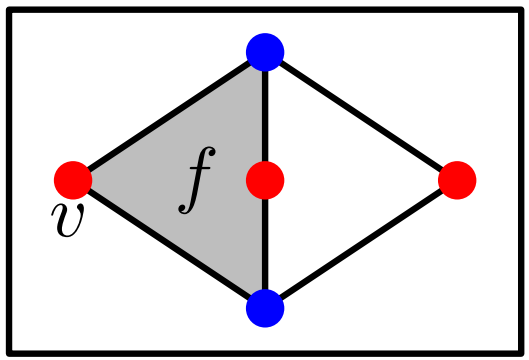
- Choose $v \in V$ (red)



Example for
1-planar drawings

Algorithm – Insertion

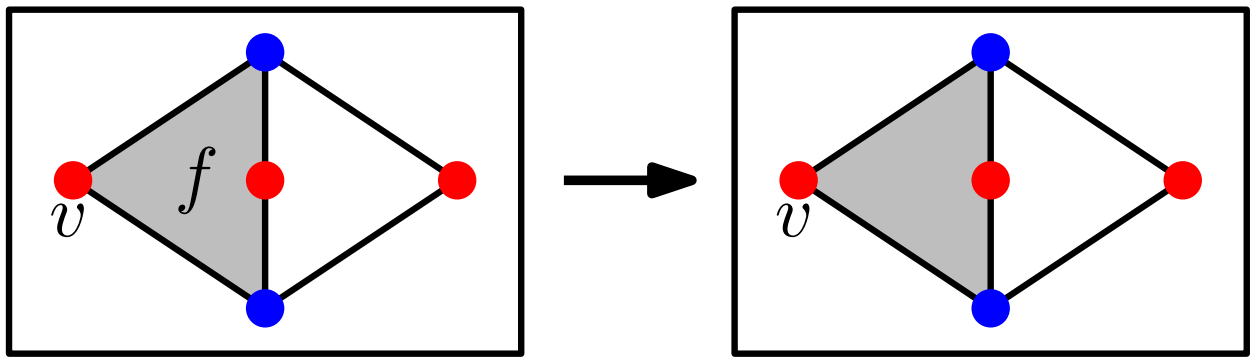
- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings



Example for
1-planar drawings

Algorithm – Insertion

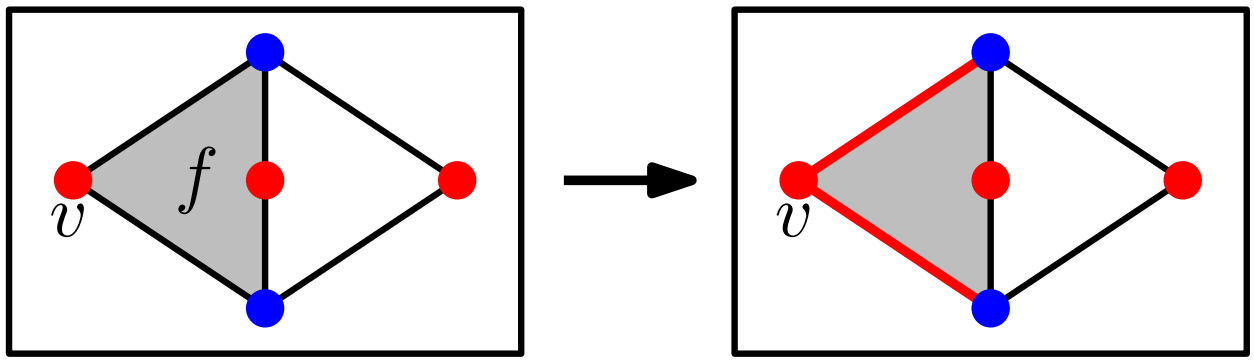
- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings



Example for
1-planar drawings

Algorithm – Insertion

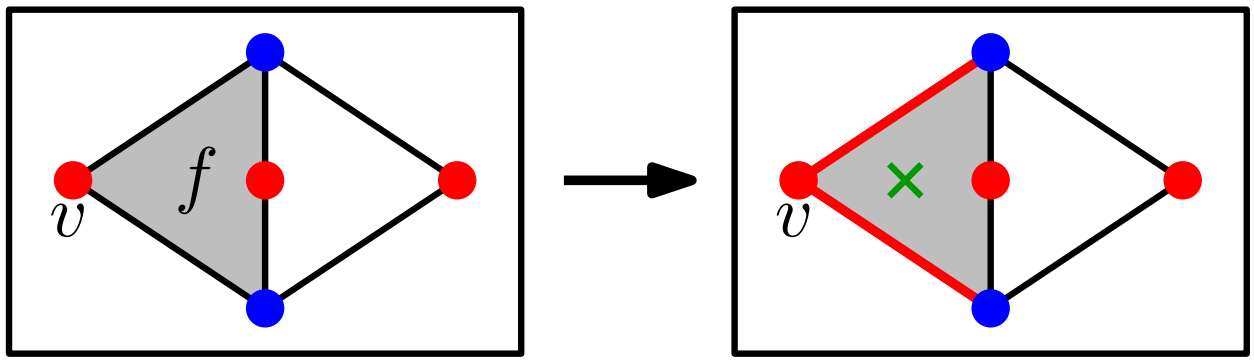
- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings



Example for
1-planar drawings

Algorithm – Insertion

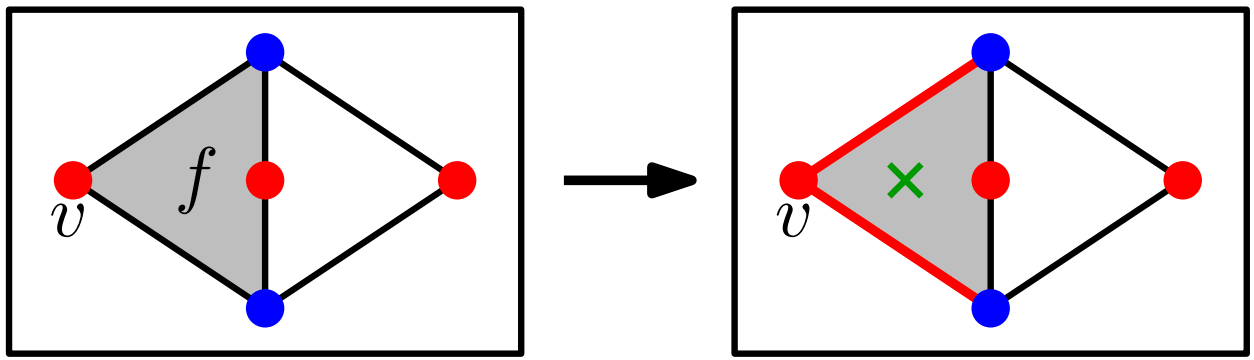
- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings



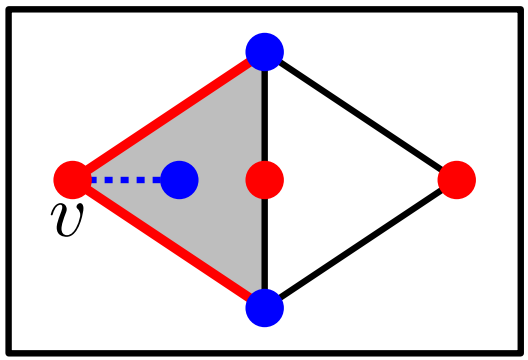
Example for
1-planar drawings

Algorithm – Insertion

- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings

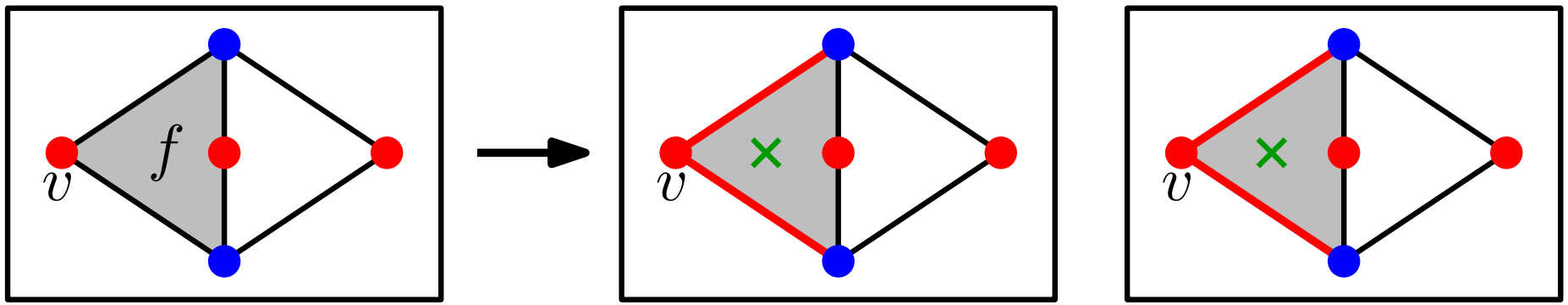


Example for
1-planar drawings

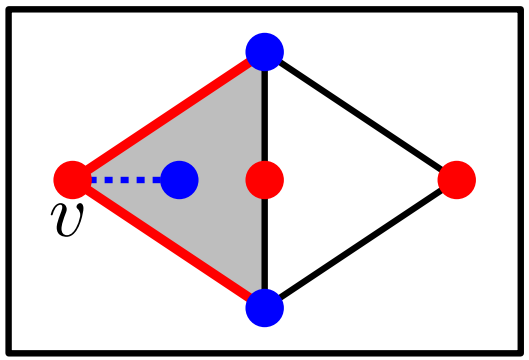


Algorithm – Insertion

- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings

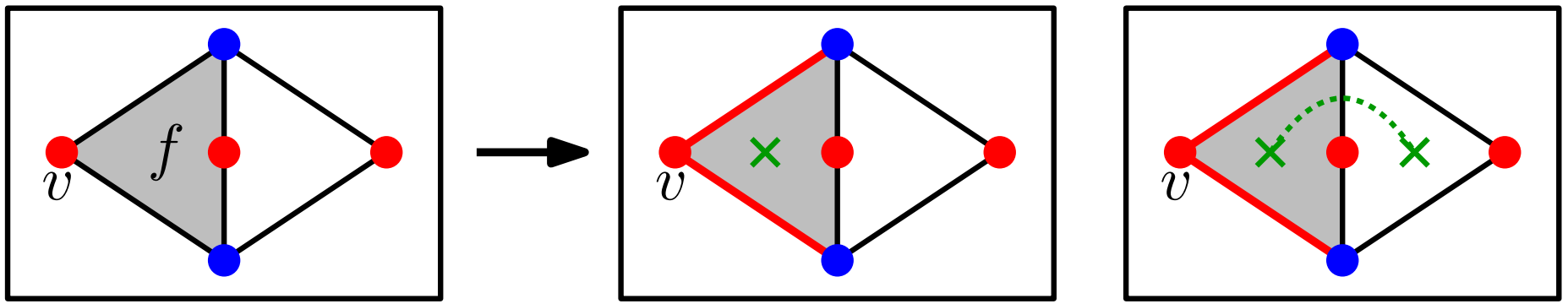


Example for
1-planar drawings

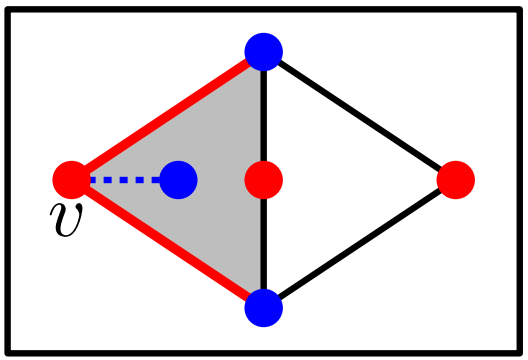


Algorithm – Insertion

- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings

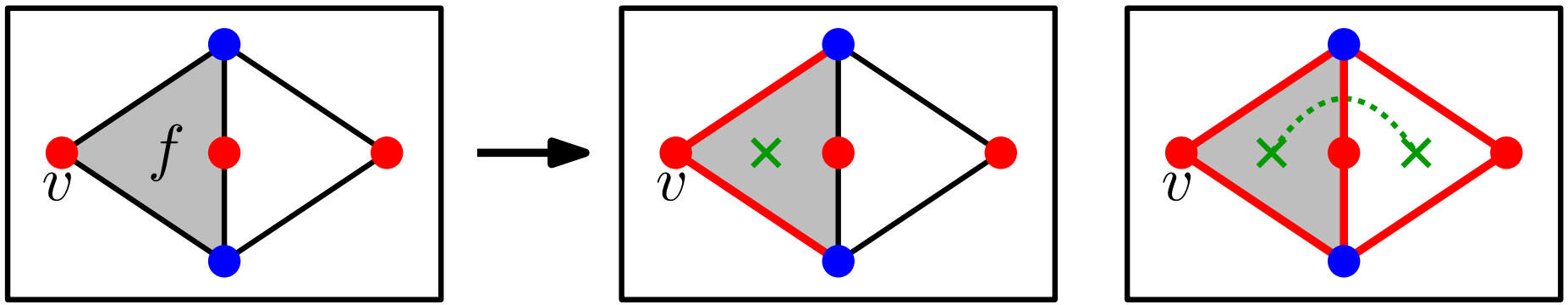


Example for
1-planar drawings

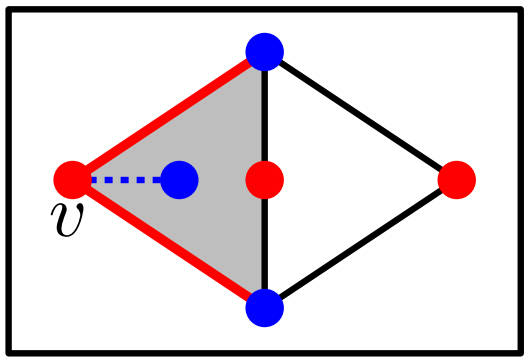


Algorithm – Insertion

- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings

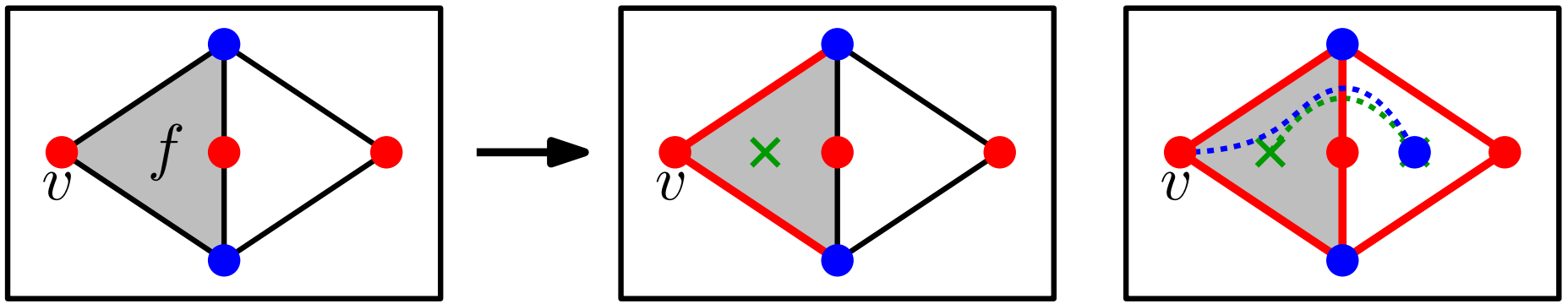


Example for
1-planar drawings

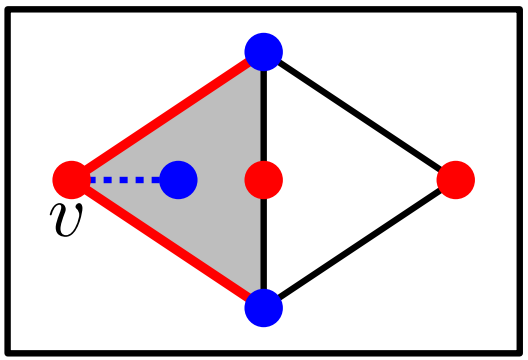


Algorithm – Insertion

- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings

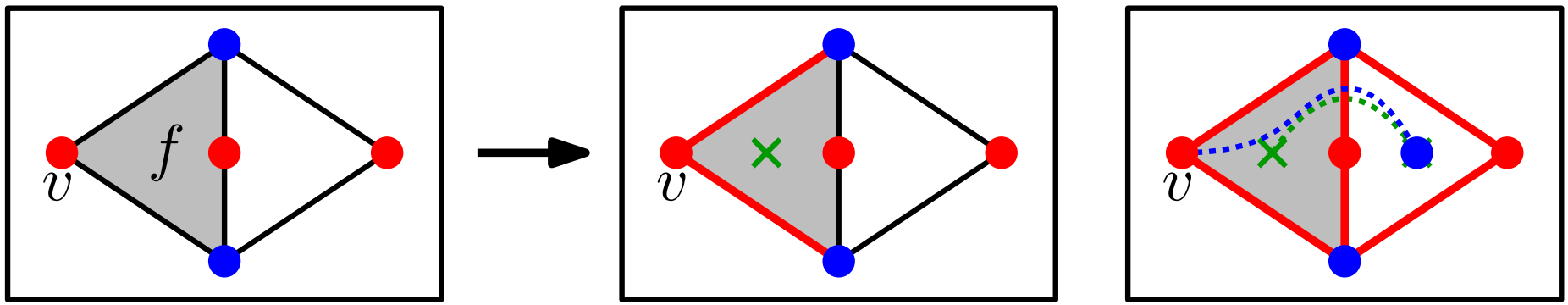


Example for
1-planar drawings

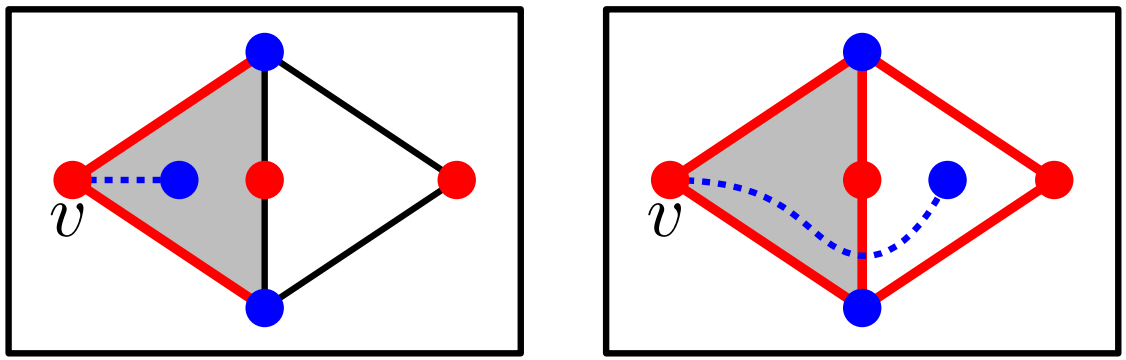


Algorithm – Insertion

- Choose $v \in V$ (red)
- **For all** faces f around v:
  search possible edges in class $\mathcal{C}$
  save valid drawings

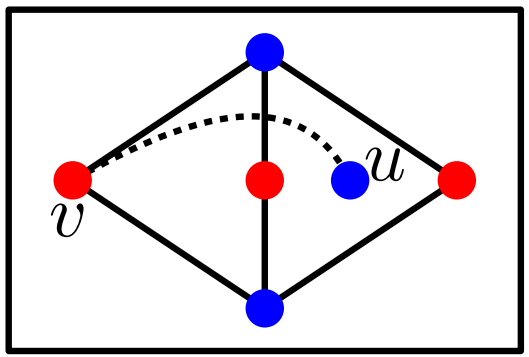


Example for
1-planar drawings



Algorithm – Insertion

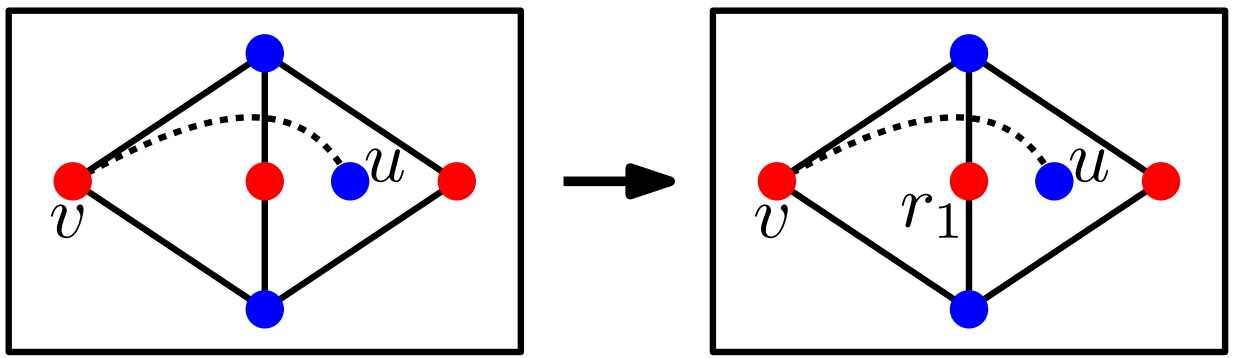
- **For all** valid drawings D:
  for all red vertices $r \neq v$
  connect u to r in all possible ways



Example for
1-planar drawings

Algorithm – Insertion

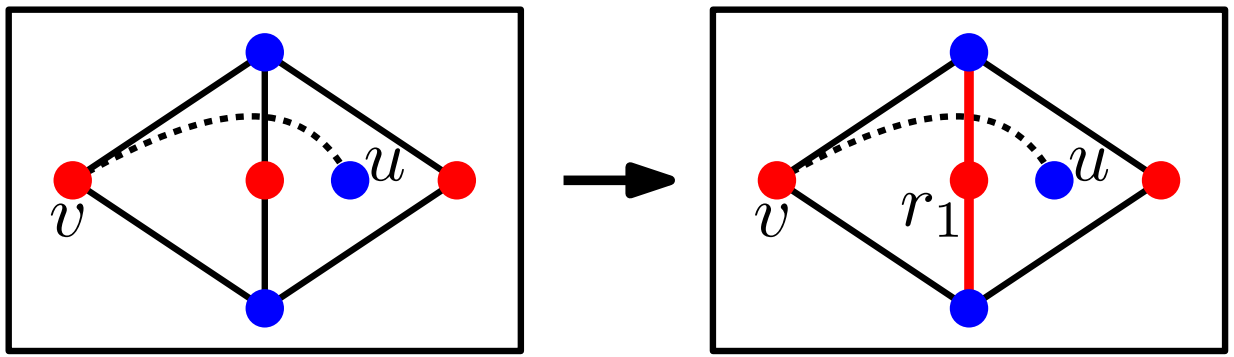
- **For all** valid drawings D:
  for all red vertices $r \neq v$
  connect u to r in all possible ways



Example for
1-planar drawings

Algorithm – Insertion

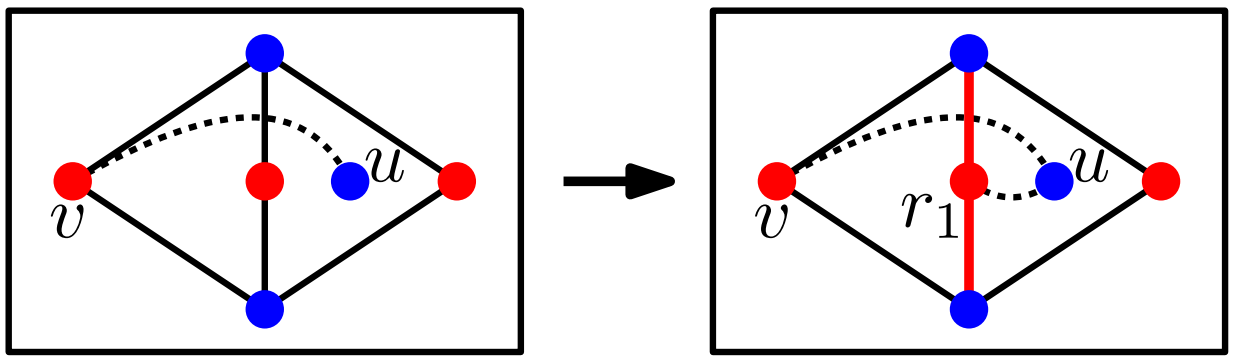
- **For all** valid drawings D:
  for all red vertices $r \neq v$
  connect u to r in all possible ways



Example for
1-planar drawings

Algorithm – Insertion

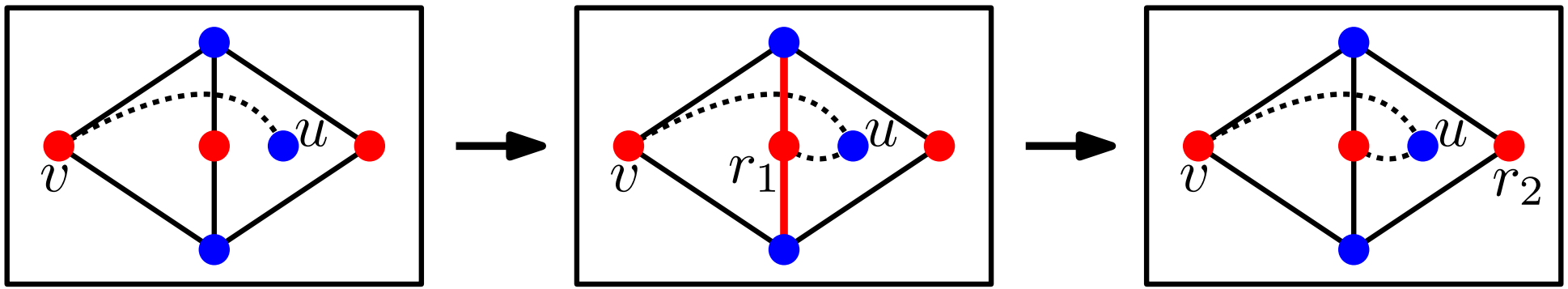
- **For all** valid drawings D:
  for all red vertices $r \neq v$
  connect u to r in all possible ways



Example for
1-planar drawings

Algorithm – Insertion

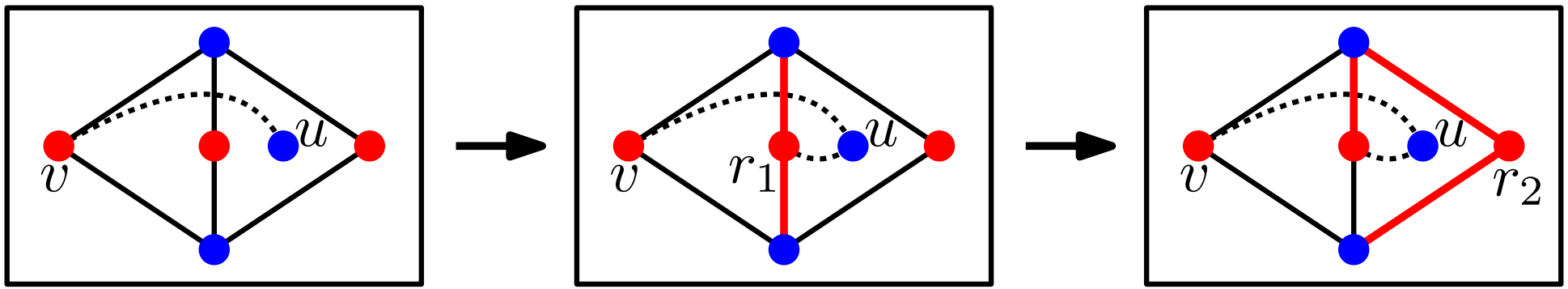
- **For all** valid drawings D:
  for all red vertices $r \neq v$
  connect u to r in all possible ways



Example for
1-planar drawings

Algorithm – Insertion

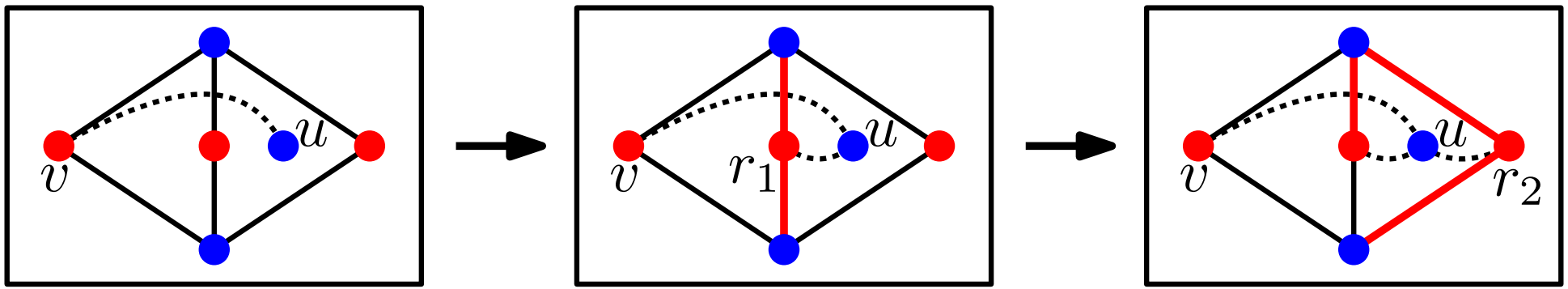
- **For all** valid drawings D:
  for all red vertices $r \neq v$
  connect u to r in all possible ways



Example for
1-planar drawings

Algorithm – Insertion

- **For all** valid drawings D:
  for all red vertices $r \neq v$
  connect u to r in all possible ways



Example for
1-planar drawings

Algorithm – Iso. Test

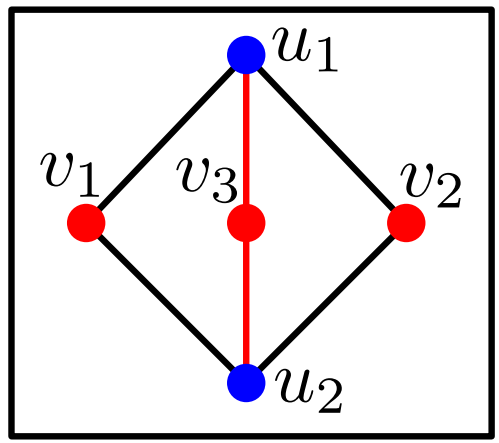
6. Delete duplicates from L'

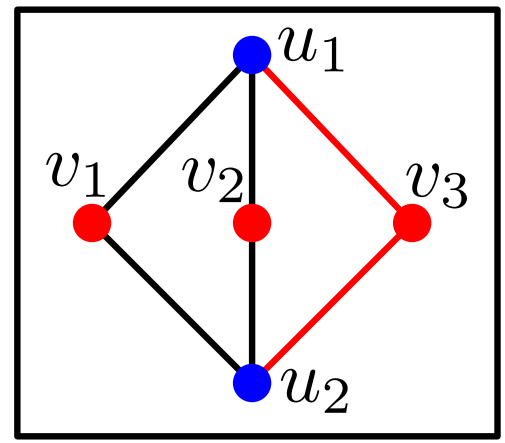
Definition: D_1 and D_2 *isomorphic* $\Leftrightarrow$ D_1 and D_2 are equivalent after renaming vertices

Algorithm – Iso. Test

6. Delete duplicates from L'

Definition: D_1 and D_2 *isomorphic* $\Leftrightarrow$ D_1 and D_2 are equivalent after renaming vertices





Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

- $(v_1, w_1) \in \Gamma_1 \mapsto (v_2, w_2) \in \Gamma_2$ & $v_1 \mapsto v_2$
  $\Longrightarrow w_1 \mapsto w_2$

Algorithm – Iso. Test

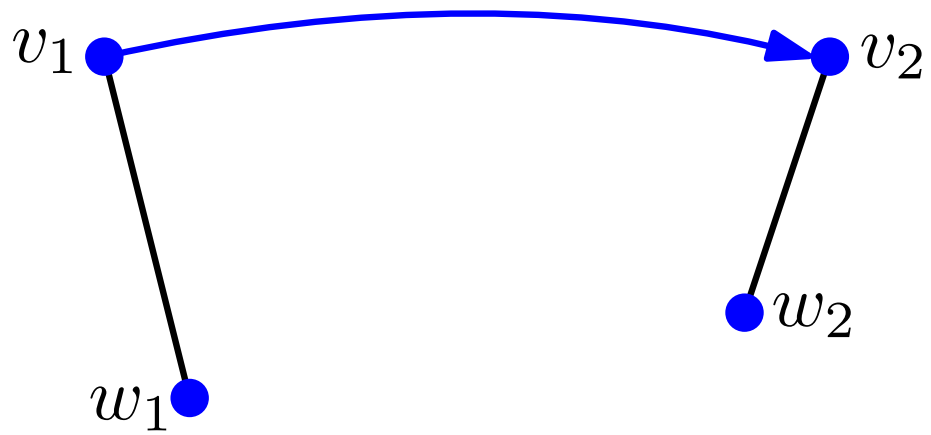
Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

- $(v_1, w_1) \in \Gamma_1 \mapsto (v_2, w_2) \in \Gamma_2$ & $v_1 \mapsto v_2$ $\Longrightarrow w_1 \mapsto w_2$

Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

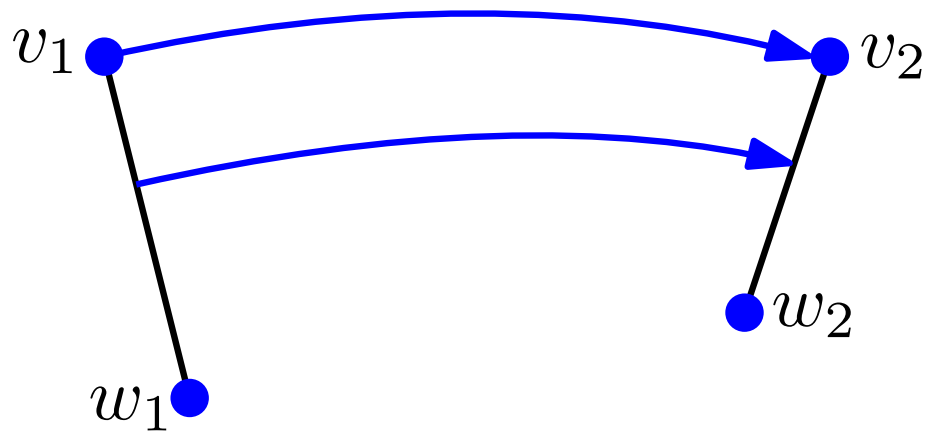
- $(v_1, w_1) \in \Gamma_1 \mapsto (v_2, w_2) \in \Gamma_2$ & $v_1 \mapsto v_2$ $\Longrightarrow w_1 \mapsto w_2$



Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

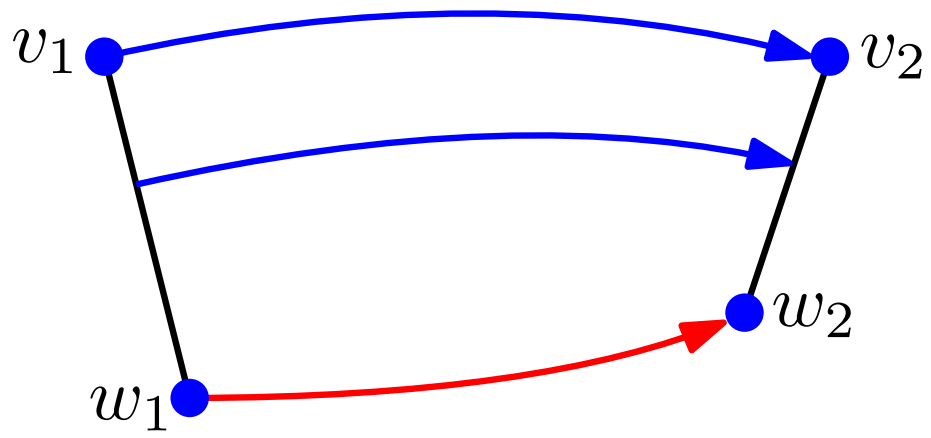
- $(v_1, w_1) \in \Gamma_1 \mapsto (v_2, w_2) \in \Gamma_2$ & $v_1 \mapsto v_2$ $\Longrightarrow w_1 \mapsto w_2$



Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

- $(v_1, w_1) \in \Gamma_1 \mapsto (v_2, w_2) \in \Gamma_2$ & $v_1 \mapsto v_2$ $\Longrightarrow w_1 \mapsto w_2$



Algorithm – Iso. Test

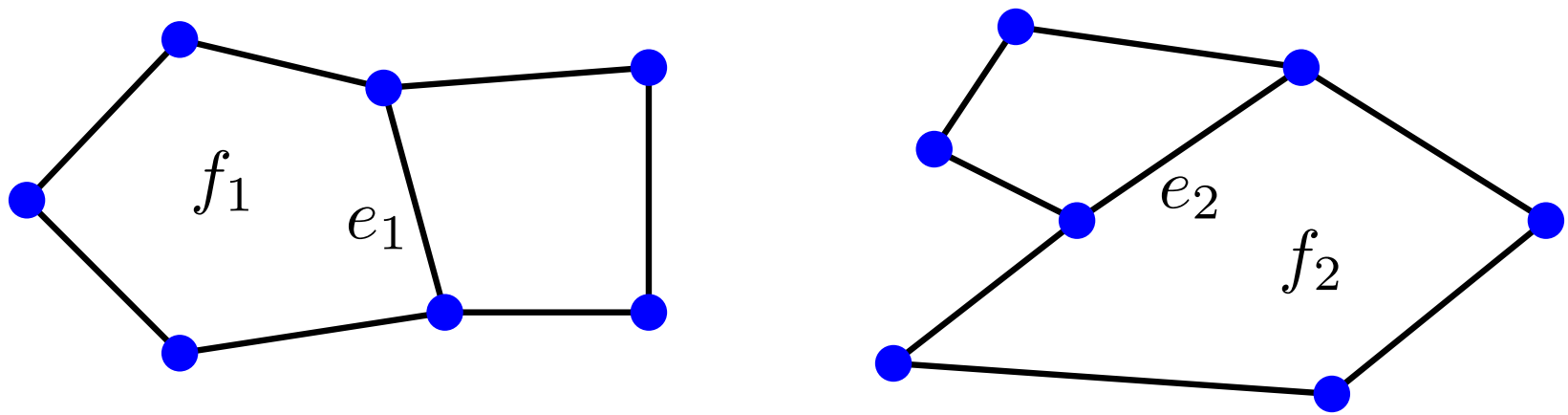
Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

- $f_i \in \Gamma_i$ faces, e_i incident edges, $f_1 \mapsto f_2$ & $e_1 \mapsto e_2$

Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

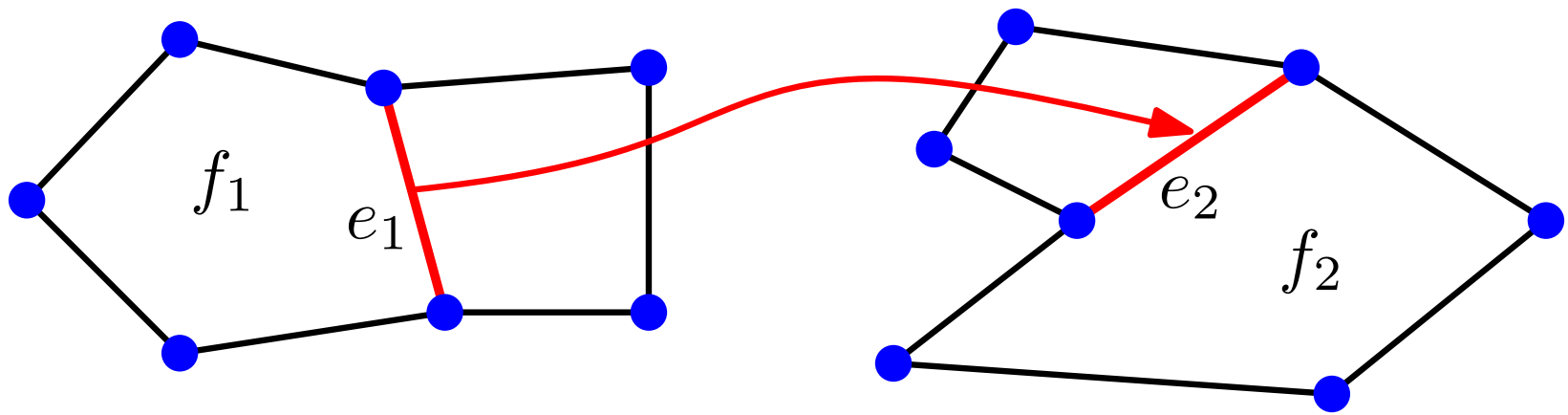
- $f_i \in \Gamma_i$ faces, e_i incident edges, $f_1 \mapsto f_2$ & $e_1 \mapsto e_2$



Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

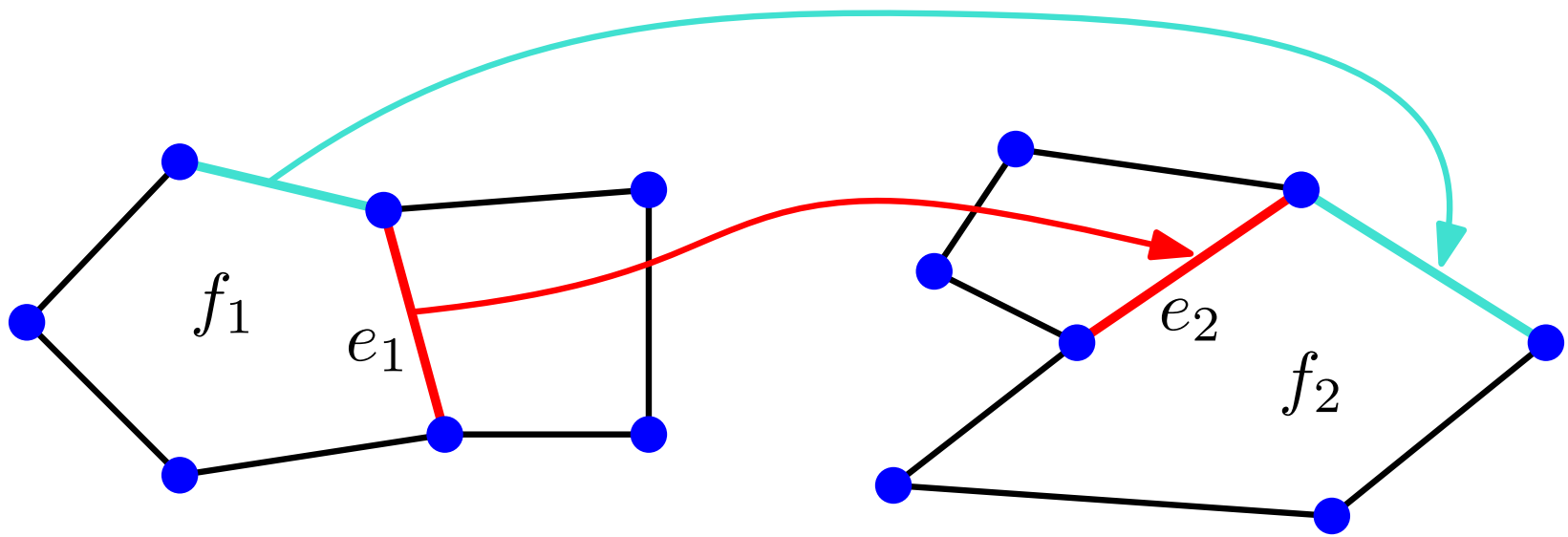
- $f_i \in \Gamma_i$ faces, e_i incident edges, $f_1 \mapsto f_2$ & $e_1 \mapsto e_2$



Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

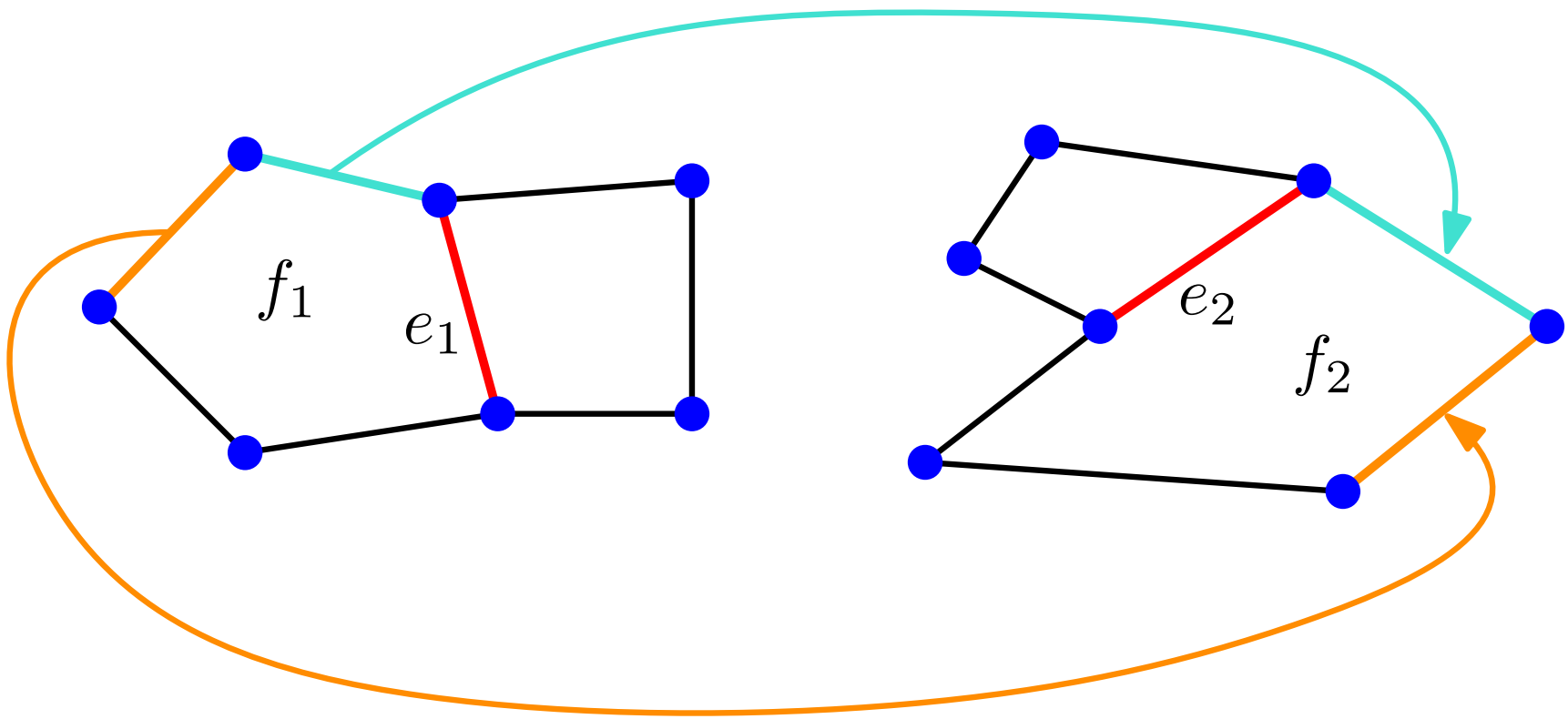
- $f_i \in \Gamma_i$ faces, e_i incident edges, $f_1 \mapsto f_2$ & $e_1 \mapsto e_2$



Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

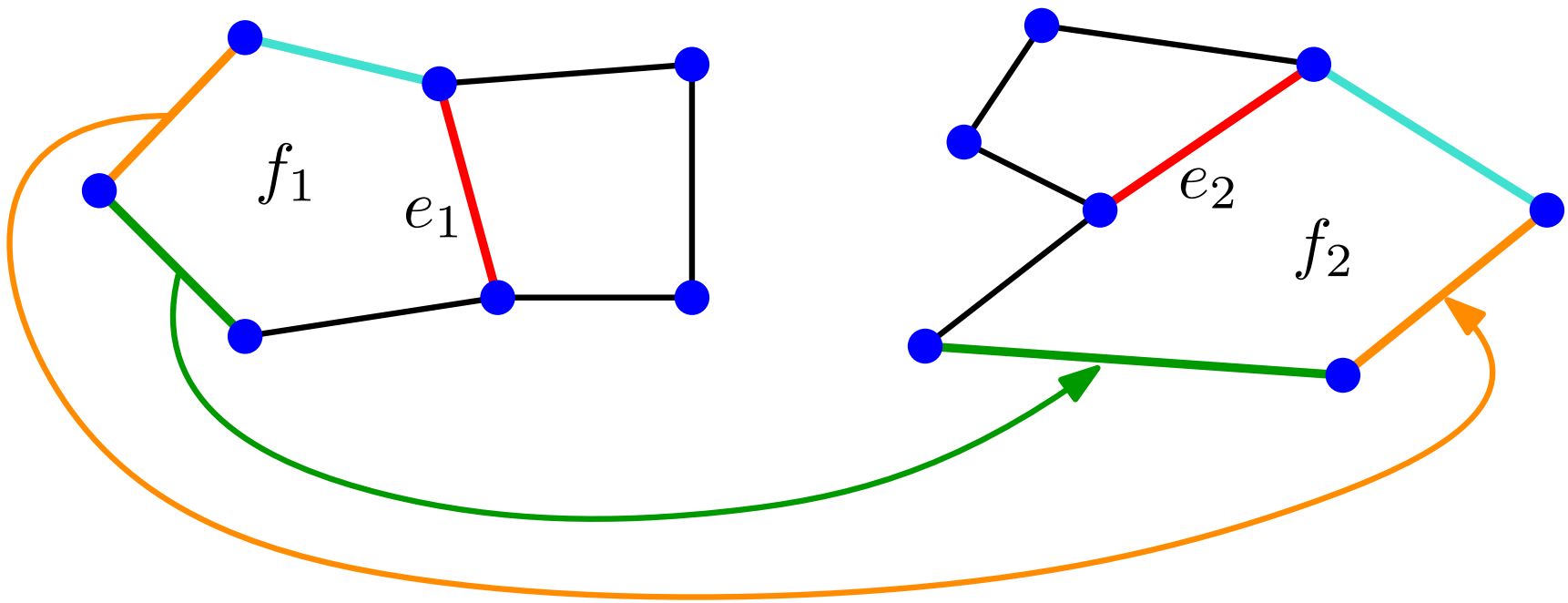
- $f_i \in \Gamma_i$ faces, e_i incident edges, $f_1 \mapsto f_2$ & $e_1 \mapsto e_2$



Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

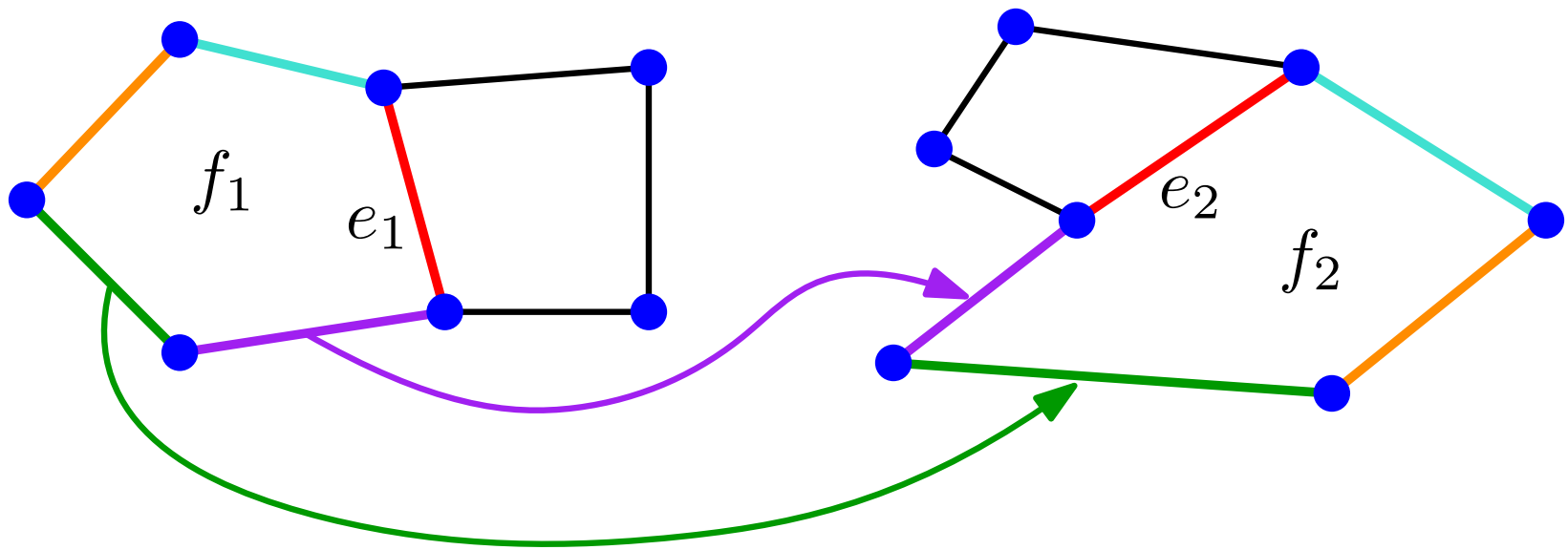
- $f_i \in \Gamma_i$ faces, e_i incident edges, $f_1 \mapsto f_2$ & $e_1 \mapsto e_2$



Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

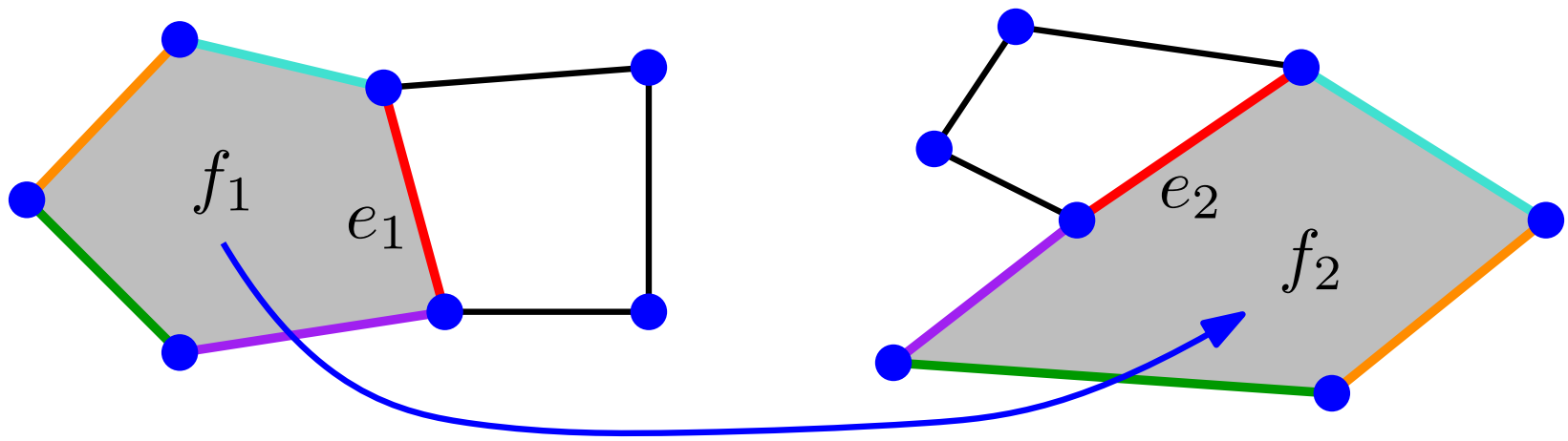
- $f_i \in \Gamma_i$ faces, e_i incident edges, $f_1 \mapsto f_2$ & $e_1 \mapsto e_2$



Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

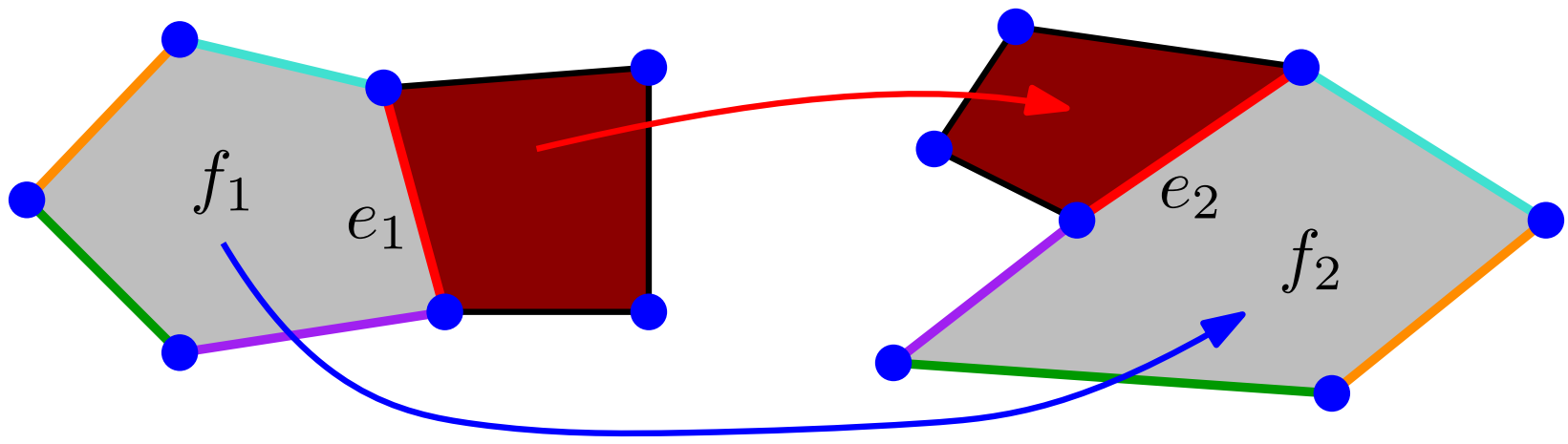
- $f_i \in \Gamma_i$ faces, e_i incident edges, $f_1 \mapsto f_2$ & $e_1 \mapsto e_2$



Algorithm – Iso. Test

Valid bijective mapping between planarizations Γ_1 of D_1 and Γ_2 of D_2 has following properties:

- $f_i \in \Gamma_i$ faces, e_i incident edges, $f_1 \mapsto f_2$ & $e_1 \mapsto e_2$



Results – Complete Graphs

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$
1-planar	K_6	K_7
2-planar	K_7	K_9
3-planar	K_8	K_{10}
4-planar	K_9	K_{11}
5-planar	K_9	K_{19}
fan-planar	K_7	K_9
fan-crossing free	K_6	K_7
gap-planar	K_8	K_9
quasi-planar	K_{10}	K_{11}

Results – Complete Graphs

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$	
1-planar	K_6	K_7	
2-planar	K_7	~~K_9~~	K_8
3-planar	K_8	~~K_{10}~~	K_9
4-planar	K_9	~~K_{11}~~	K_{10}
5-planar	K_9	~~K_{19}~~	K_{10}
fan-planar	K_7	~~K_9~~	K_8
fan-crossing free	K_6	K_7	
gap-planar	K_8	K_9	
quasi-planar	K_{10}	K_{11}	

Results – Complete Graphs

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$	
1-planar	K_6	K_7	
2-planar	K_7	~~K_9~~	K_8
3-planar	K_8	~~K_{10}~~	K_9
4-planar	K_9	~~K_{11}~~	K_{10}
5-planar	K_9	~~K_{19}~~	K_{10}
fan-planar	K_7	~~K_9~~	K_8
fan-crossing free	K_6	K_7	
gap-planar	K_8	K_9	
quasi-planar	K_{10}	K_{11}	

Results – Complete Graphs

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$	
1-planar	K_6	K_7	
2-planar	K_7	~~K_9~~	K_8
3-planar	K_8	~~K_{10}~~	K_9
4-planar	K_9	~~K_{11}~~	K_{10}
5-planar	K_9	~~K_{10}~~	K_{10}
fan-planar	K_7	~~K_9~~	K_8
fan-crossing free	K_6	K_7	
gap-planar	K_8	K_9	
quasi-planar	K_{10}	K_{11}	

$\rightarrow$ Answers a question by Bae et al (2018):
gap-planar $\not\subset$ fan-planar

Results – Complete Bipartite Graphs

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$
1-planar	$K_{3,6}$	$K_{3,7}$
	$K_{4,4}$	$K_{4,5}$
2-planar	$K_{3,10}$	$K_{3,11}$
	$K_{4,6}$	$K_{4,7}$
	$K_{4,5}$	$K_{5,5}$
3-planar	$K_{3,14}$	$K_{3,15}$
	$K_{4,9}$	$K_{4,10}$
	$K_{5,6}$	$K_{5,7}$
	$K_{5,6}$	$K_{6,6}$

Results – Complete Bipartite Graphs

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$
fan-planar	$K_{4,n}$	$K_{5,5}$
fan-crossing free	$K_{3,6}$	$K_{3,7}$
	$K_{4,6}$	$K_{4,7}$
	$K_{4,5}$	$K_{5,5}$
gap-planar	$K_{3,12}$	$K_{3,14}$
	$K_{4,8}$	$K_{4,10}$
	$K_{5,6}$	$K_{5,7}$
	$K_{5,6}$	$K_{6,6}$

Results – Complete Bipartite Graphs

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$
fan-planar	$K_{4,n}$	$K_{5,5}$
fan-crossing free	$K_{3,6}$	$K_{3,7}$
	$K_{4,6}$	$K_{4,7}$
	$K_{4,5}$	$K_{5,5}$
gap-planar	$K_{3,12}$	$K_{3,14}$
	$K_{4,8}$	~~$K_{4,10}$~~ $K_{4,9}$
	$K_{5,6}$	$K_{5,7}$
	$K_{5,6}$	$K_{6,6}$

Results – Complete Bipartite Graphs

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$
fan-planar	$K_{4,n}$	$K_{5,5}$
fan-crossing free	$K_{3,6}$	$K_{3,7}$
	$K_{4,6}$	$K_{4,7}$
	$K_{4,5}$	$K_{5,5}$
gap-planar	$K_{3,12}$	$K_{3,14}$
	$K_{4,8}$	~~$K_{4,10}$~~ $K_{4,9}$
	$K_{5,6}$	$K_{5,7}$
	$K_{5,6}$	$K_{6,6}$

$\rightarrow$ Answers a question by Bae et al (2018)

Results – Complete Bipartite Graphs

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$
fan-planar	$K_{4,n}$	$K_{5,5}$
fan-crossing free	$K_{3,6}$	$K_{3,7}$
	$K_{4,6}$	$K_{4,7}$
	$K_{4,5}$	$K_{5,5}$
gap-planar	$K_{3,12}$	$K_{3,14}$
	$K_{4,8}$	~~$K_{4,10}$~~ $K_{4,9}$
	$K_{5,6}$	$K_{5,7}$
	$K_{5,6}$	$K_{6,6}$

Still open: $K_{3,13}$

Effect of Isomorphic Test

test	with iso. test		without iso. test	
	drawings	time	drawings	time
1-planar K_7	25	$\approx 0.1s$	158	$\approx 1.4s$
1-planar $K_{4,5}$	69	$\approx 0.2s$	2662	$\approx 24.3s$
2-planar K_8	191	$\approx 0.5s$	11366	$\approx 113s$
2-planar $K_{5,5}$	1473	$\approx 5s$	1423684	$\approx 6h$

Effect of Isomorphic Test

test	with iso. test		without iso. test	
	drawings	time	drawings	time
1-planar K_7	25	$\approx 0.1s$	158	$\approx 1.4s$
1-planar $K_{4,5}$	69	$\approx 0.2s$	2662	$\approx 24.3s$
2-planar K_8	191	$\approx 0.5s$	11366	$\approx 113s$
2-planar $K_{5,5}$	1473	$\approx 5s$	1423684	$\approx 6h$

Statistics

2-planar	non-iso	time	3-planar	non-iso	time
$K_{2,3}$	6	90	$K_{2,3}$	6	234
$K_{3,3}$	19	254	$K_{3,3}$	69	1802
$K_{3,4}$	71	1458	$K_{3,4}$	1188	16969
$K_{4,4}$	38	1152	$K_{4,4}$	2704	97801
$K_{4,5}$	37	1826	$K_{4,5}$	7653	310194
$K_{5,5}$	0	357	$K_{5,5}$	1899	199908
			$K_{5,6}$	438	47396
			$K_{6,6}$	0	4822
Sum	171	5137	Sum	13957	679126

Time in milliseconds

Statistics

2-planar	non-iso	time	3-planar	non-iso	time
$K_{2,3}$	6	90	$K_{2,3}$	6	234
$K_{3,3}$	19	254	$K_{3,3}$	69	1802
$K_{3,4}$	71	1458	$K_{3,4}$	1188	16969
$K_{4,4}$	38	1152	$K_{4,4}$	2704	97801
$K_{4,5}$	37	1826	$K_{4,5}$	7653	310194
$K_{5,5}$	0	357	$K_{5,5}$	1899	199908
			$K_{5,6}$	438	47396
			$K_{6,6}$	0	4822
Sum	171	5137	Sum	13957	679126
		$\approx$ 5 s			$\approx$ 11 min

Time in milliseconds

Statistics

2-planar	non-iso	time	3-planar	non-iso	time
$K_{2,3}$	6	90	$K_{2,3}$	6	234
$K_{3,3}$	19	254	$K_{3,3}$	69	1802
$K_{3,4}$	71	1458	$K_{3,4}$	1188	16969
$K_{4,4}$	38	1152	$K_{4,4}$	2704	97801
$K_{4,5}$	37	1826	$K_{4,5}$	7653	310194
$K_{5,5}$	0	357	$K_{5,5}$	1899	199908
			$K_{5,6}$	438	47396
			$K_{6,6}$	0	4822
Sum	171	5137	Sum	13957	679126
		$\approx$ 5 s			$\approx$ 11 min

Time in milliseconds

Limits

4-planar	non-iso	time
$K_{2,3}$	6	108
$K_{3,3}$	102	2146
$K_{3,4}$	6194	163000
$K_{4,4}$	81817	34096183
$K_{4,5}$	? ? ?	

Limits

4-planar	non-iso	time
$K_{2,3}$	6	108
$K_{3,3}$	102	2146
$K_{3,4}$	6194	163000
$K_{4,4}$	81817	34096183
$K_{4,5}$	? ? ?	

$>$ 11 hours !!!

Limits

4-planar	non-iso	time
$K_{2,3}$	6	108
$K_{3,3}$	102	2146
$K_{3,4}$	6194	163000
$K_{4,4}$	81817	34096183
$K_{4,5}$	? ? ?	

$>$ 11 hours !!!

- long time before you have a 'good' (many vertices) drawing

Limits

4-planar	non-iso	time
$K_{2,3}$	6	108
$K_{3,3}$	102	2146
$K_{3,4}$	6194	163000
$K_{4,4}$	81817	34096183 $>$ 11 hours !!!
$K_{4,5}$	? ? ?	

- long time before you have a 'good' (many vertices) drawing
- **New idea:** search only for drawings

Limits

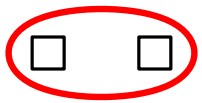
4-planar	non-iso	time
$K_{2,3}$	6	108
$K_{3,3}$	102	2146
$K_{3,4}$	6194	163000
$K_{4,4}$	81817	34096183
$K_{4,5}$	? ? ?	

$>$ 11 hours !!!

- long time before you have a 'good' (many vertices) drawing
- **New idea:** search only for drawings
- change original algorithm slightly
  $\Rightarrow$ DFS variant

DFS Variant of the Algorithm

Current Variant (BFS)



$K_{2,2}$

$K_{2,3}$

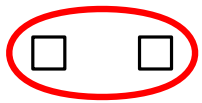
$K_{3,3}$

$K_{3,4}$

$K_{4,4}$

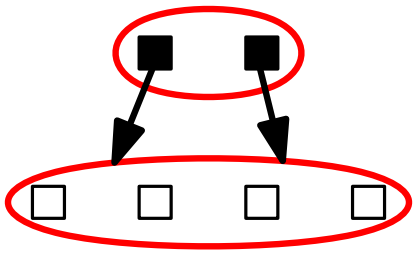
$K_{4,5}$

New Variant (DFS)

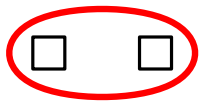


DFS Variant of the Algorithm

Current Variant (BFS)



New Variant (DFS)



$K_{2,2}$

$K_{2,3}$

$K_{3,3}$

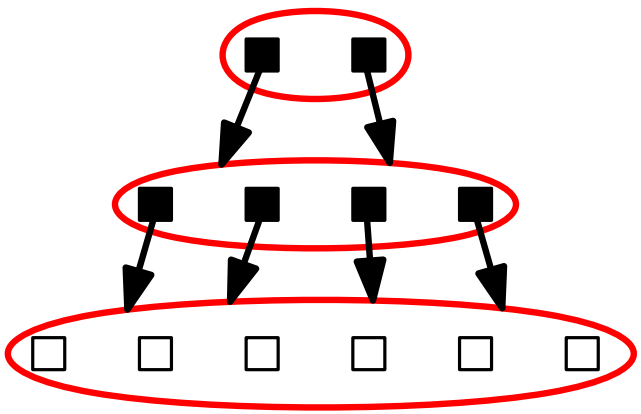
$K_{3,4}$

$K_{4,4}$

$K_{4,5}$

DFS Variant of the Algorithm

Current Variant (BFS)



$K_{2,2}$

$K_{2,3}$

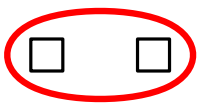
$K_{3,3}$

$K_{3,4}$

$K_{4,4}$

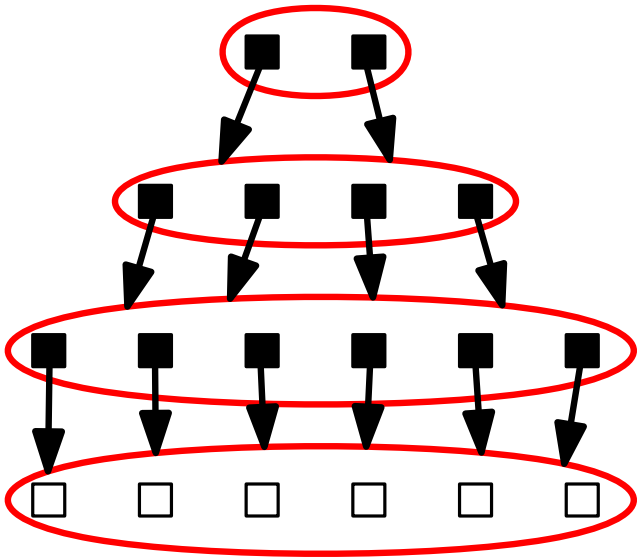
$K_{4,5}$

New Variant (DFS)



DFS Variant of the Algorithm

Current Variant (BFS)



$K_{2,2}$

$K_{2,3}$

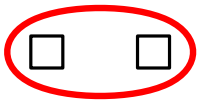
$K_{3,3}$

$K_{3,4}$

$K_{4,4}$

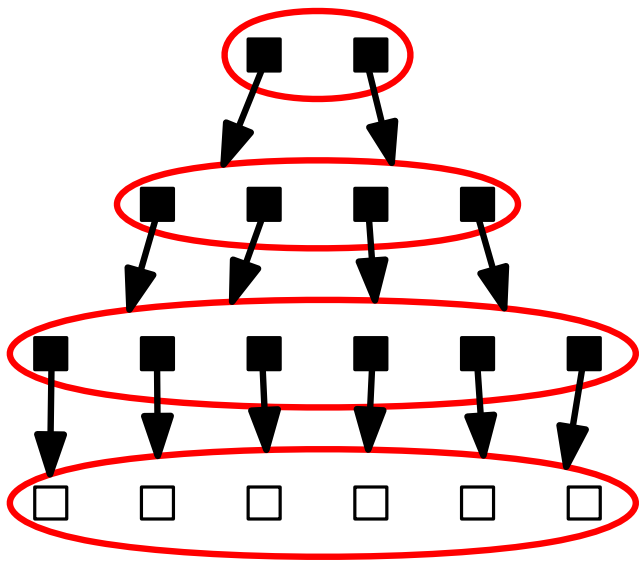
$K_{4,5}$

New Variant (DFS)



DFS Variant of the Algorithm

Current Variant (BFS)



$K_{2,2}$

$K_{2,3}$

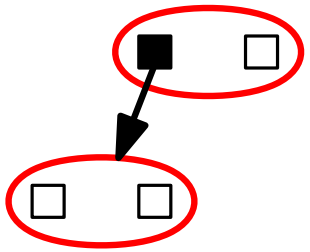
$K_{3,3}$

$K_{3,4}$

$K_{4,4}$

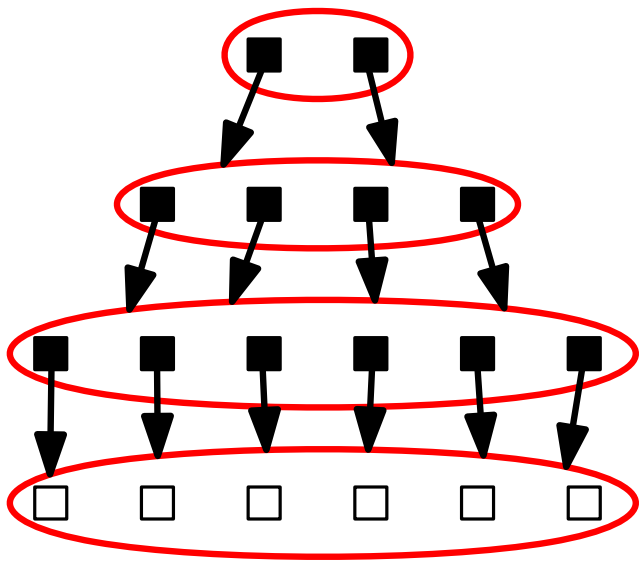
$K_{4,5}$

New Variant (DFS)



DFS Variant of the Algorithm

Current Variant (BFS)



$K_{2,2}$

$K_{2,3}$

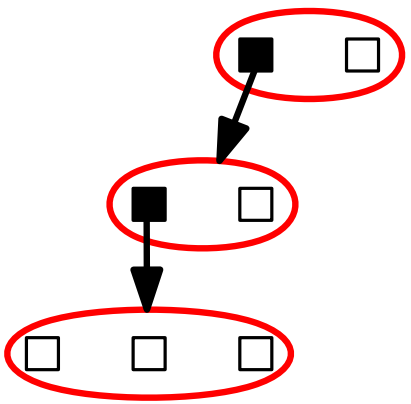
$K_{3,3}$

$K_{3,4}$

$K_{4,4}$

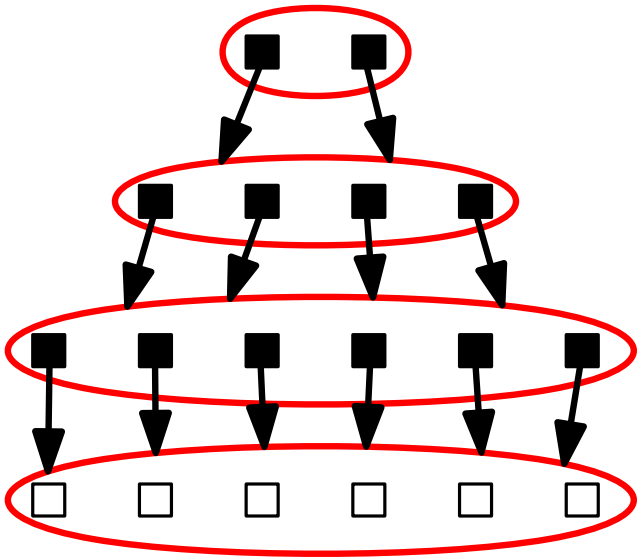
$K_{4,5}$

New Variant (DFS)



DFS Variant of the Algorithm

Current Variant (BFS)



$K_{2,2}$

$K_{2,3}$

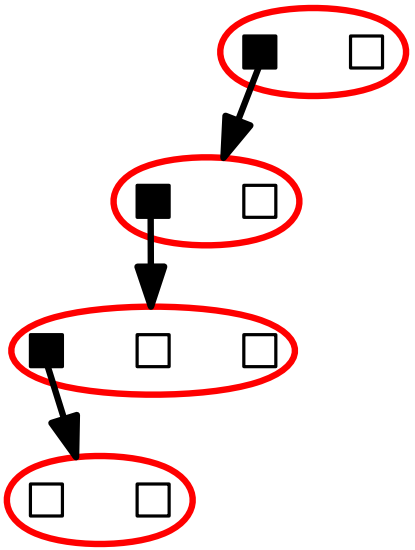
$K_{3,3}$

$K_{3,4}$

$K_{4,4}$

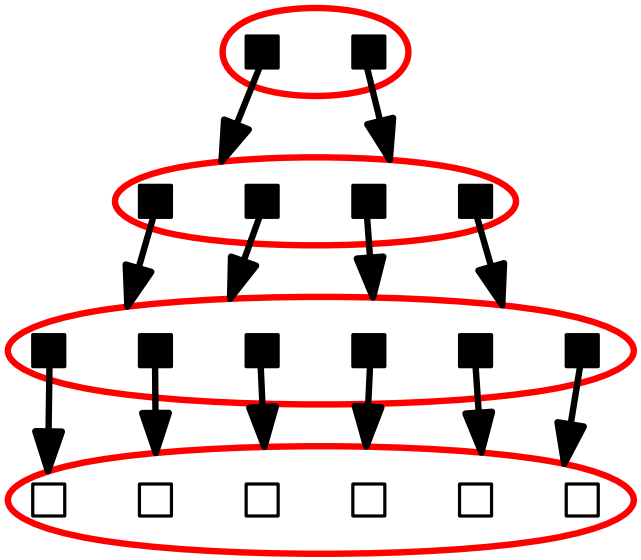
$K_{4,5}$

New Variant (DFS)



DFS Variant of the Algorithm

Current Variant (BFS)



$K_{2,2}$

$K_{2,3}$

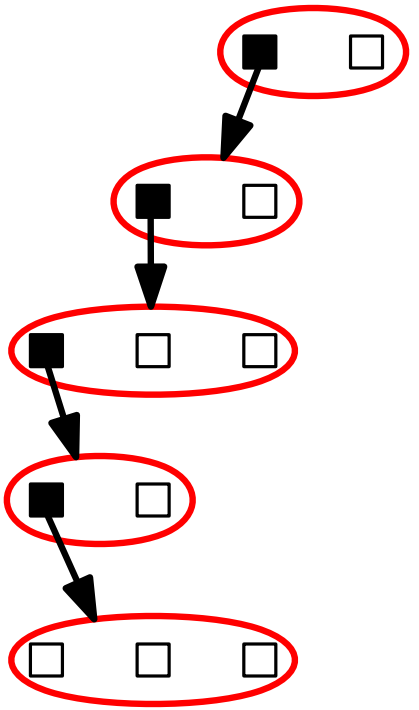
$K_{3,3}$

$K_{3,4}$

$K_{4,4}$

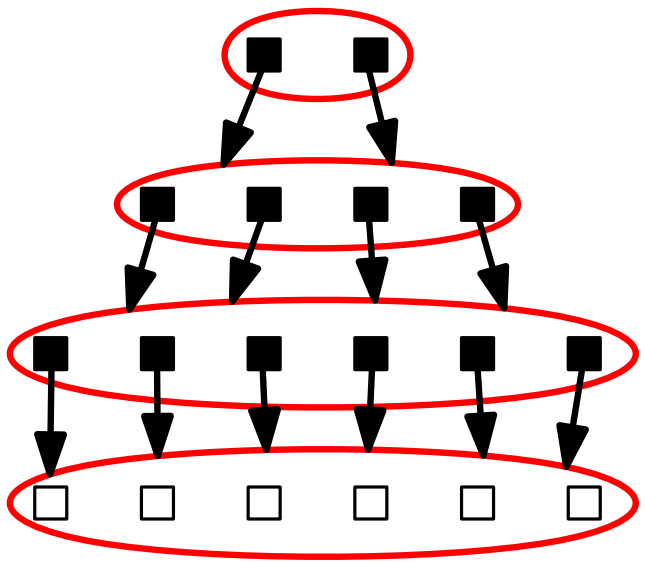
$K_{4,5}$

New Variant (DFS)



DFS Variant of the Algorithm

Current Variant (BFS)



$K_{2,2}$

$K_{2,3}$

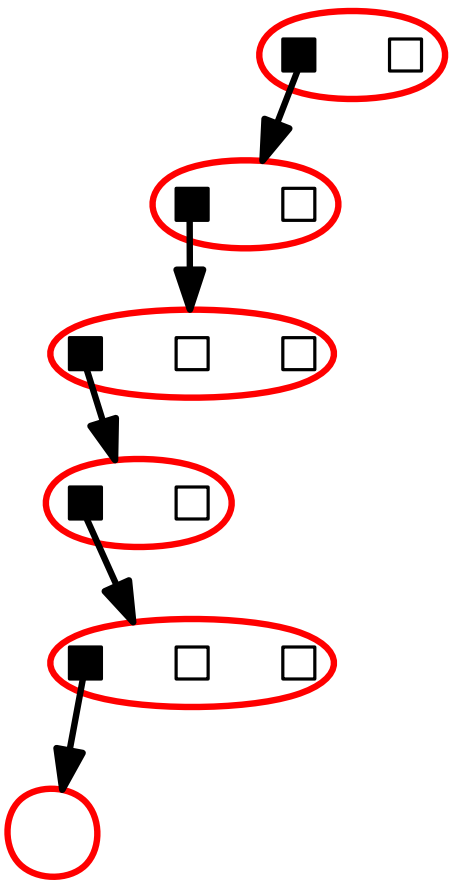
$K_{3,3}$

$K_{3,4}$

$K_{4,4}$

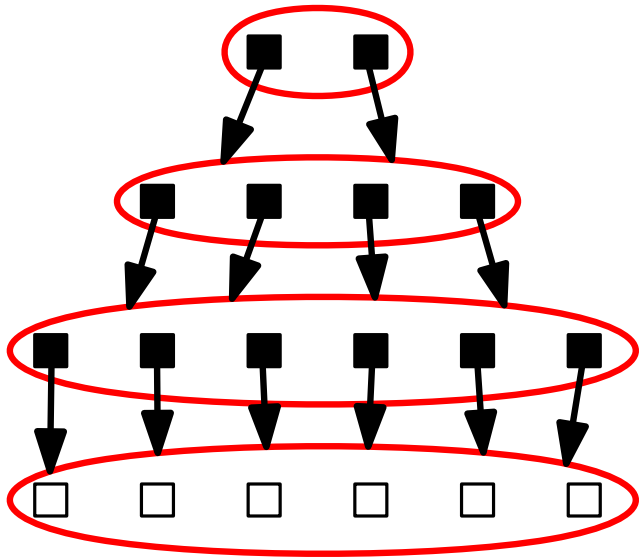
$K_{4,5}$

New Variant (DFS)



DFS Variant of the Algorithm

Current Variant (BFS)



$K_{2,2}$

$K_{2,3}$

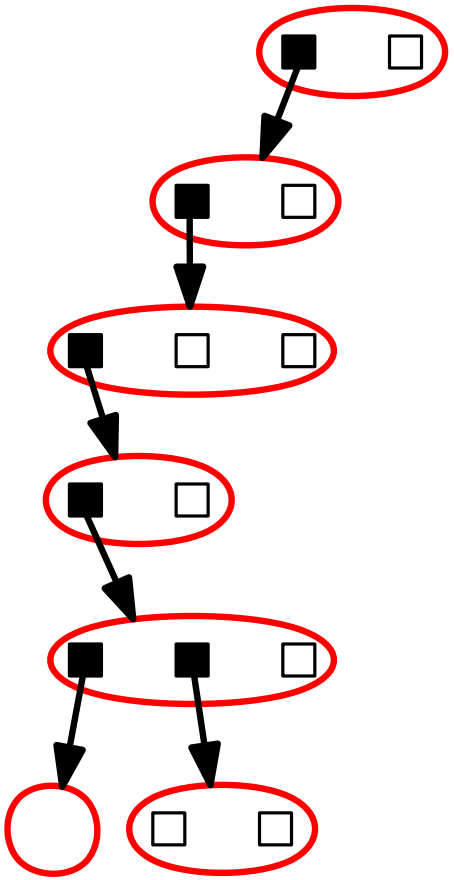
$K_{3,3}$

$K_{3,4}$

$K_{4,4}$

$K_{4,5}$

New Variant (DFS)



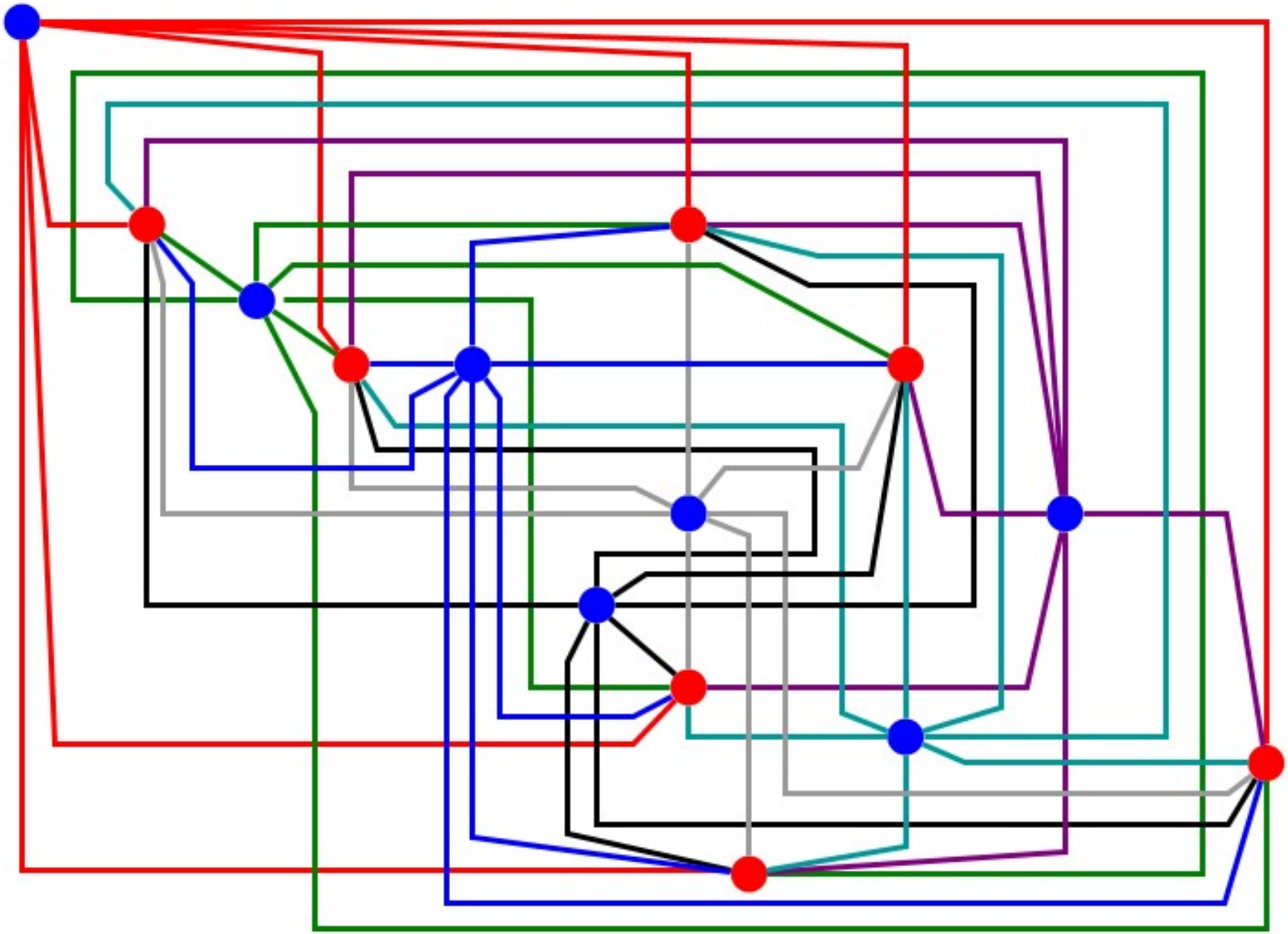
Results

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$
4-planar	$K_{3,18}$	$K_{3,19}$
	$K_{4,11}$	$K_{4,19}$
	$K_{5,8}$	$K_{5,19}$
	$K_{6,6}$	$K_{6,19}$
quasi-planar	$K_{4,n}$	–
	$K_{5,18}$	?
	$K_{6,10}$	?
	$K_{7,7}$	$K_{7,52}$

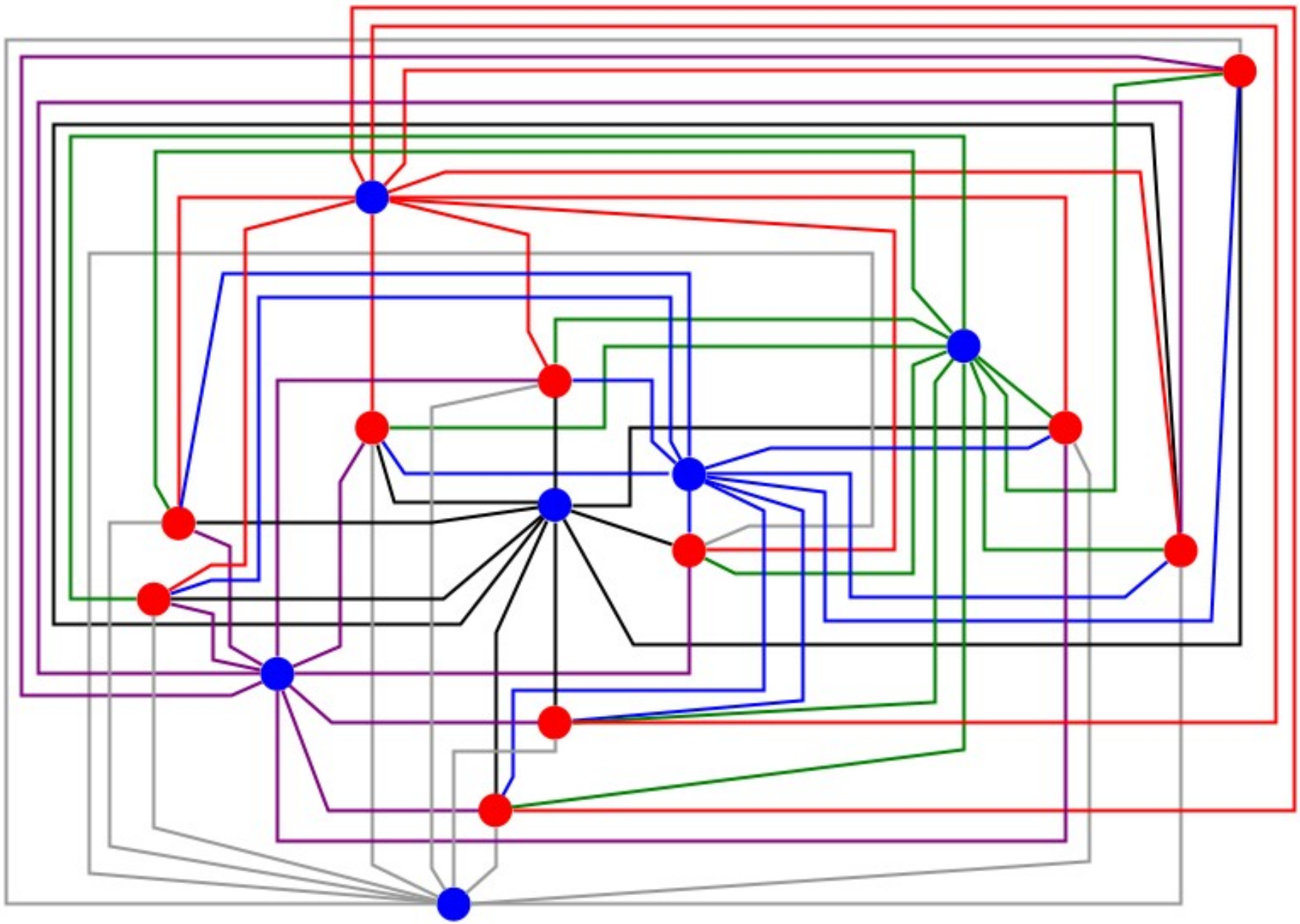
Results

class $\mathcal{C}$	$\in \mathcal{C}$	$\notin \mathcal{C}$
4-planar	$K_{3,18}$	$K_{3,19}$
	$K_{4,11}$	$K_{4,19}$
	$K_{5,8}$	$K_{5,19}$
	$K_{6,6}$	$K_{6,19}$
quasi-planar	$K_{4,n}$	–
	$K_{5,18}$	?
	$K_{6,10}$	?
	$K_{7,7}$	$K_{7,52}$

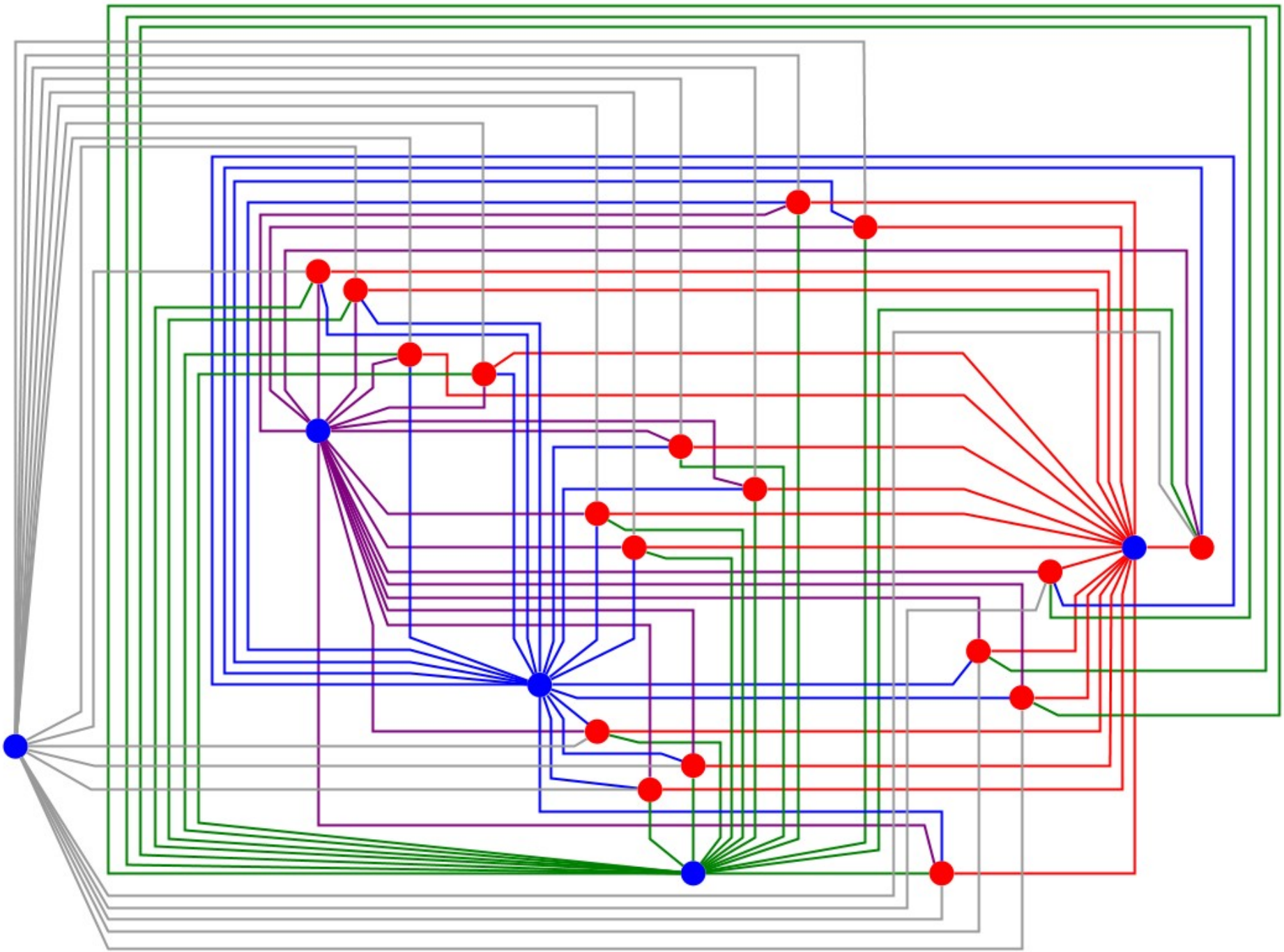
Quasiplanar Drawing of $K_{7,7}$



Quasiplanar Drawing of $K_{6,10}$



Quasiplanar Drawing of $K_{5,18}$



Open Problems

- Is it possible to extend the algorithm to other graphs (not complete and complete bipartite)?

Open Problems

- Is it possible to extend the algorithm to other graphs (not complete and complete bipartite)?
- Are there other variants of the algorithm that can reduce the search space even more?

Open Problems

- Is it possible to extend the algorithm to other graphs (not complete and complete bipartite)?
- Are there other variants of the algorithm that can reduce the search space even more?
- Can you draw 'bigger' complete (bipartite) graphs, if you allow that adjacent edges cross?

Open Problems

- Is it possible to extend the algorithm to other graphs (not complete and complete bipartite)?
- Are there other variants of the algorithm that can reduce the search space even more?
- Can you draw 'bigger' complete (bipartite) graphs, if you allow that adjacent edges cross?

Thank you for your attention !