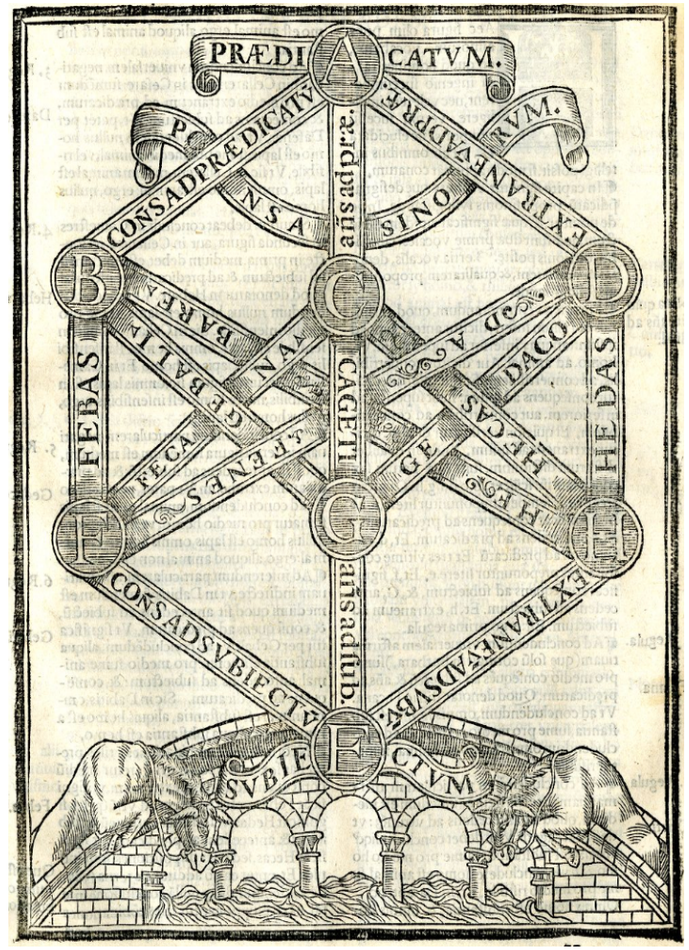
Minimizing Switches in Cased Graph Drawings

Master Thesis

Sebastian Röder

23.10.25

Supervisors: Robert Ganian, Martin Nöllenburg



Source: J. Pablos, "Recognitio, summularum" p. 116, 1554
Primeros Libros de las Américas, Biblioteca Nacional de México

TU WIEN

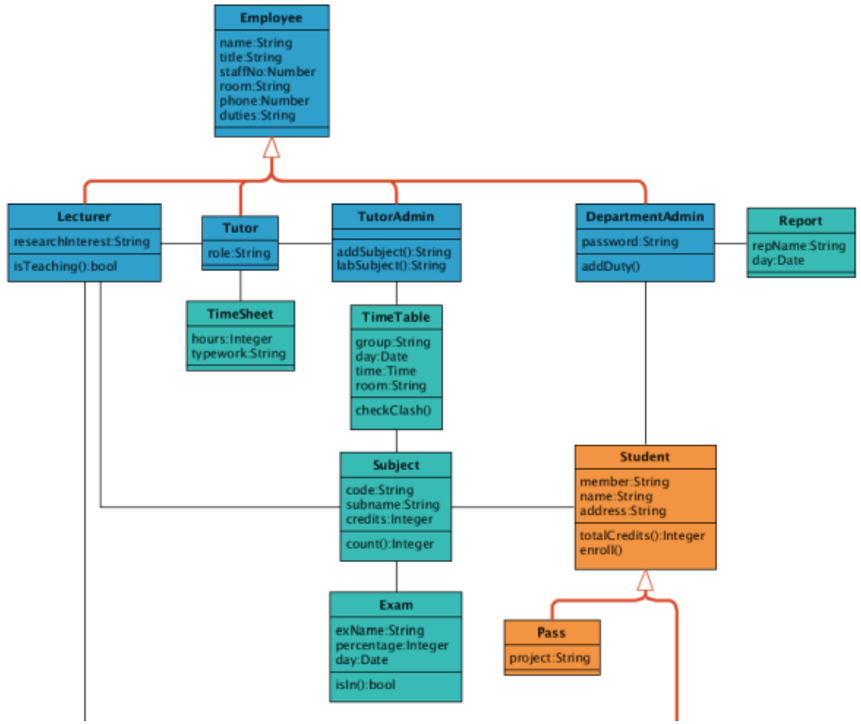
ALGORITHMS AND COMPLEXITY GROUP

Graph Drawing

- Study of creating geometric representations of graphs

Graph Drawing

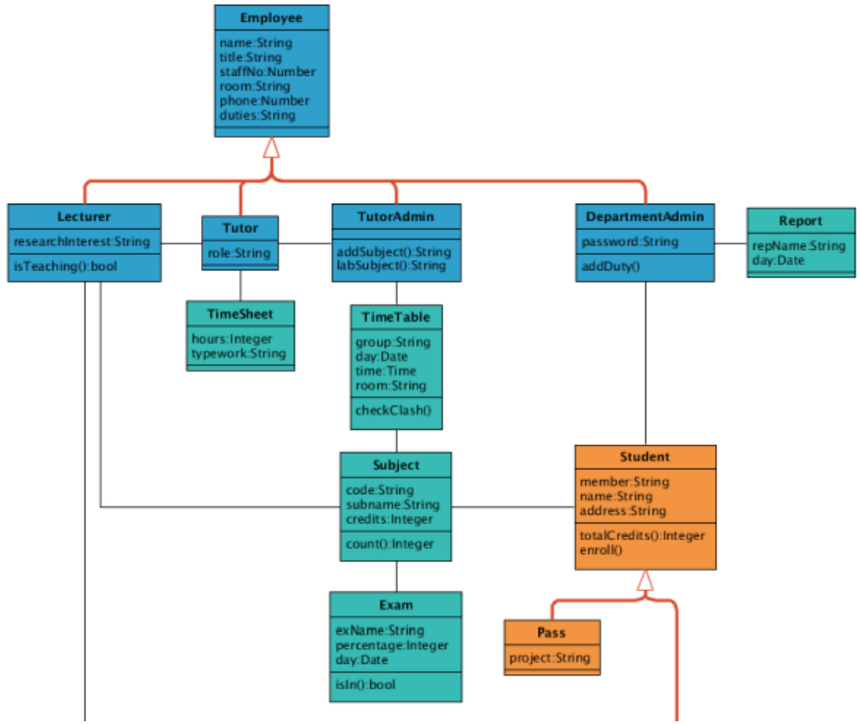
- Study of creating geometric representations of graphs



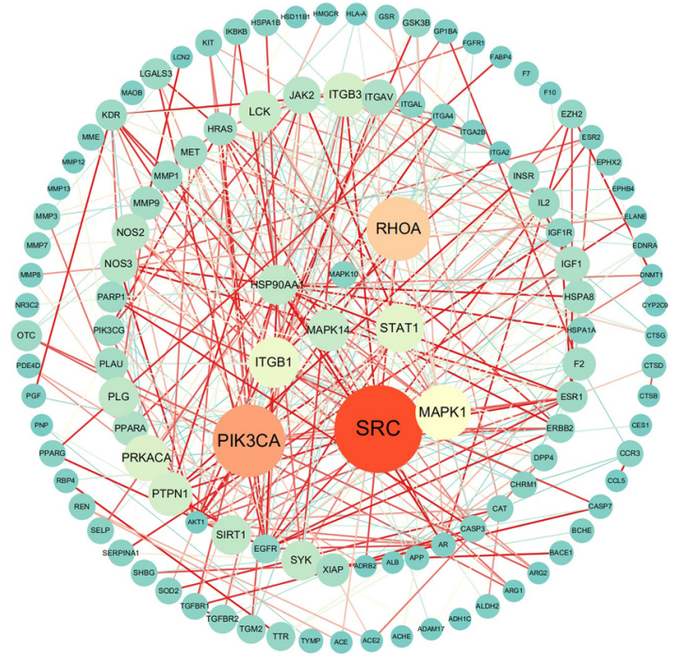
[yworks.com]

Graph Drawing

- Study of creating geometric representations of graphs



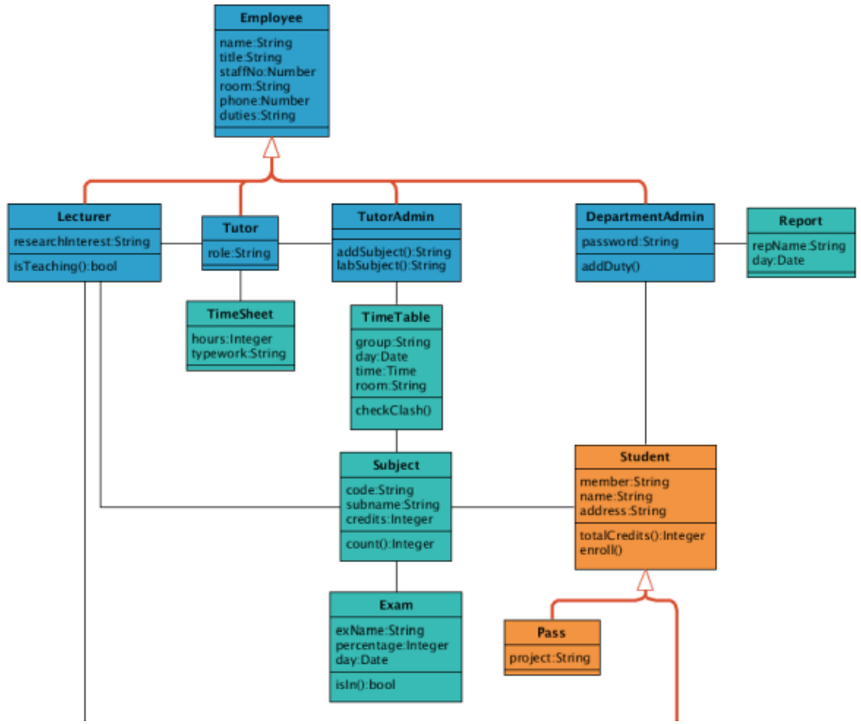
[yworks.com]



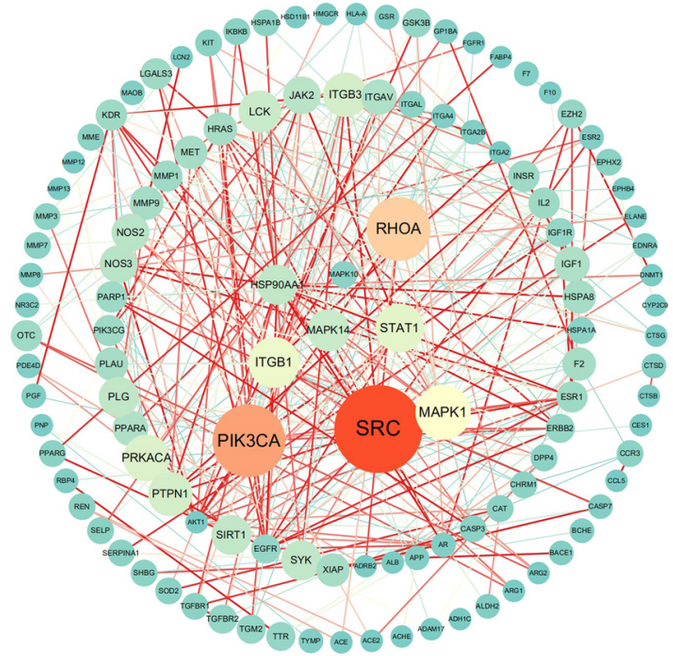
[Lv et al., 2022]

Graph Drawing

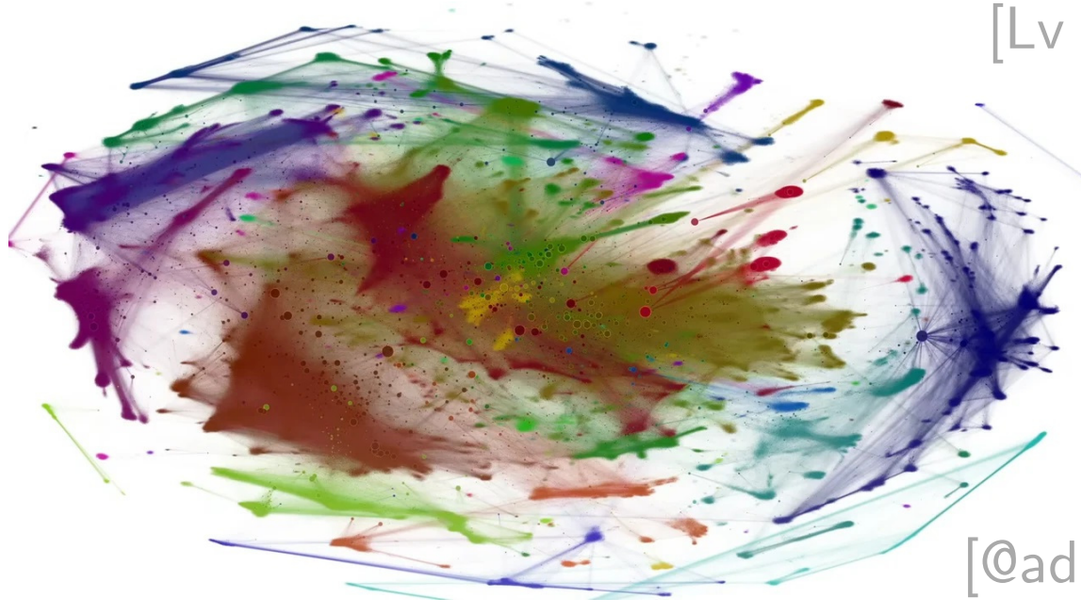
- Study of creating geometric representations of graphs



[yworks.com]

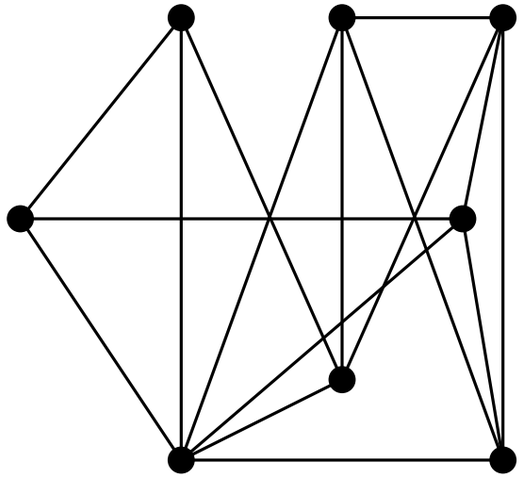


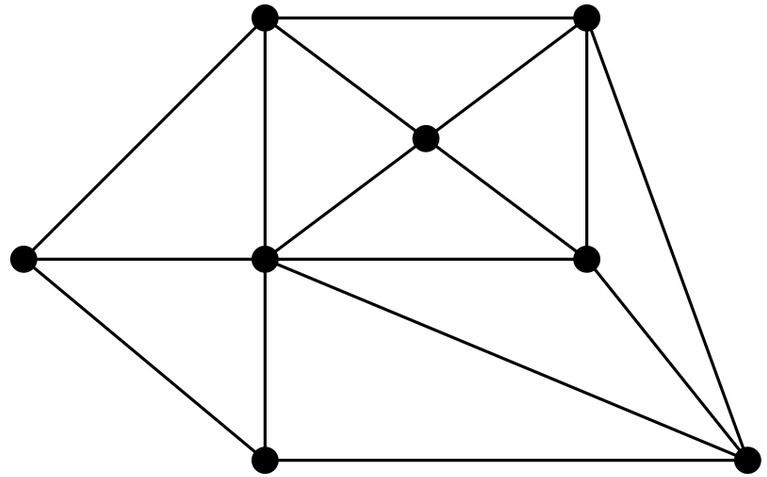
[Lv et al., 2022]



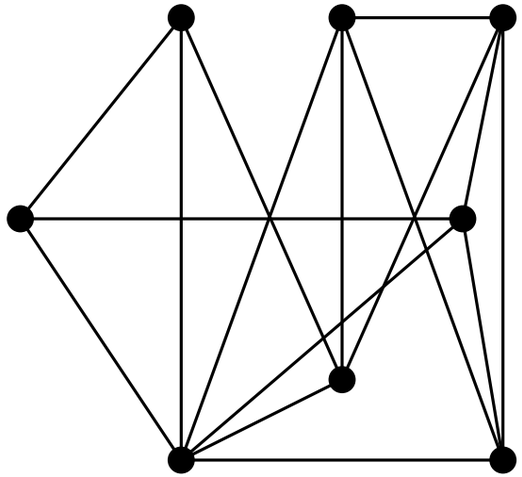
[@adum_codes]

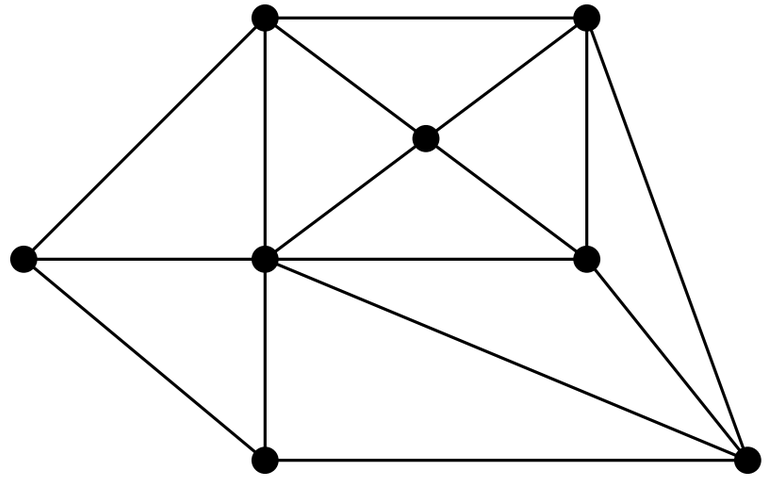
Edge Crossings





Edge Crossings

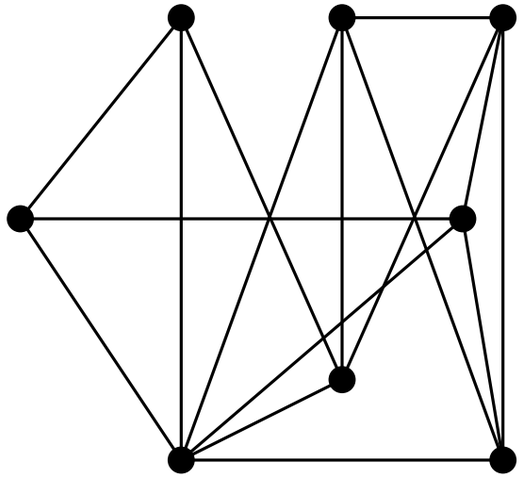


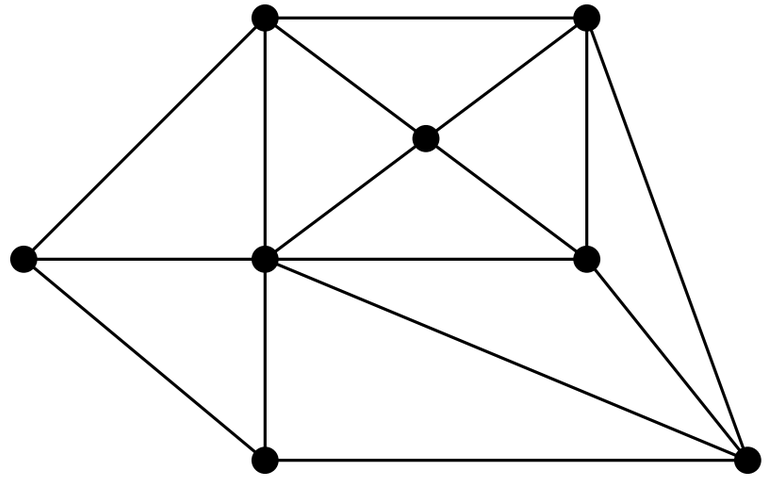


However:

- Minimizing crossings is NP-complete

Edge Crossings





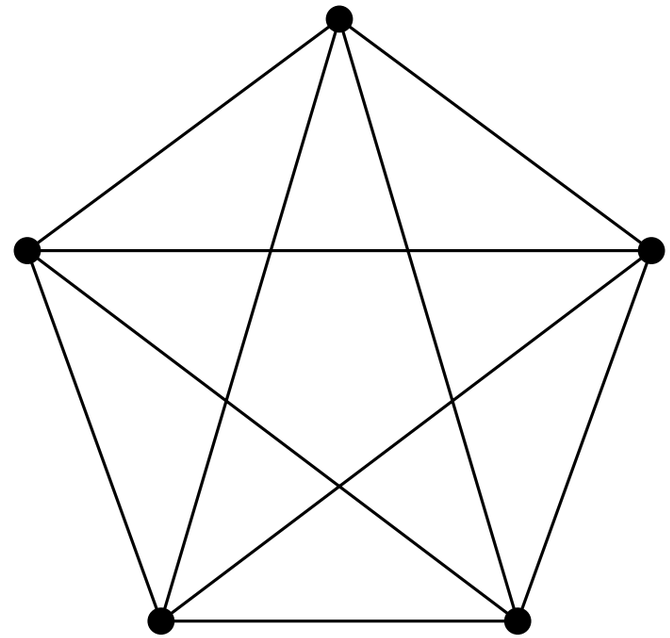
However:

- Minimizing crossings is NP-complete
- Not always possible to remove crossings

Beyond Planarity

Possible solutions:

K_5

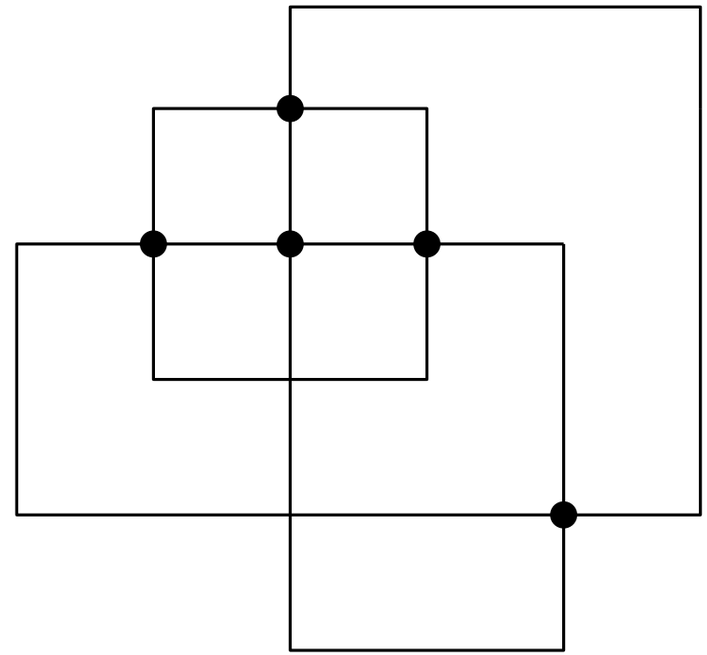


Beyond Planarity

Possible solutions:

- Orthogonal crossings

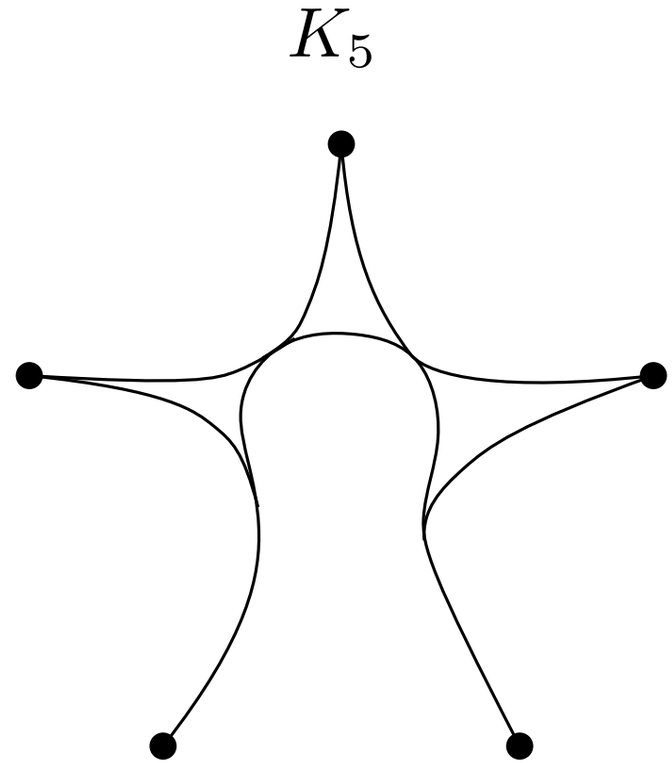
K_5



Beyond Planarity

Possible solutions:

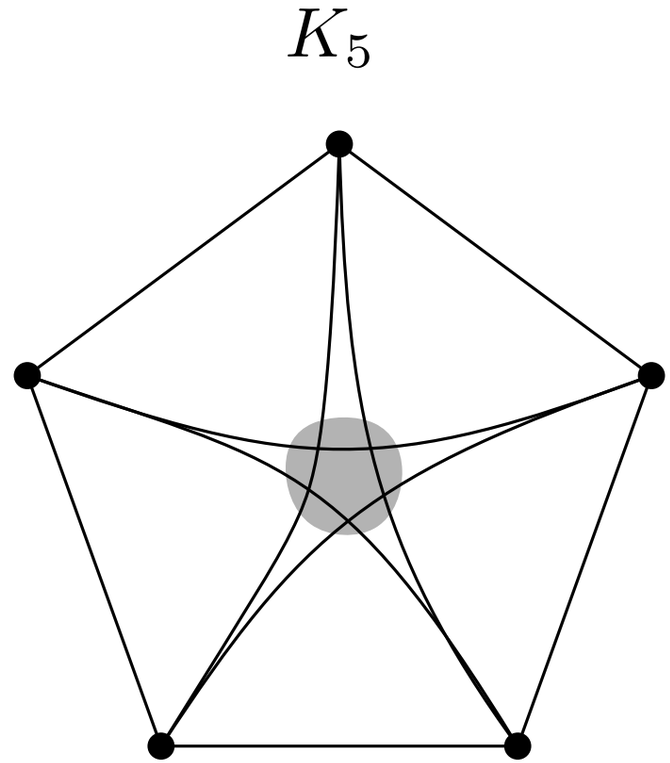
- Orthogonal crossings
- Confluent drawings



Beyond Planarity

Possible solutions:

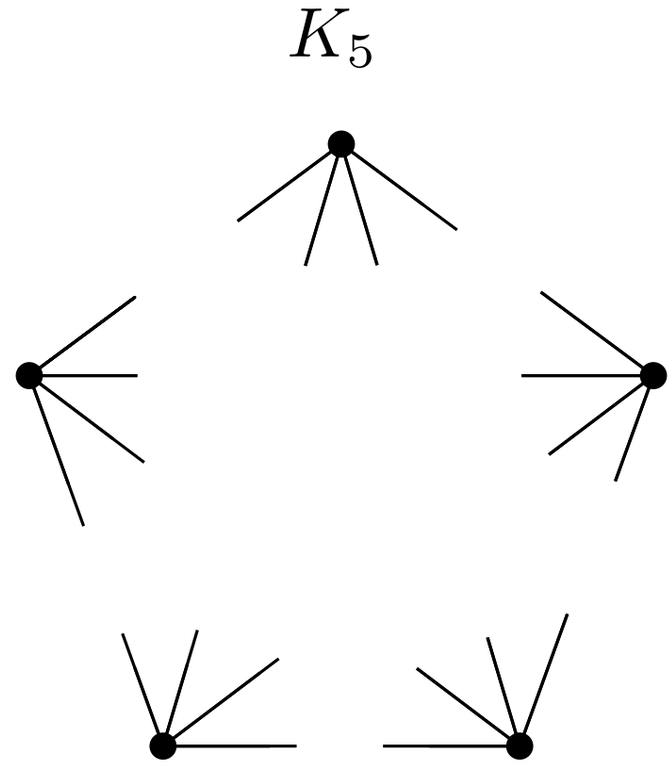
- Orthogonal crossings
- Confluent drawings
- Bundled crossings



Beyond Planarity

Possible solutions:

- Orthogonal crossings
- Confluent drawings
- Bundled crossings
- Partial edge drawings

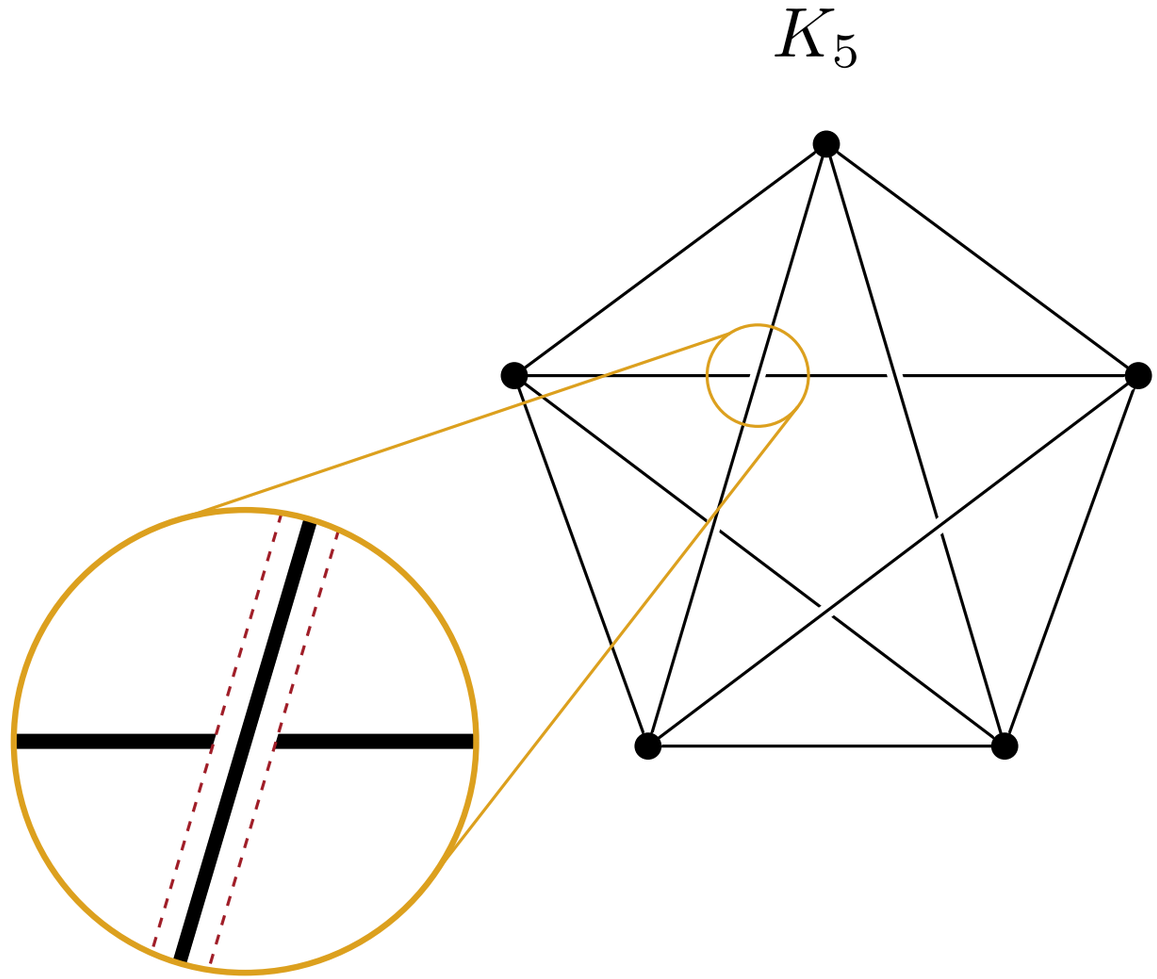


Beyond Planarity

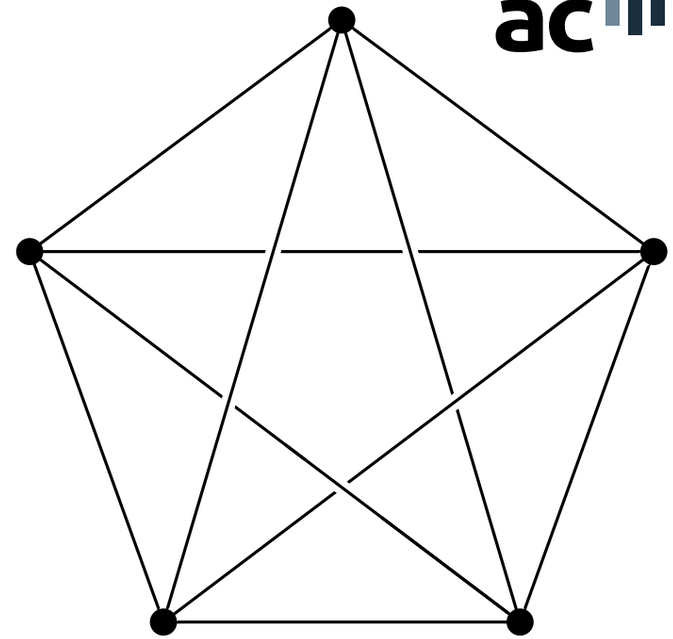
Possible solutions:

- Orthogonal crossings
- Confluent drawings
- Bundled crossings
- Partial edge drawings
- Cased drawings

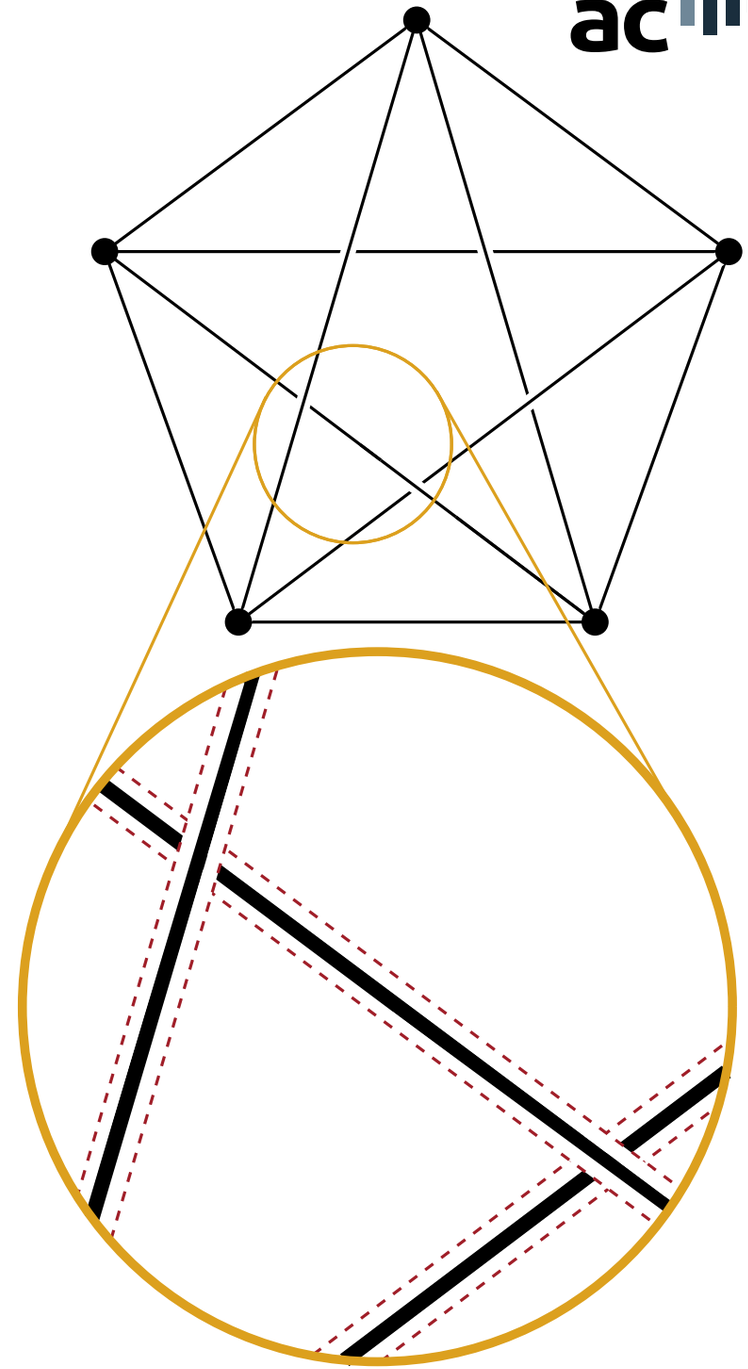
[Eppstein et al., 2007]



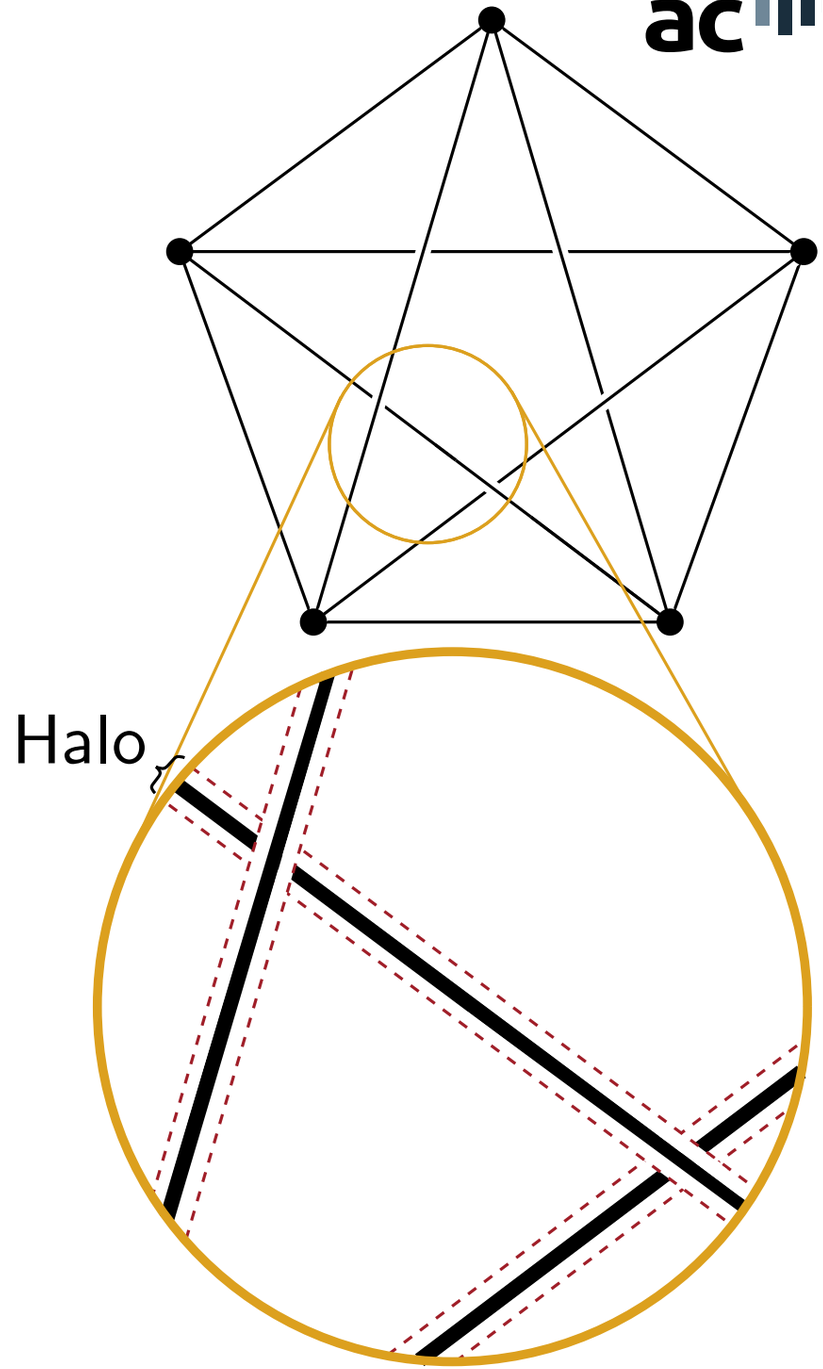
Cased drawings



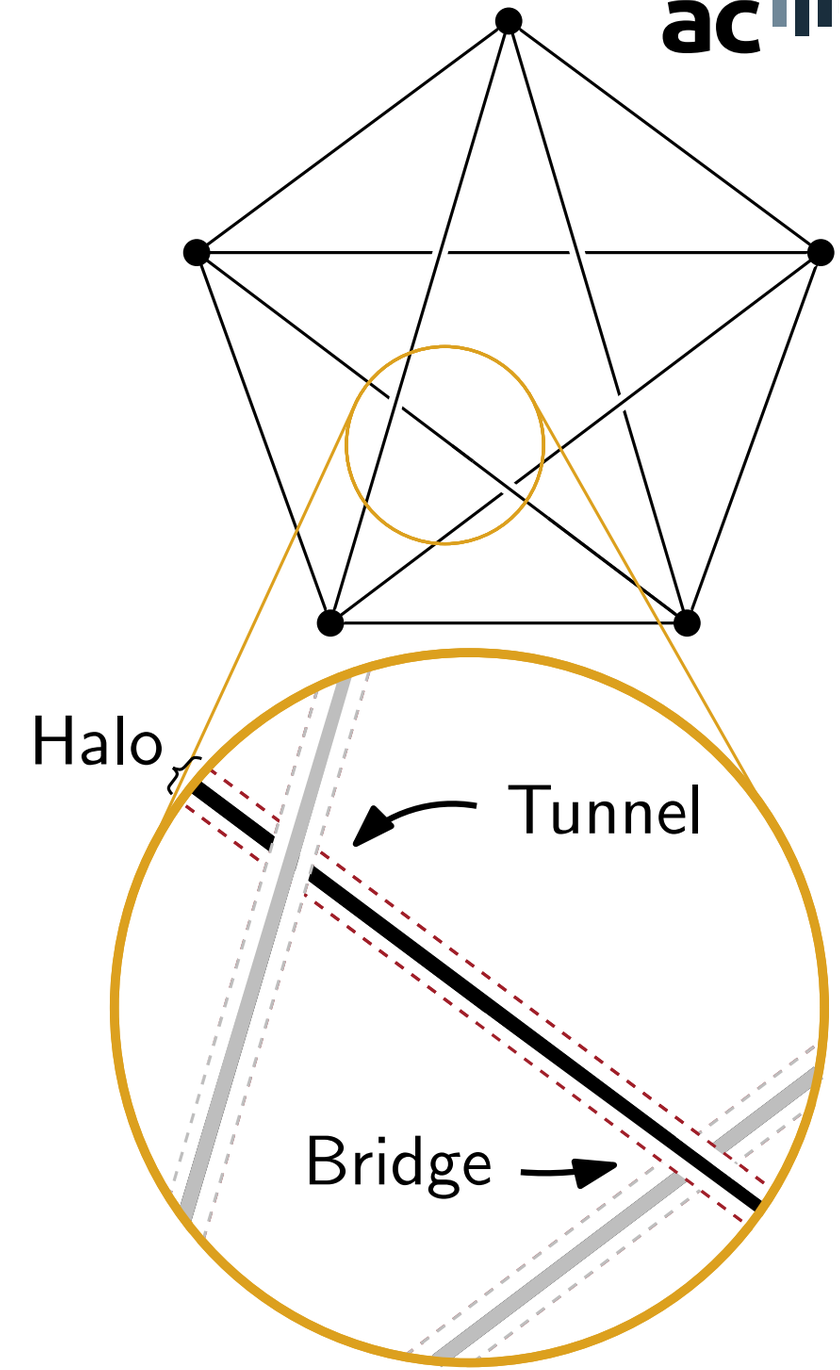
Cased drawings



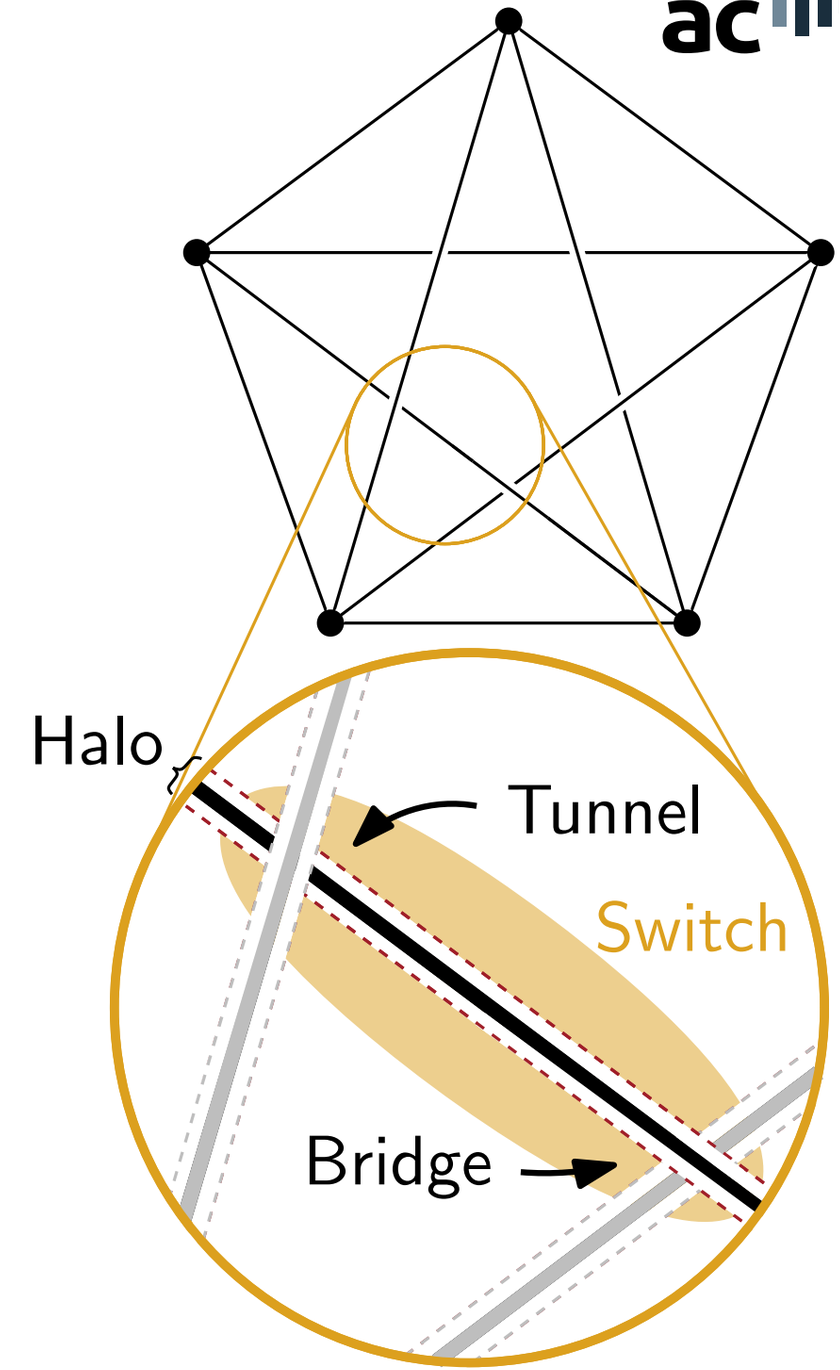
Cased drawings



Cased drawings

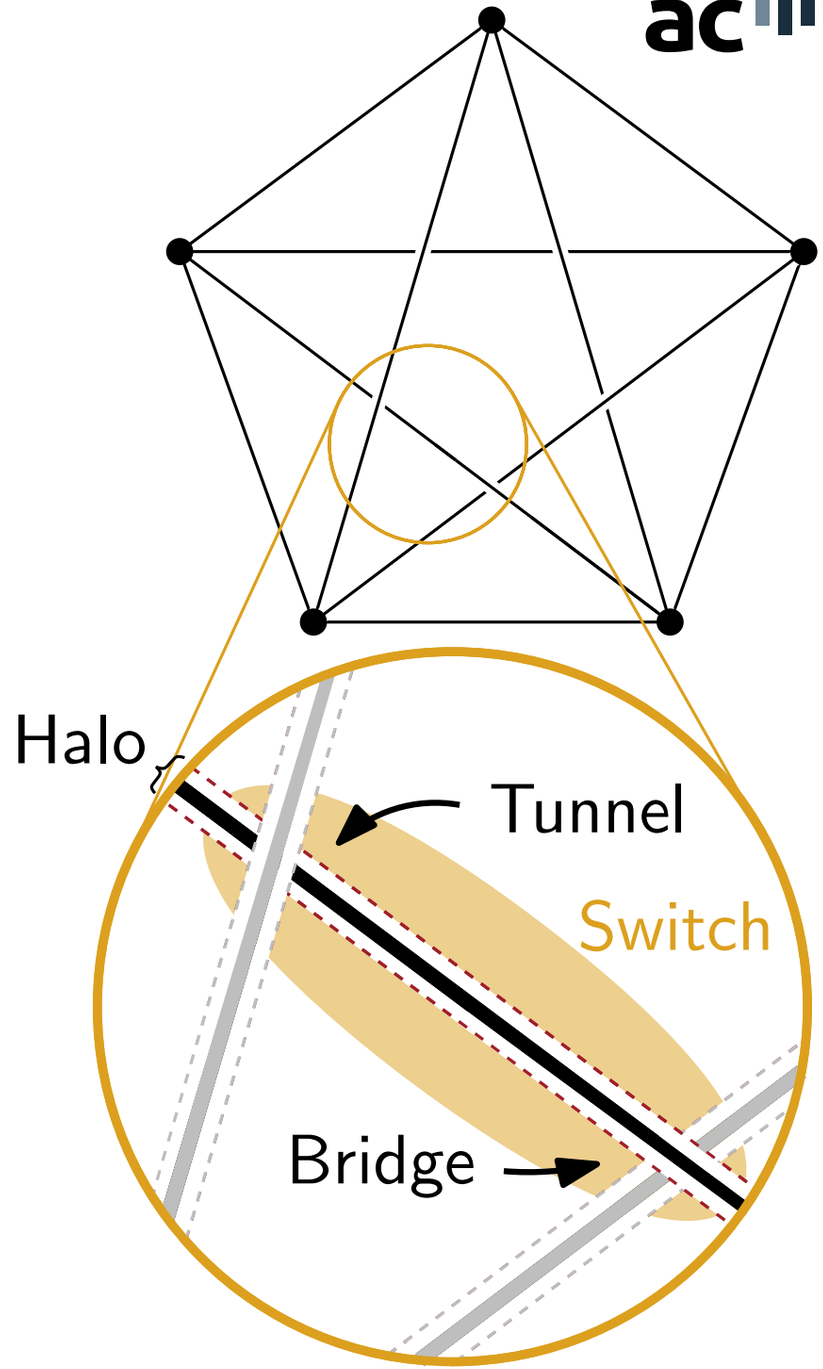


Cased drawings



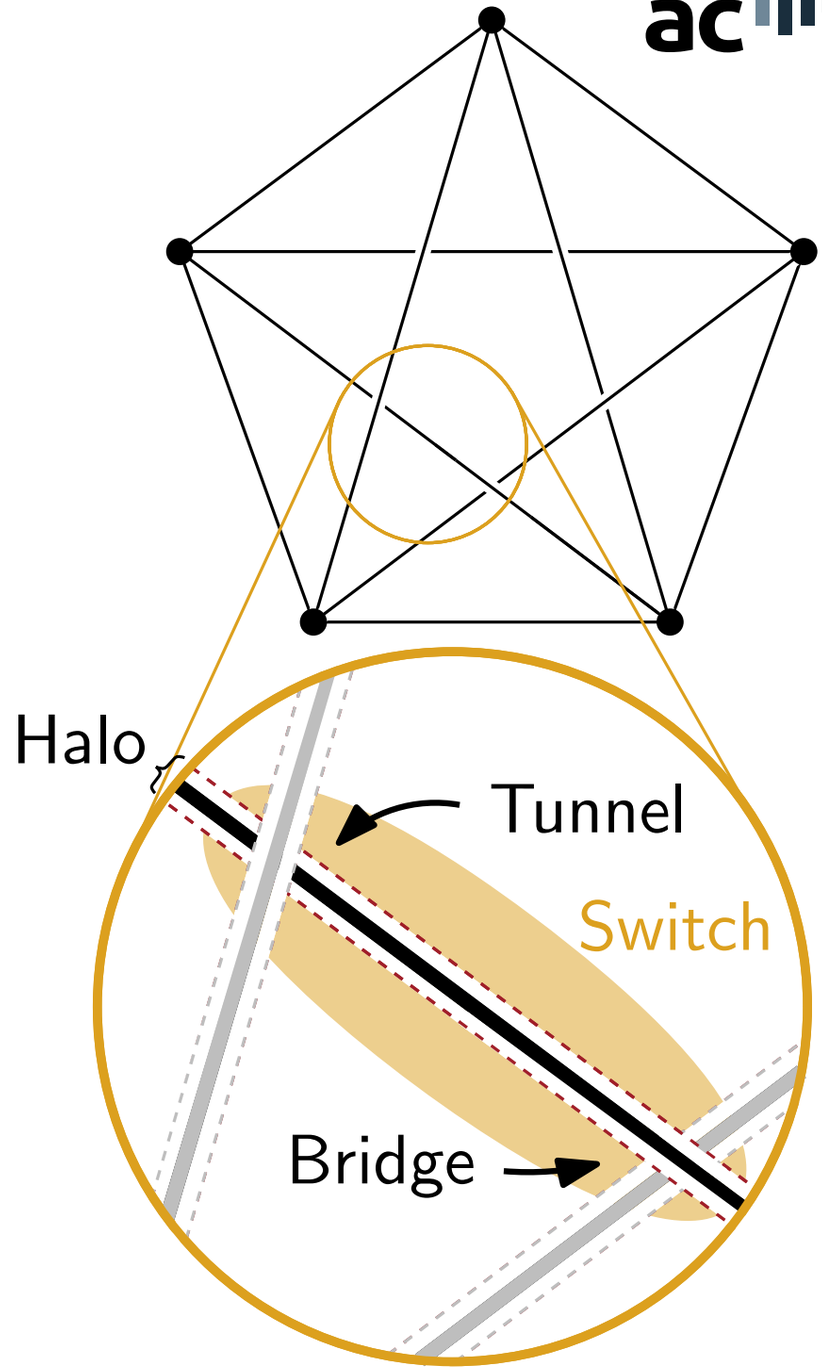
Cased drawings

- Post-processing on fixed drawing



Cased drawings

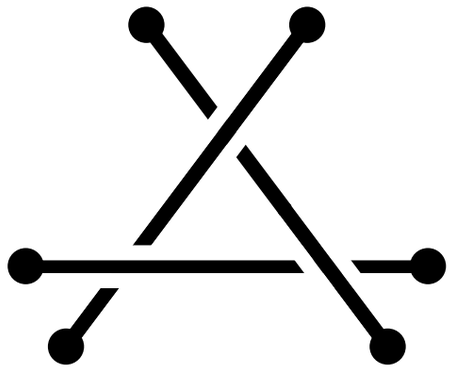
- Post-processing on fixed drawing
- Straight-line drawings

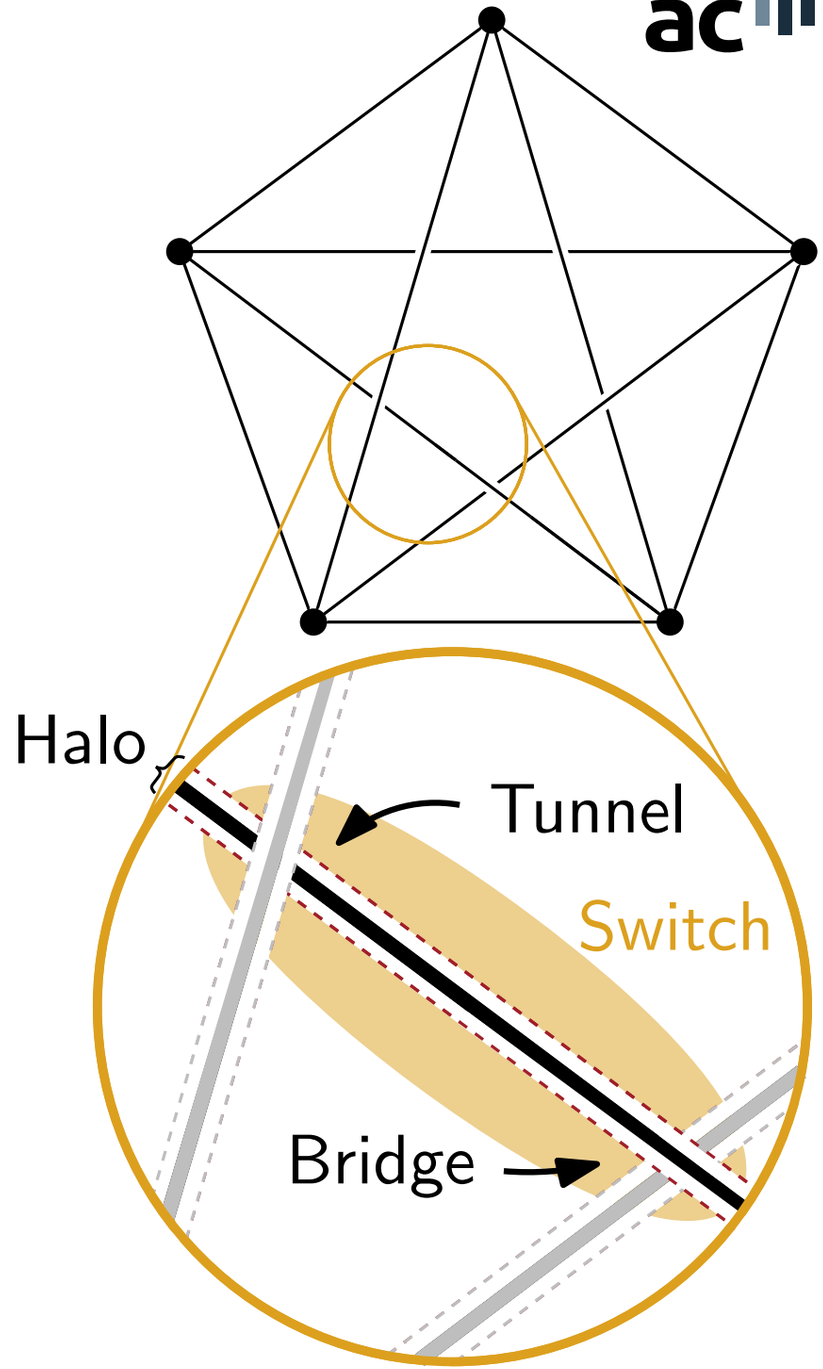


Cased drawings

- Post-processing on fixed drawing
- Straight-line drawings

Weaving model

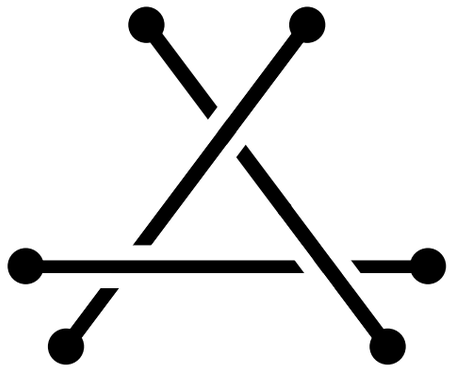




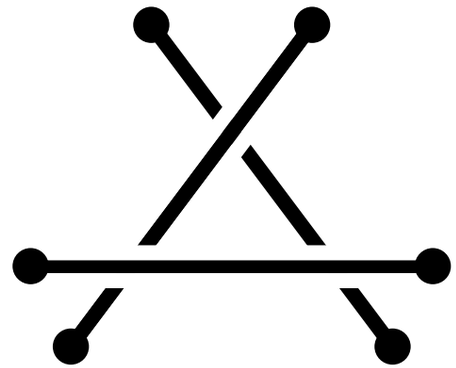
Cased drawings

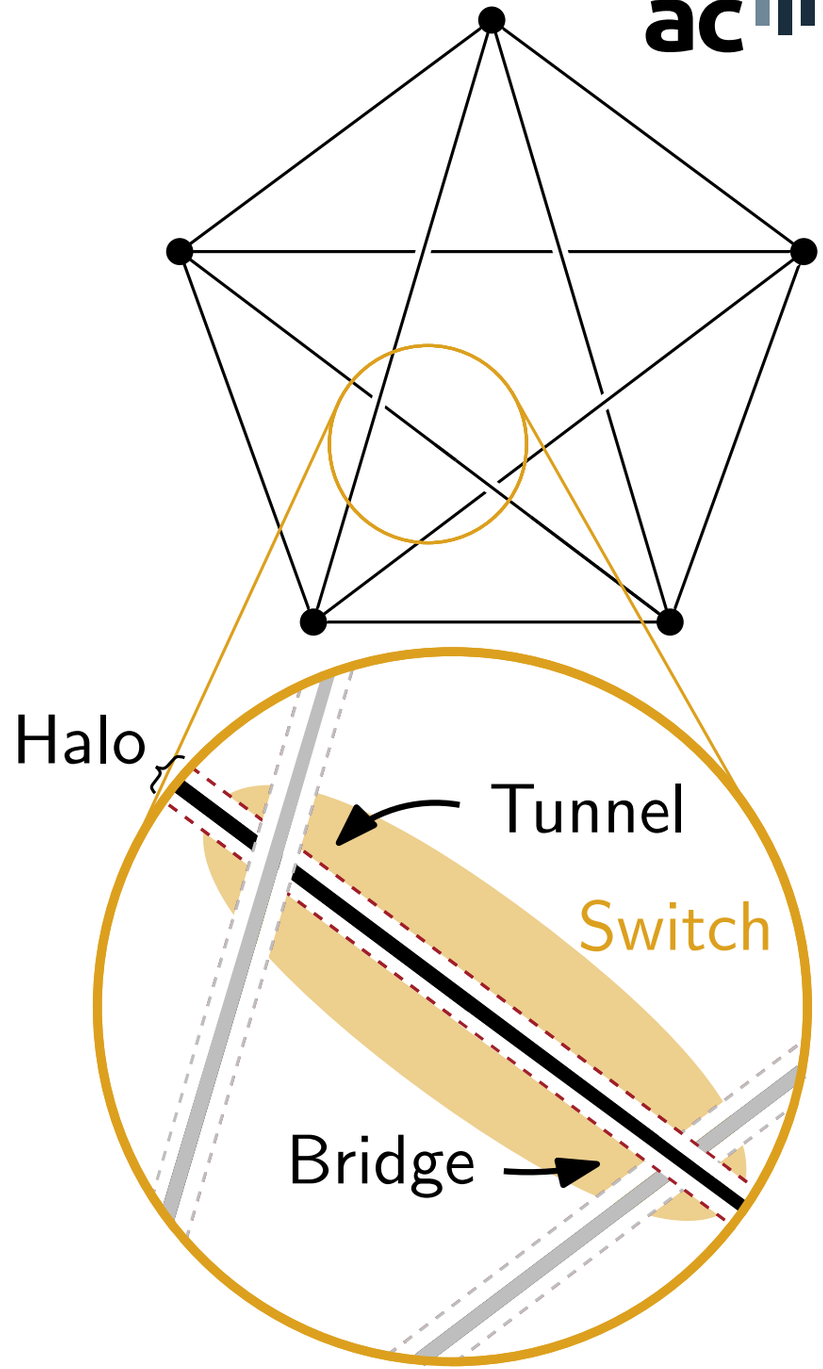
- Post-processing on fixed drawing
- Straight-line drawings

Weaving model



Stacking model

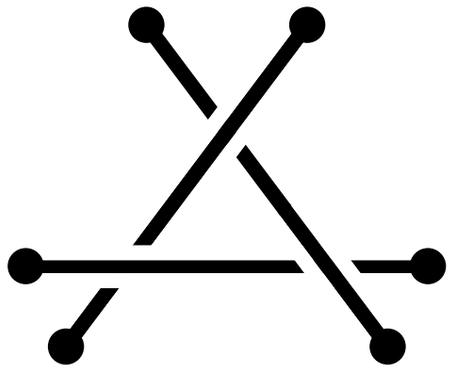




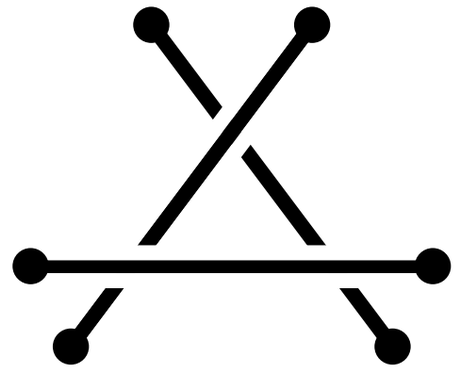
Cased drawings

- Post-processing on fixed drawing
- Straight-line drawings

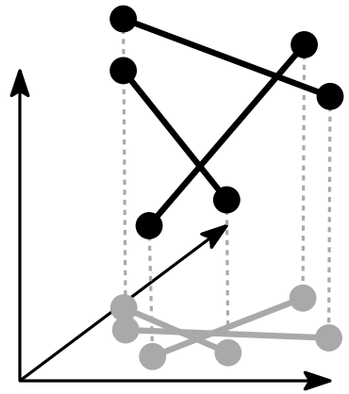
Weaving model

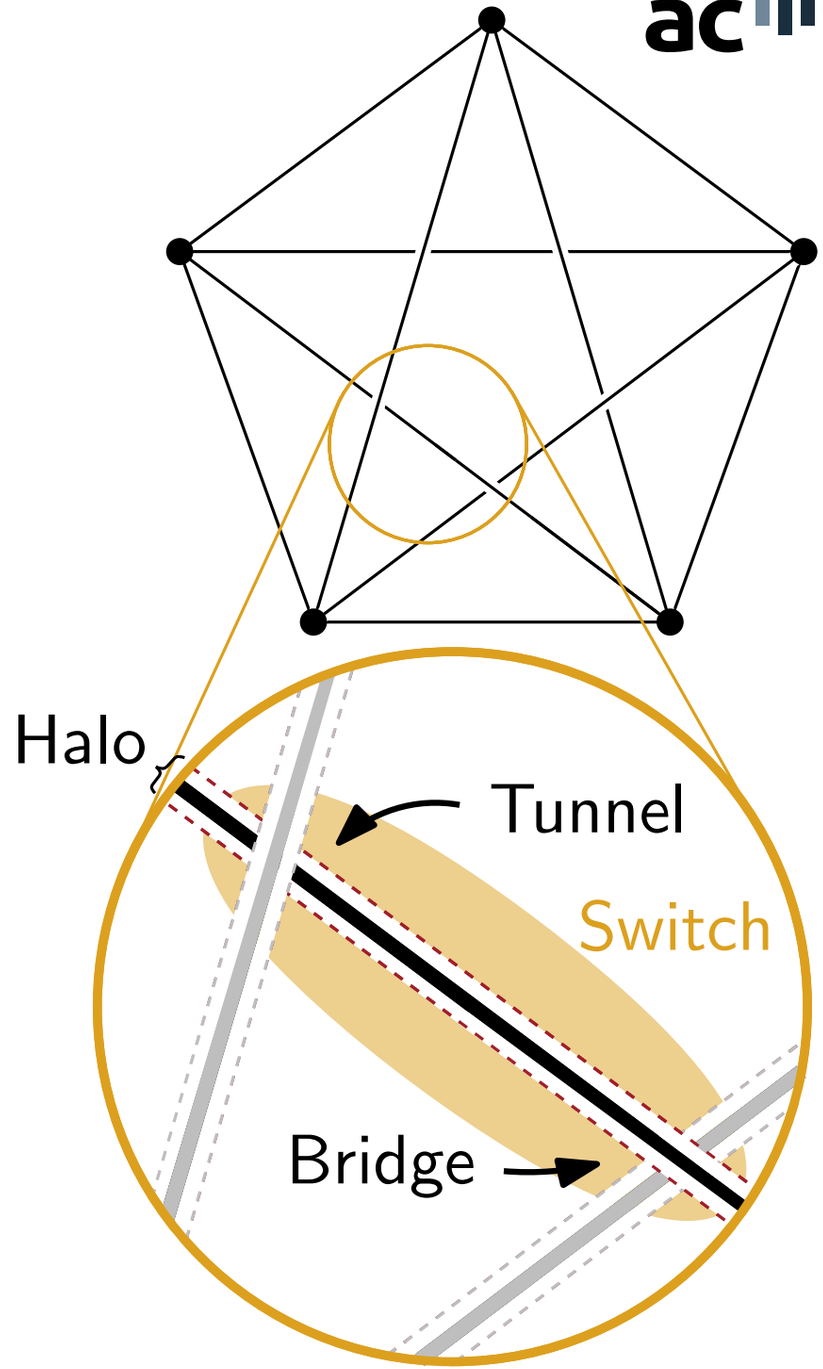


Stacking model



Realizable model



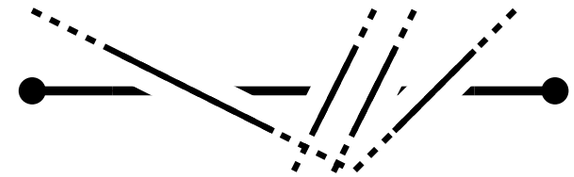


Quality of casings

- Prevent:

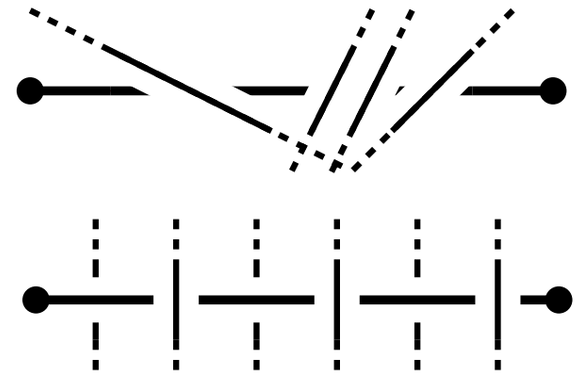
Quality of casings

- Prevent: 1. Edges hidden by too many tunnels



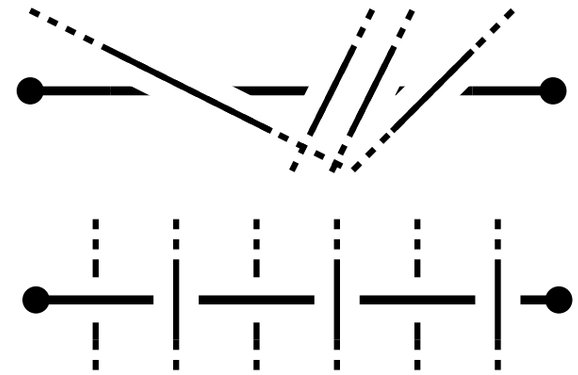
Quality of casings

- Prevent:
  1. Edges hidden by too many tunnels
  2. Too many switches



Quality of casings

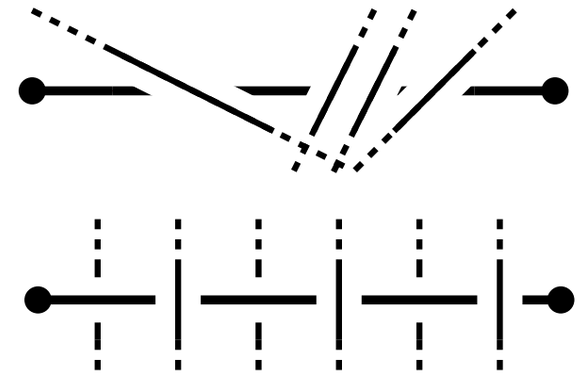
- Prevent:
  1. Edges hidden by too many tunnels
  2. Too many switches



- MinTotalSwitches
- MaxTotalSwitches
- MinMaxSwitches
- MinSwitchEdges

Quality of casings

- Prevent:
  1. Edges hidden by too many tunnels
  2. Too many switches



	Stacking	**Weaving**	**Realizable**
MinTotalSwitches	*open*	$O(m^4)$	*open*
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	*open*	*open*
MinSwitchEdges	*open*	*open*	*open*

Results

Hardness	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	$O(m^4)$	*open*
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	*open*	*open*
MinSwitchEdges	*open*	*open*	*open*

Results

Hardness	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	$O(m^4)$	NP-hard
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Results

Hardness	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

* with restrictions

Results

Hardness	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	*open*	*open*
MinSwitchEdges	*open*	*open*	*open*

* with restrictions

Results

Hardness	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	VC(χ_G)	*open*
MinSwitchEdges	tw(K_{δ_G})	tw(K_{δ_G})	tw(K_{δ_G})

* with restrictions

Results

Hardness	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	VC(χ_G)	*open*
MinSwitchEdges	tw(K_{δ_G})	tw(K_{δ_G})	tw(K_{δ_G})

- k-switch-planar graphs

* with restrictions

Results

Hardness	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	VC(χ_G)	*open*
MinSwitchEdges	tw(K_{δ_G})	tw(K_{δ_G})	tw(K_{δ_G})

- k-switch-planar graphs
- MILP implementation

* with restrictions

Results

Hardness	Stacking	**Weaving**	Realizable
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	$\mathsf{VC}(\chi_G)$	*open*
MinSwitchEdges	$\mathsf{tw}(K_{\delta_G})$	$\mathsf{tw}(K_{\delta_G})$	$\mathsf{tw}(K_{\delta_G})$

- k-switch-planar graphs
- MILP implementation

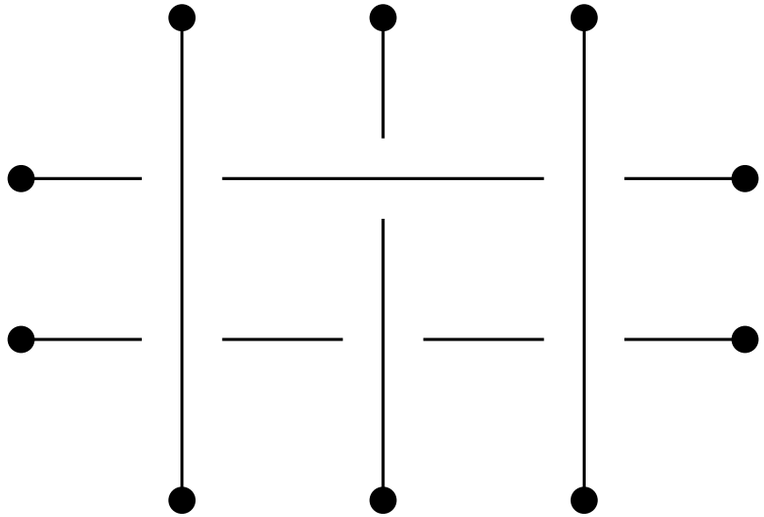
* with restrictions

MinMaxSwitches is NP-complete

- Minimize maximum number of switches per edge

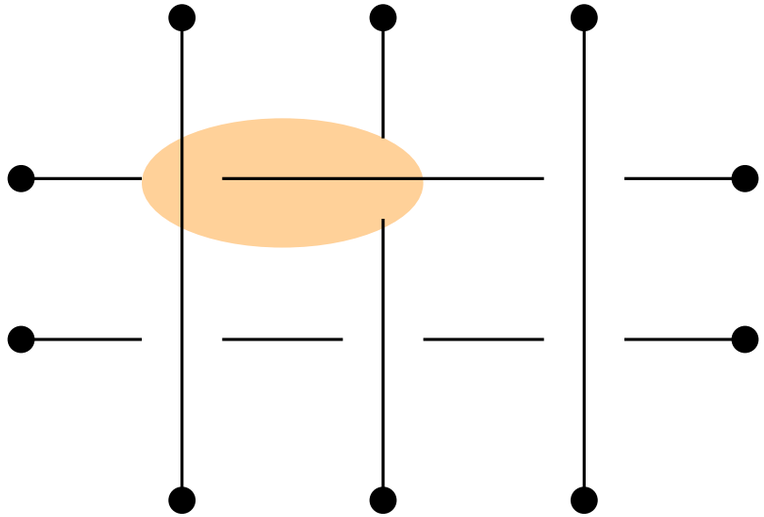
MinMaxSwitches is NP-complete

- Minimize maximum number of switches per edge



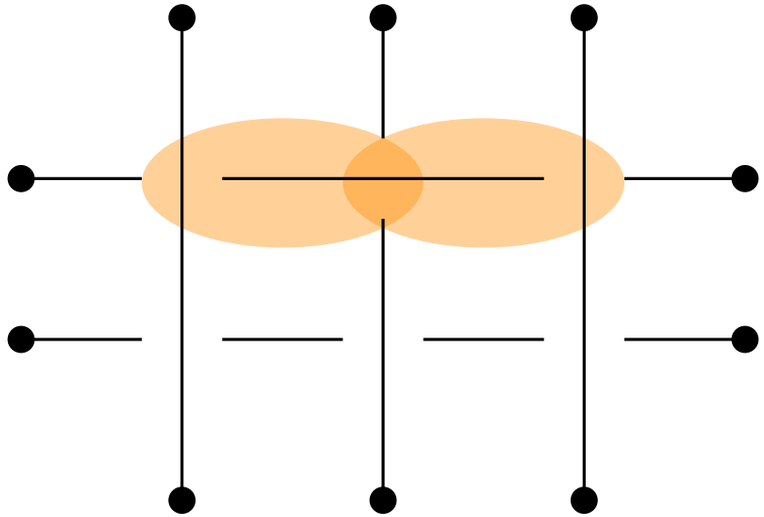
MinMaxSwitches is NP-complete

- Minimize maximum number of switches per edge



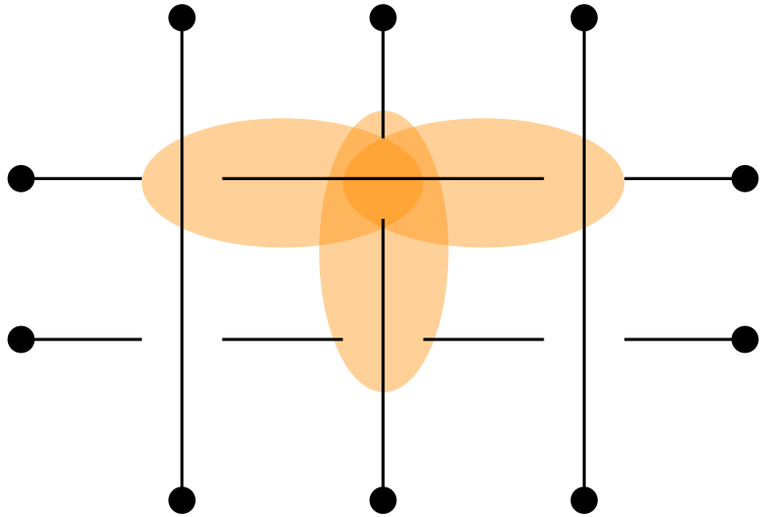
MinMaxSwitches is NP-complete

- Minimize maximum number of switches per edge



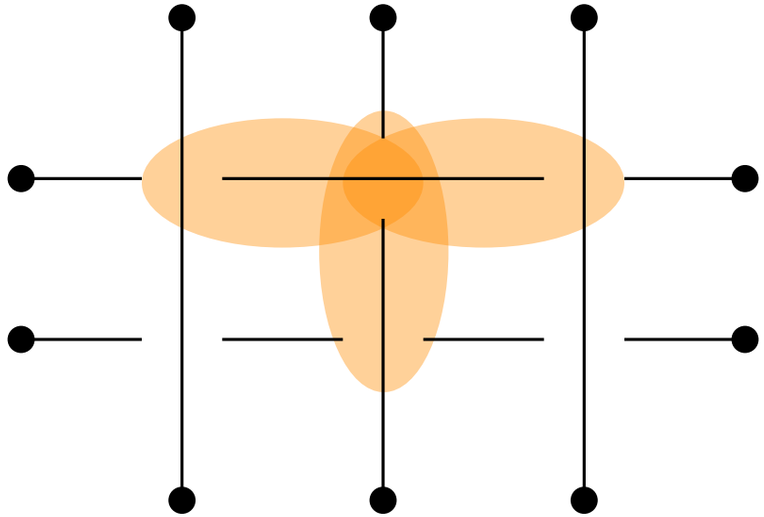
MinMaxSwitches is NP-complete

- Minimize maximum number of switches per edge



MinMaxSwitches is NP-complete

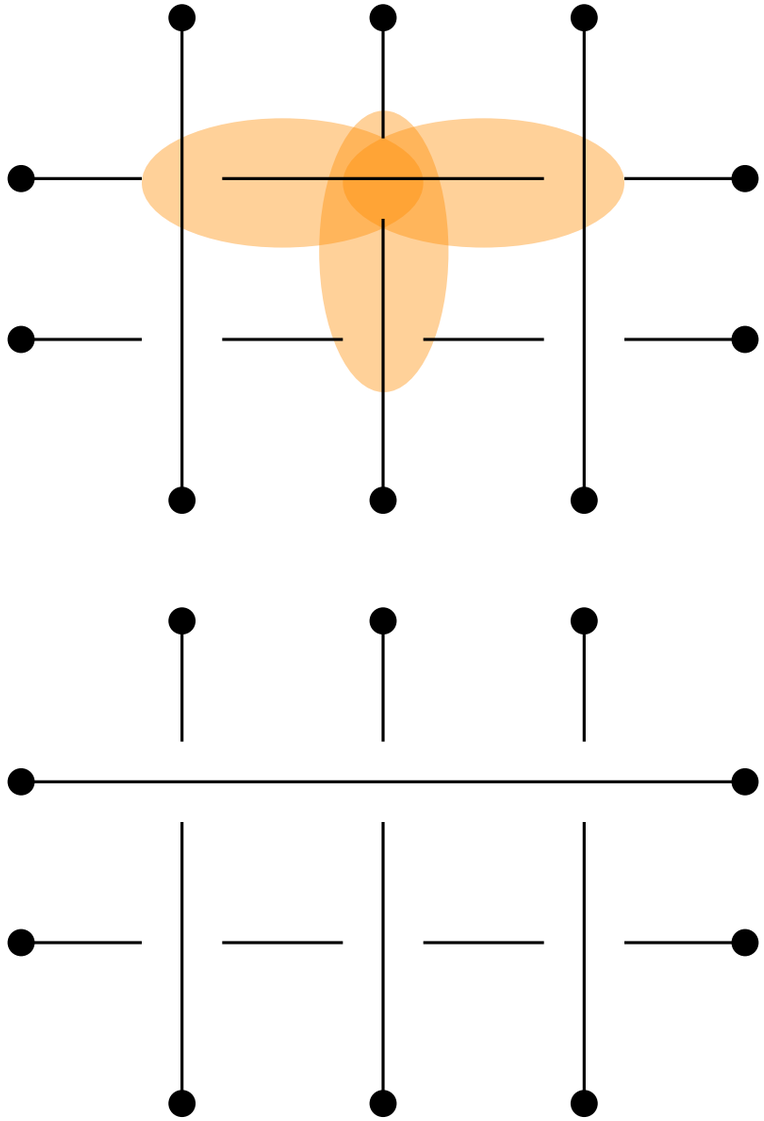
- Minimize maximum number of switches per edge



3 total switches
max. 2 switches per edge

MinMaxSwitches is NP-complete

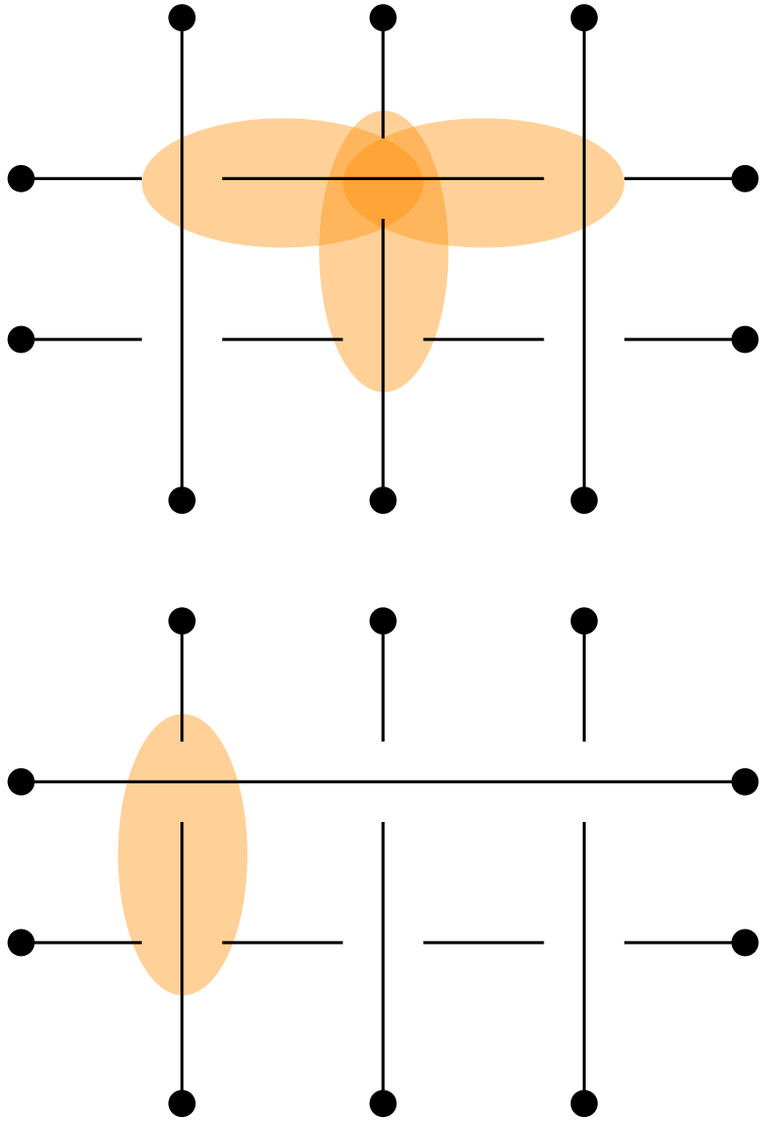
- Minimize maximum number of switches per edge



3 total switches
max. 2 switches per edge

MinMaxSwitches is NP-complete

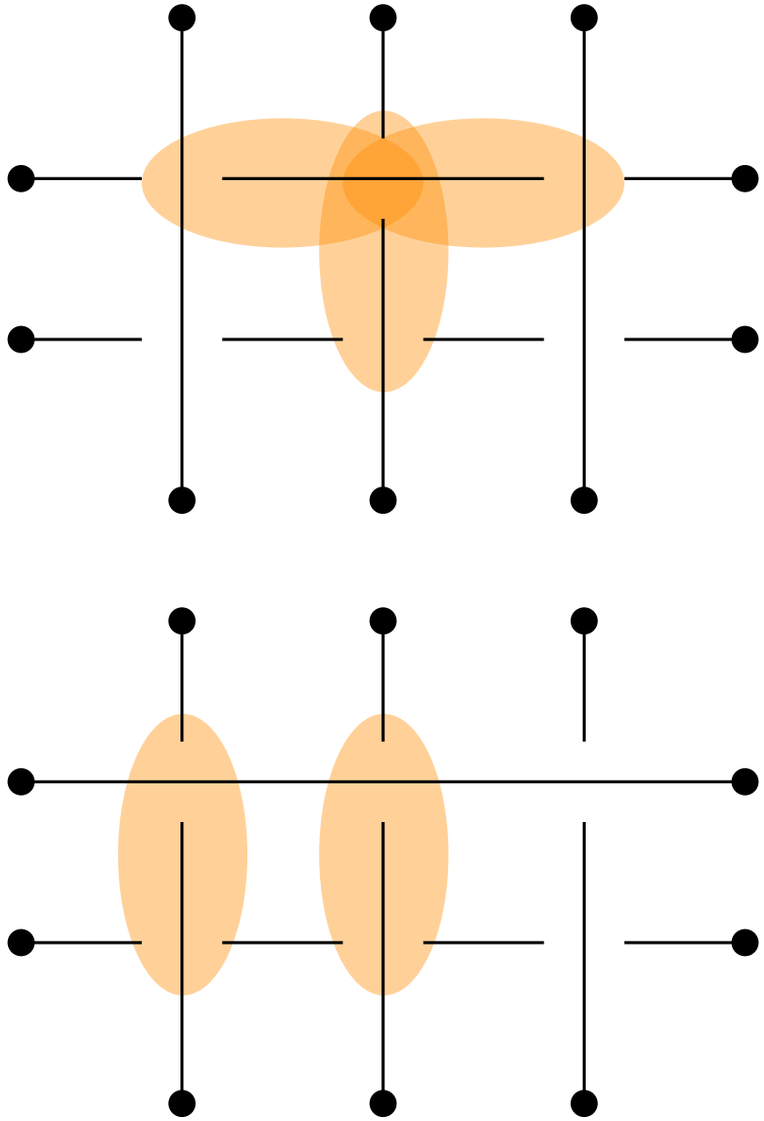
- Minimize maximum number of switches per edge



3 total switches
max. 2 switches per edge

MinMaxSwitches is NP-complete

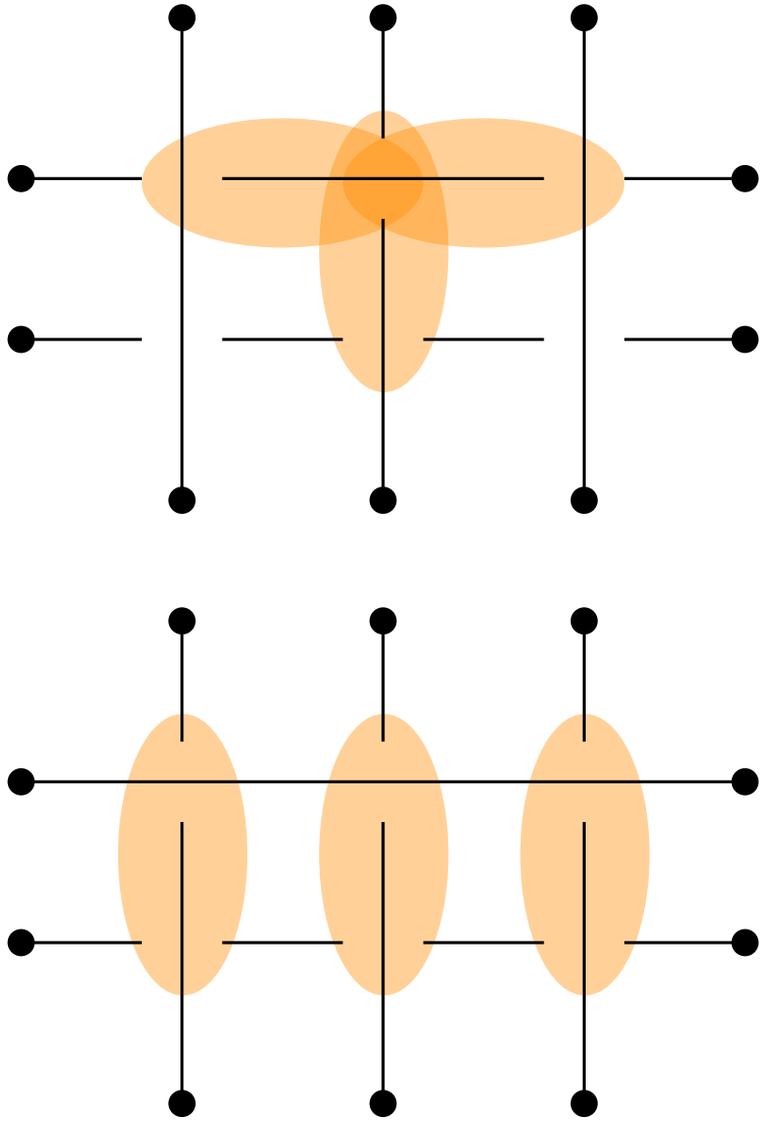
- Minimize maximum number of switches per edge



3 total switches
max. 2 switches per edge

MinMaxSwitches is NP-complete

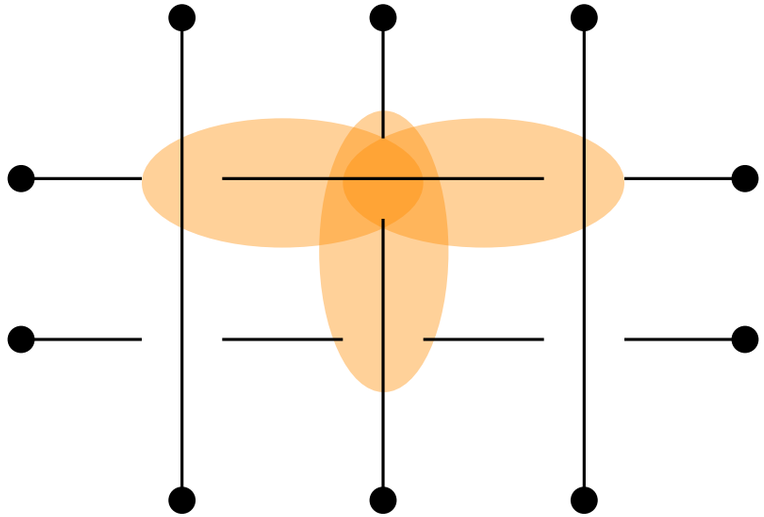
- Minimize maximum number of switches per edge



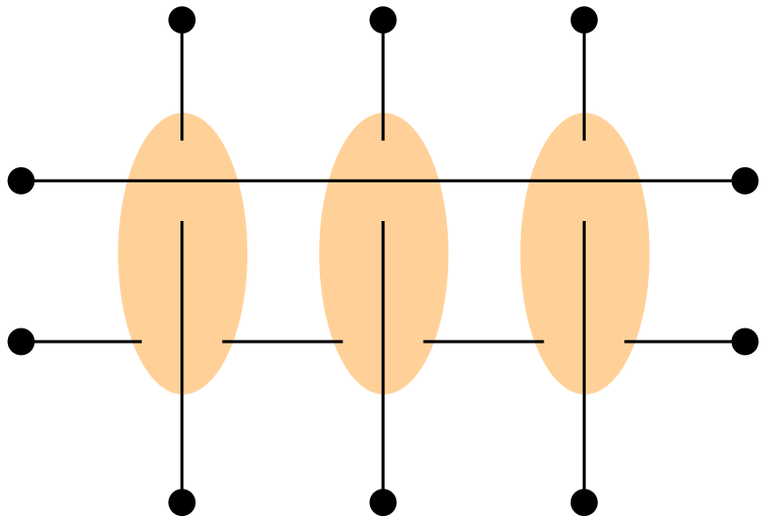
3 total switches
max. 2 switches per edge

MinMaxSwitches is NP-complete

- Minimize maximum number of switches per edge



3 total switches
max. 2 switches per edge



3 total switches
max. 1 switches per edge

MinMaxSwitches is NP-complete

- Reduction from NAE 3SAT

MinMaxSwitches is NP-complete

- Reduction from NAE 3SAT

Not-all-equal 3SAT (NAE 3SAT)

$$\varphi := \bigwedge_{i=1}^{m} \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$$

$$\mathsf{NAE}(x, y, z) := x \neq y \vee y \neq z \vee z \neq x$$

MinMaxSwitches is NP-complete

- Reduction from NAE 3SAT

Not-all-equal 3SAT (NAE 3SAT)

$$\varphi := \bigwedge_{i=1}^{m} \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$$

$$\mathsf{NAE}(x, y, z) := x \neq y \vee y \neq z \vee z \neq x$$

- Proof idea: $\varphi = \mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$

MinMaxSwitches is NP-complete

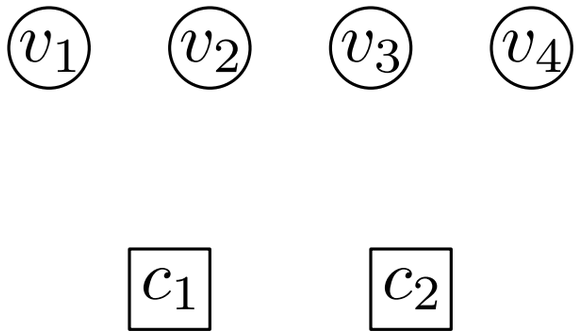
- Reduction from NAE 3SAT

Not-all-equal 3SAT (NAE 3SAT)

$$\varphi := \bigwedge_{i=1}^{m} \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$$

$$\mathsf{NAE}(x, y, z) := x \neq y \vee y \neq z \vee z \neq x$$

- Proof idea: $\varphi = \mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$



MinMaxSwitches is NP-complete

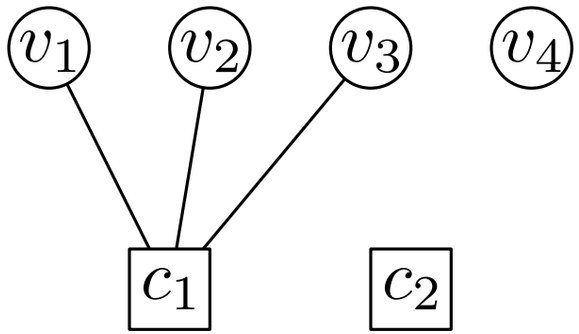
- Reduction from NAE 3SAT

Not-all-equal 3SAT (NAE 3SAT)

$$\varphi := \bigwedge_{i=1}^{m} \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$$

$$\mathsf{NAE}(x, y, z) := x \neq y \vee y \neq z \vee z \neq x$$

- Proof idea: $\varphi = \mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$



MinMaxSwitches is NP-complete

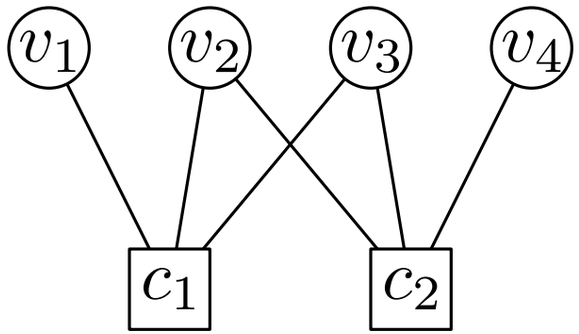
- Reduction from NAE 3SAT

Not-all-equal 3SAT (NAE 3SAT)

$$\varphi := \bigwedge_{i=1}^{m} \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$$

$$\mathsf{NAE}(x, y, z) := x \neq y \vee y \neq z \vee z \neq x$$

- Proof idea: $\varphi = \mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$



MinMaxSwitches is NP-complete

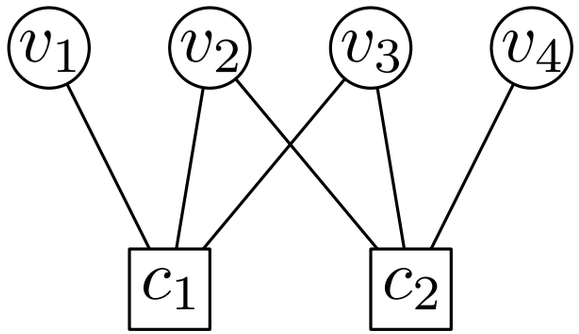
- Reduction from NAE 3SAT

Not-all-equal 3SAT (NAE 3SAT)

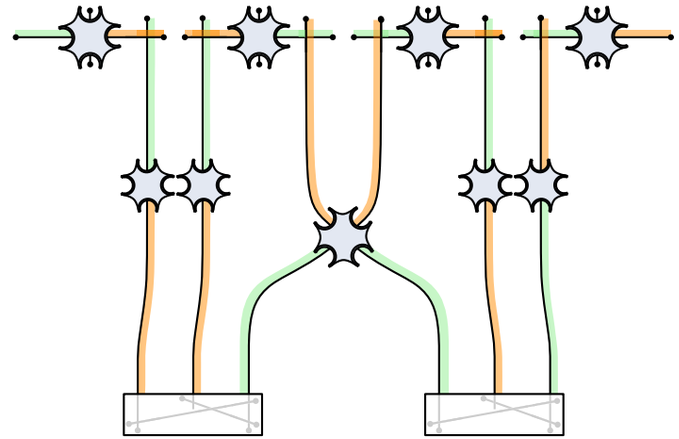
$$\varphi := \bigwedge_{i=1}^{m} \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$$

$$\mathsf{NAE}(x, y, z) := x \neq y \vee y \neq z \vee z \neq x$$

- Proof idea: $\varphi = \mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$



$\Leftrightarrow$



MinMaxSwitches is NP-complete

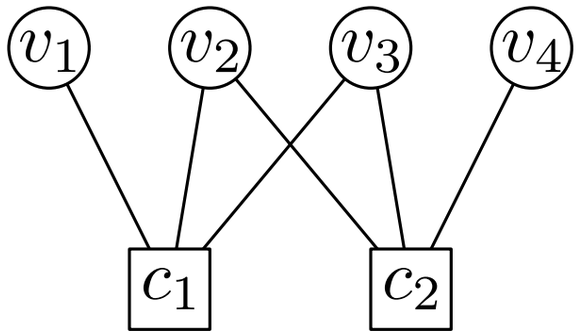
- Reduction from NAE 3SAT

Not-all-equal 3SAT (NAE 3SAT)

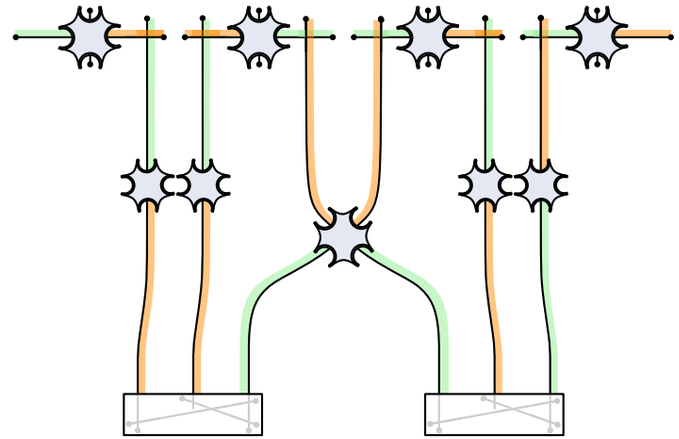
$$\varphi := \bigwedge_{i=1}^{m} \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$$

$$\mathsf{NAE}(x, y, z) := x \neq y \vee y \neq z \vee z \neq x$$

- Proof idea: $\varphi = \mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$

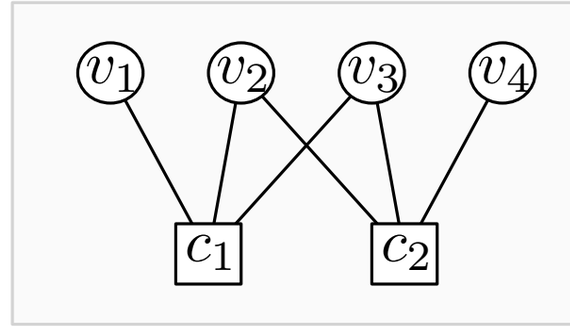


$\Leftrightarrow$

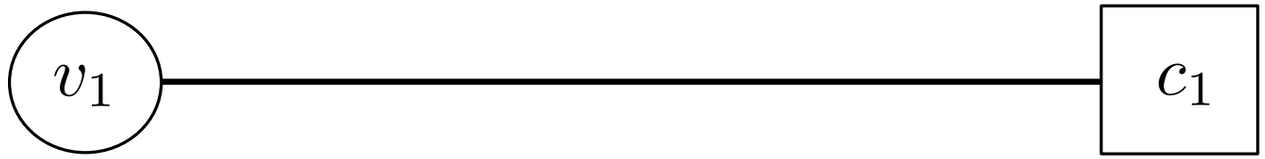


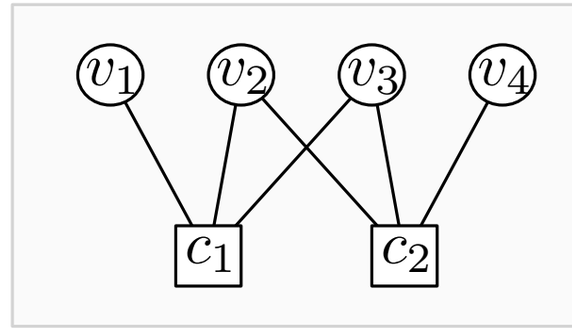
φ sat. $\Leftrightarrow$ δ_G embeddable with 1 switch per edge

Proof overview

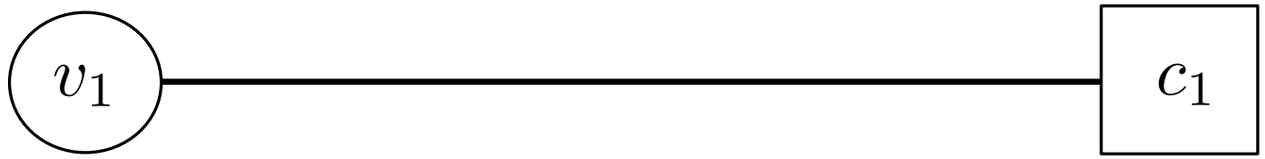


Proof overview

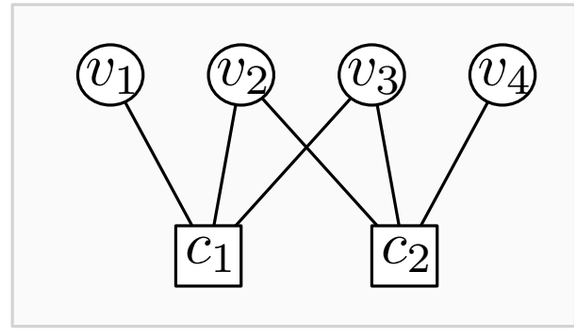




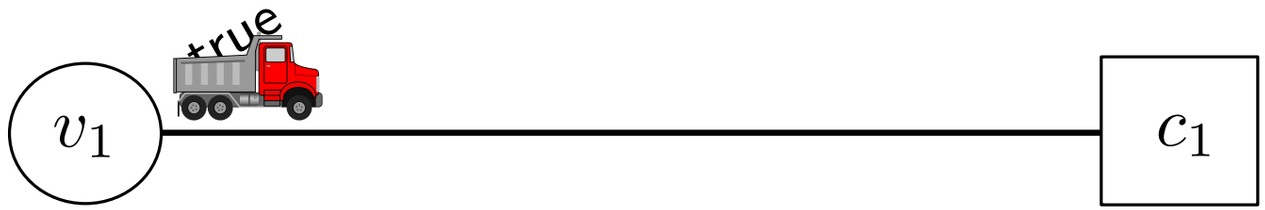
Proof overview



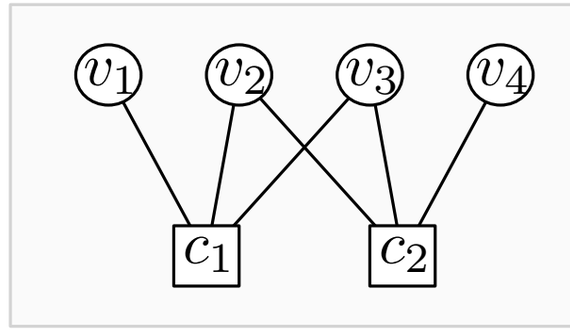
$v_1 \in \{\text{true}, \text{false}\}$



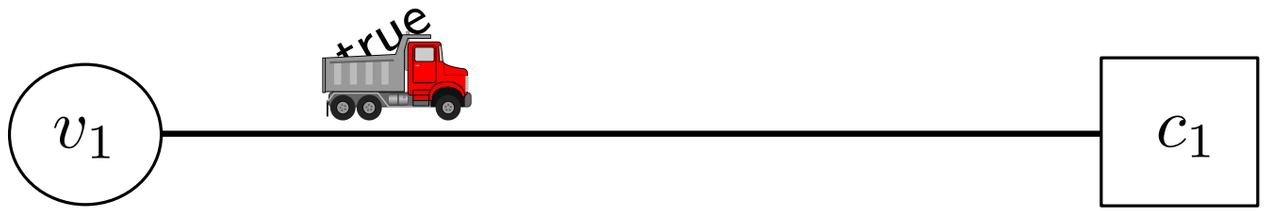
Proof overview



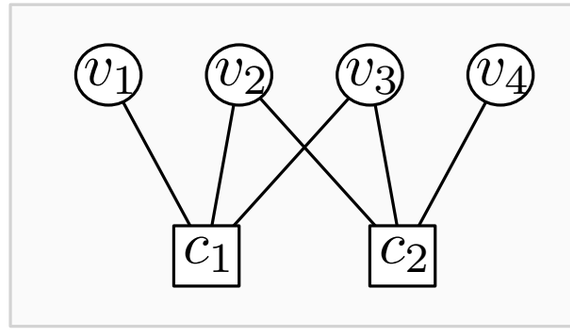
$v_1 \in \{\text{true}, \text{false}\}$



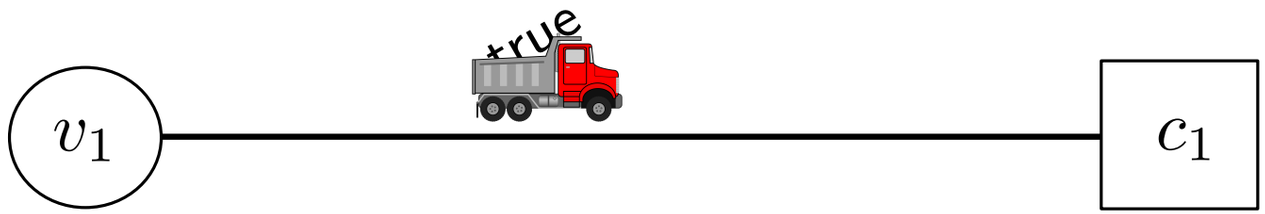
Proof overview



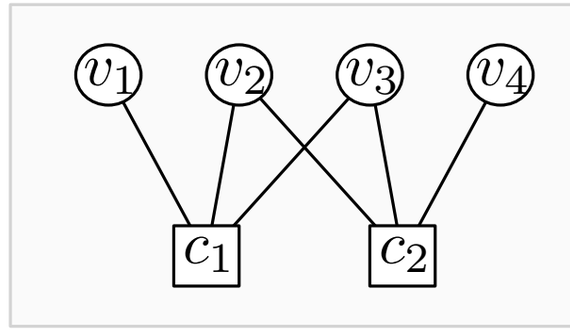
$v_1 \in \{\text{true}, \text{false}\}$



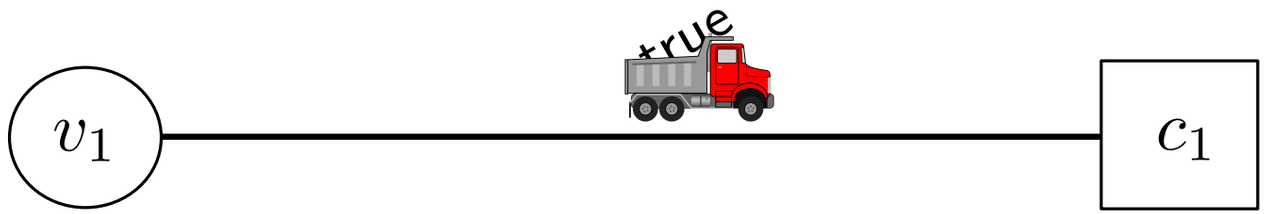
Proof overview



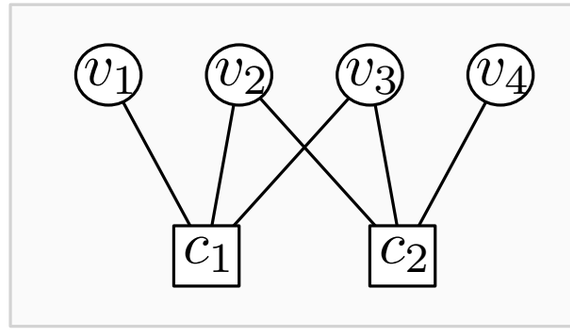
$v_1 \in \{\text{true}, \text{false}\}$



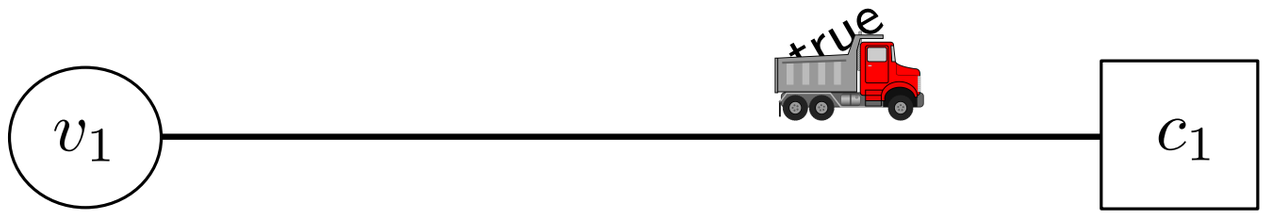
Proof overview



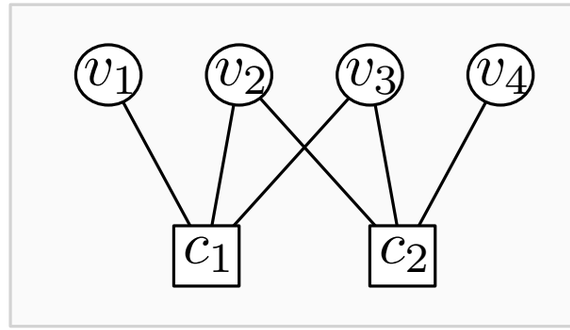
$v_1 \in \{\text{true}, \text{false}\}$



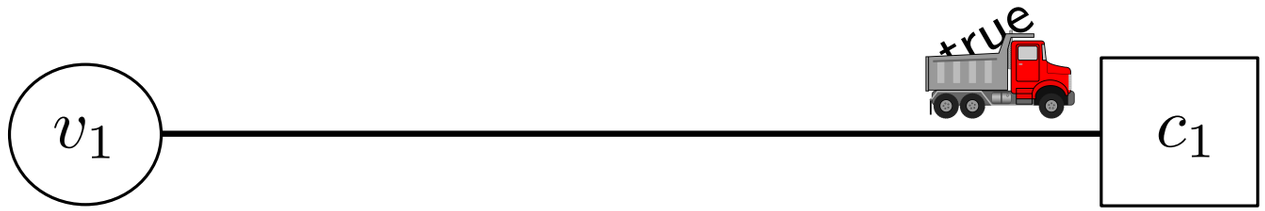
Proof overview



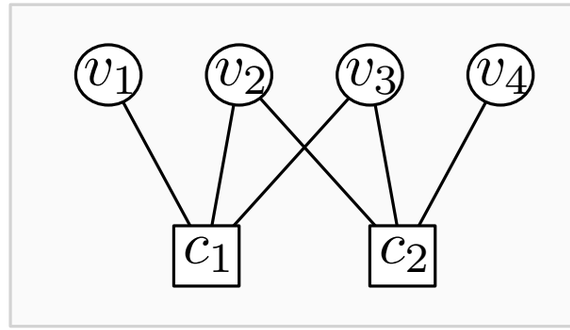
$v_1 \in \{\text{true}, \text{false}\}$



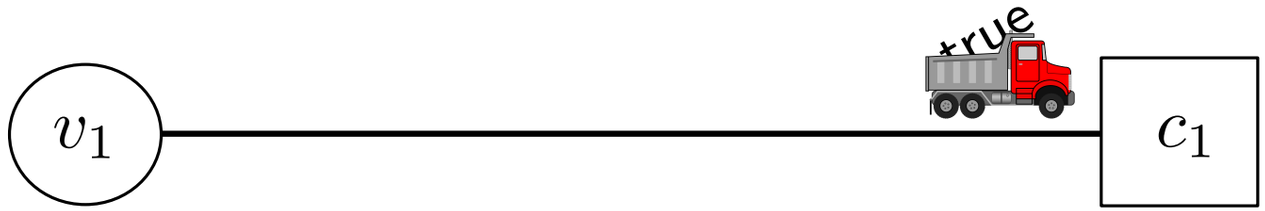
Proof overview



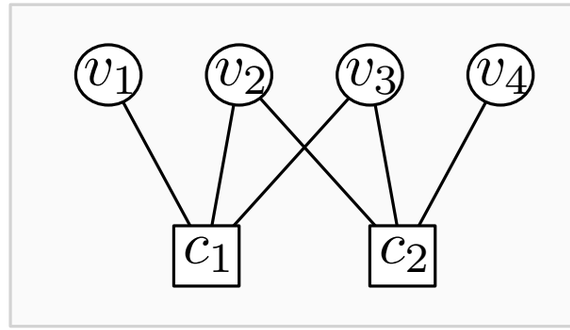
$v_1 \in \{\text{true}, \text{false}\}$

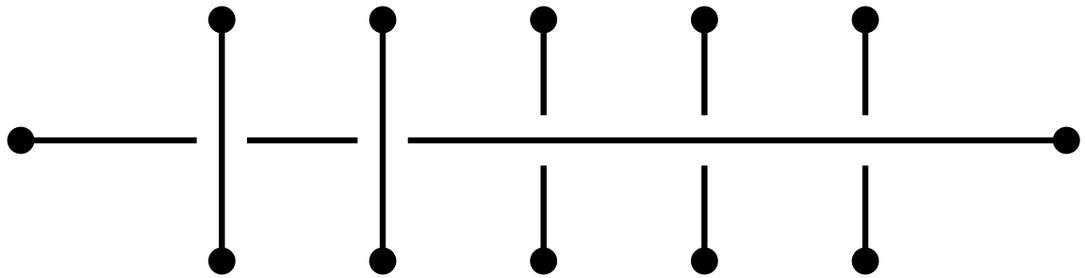


Proof overview

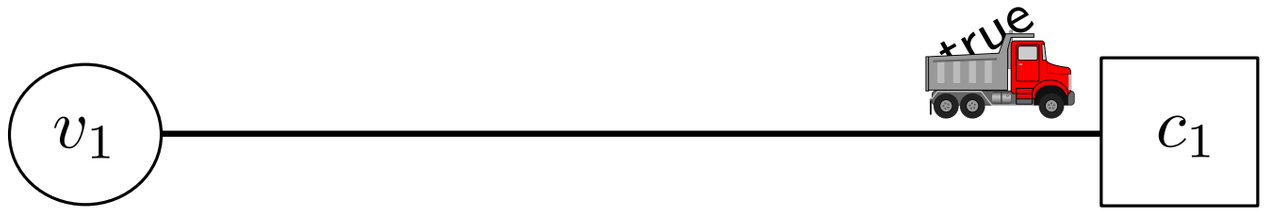


$v_1 \in \{\text{true}, \text{false}\}$

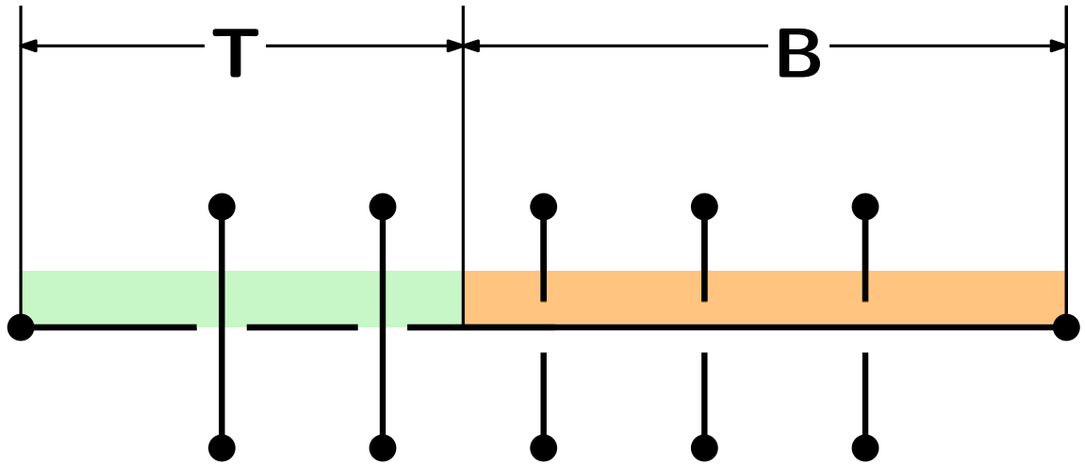


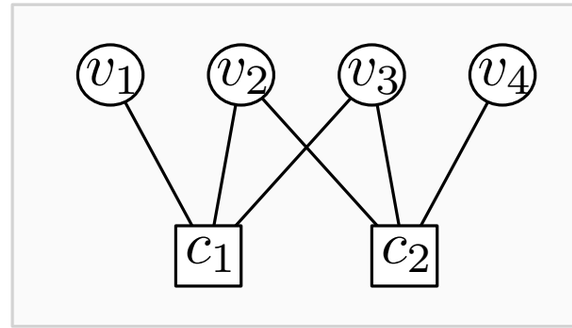


Proof overview



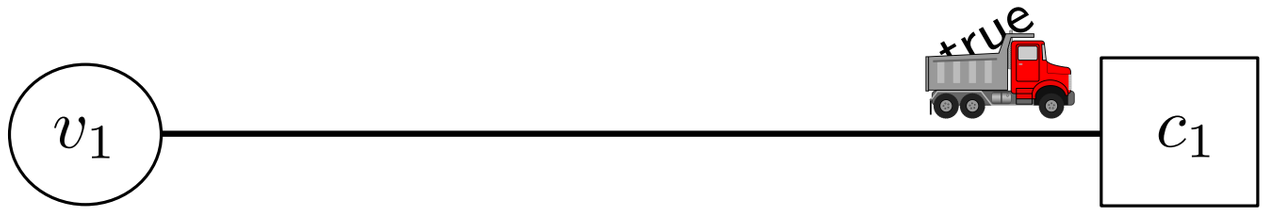
$v_1 \in \{\text{true}, \text{false}\}$



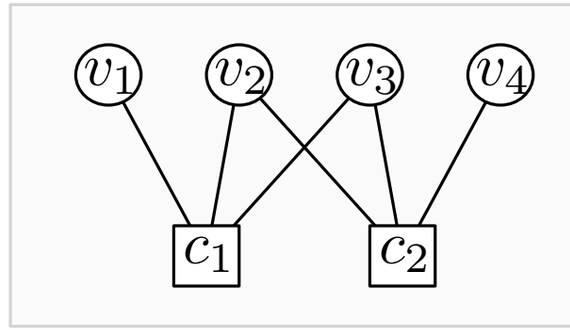


T . . . tunnel
B . . . bridge

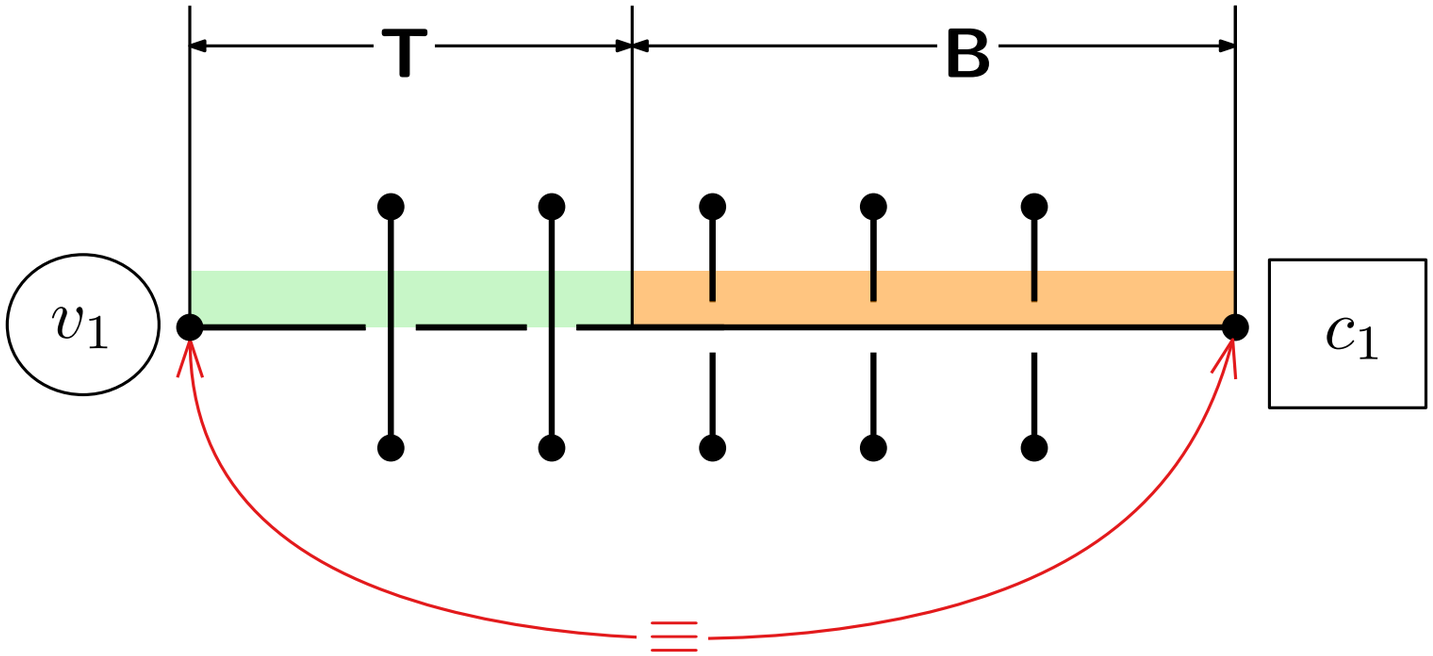
Proof overview



$v_1 \in \{\text{true}, \text{false}\}$

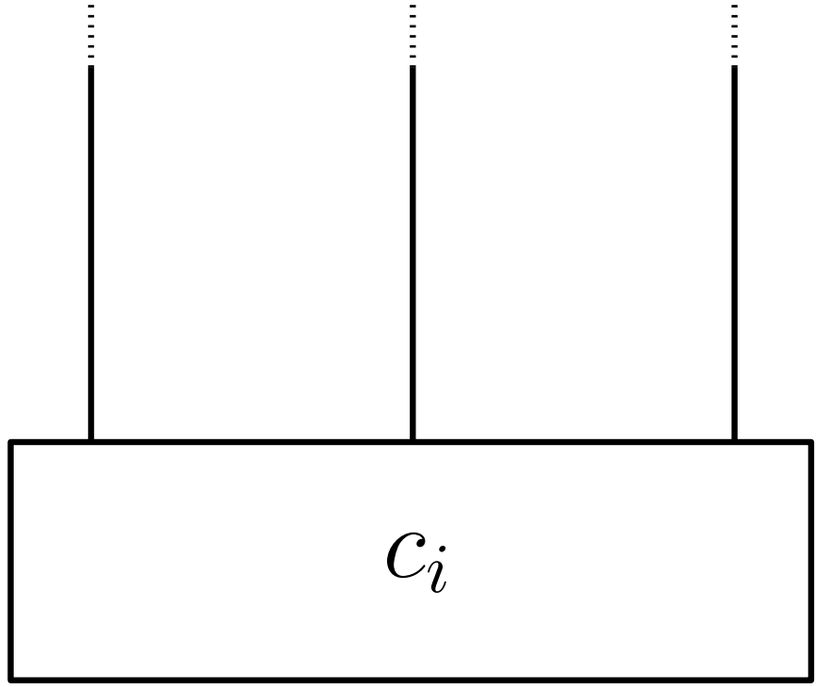


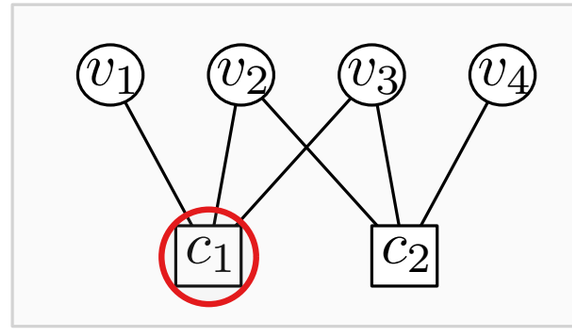
T . . . tunnel
B . . . bridge



Clause gadget

- $c_i = \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$

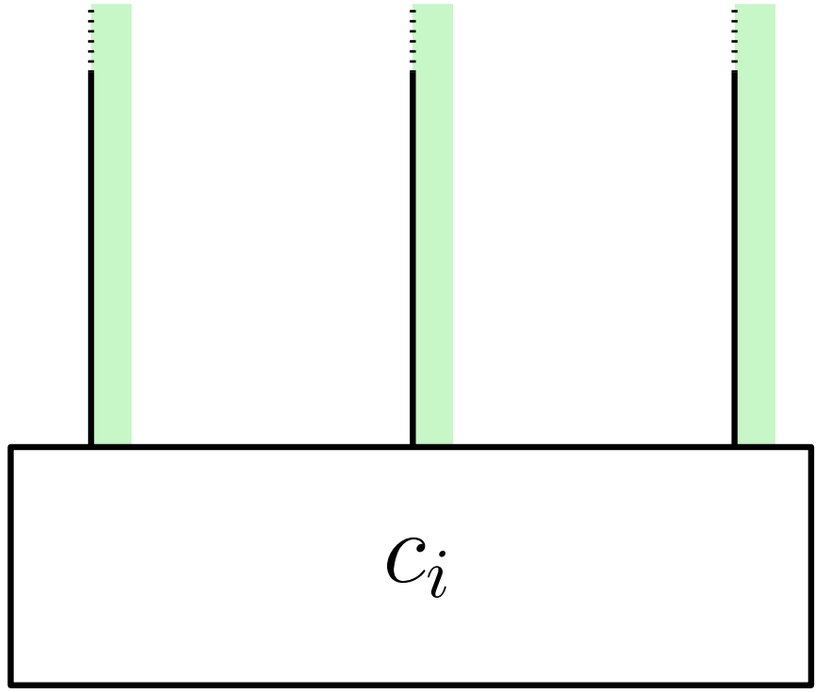


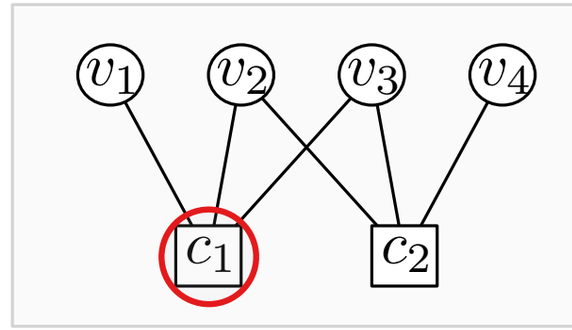


T . . . tunnel
B . . . bridge

Clause gadget

- $c_i = \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$



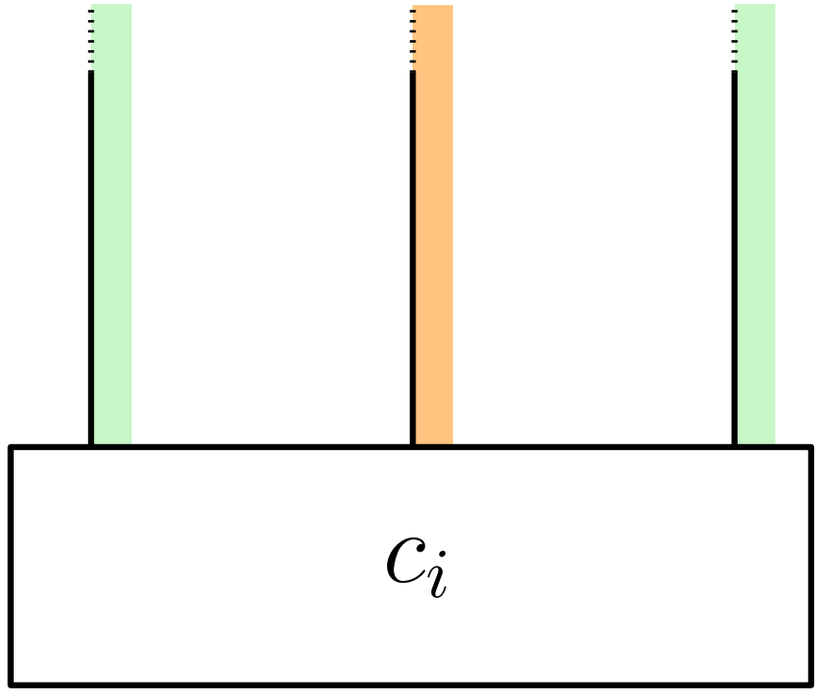


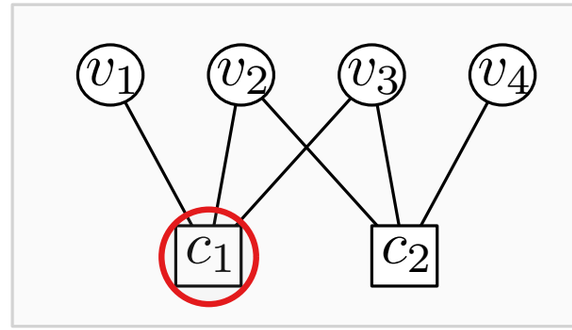
T . . . tunnel
B . . . bridge

- All equal: ≥ 2 switches necessary

Clause gadget

- $c_i = \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$



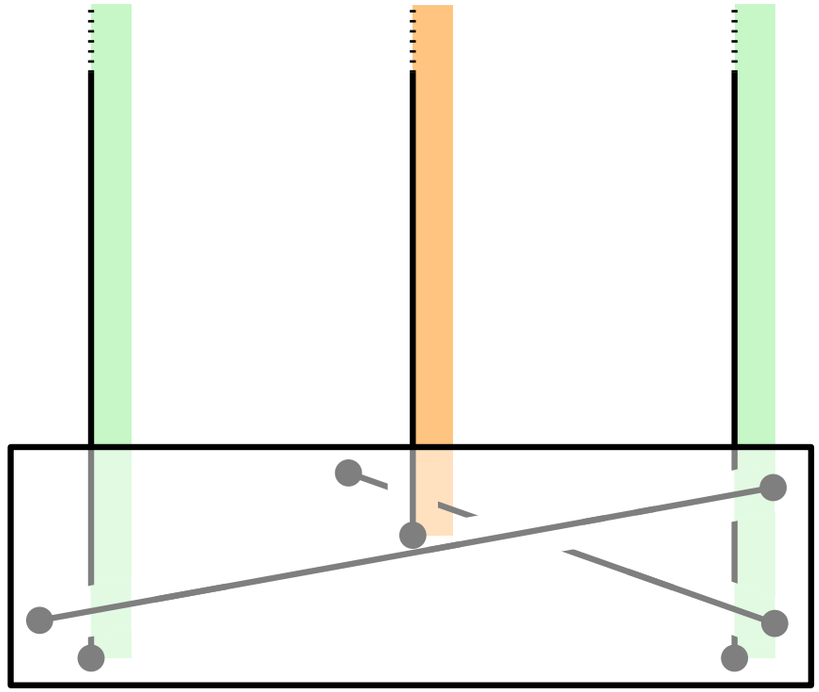


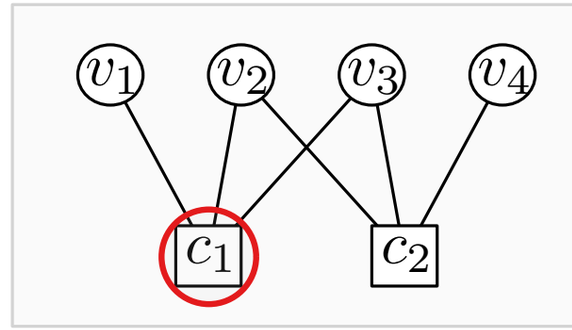
T . . . tunnel
B . . . bridge

- All equal: ≥ 2 switches necessary
- Not all equal: possible with 1 switch

Clause gadget

- $c_i = \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$



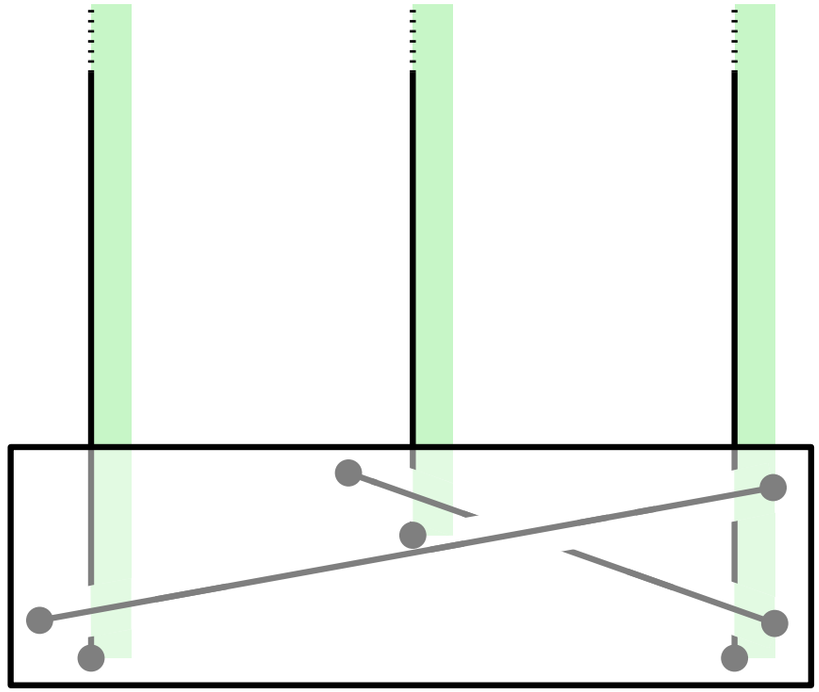


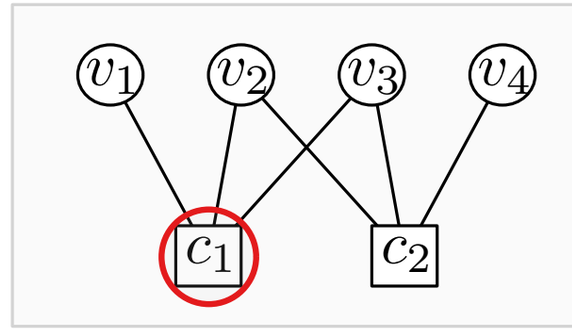
T . . . tunnel
B . . . bridge

- All equal: ≥ 2 switches necessary
- Not all equal: possible with 1 switch

Clause gadget

- $c_i = \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$



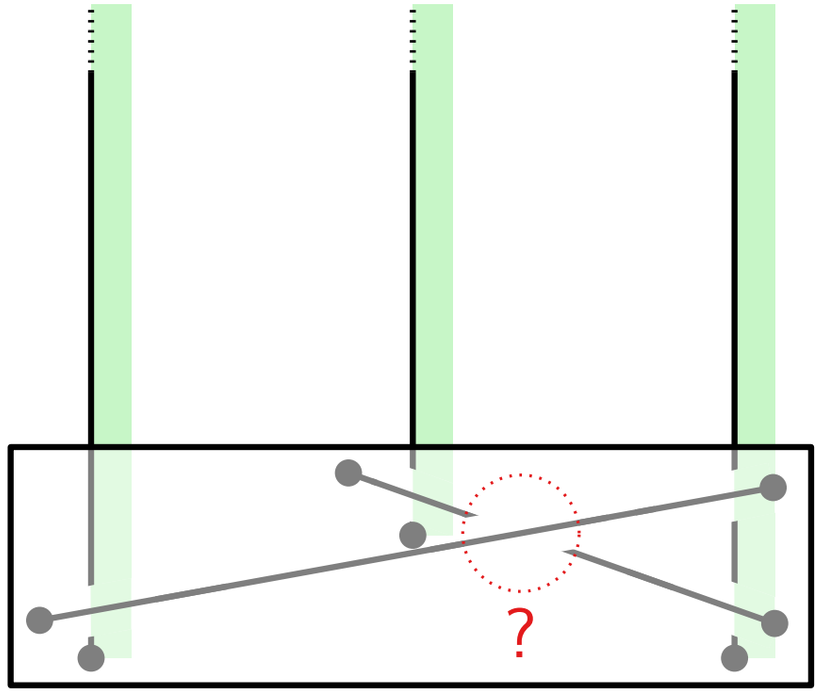


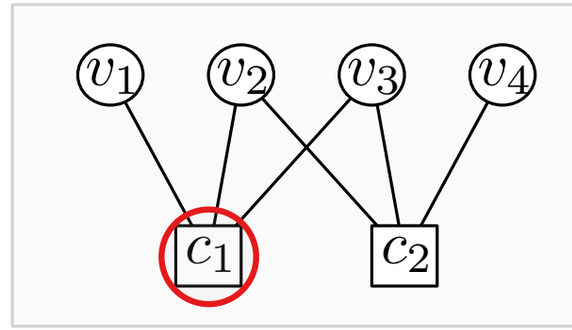
T . . . tunnel
B . . . bridge

- All equal: ≥ 2 switches necessary
- Not all equal: possible with 1 switch

Clause gadget

- $c_i = \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$



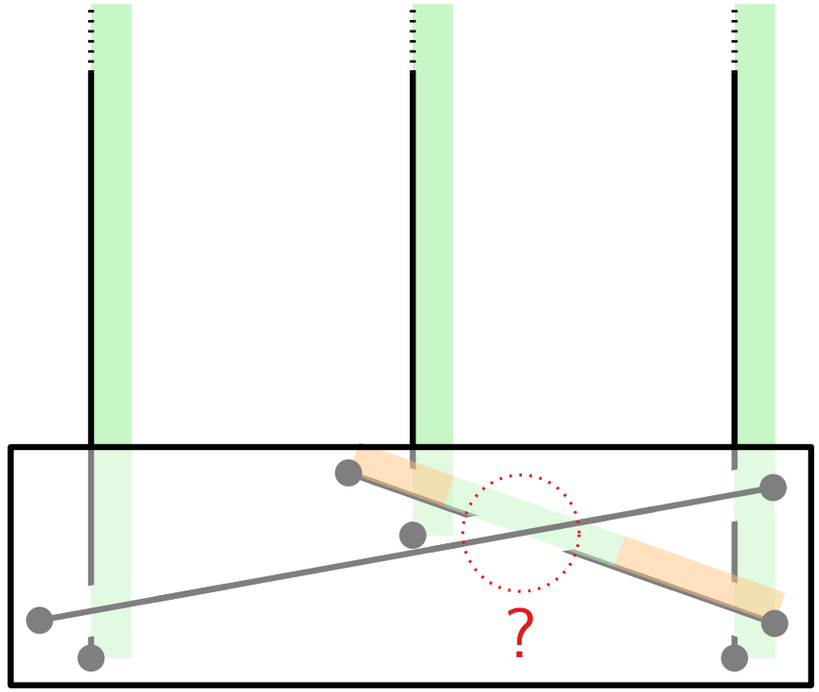


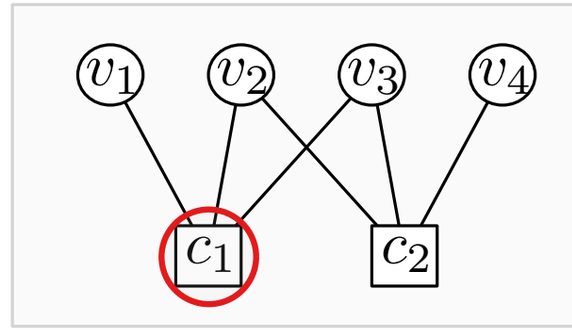
T . . . tunnel
B . . . bridge

- All equal: ≥ 2 switches necessary
- Not all equal: possible with 1 switch

Clause gadget

- $c_i = \mathsf{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$



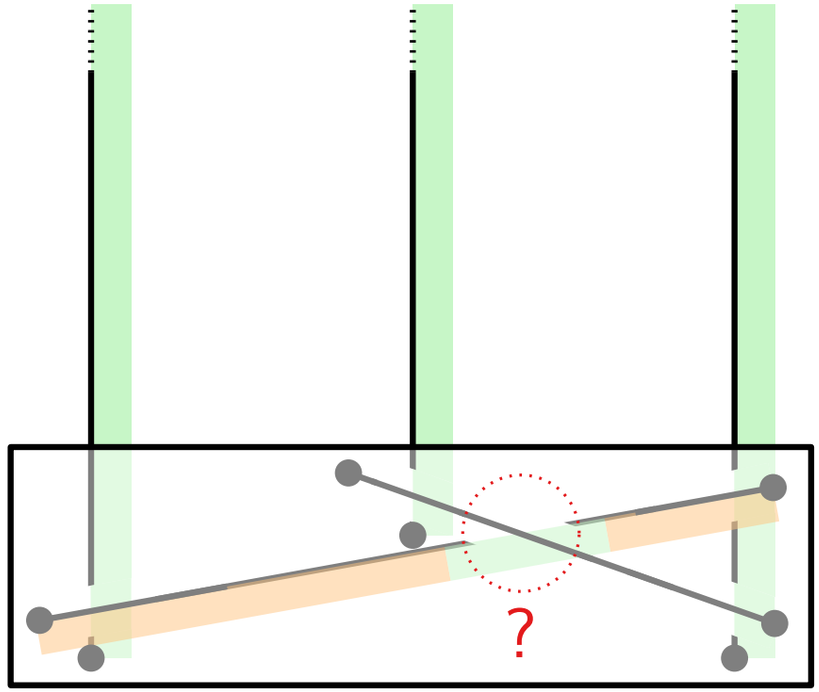


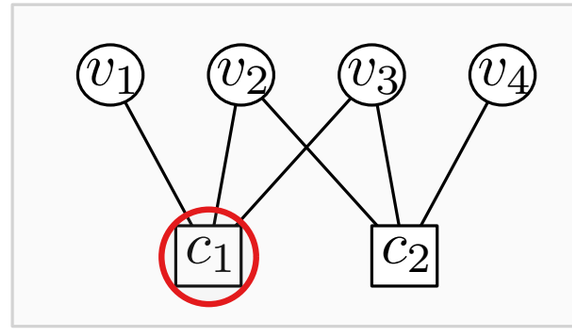
T . . . tunnel
B . . . bridge

- All equal: ≥ 2 switches necessary
- Not all equal: possible with 1 switch

Clause gadget

- $c_i = \text{NAE}(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$

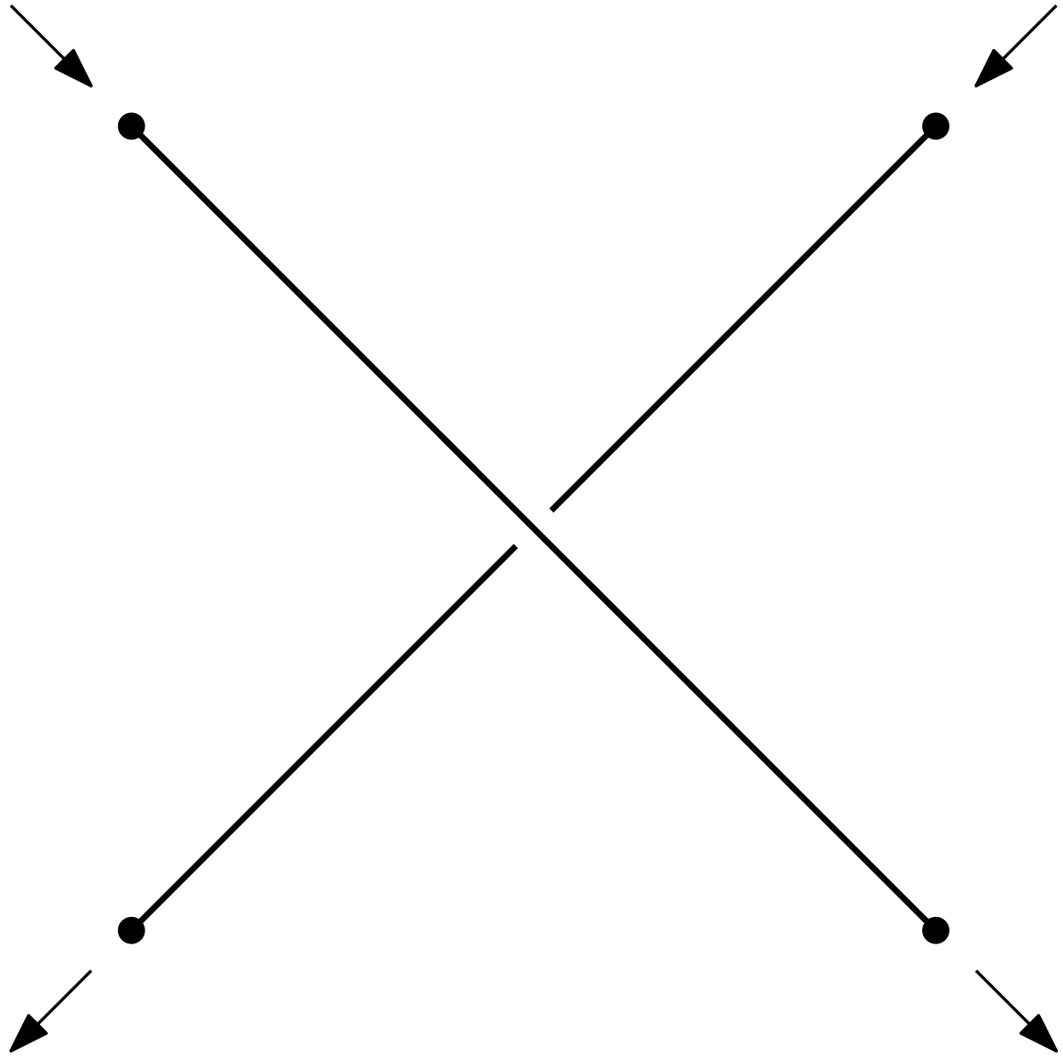


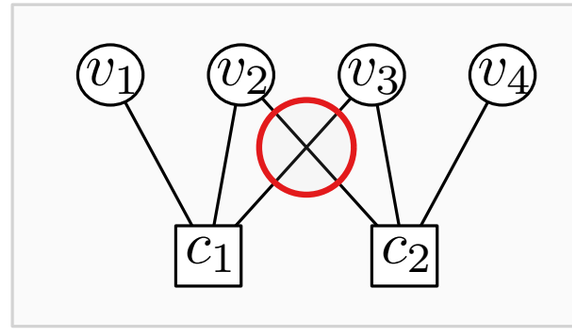


T . . . tunnel
B . . . bridge

- All equal: ≥ 2 switches necessary
- Not all equal: possible with 1 switch

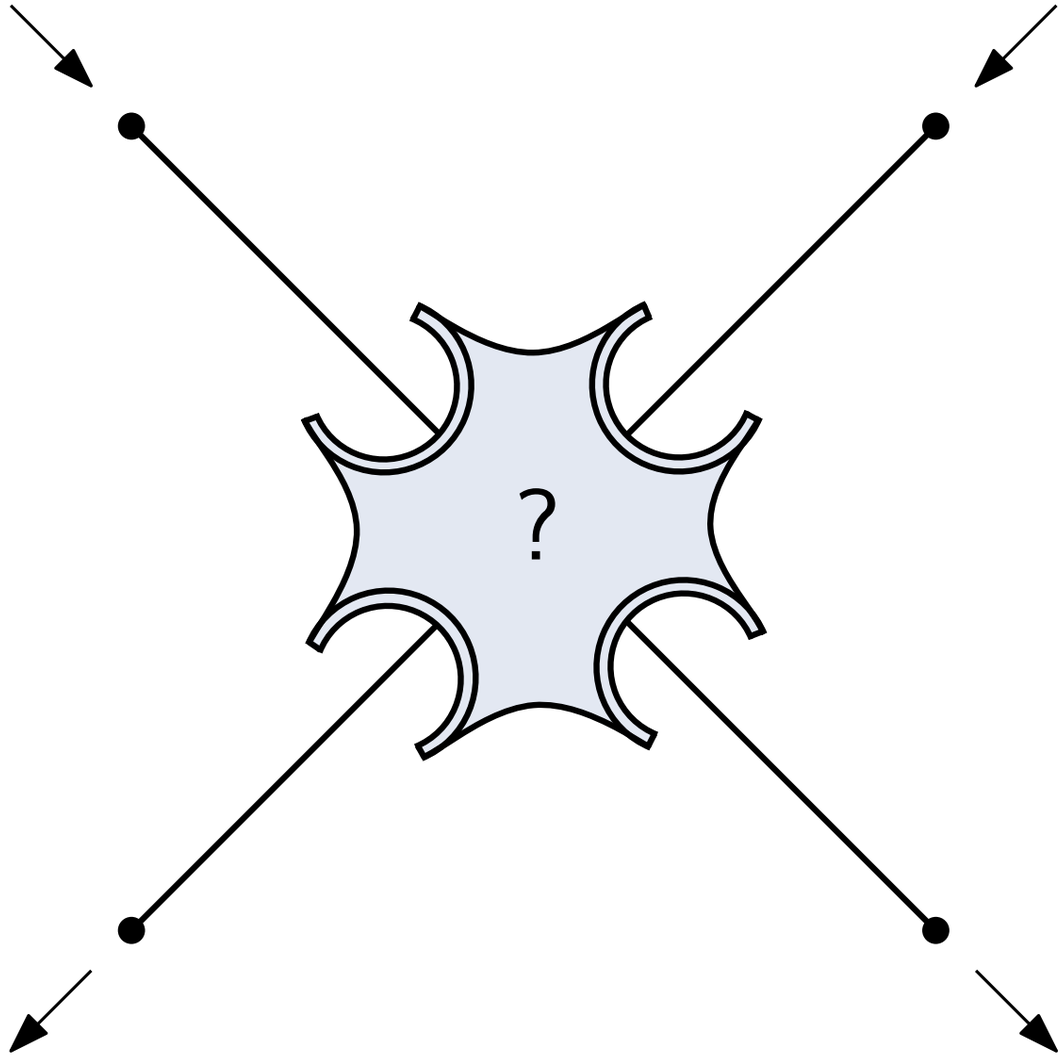
Crossing gadget

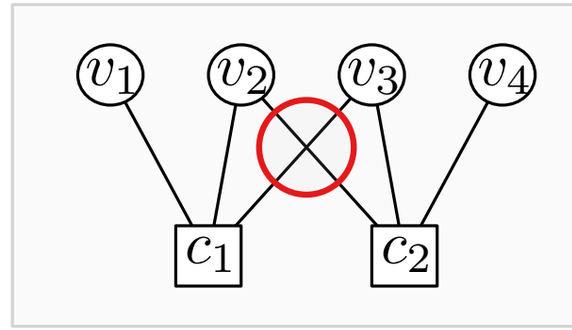




T . . . tunnel
B . . . bridge

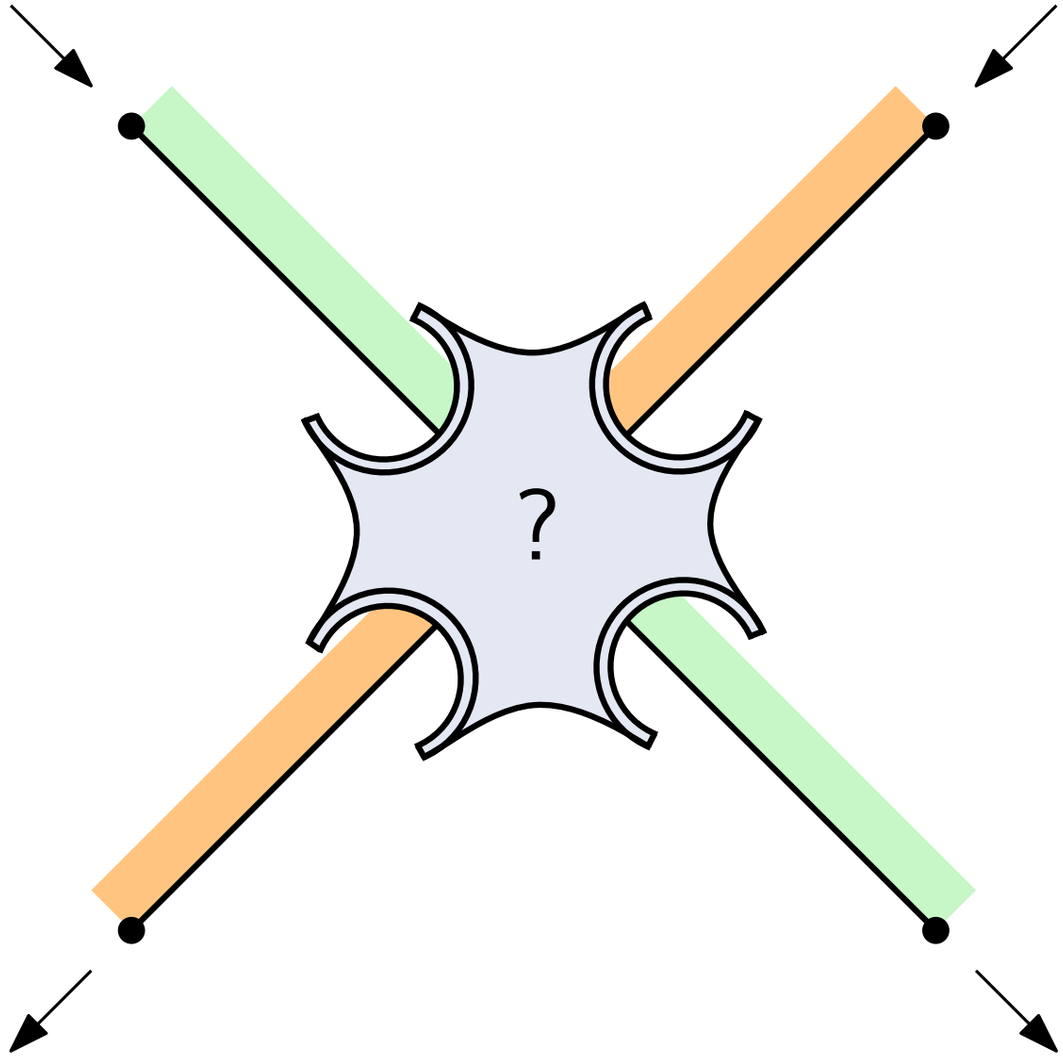
Crossing gadget

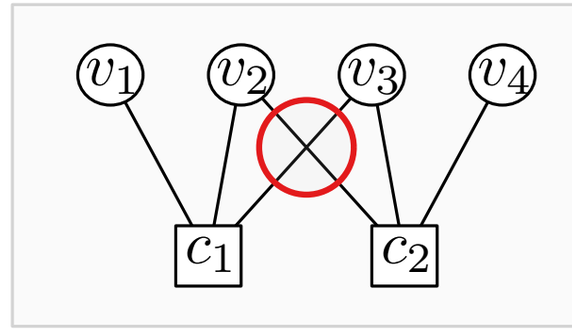




T . . . tunnel
B . . . bridge

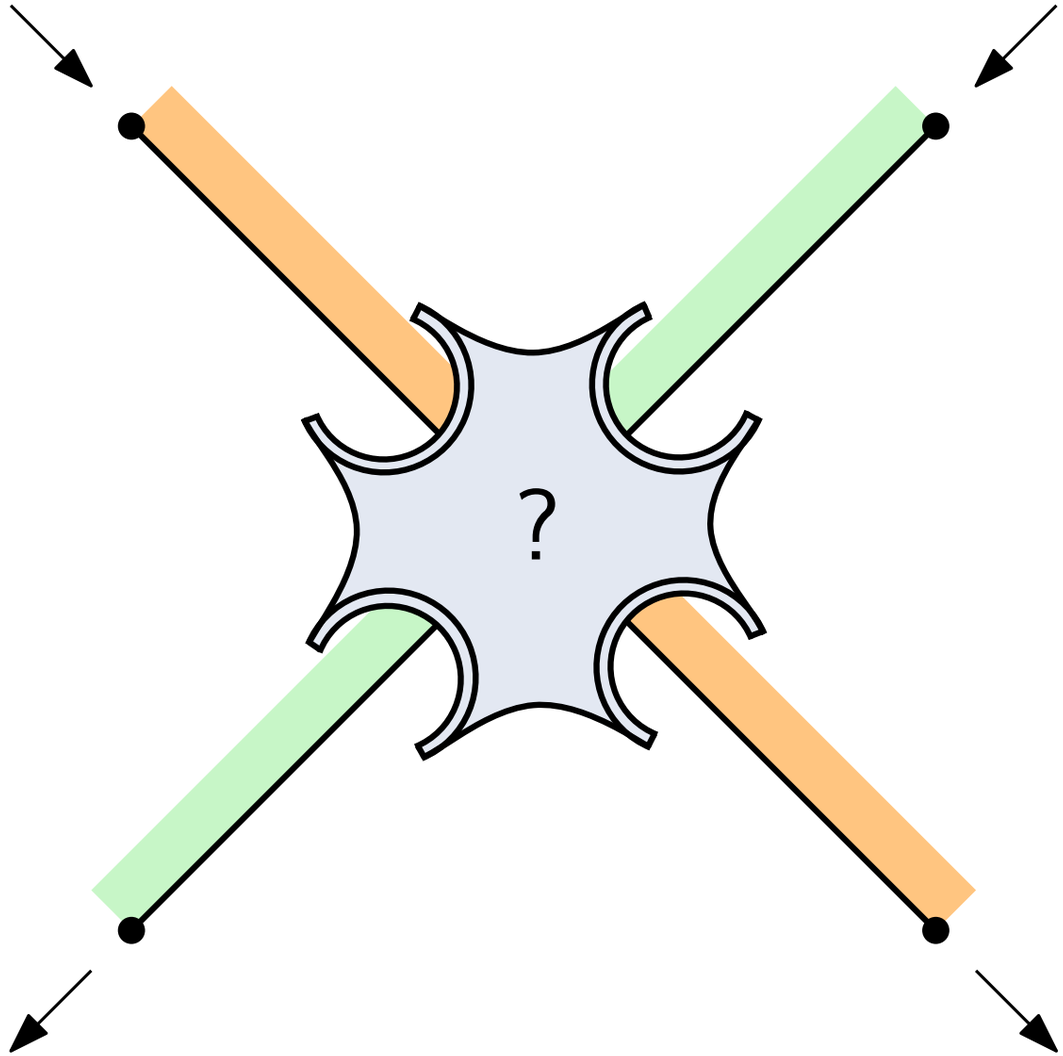
Crossing gadget

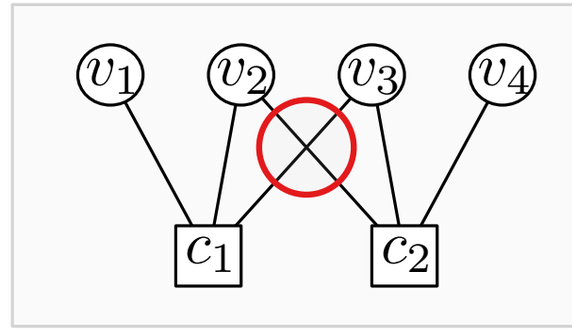




T . . . tunnel
B . . . bridge

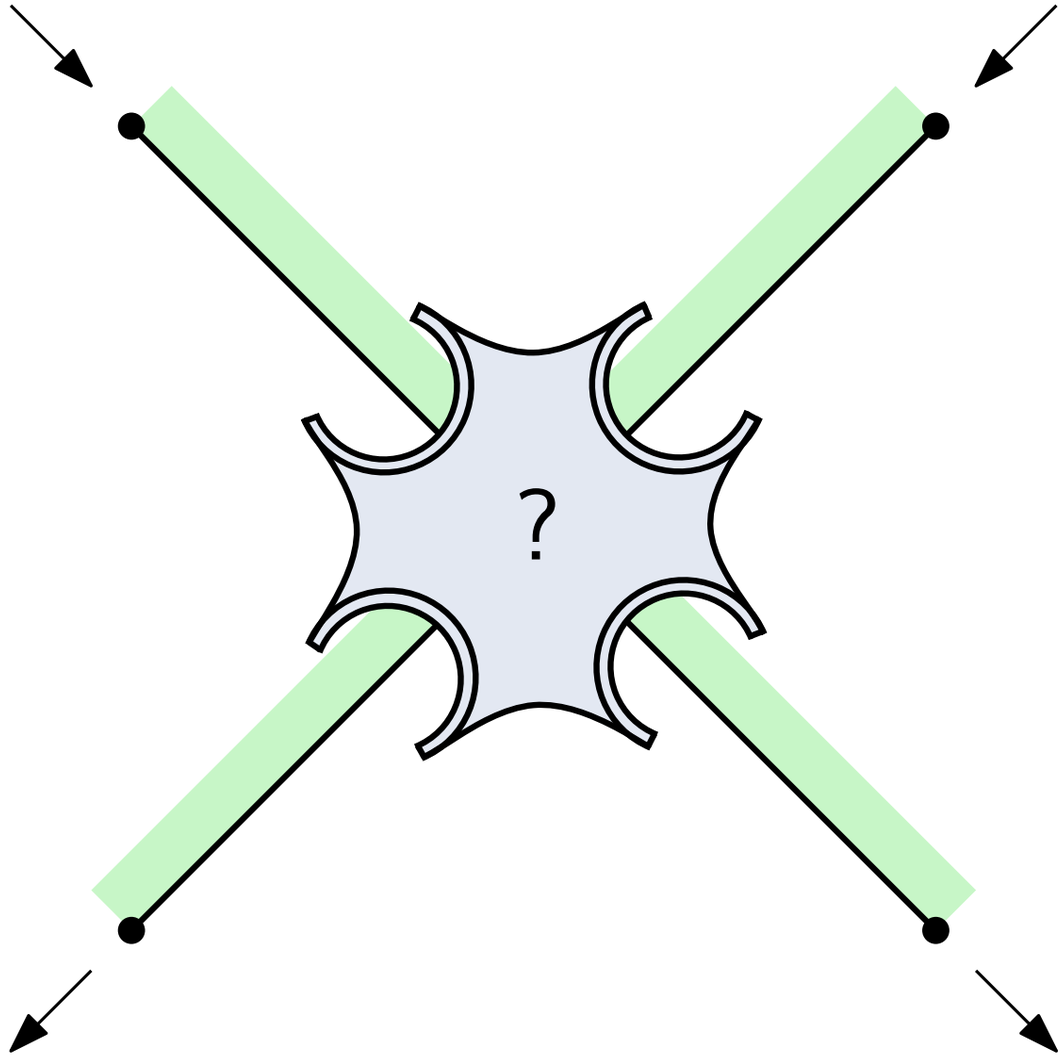
Crossing gadget

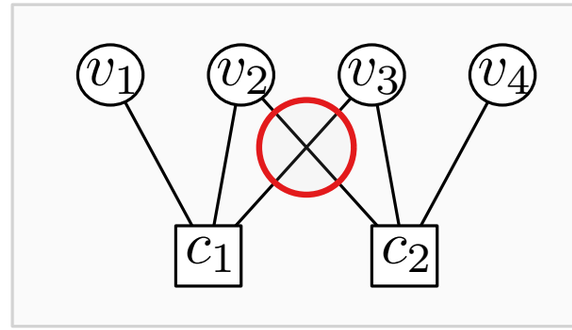




T . . . tunnel
B . . . bridge

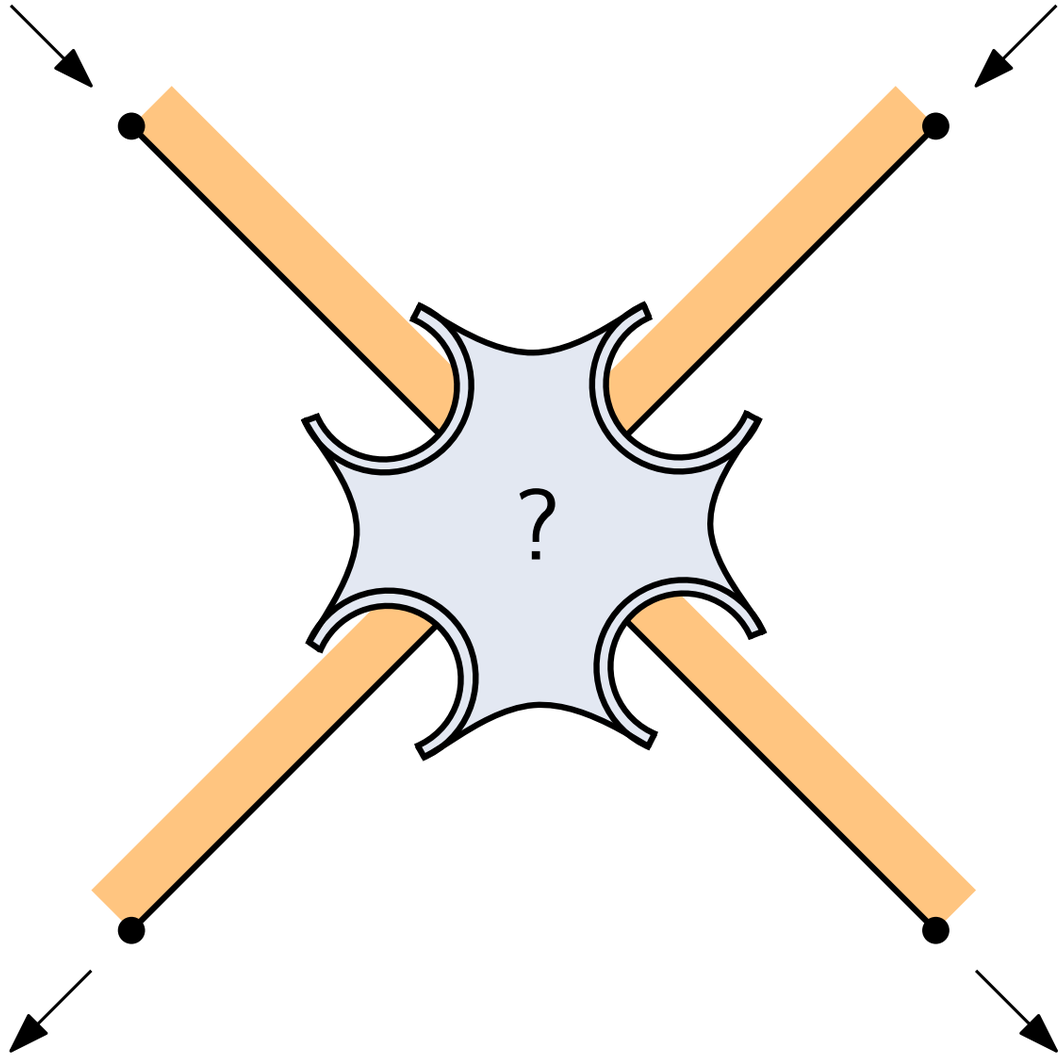
Crossing gadget

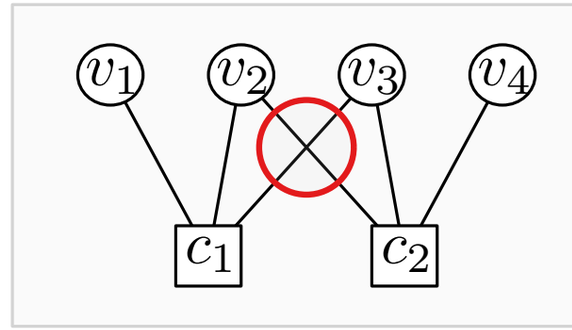




T . . . tunnel
B . . . bridge

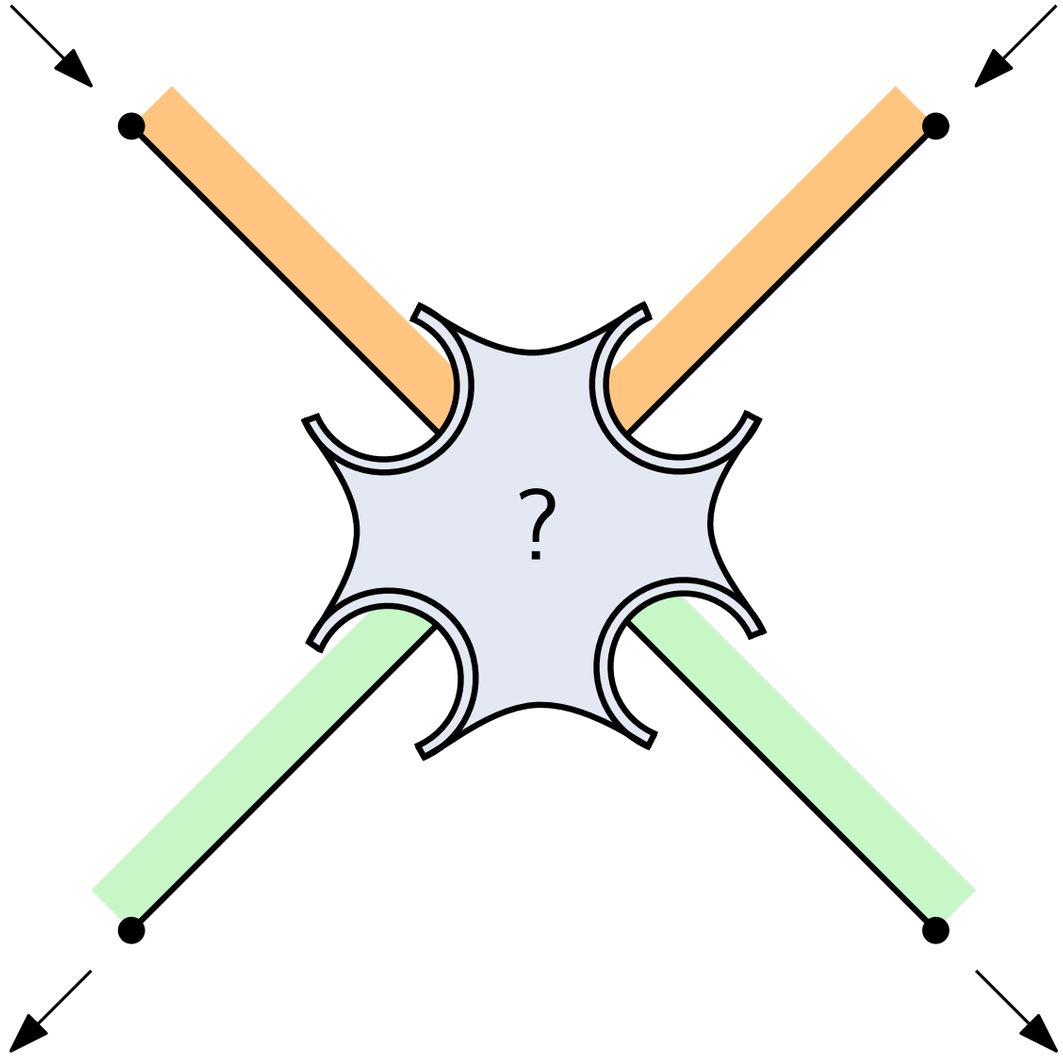
Crossing gadget

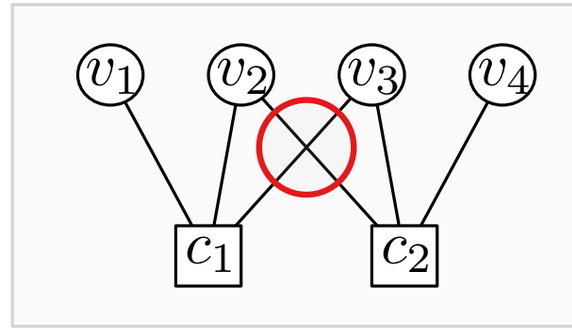




T . . . tunnel
B . . . bridge

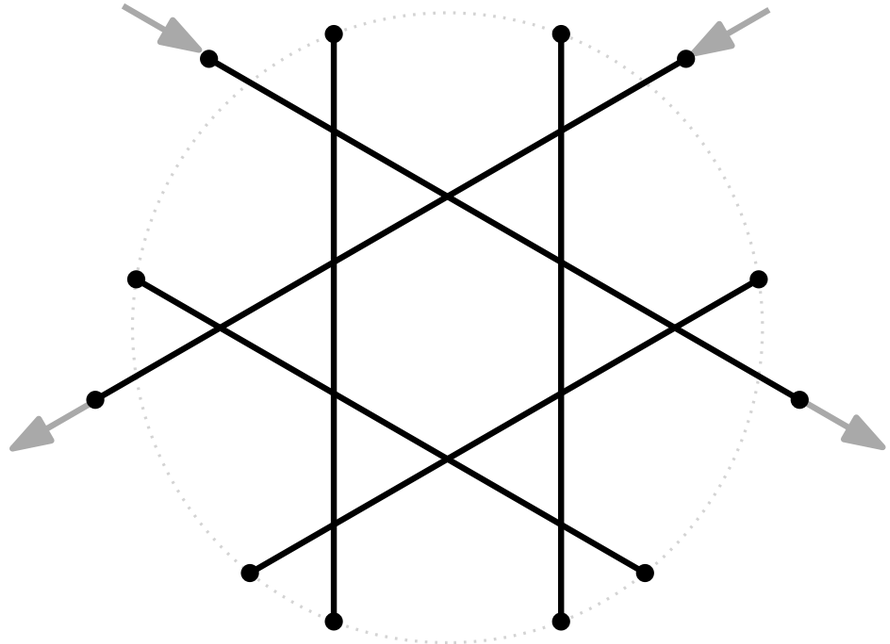
Crossing gadget

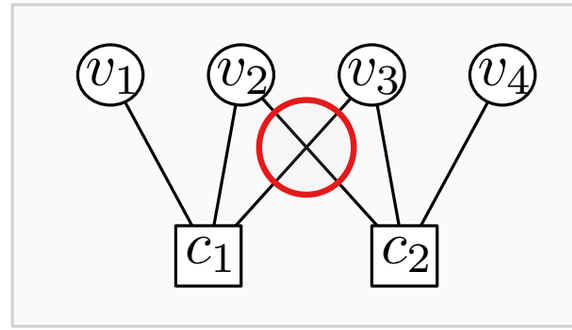




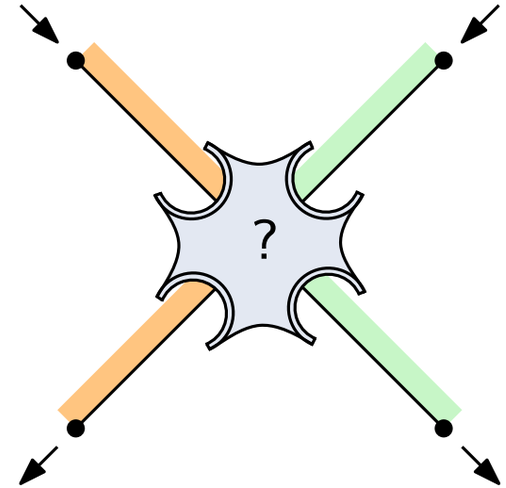
T . . . tunnel
B . . . bridge

Crossing gadget

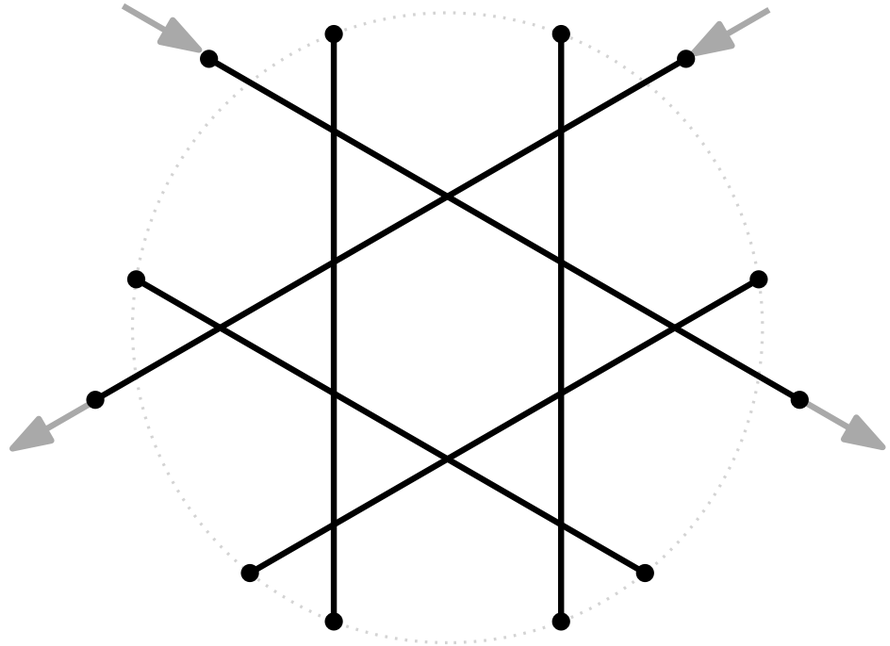


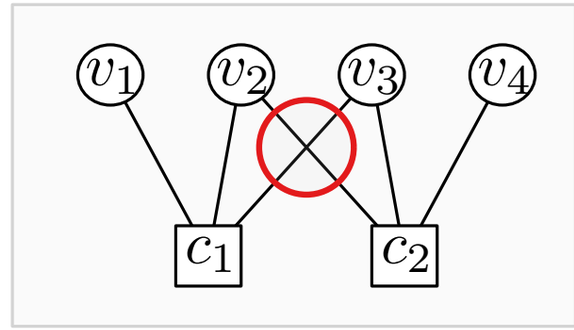


T . . . tunnel
B . . . bridge

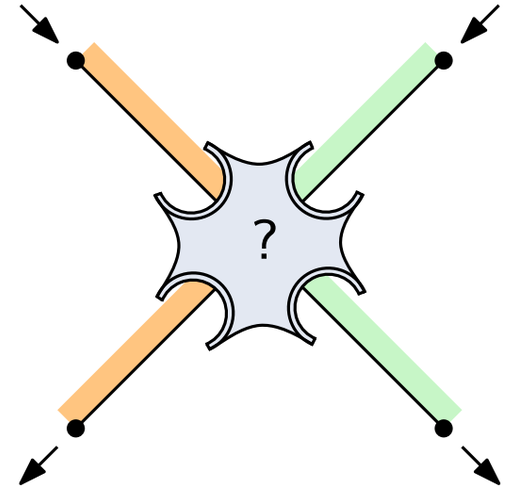


Crossing gadget

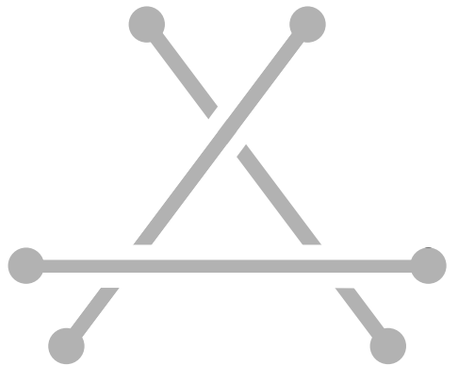




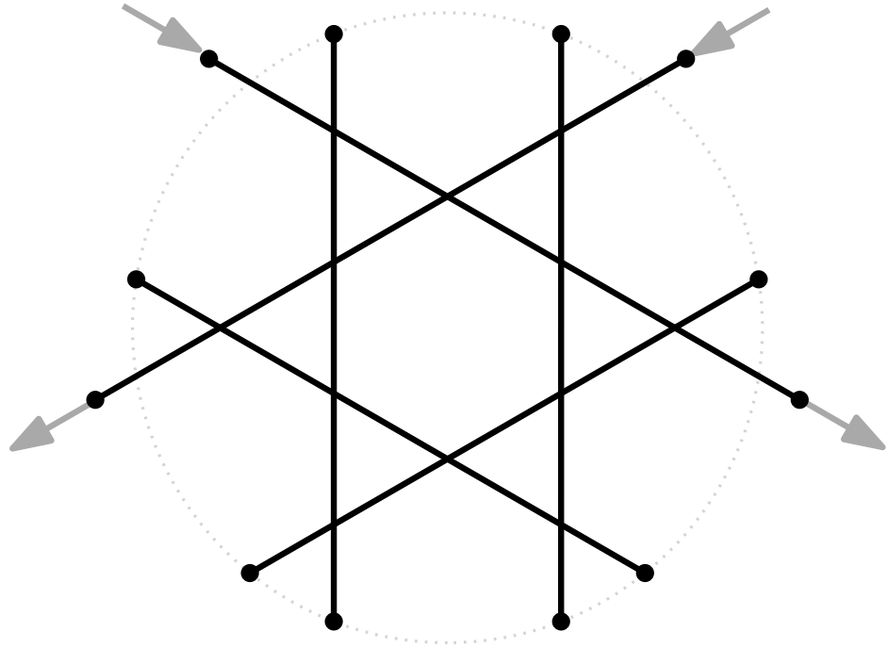
T . . . tunnel
B . . . bridge

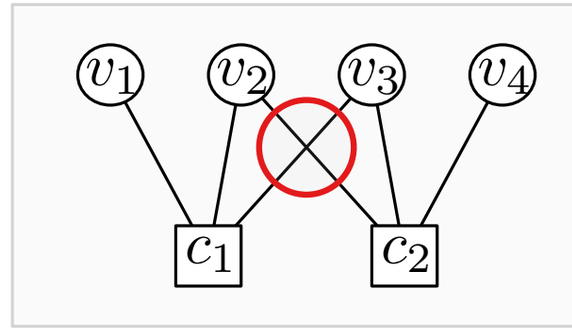


- **Obs. 1** Triangles induce at least 1 switch

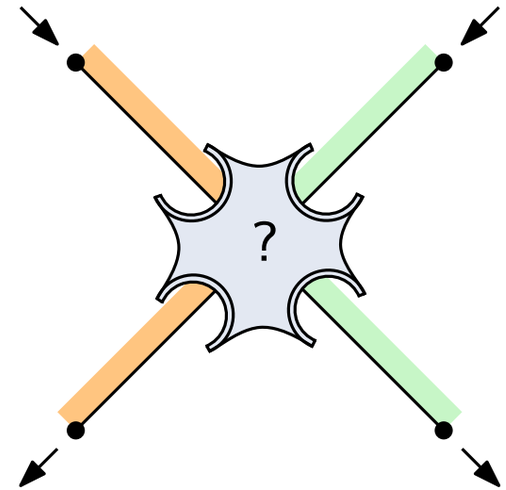


Crossing gadget

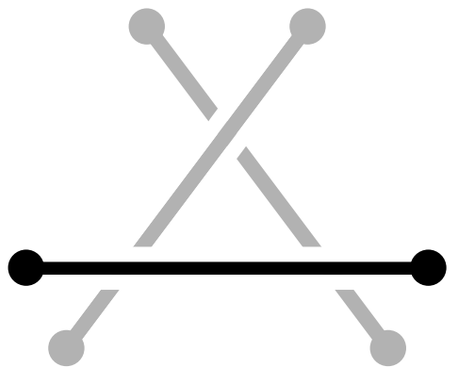




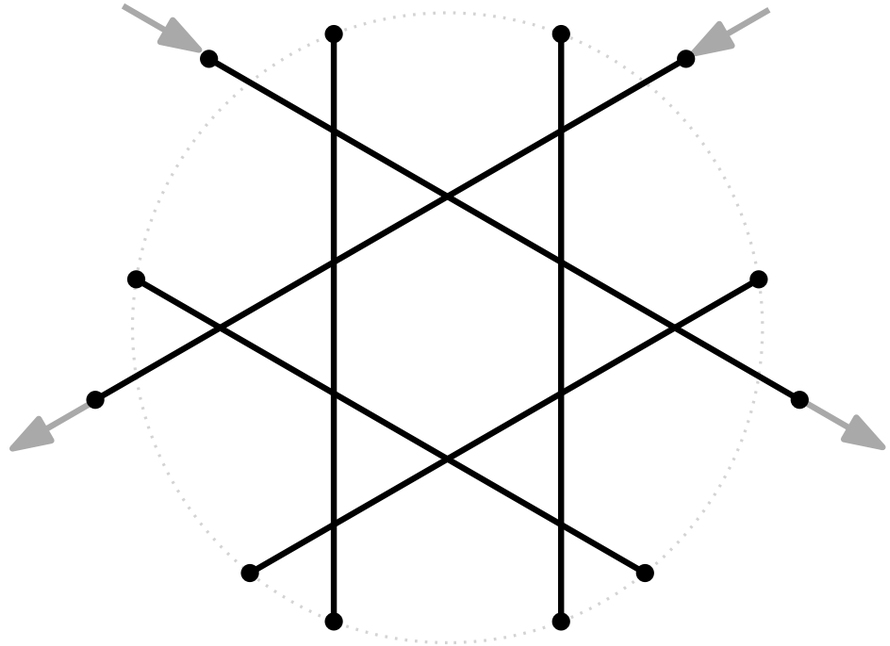
T . . . tunnel
B . . . bridge

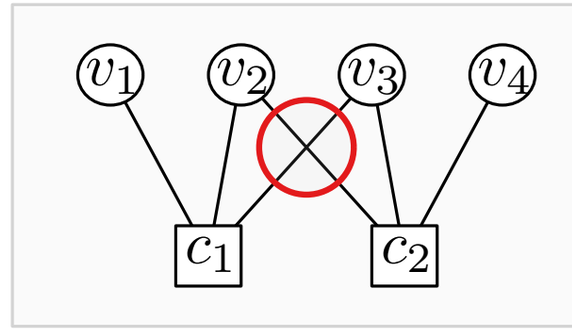


- **Obs. 1** Triangles induce at least 1 switch

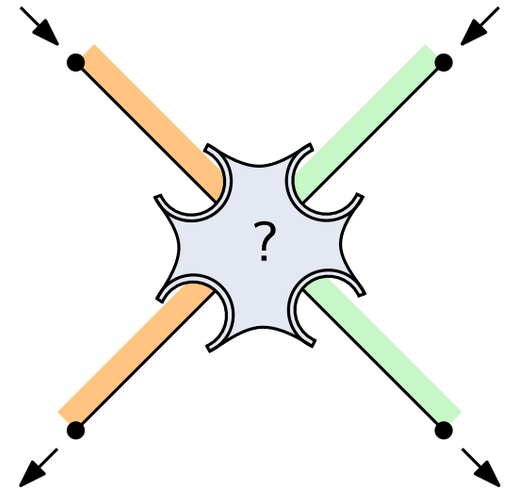


Crossing gadget

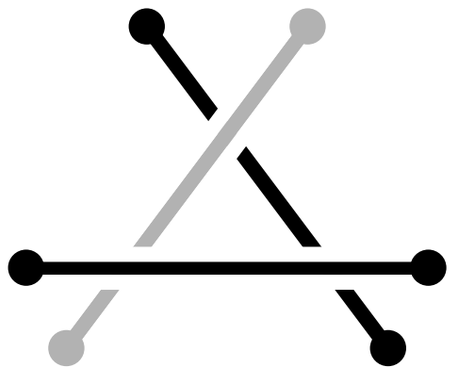




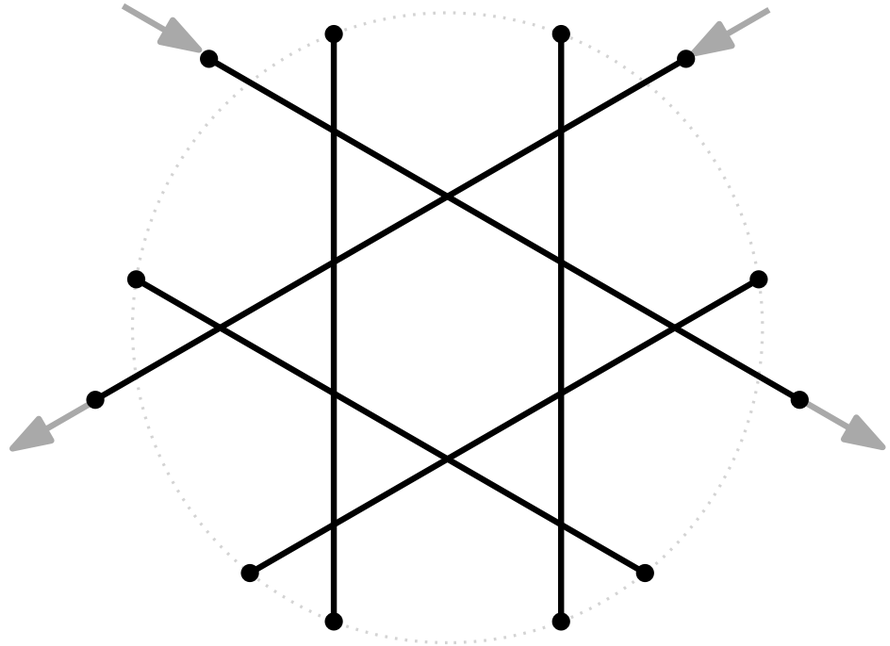
T . . . tunnel
B . . . bridge

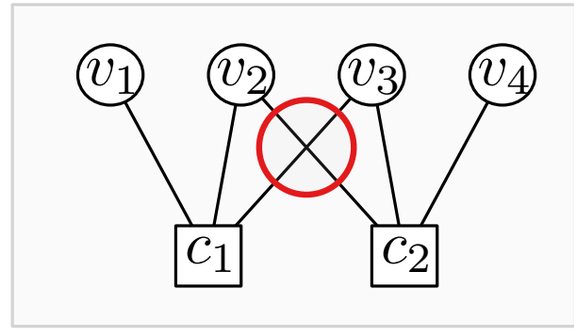


- **Obs. 1** Triangles induce at least 1 switch

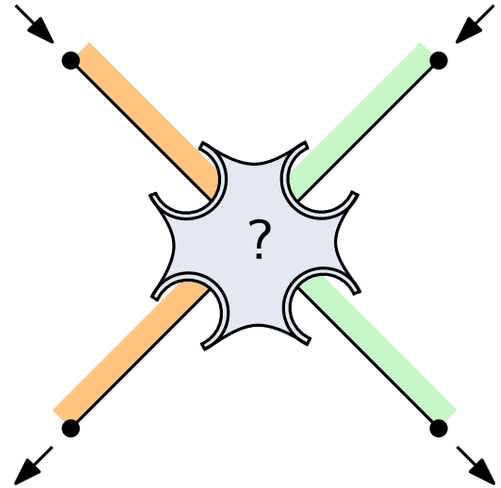


Crossing gadget

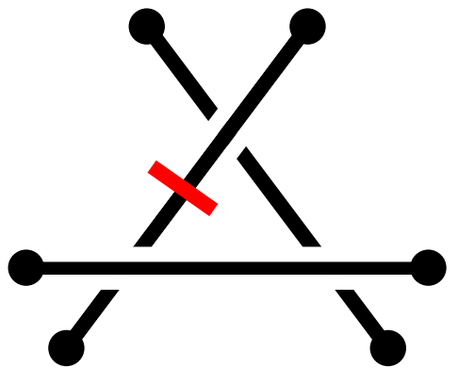




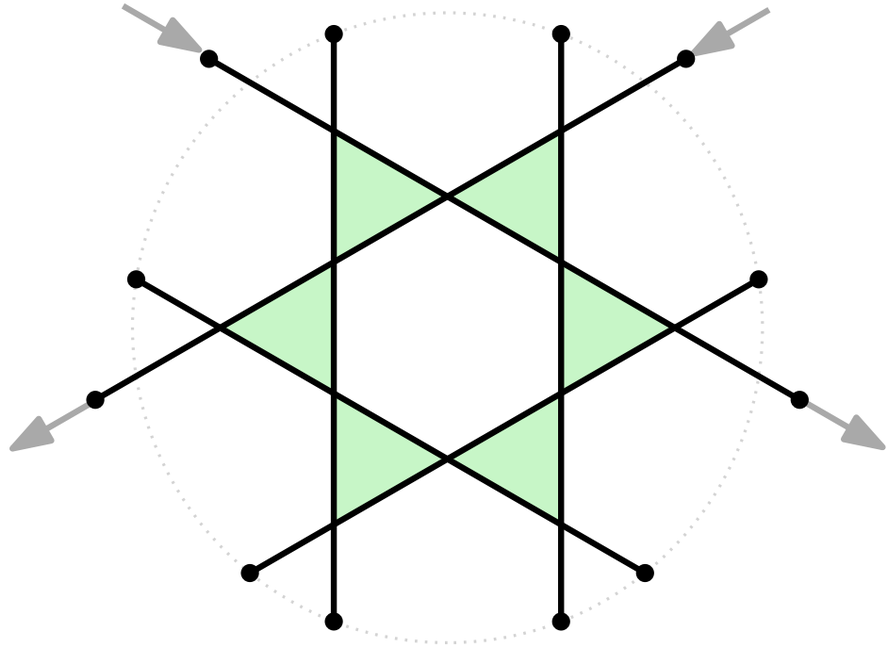
T . . . tunnel
B . . . bridge

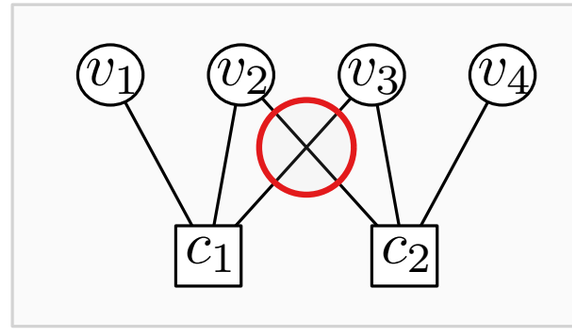


- **Obs. 1** Triangles induce at least 1 switch

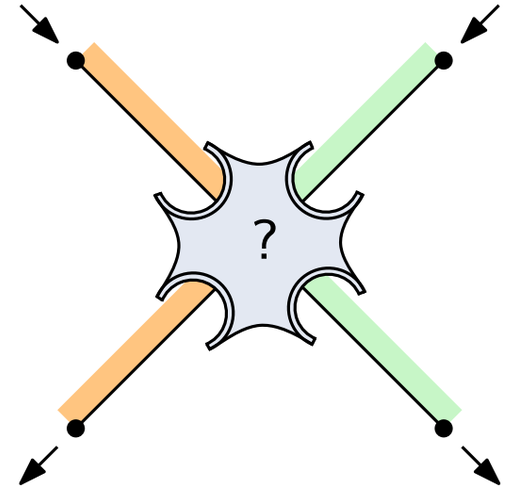


Crossing gadget



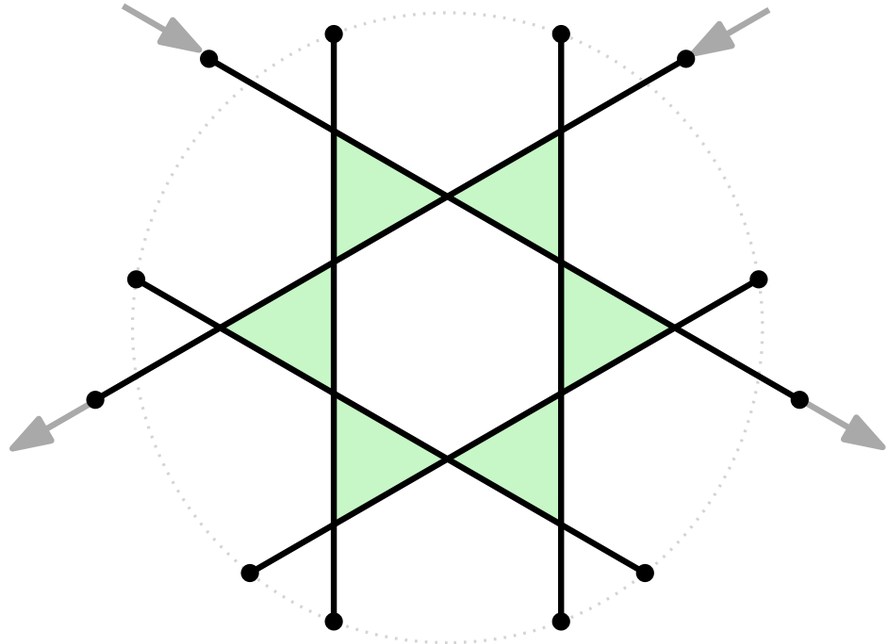


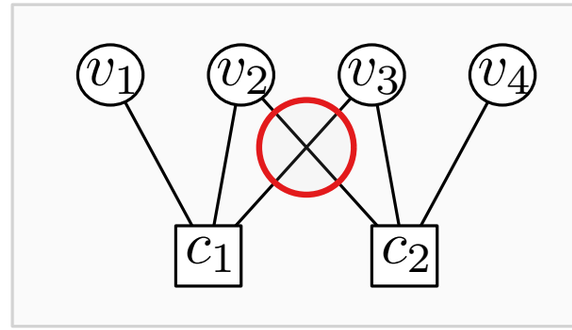
T . . . tunnel
B . . . bridge



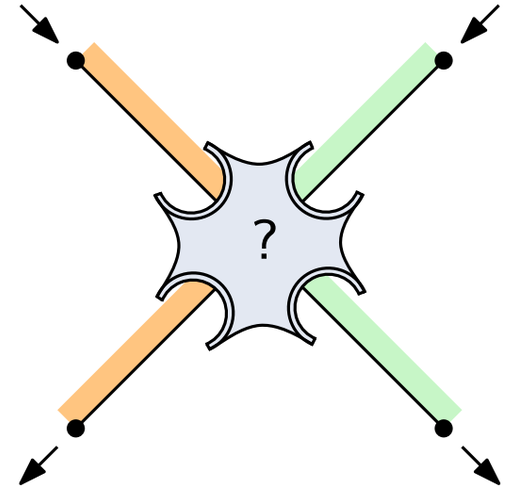
- **Obs. 1** Triangles induce at least 1 switch
- **Obs. 2** 6 triangles in gadget

Crossing gadget



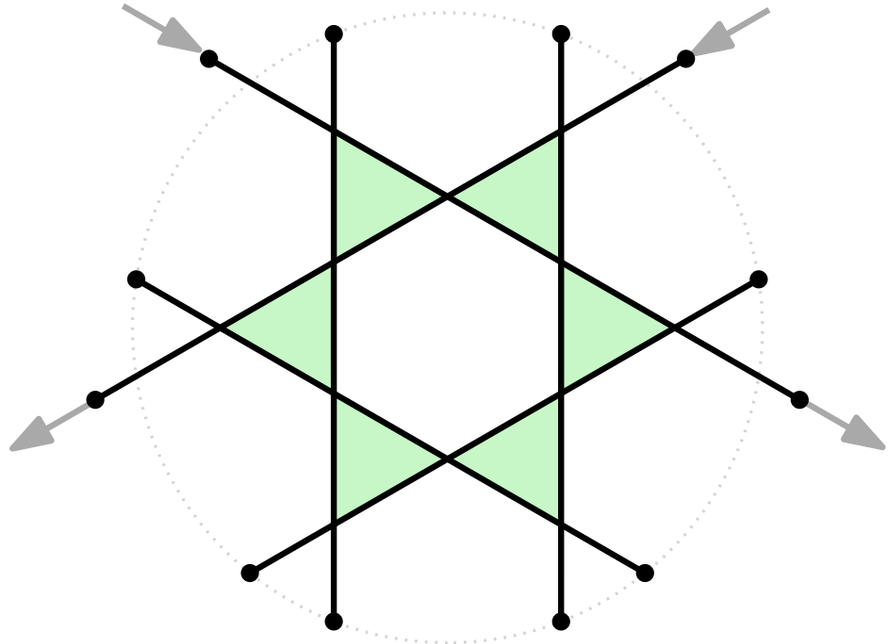


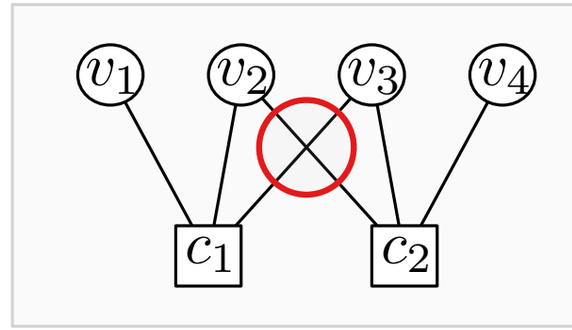
T . . . tunnel
B . . . bridge



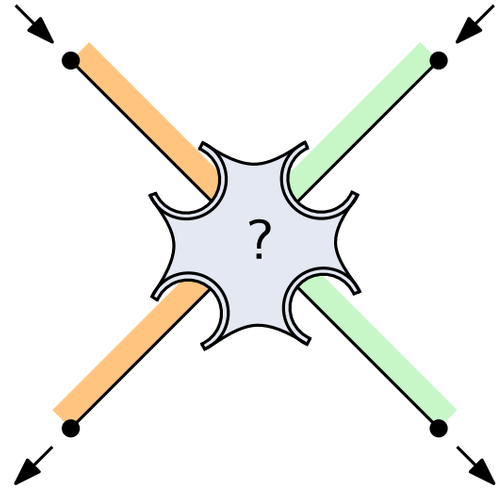
- **Obs. 1** Triangles induce at least 1 switch
- **Obs. 2** 6 triangles in gadget
- **Obs. 3** 6 edges in gadget

Crossing gadget



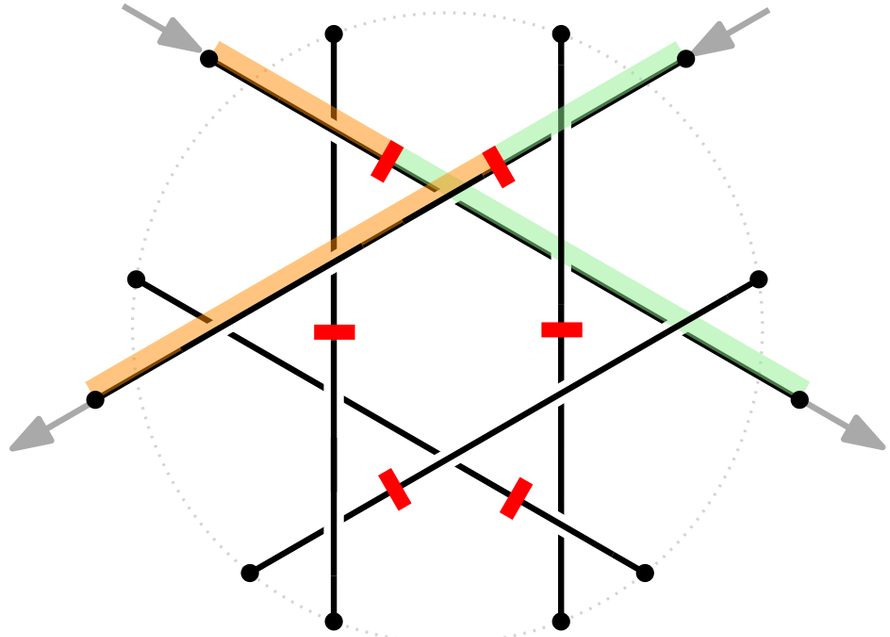


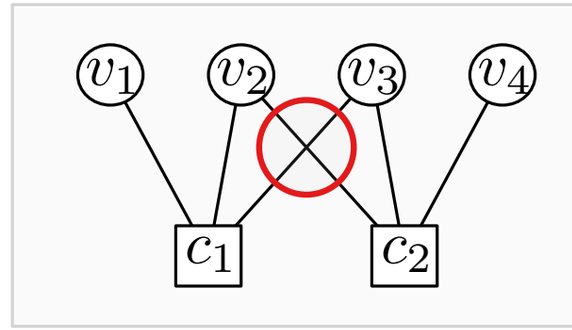
T . . . tunnel
B . . . bridge



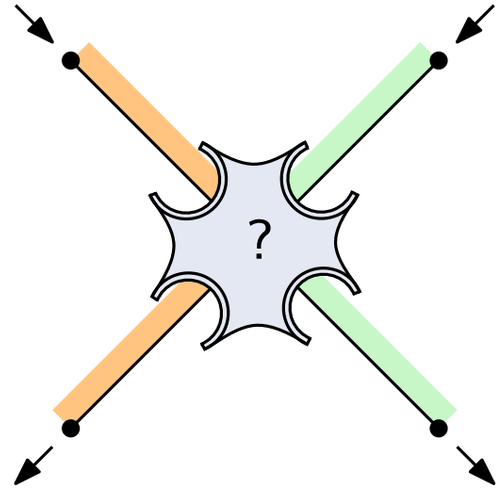
- **Obs. 1** Triangles induce at least 1 switch
- **Obs. 2** 6 triangles in gadget
- **Obs. 3** 6 edges in gadget $\Rightarrow$ at least 1 switch per edge

Crossing gadget



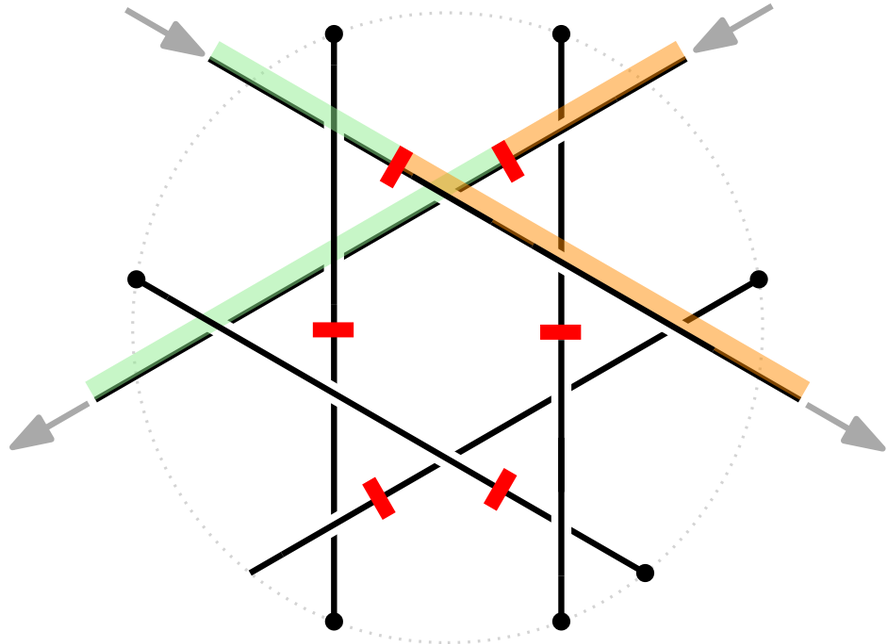


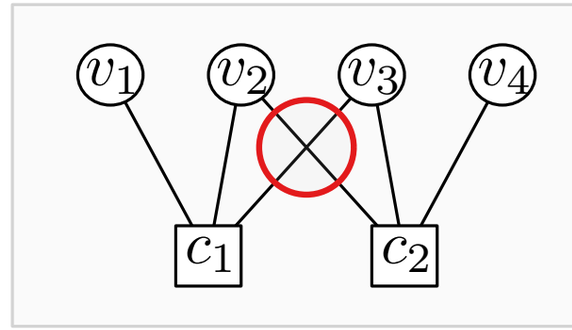
T . . . tunnel
B . . . bridge



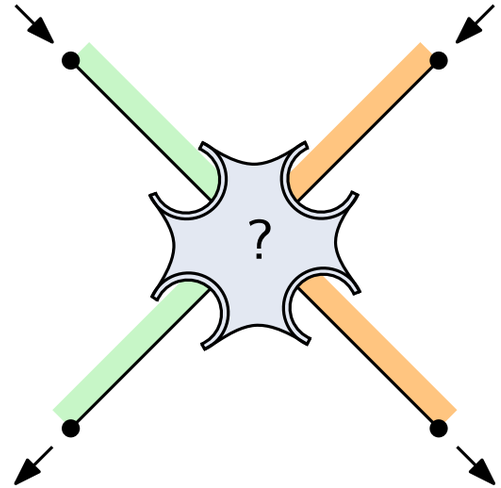
- **Obs. 1** Triangles induce at least 1 switch
- **Obs. 2** 6 triangles in gadget
- **Obs. 3** 6 edges in gadget $\Rightarrow$ at least 1 switch per edge

Crossing gadget



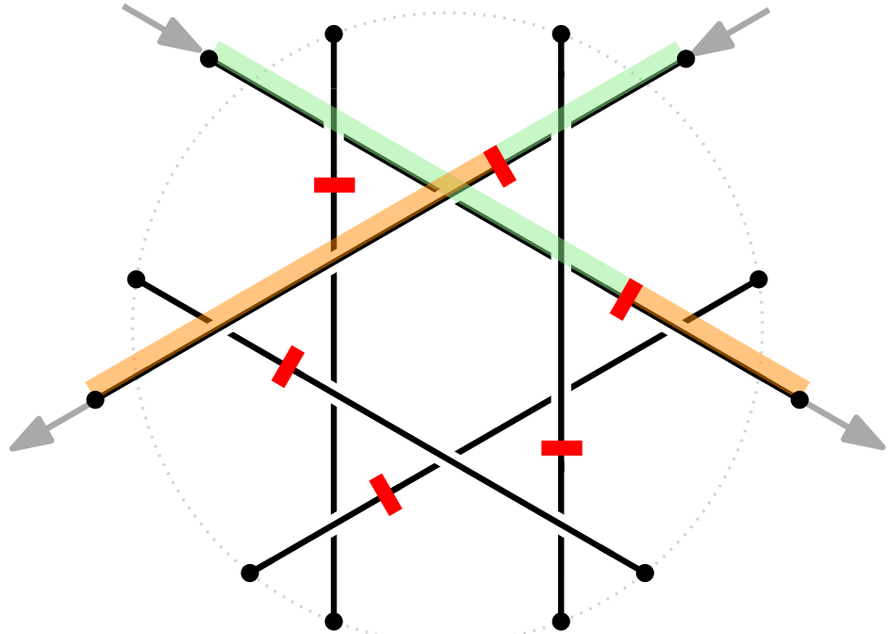


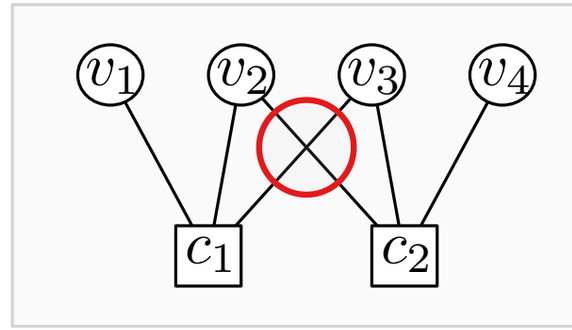
T . . . tunnel
B . . . bridge



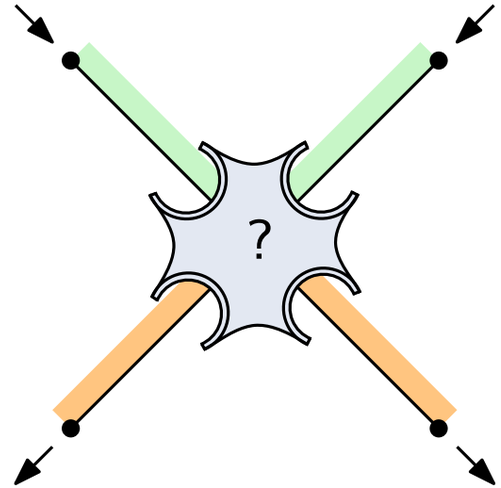
- **Obs. 1** Triangles induce at least 1 switch
- **Obs. 2** 6 triangles in gadget
- **Obs. 3** 6 edges in gadget $\Rightarrow$ at least 1 switch per edge

Crossing gadget



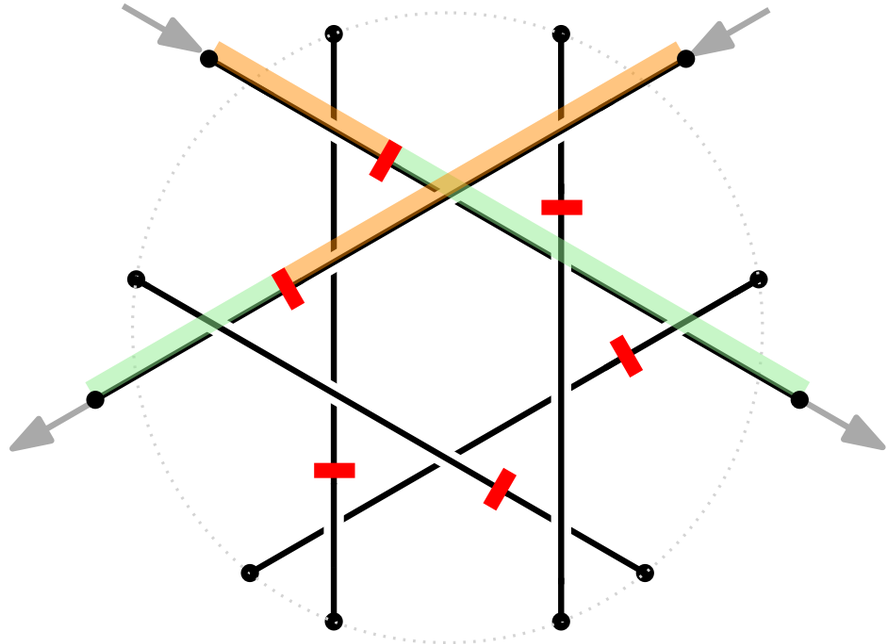


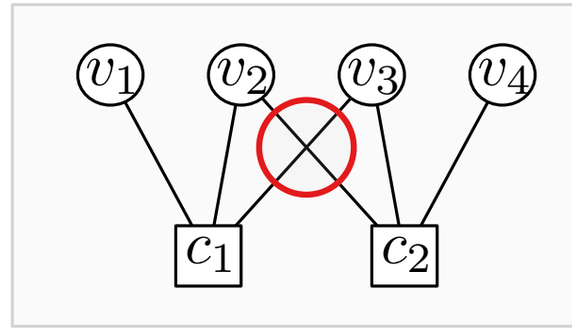
T . . . tunnel
B . . . bridge



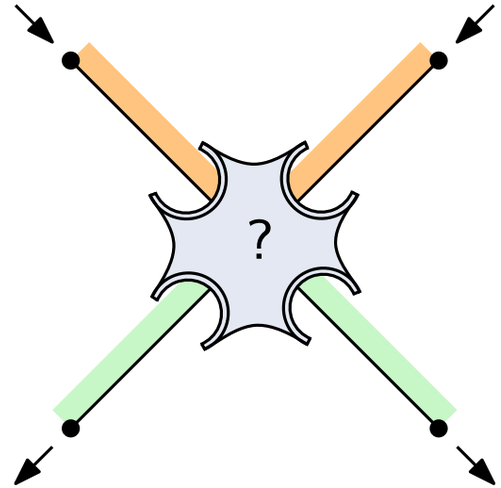
- **Obs. 1** Triangles induce at least 1 switch
- **Obs. 2** 6 triangles in gadget
- **Obs. 3** 6 edges in gadget $\Rightarrow$ at least 1 switch per edge

Crossing gadget



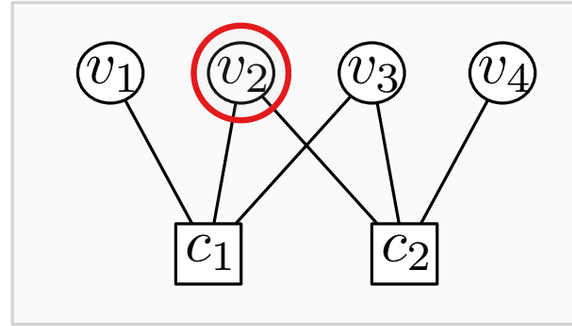


T . . . tunnel
B . . . bridge



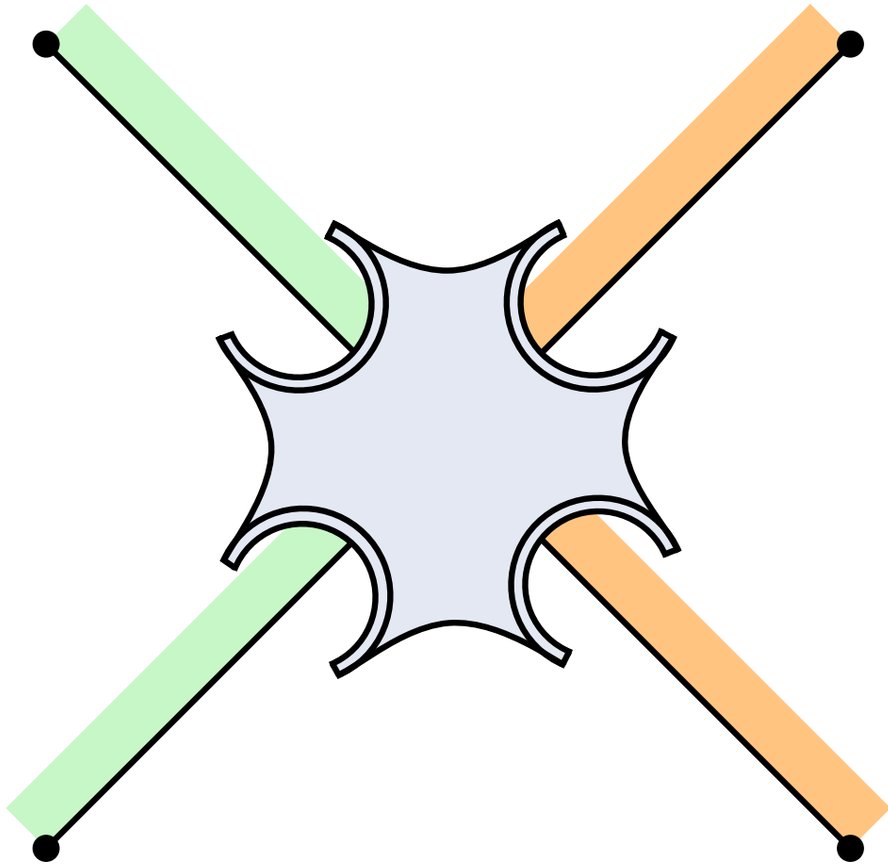
- **Obs. 1** Triangles induce at least 1 switch
- **Obs. 2** 6 triangles in gadget
- **Obs. 3** 6 edges in gadget $\Rightarrow$ at least 1 switch per edge

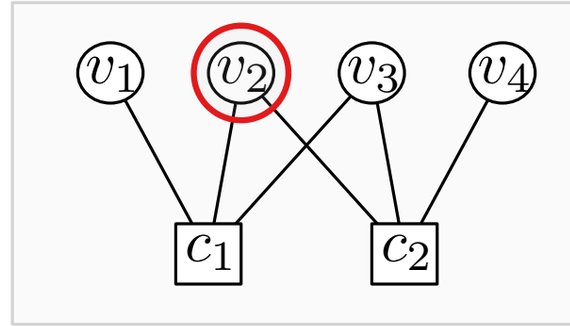
Variable gadgets



T . . . tunnel
B . . . bridge

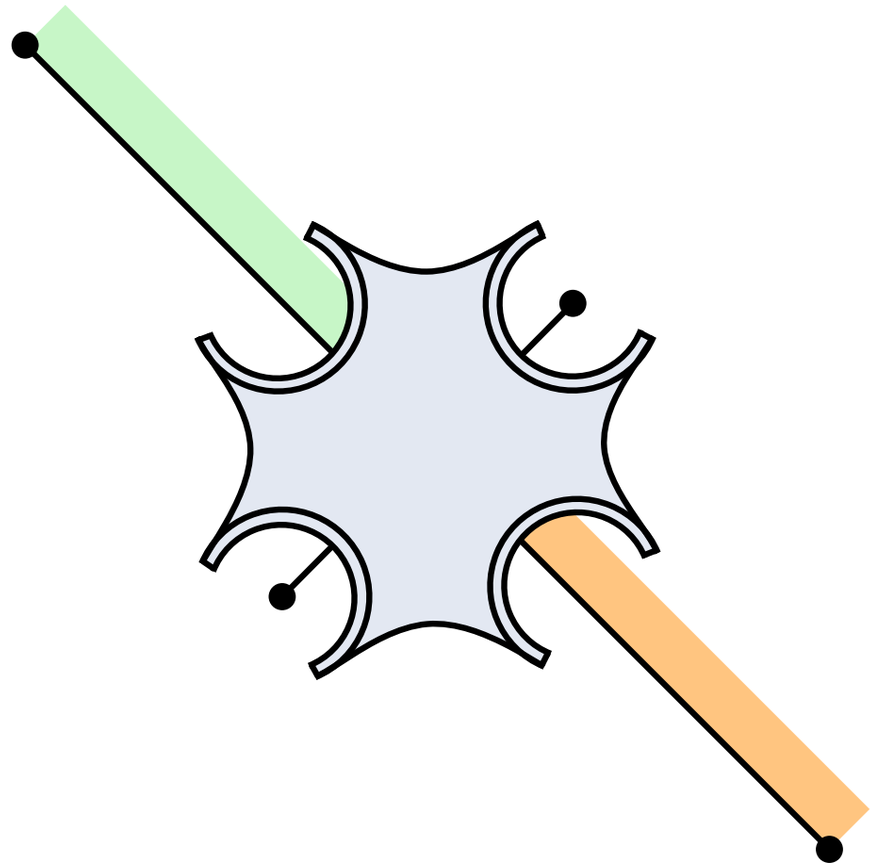
Variable gadgets

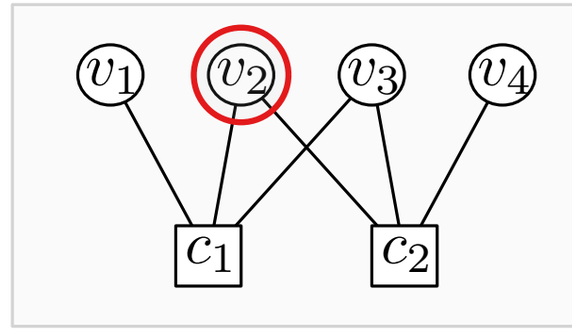




T . . . tunnel
B . . . bridge

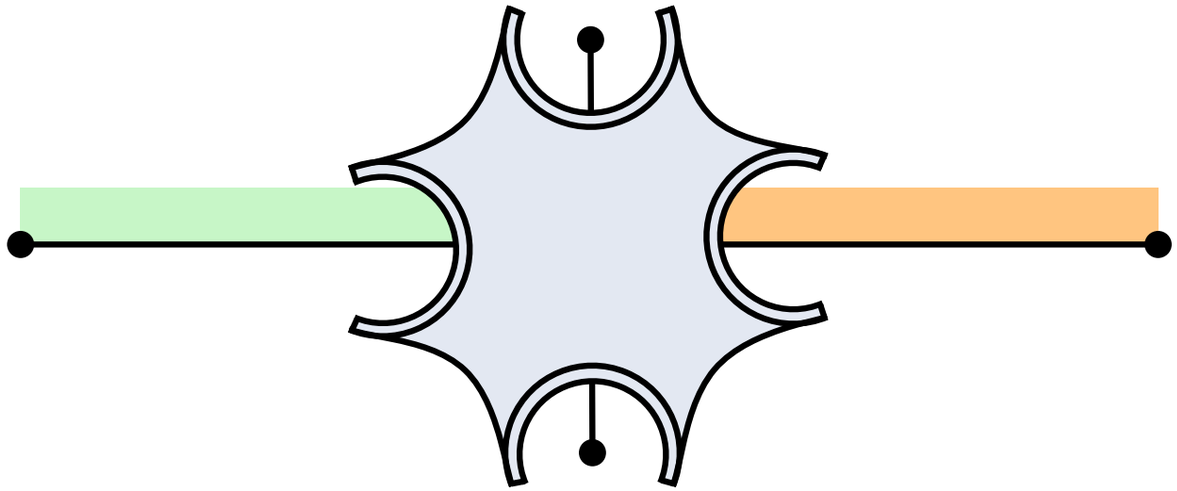
Variable gadgets

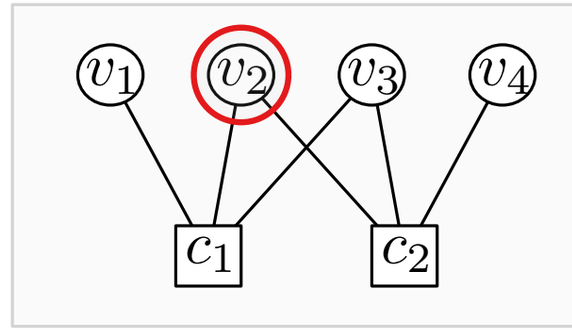




T . . . tunnel
B . . . bridge

Variable gadgets

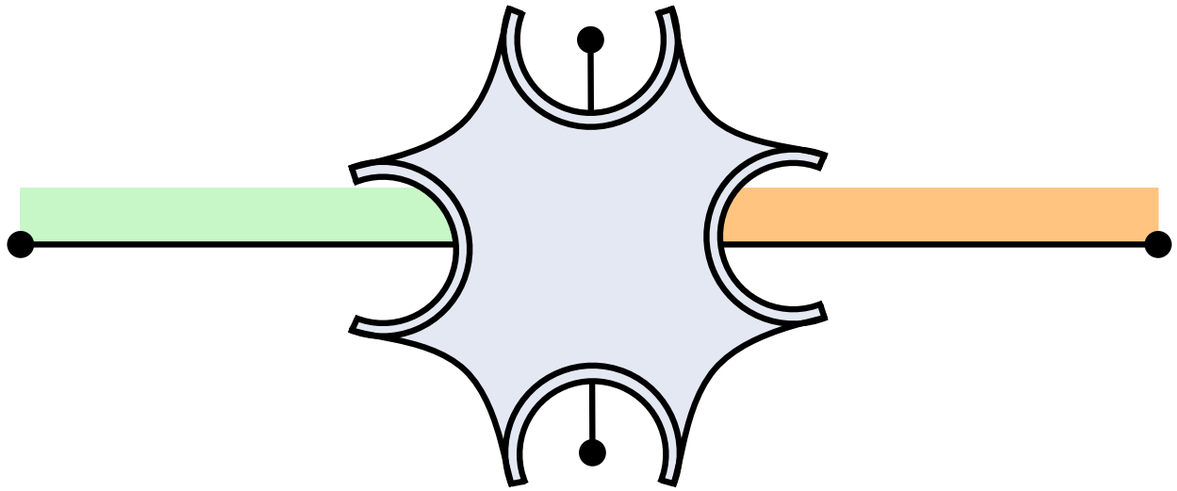


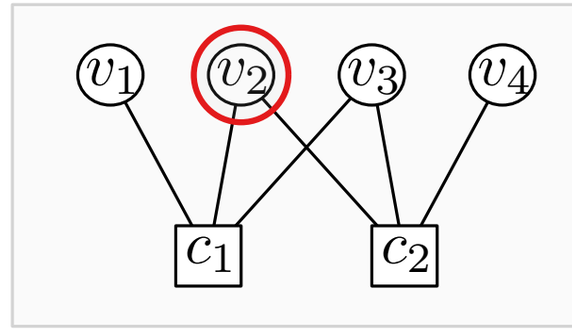


T . . . tunnel
B . . . bridge

Variable gadgets

- v_i = true

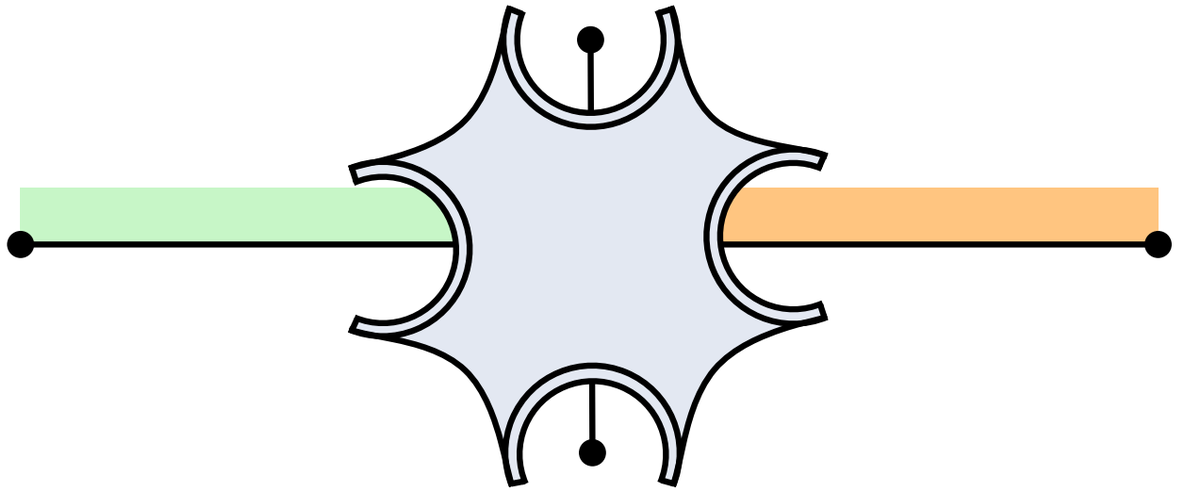




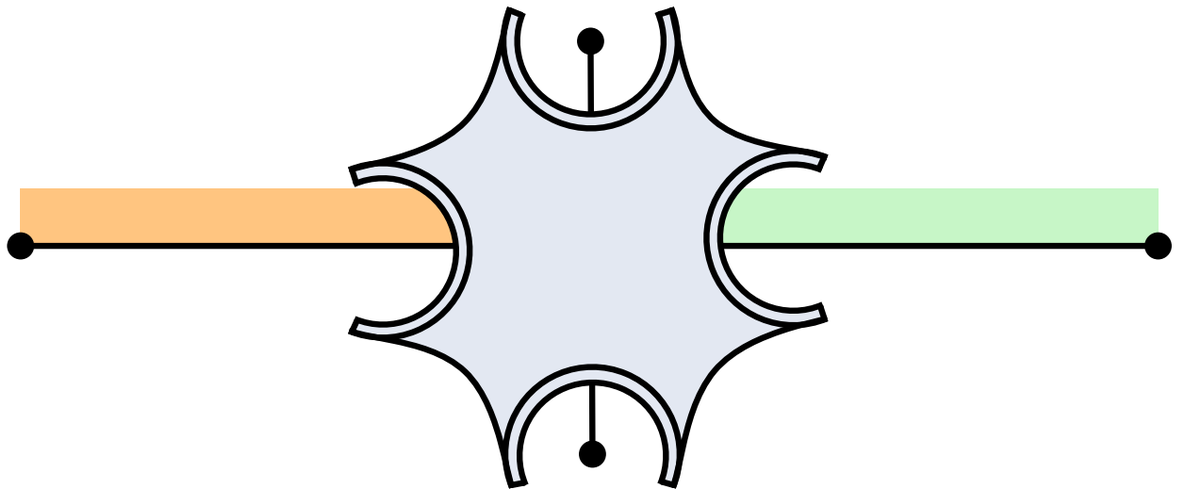
T . . . tunnel
B . . . bridge

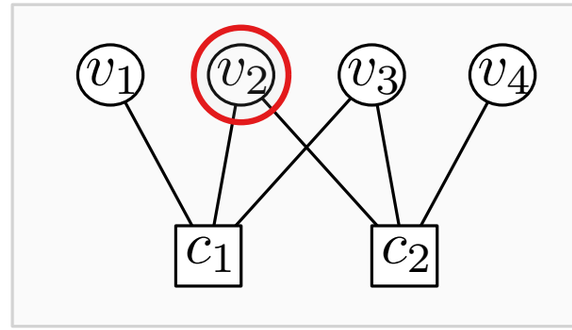
Variable gadgets

- v_i = true



- v_i = false

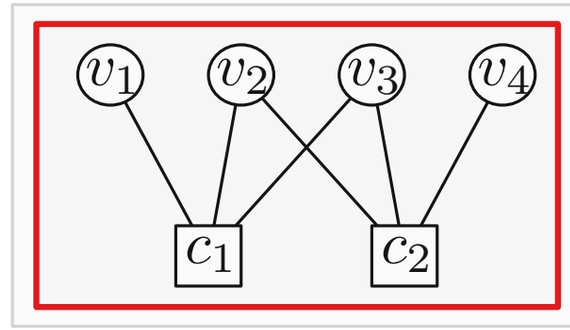




T . . . tunnel
B . . . bridge

Reduction

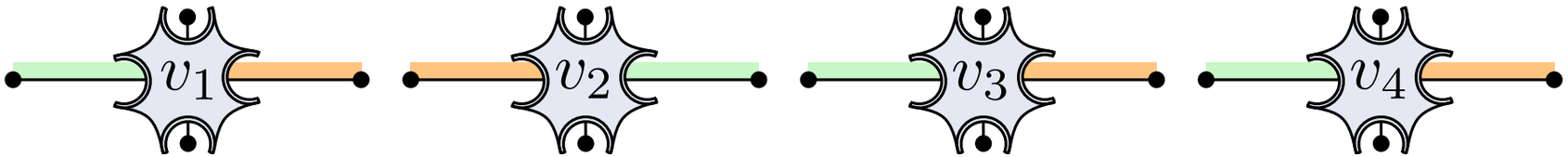
- $\mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$

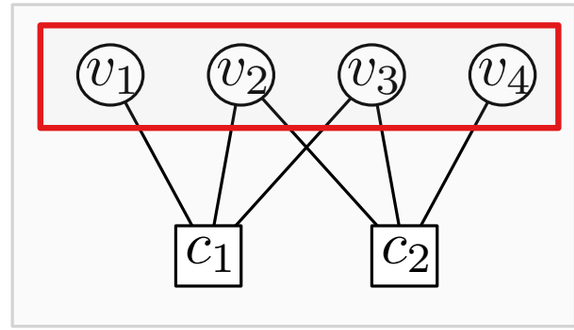


T . . . tunnel
B . . . bridge

Reduction

- $\mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$

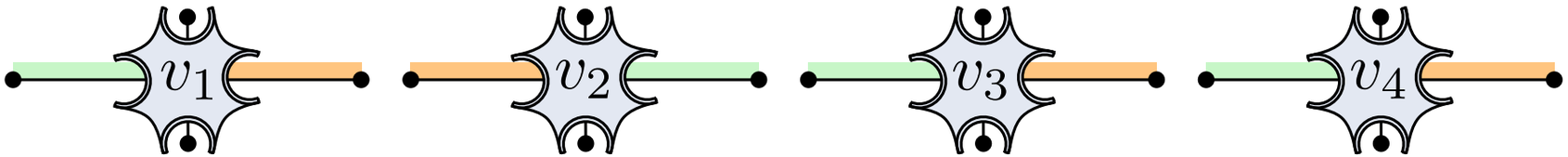


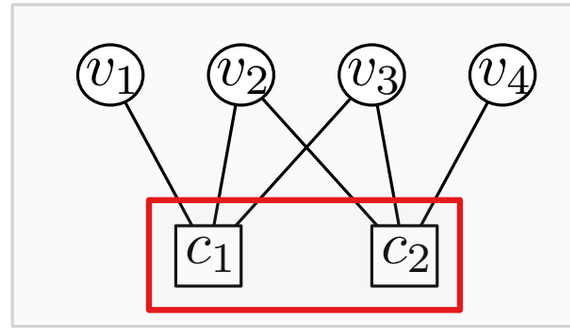


T . . . tunnel
B . . . bridge

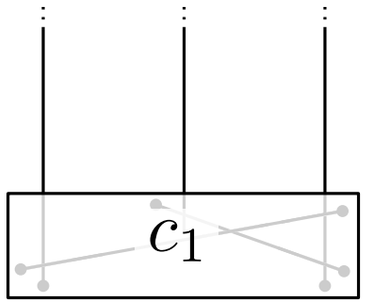
Reduction

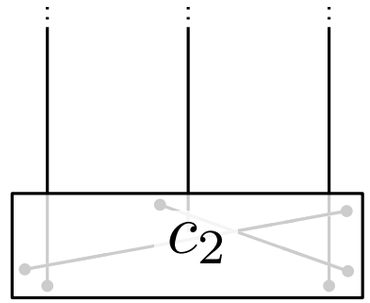
- $\mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$





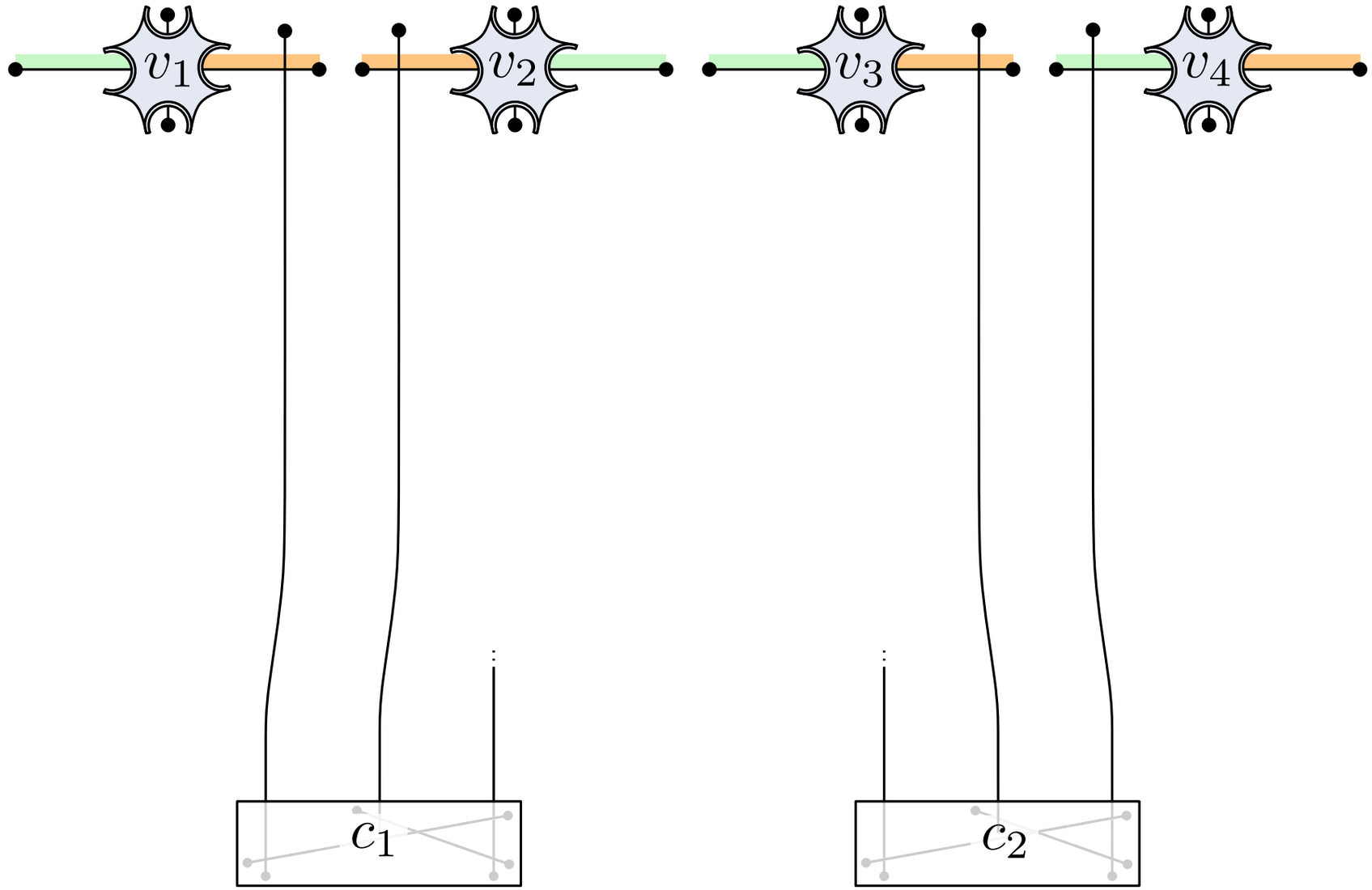
T . . . tunnel
B . . . bridge

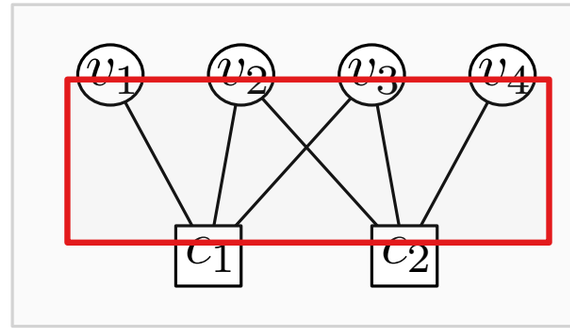




Reduction

- $\mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$

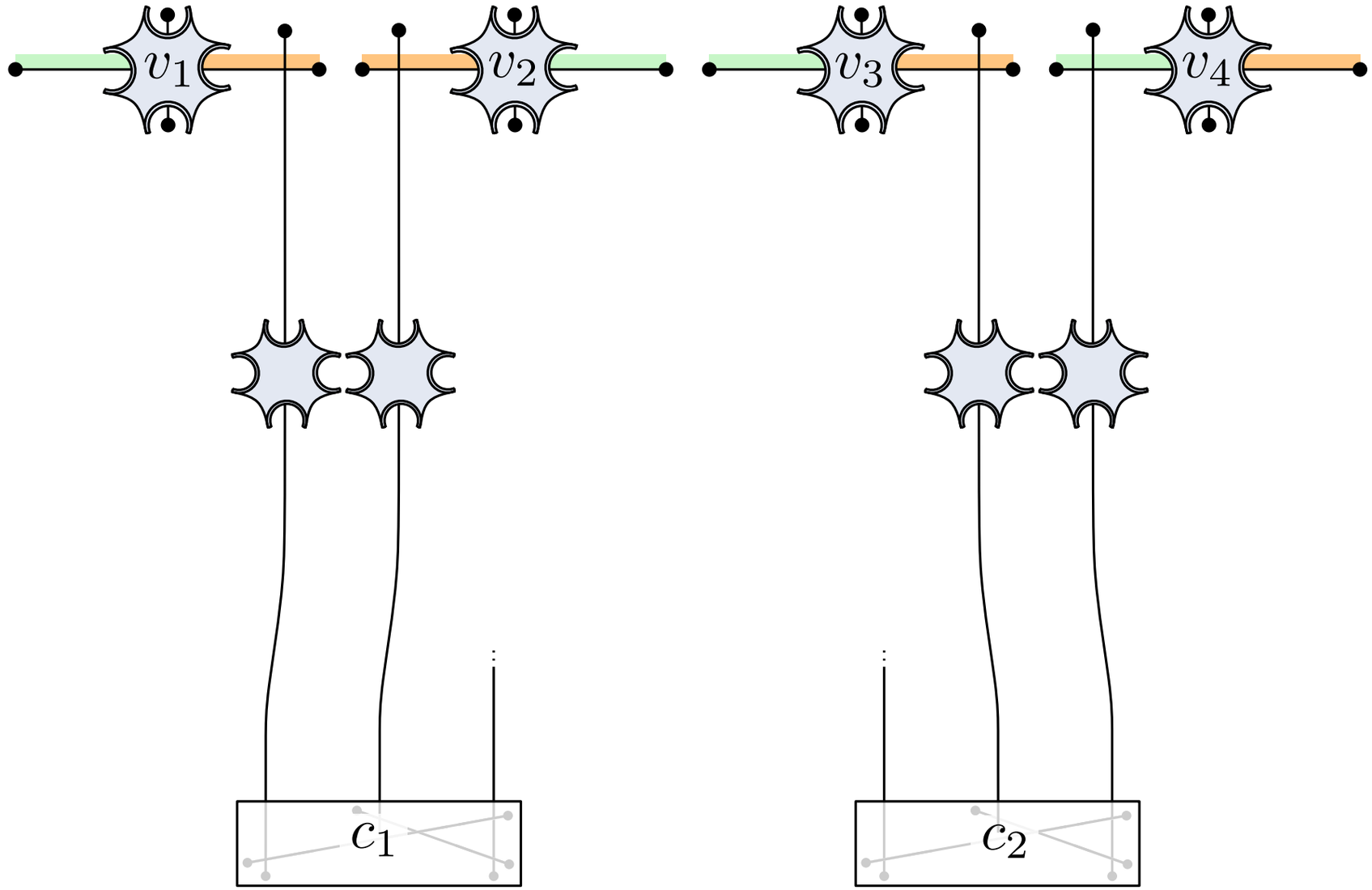


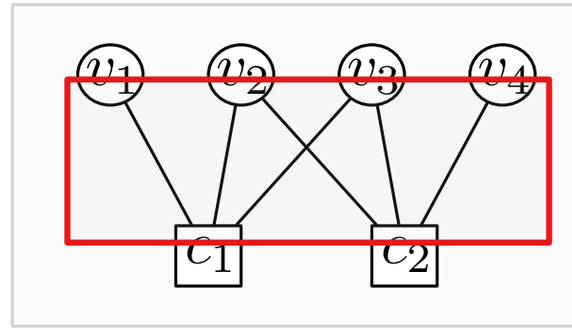


T . . . tunnel
B . . . bridge

Reduction

- NAE$(\neg v_1 \vee v_2 \vee v_3) \wedge$ NAE$(\neg v_2 \vee \neg v_3 \vee v_4)$

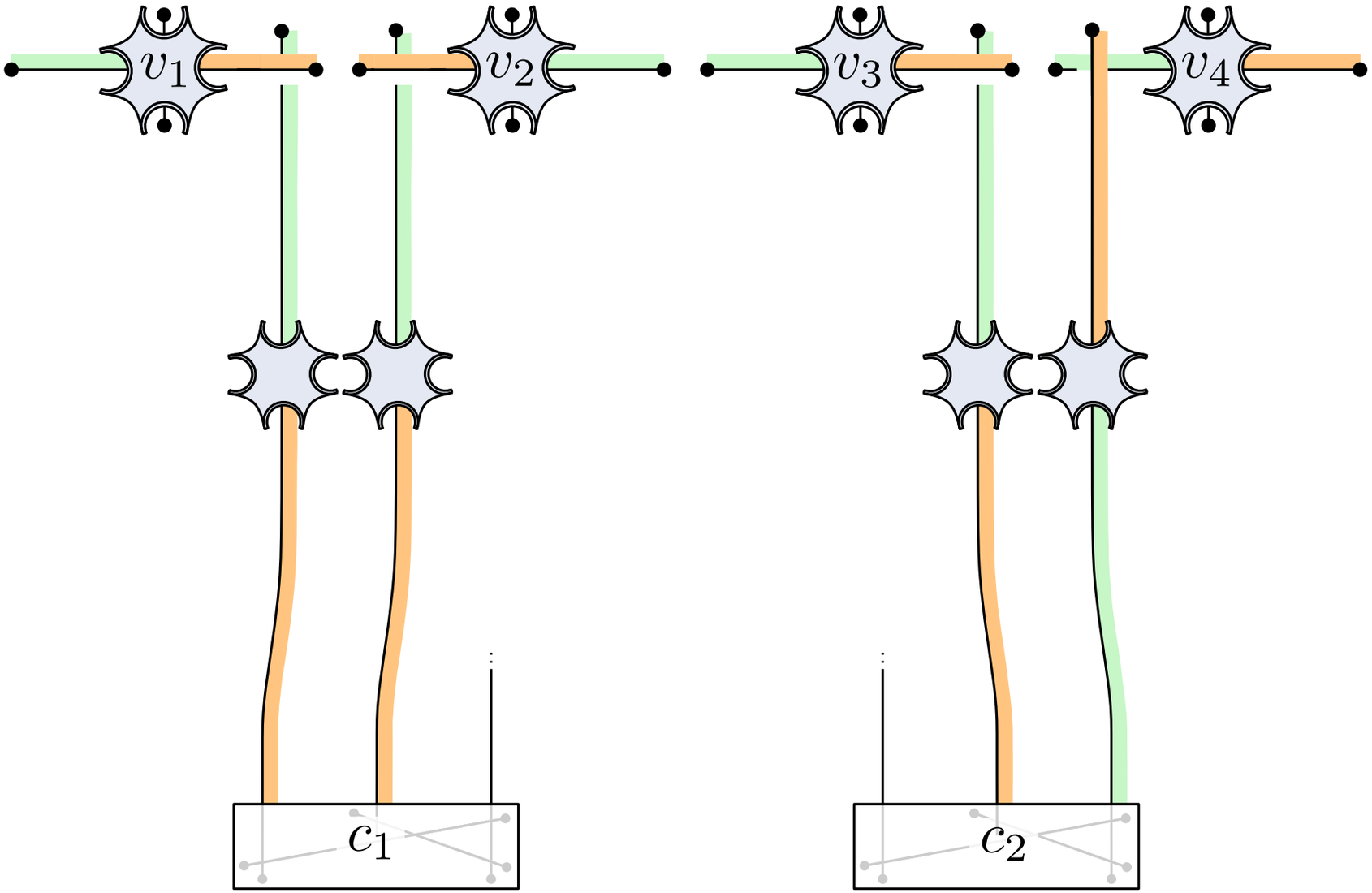


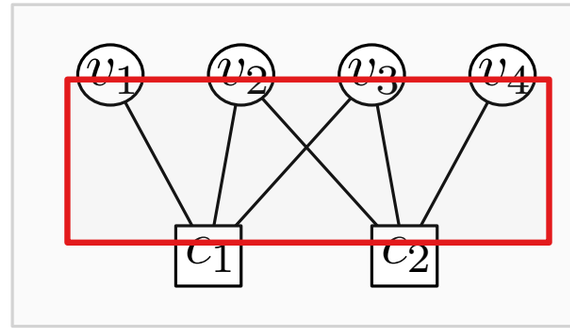


T . . . tunnel
B . . . bridge

Reduction

- NAE$(\neg v_1 \vee v_2 \vee v_3) \wedge$ NAE$(\neg v_2 \vee \neg v_3 \vee v_4)$

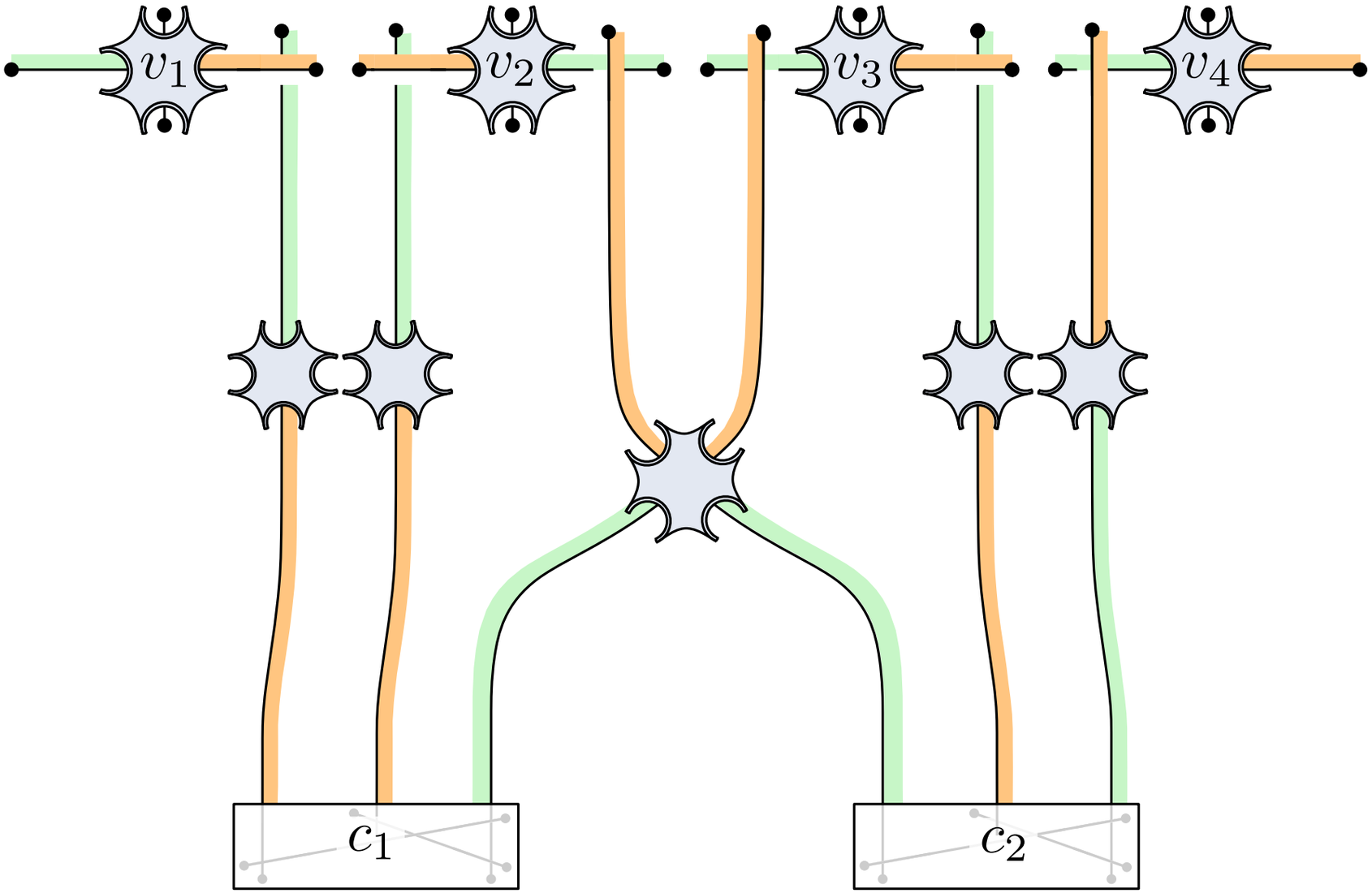


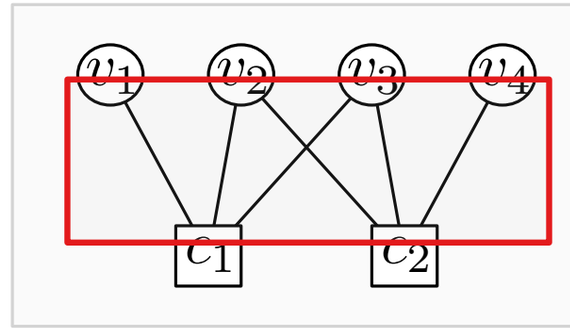


T . . . tunnel
B . . . bridge

Reduction

- NAE$(\neg v_1 \vee v_2 \vee v_3) \wedge$ NAE$(\neg v_2 \vee \neg v_3 \vee v_4)$

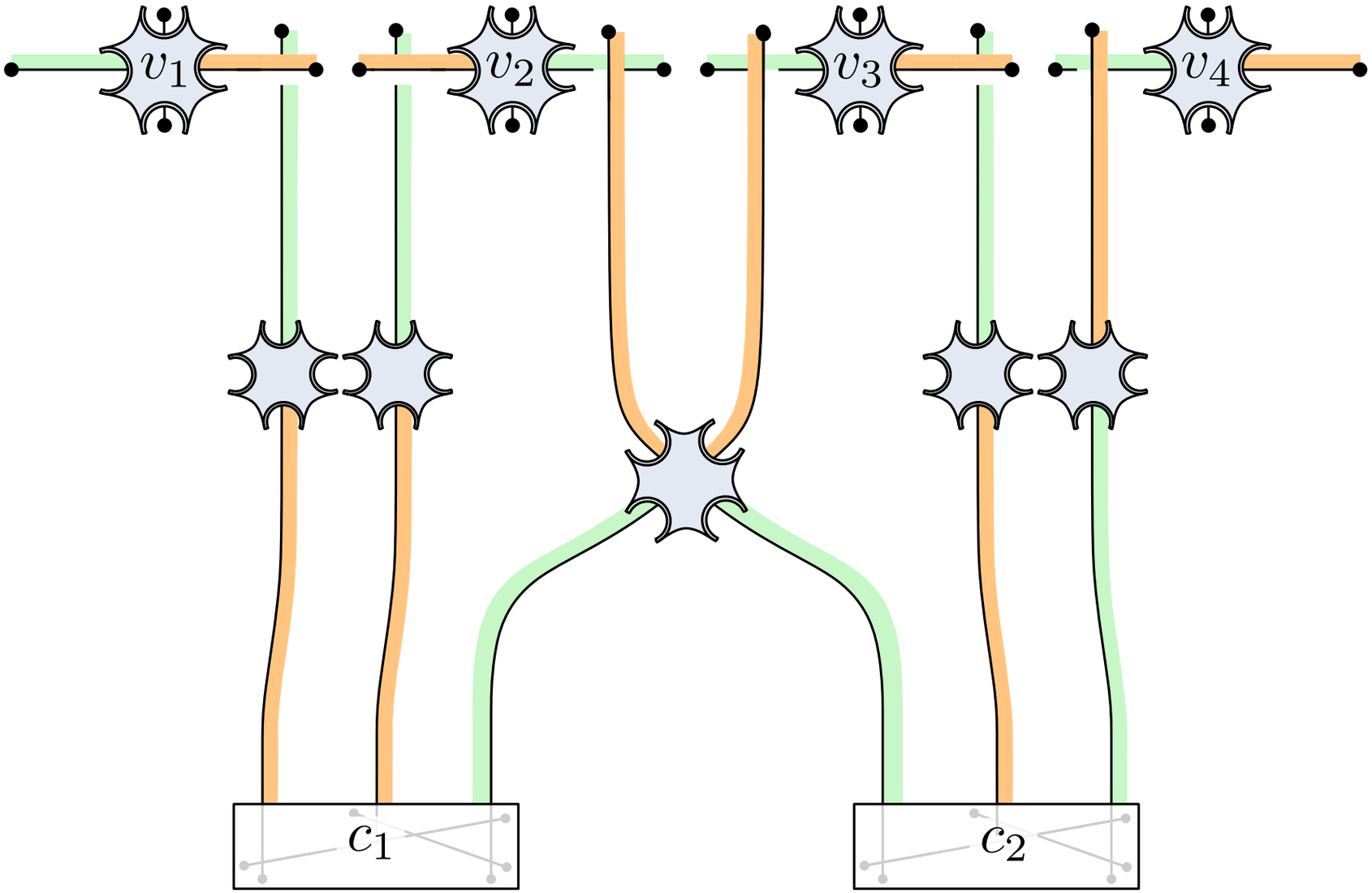


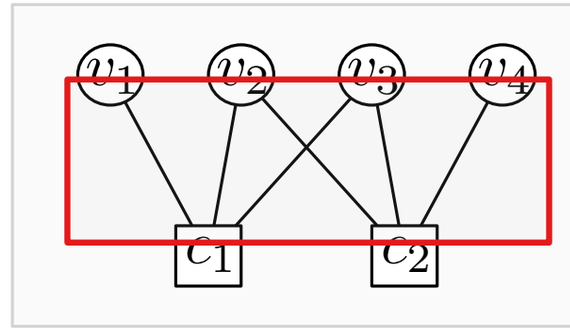


T . . . tunnel
B . . . bridge

Reduction

- $\mathsf{NAE}(\neg v_1 \vee v_2 \vee v_3) \wedge \mathsf{NAE}(\neg v_2 \vee \neg v_3 \vee v_4)$





T . . . tunnel
B . . . bridge

$v_1 = \text{true}$
$v_2 = \text{true}$
$v_3 = \text{false}$
$v_4 = \text{true}$

Results

Hardness	Stacking	**Weaving**	Realizable
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	$\mathsf{VC}(\chi_G)$	*open*
MinSwitchEdges	$\mathsf{tw}(K_{\delta_G})$	$\mathsf{tw}(K_{\delta_G})$	$\mathsf{tw}(K_{\delta_G})$

- k-switch-planar graphs
- MILP implementation

* with restrictions

Results

Hardness	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	VC(χ_G)	*open*
MinSwitchEdges	tw(K_{δ_G})	tw(K_{δ_G})	tw(K_{δ_G})

- k-switch-planar graphs
- MILP implementation

* with restrictions

Results

Hardness	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	$\mathsf{VC}(\chi_G)$	*open*
MinSwitchEdges	$\mathsf{tw}(K_{\delta_G})$	$\mathsf{tw}(K_{\delta_G})$	$\mathsf{tw}(K_{\delta_G})$

- k-switch-planar graphs
- MILP implementation

* with restrictions

On Graph Classes

- k-gap-planar graphs [Baffer et al., 2018]

On Graph Classes

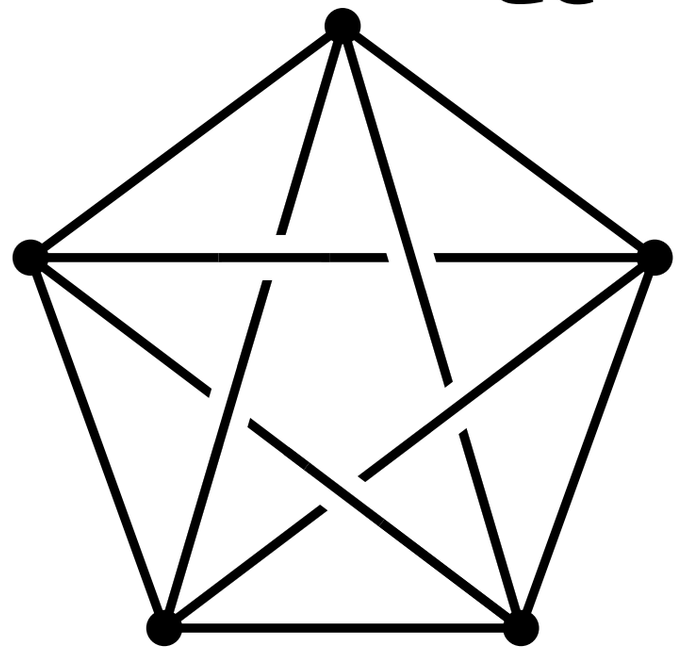
- k-gap-planar graphs [Baffer et al., 2018]

  ... all graphs that can be embedded crossing-free by adding at most k gaps per edge ...

On Graph Classes

- k-gap-planar graphs [Baffer et al., 2018]

  ... all graphs that can be embedded crossing-free by adding at most k gaps per edge ...



K_5 is 1-gap-planar

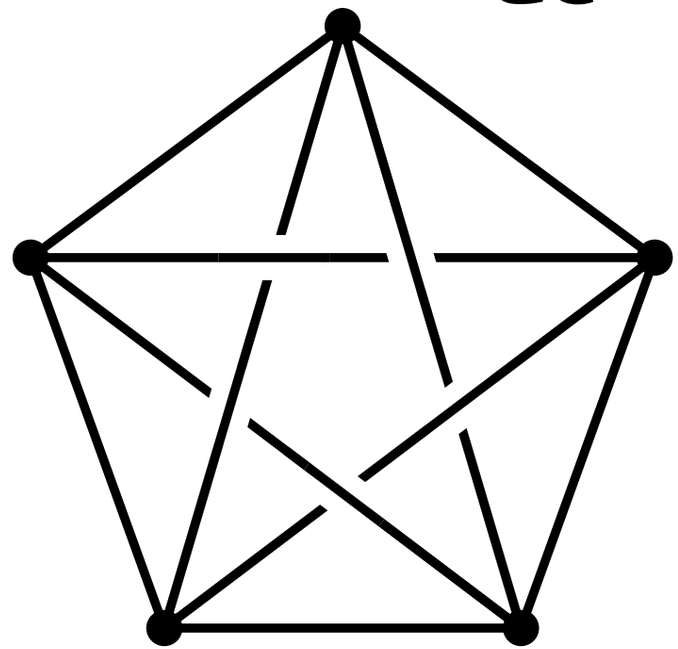
On Graph Classes

- k-gap-planar graphs [Baffer et al., 2018]

  ... all graphs that can be embedded crossing-free by adding at most k gaps per edge ...

- k-switch-planar graphs

  ... all graphs that can be encased with at most k switches per edge ...



K_5 is 1-gap-planar

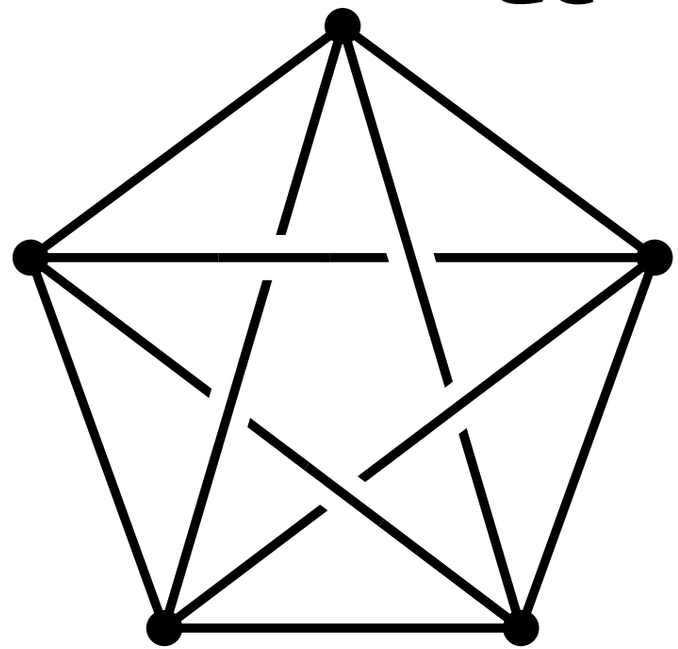
On Graph Classes

- k-gap-planar graphs [Baffer et al., 2018]

  ... all graphs that can be embedded crossing-free by adding at most k gaps per edge ...

- k-switch-planar graphs

  ... all graphs that can be encased with at most k switches per edge ...



K_5 is 1-gap-planar

K_5 is 1-switch-planar

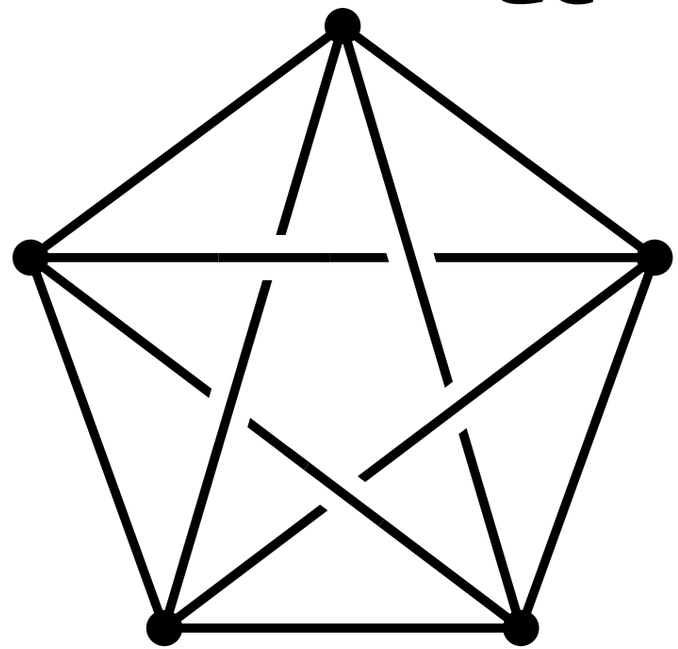
On Graph Classes

- k-gap-planar graphs [Baffer et al., 2018]

  ... all graphs that can be embedded crossing-free by adding at most k gaps per edge ...

- b-bend k-switch-planar graphs

  ... all graphs that can be encased with at most k switches and b bends per edge ...



K_5 is 1-gap-planar

K_5 is 1-switch-planar

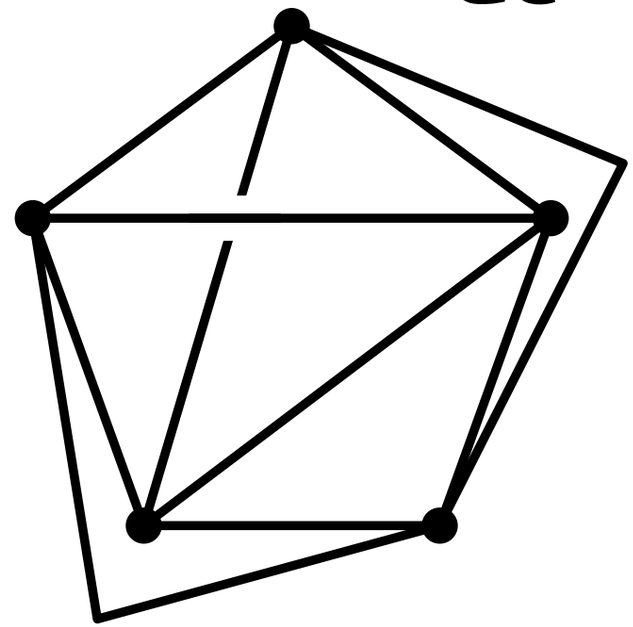
On Graph Classes

- k-gap-planar graphs [Baffer et al., 2018]

  ... all graphs that can be embedded crossing-free by adding at most k gaps per edge ...

- b-bend k-switch-planar graphs

  ... all graphs that can be encased with at most k switches and b bends per edge ...



K_5 is 1-gap-planar

K_5 is 1-bend 0-switch planar

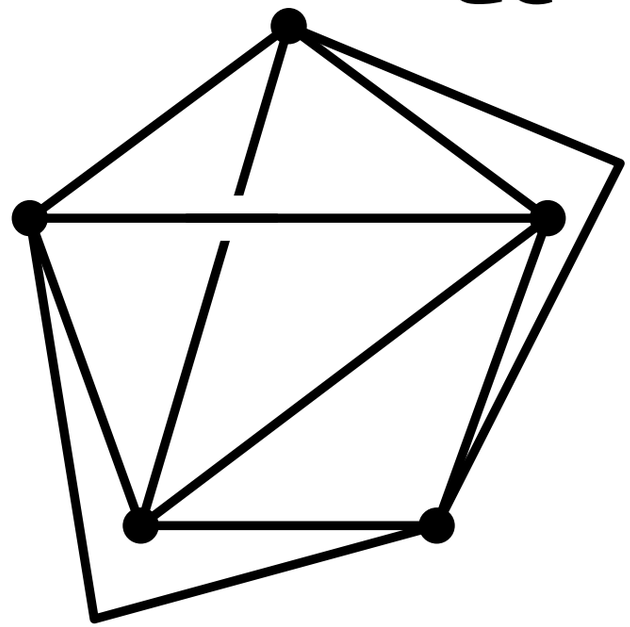
On Graph Classes

- k-gap-planar graphs [Baffer et al., 2018]

  ... all graphs that can be embedded crossing-free by adding at most k gaps per edge ...

- b-bend k-switch-planar graphs

  ... all graphs that can be encased with at most k switches and b bends per edge ...



K_5 is 1-gap-planar

K_5 is 1-bend 0-switch planar

Theorem. Any graph is 2-bend 1-switch-planar.

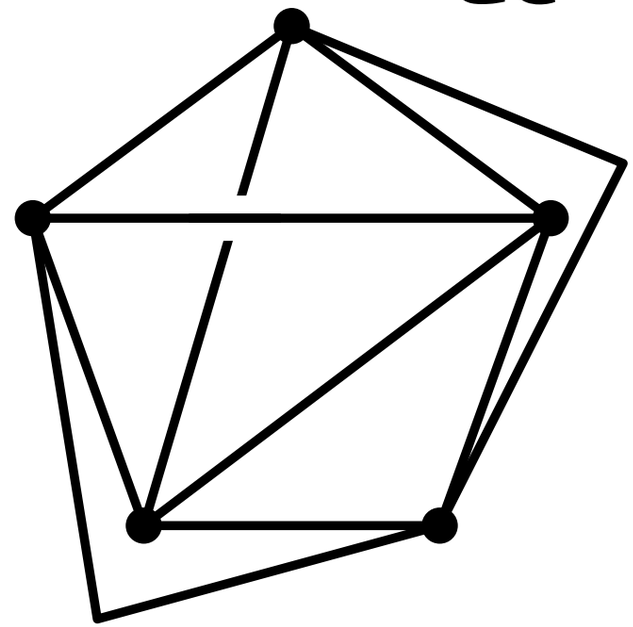
On Graph Classes

- k-gap-planar graphs [Baffer et al., 2018]

  ... all graphs that can be embedded crossing-free by adding at most k gaps per edge ...

- b-bend k-switch-planar graphs

  ... all graphs that can be encased with at most k switches and b bends per edge ...



K_5 is 1-gap-planar

K_5 is 1-bend 0-switch planar

Theorem. Any graph is 2-bend 1-switch-planar.

Lemma. Any bipartite graph is 1-bend 1-switch-planar.

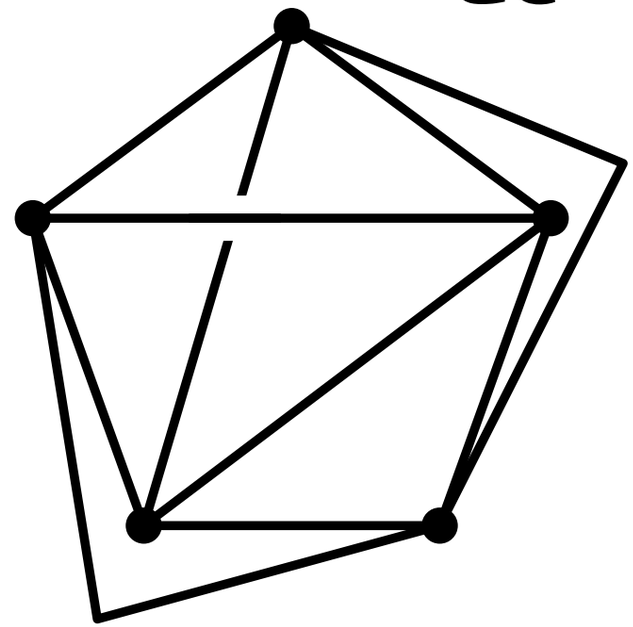
On Graph Classes

- k-gap-planar graphs [Baffer et al., 2018]

  ... all graphs that can be embedded crossing-free by adding at most k gaps per edge ...

- b-bend k-switch-planar graphs

  ... all graphs that can be encased with at most k switches and b bends per edge ...



K_5 is 1-gap-planar

K_5 is 1-bend 0-switch planar

Theorem. Any graph is 2-bend 1-switch-planar.

Lemma. Any bipartite graph is 1-bend 1-switch-planar.

Lemma. A graph is 0-bend 0-switch-planar *iff* it is geometric biplanar.

Results

Hardness	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	VC(χ_G)	*open*
MinSwitchEdges	tw(K_{δ_G})	tw(K_{δ_G})	tw(K_{δ_G})

- k-switch-planar graphs
- MILP implementation

* with restrictions

Results

Hardness

	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	$O(m^4)$*	NP-hard *
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability

	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	$\mathsf{VC}(\chi_G)$	*open*
MinSwitchEdges	$\mathsf{tw}(K_{\delta_G})$	$\mathsf{tw}(K_{\delta_G})$	$\mathsf{tw}(K_{\delta_G})$

- k-switch-planar graphs
- MILP implementation

* with restrictions

MILP implementation

- 250 random graphs with spring embedding

MILP implementation

- 250 random graphs with spring embedding



Small: ≤ 50 crossings



Large: $50 - 150$ crossings

MILP implementation

- 250 random graphs with spring embedding

Small: ≤ 50 crossings

Large: $50 - 150$ crossings

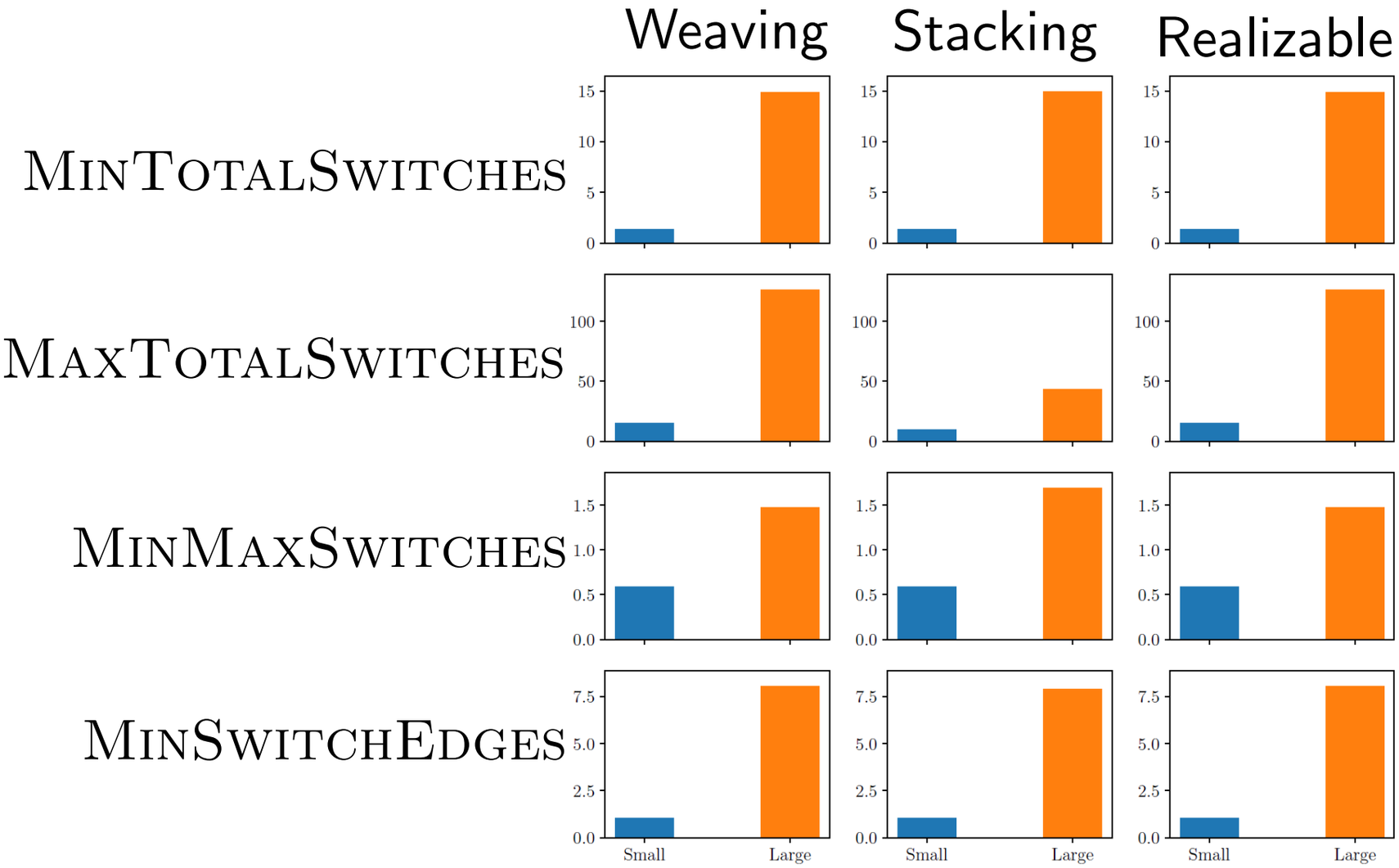


MILP implementation

- 250 random graphs with spring embedding

Small: ≤ 50 crossings

Large: $50 - 150$ crossings

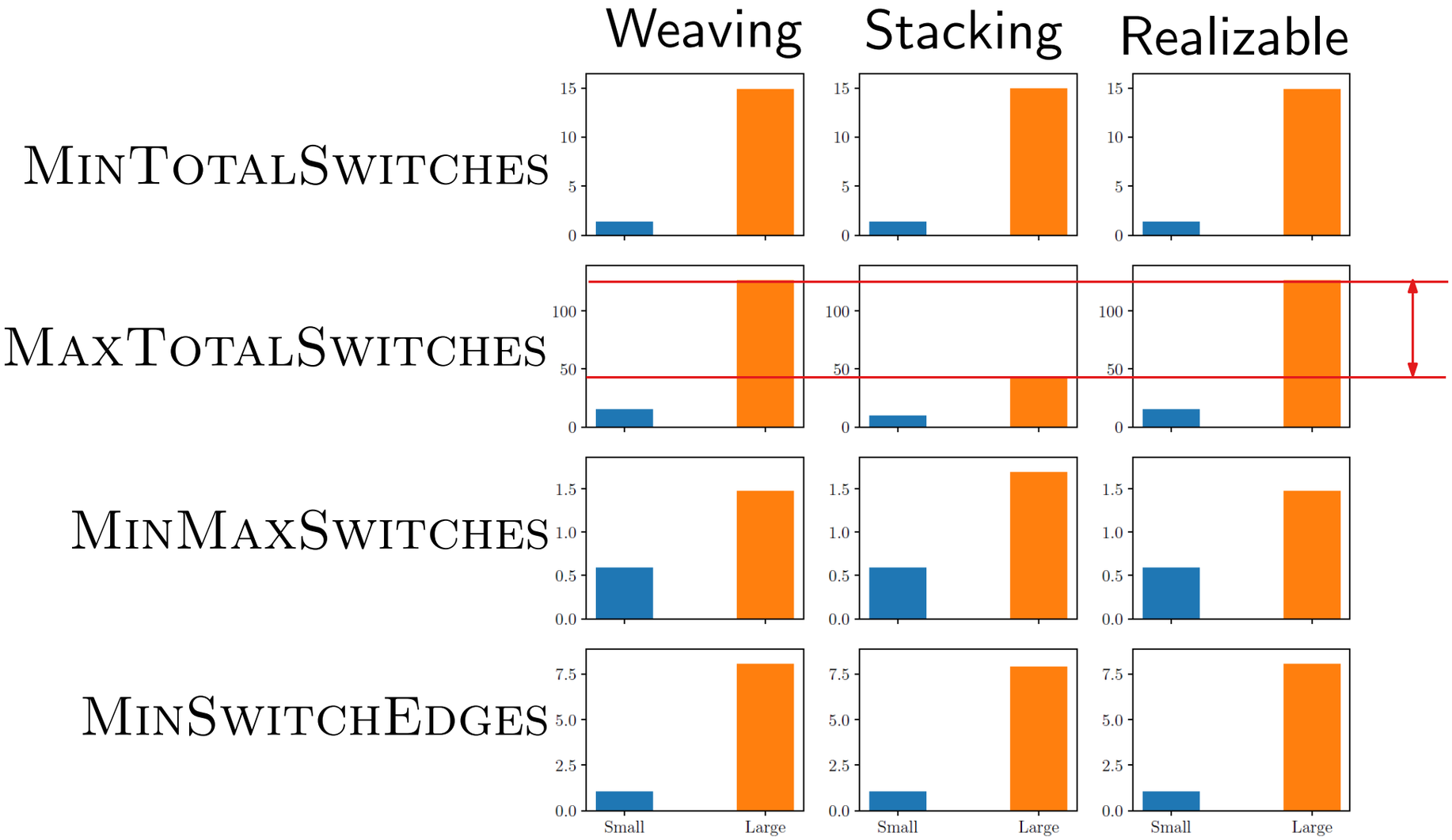


MILP implementation

- 250 random graphs with spring embedding

Small: ≤ 50 crossings

Large: $50 - 150$ crossings

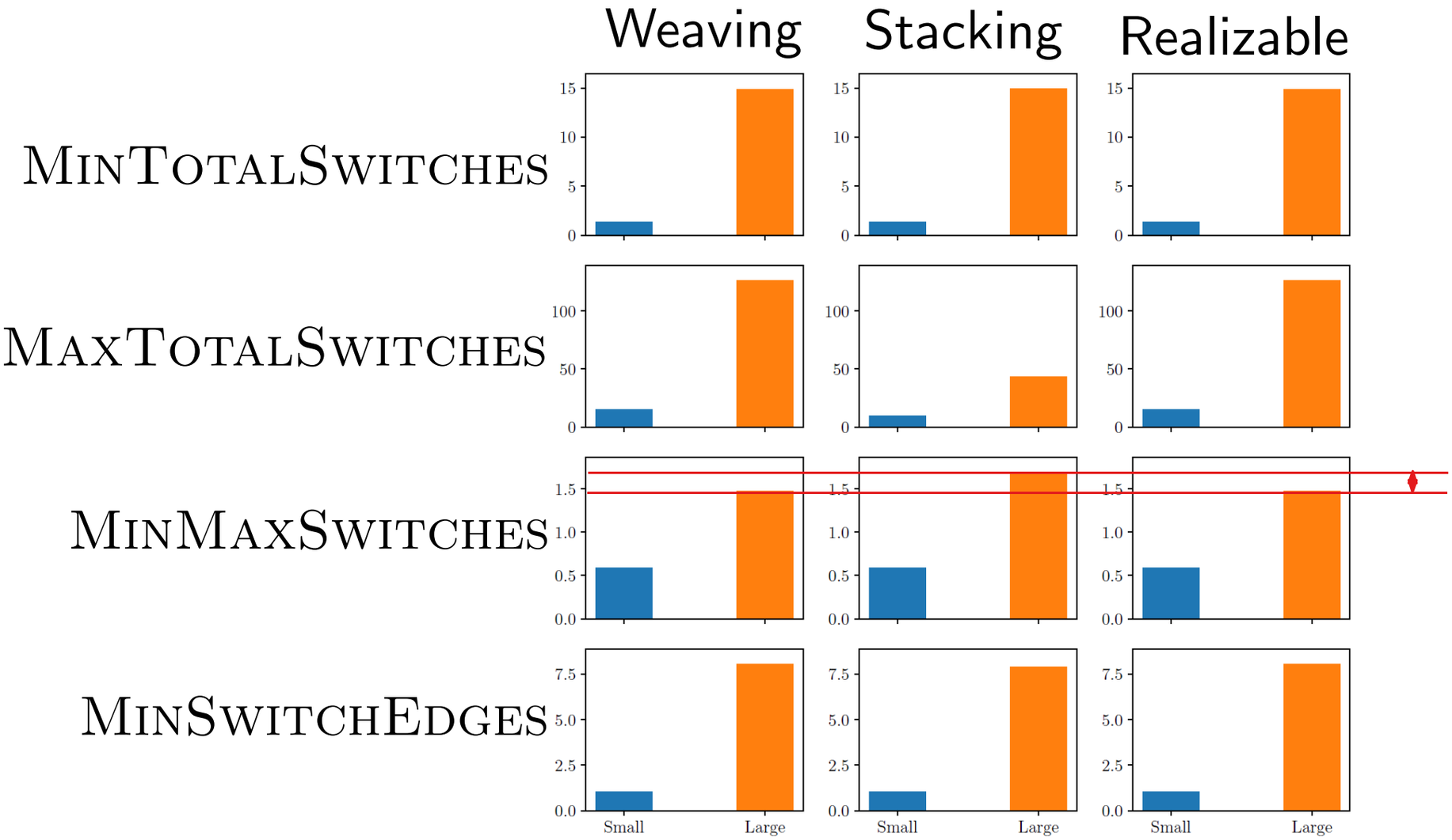


MILP implementation

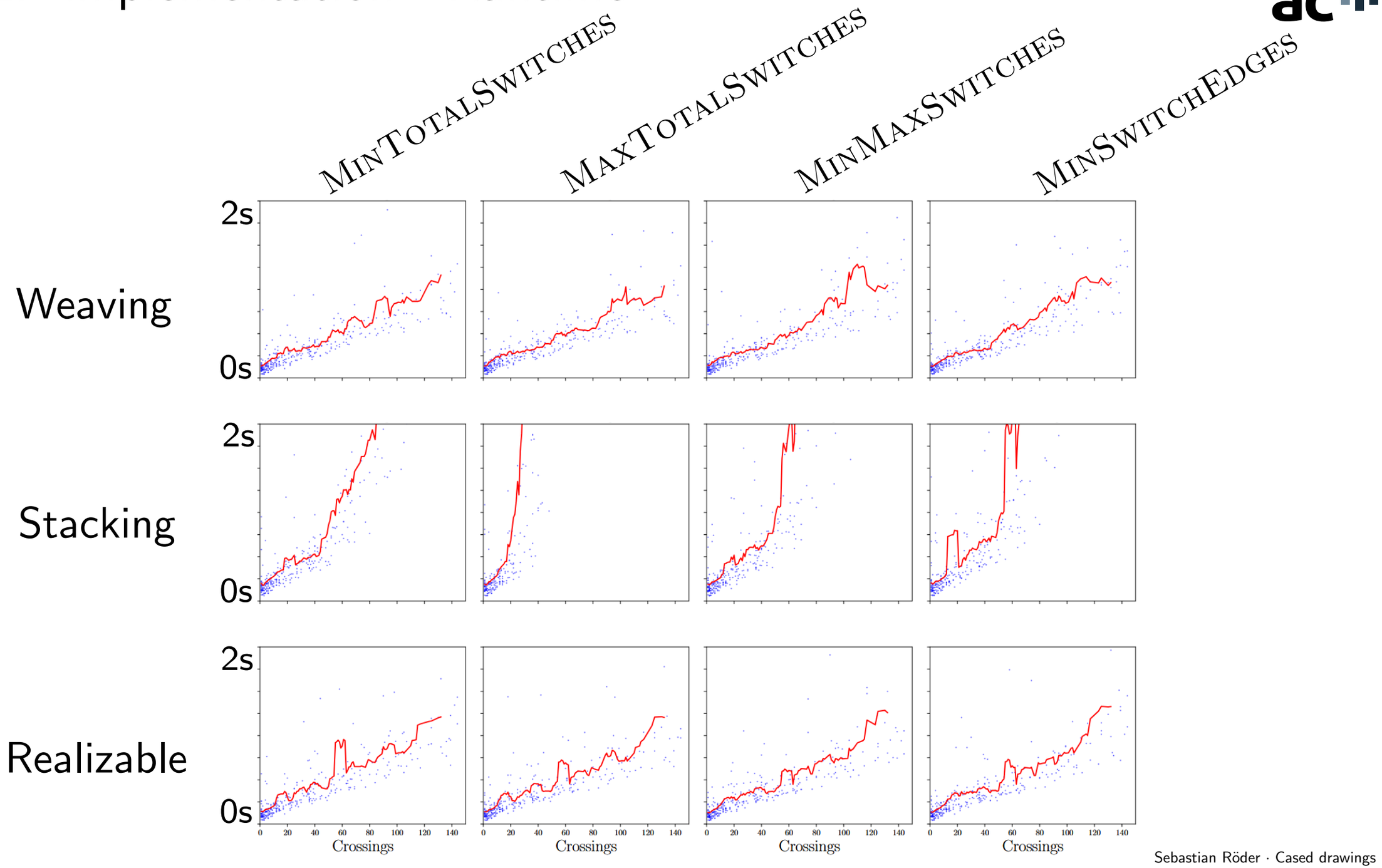
- 250 random graphs with spring embedding

Small: ≤ 50 crossings

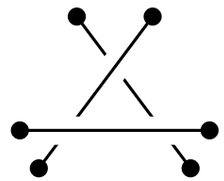
Large: $50 - 150$ crossings

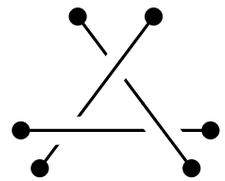


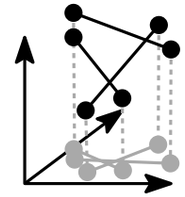
MILP implementation - Runtime



Outlook

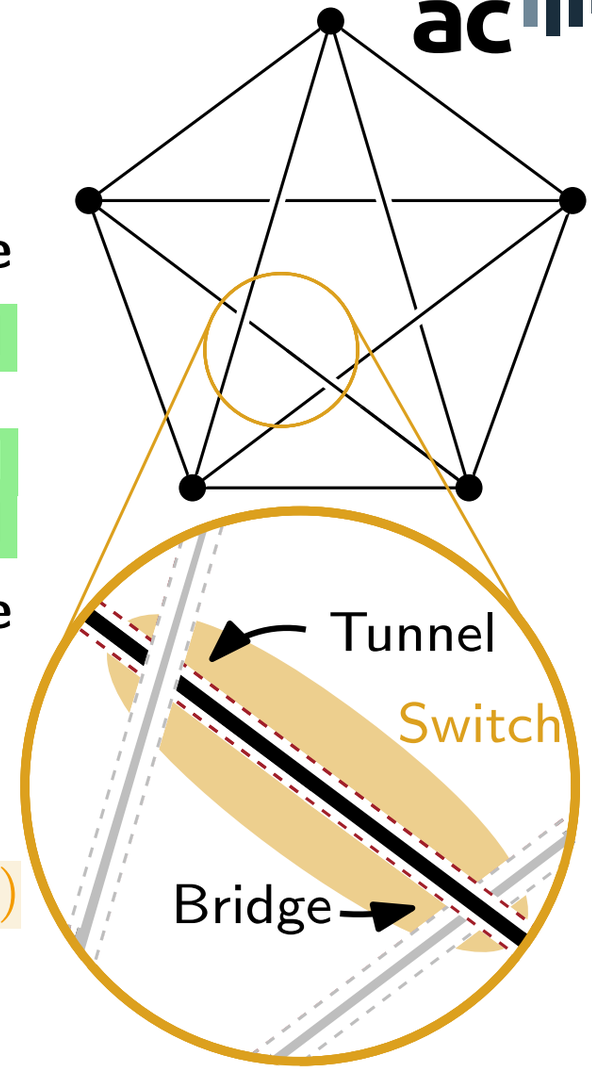






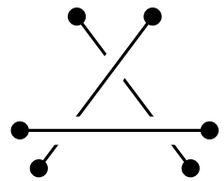
Hardness	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	$O(m^4)$	NP-hard
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

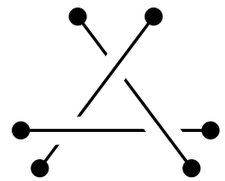
Tractability	**Stacking**	**Weaving**	**Realizable**
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	VC(χ_G)	*open*
MinSwitchEdges	tw(K_{δ_G})	tw(K_{δ_G})	tw(K_{δ_G})

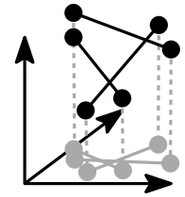


- k-switch-planar graphs
- MILP implementation

Outlook





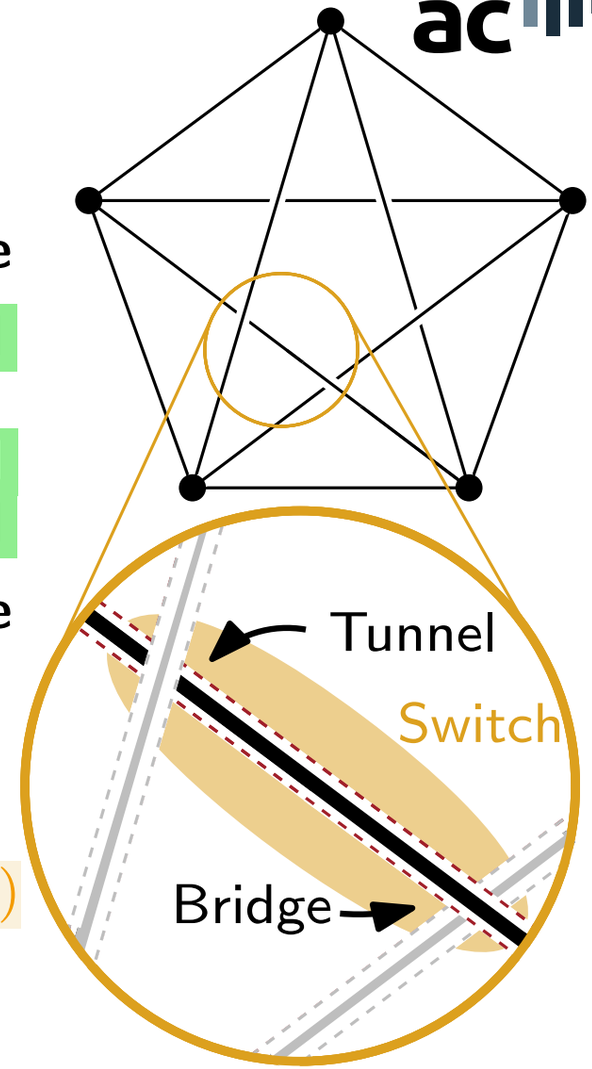


Hardness

	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	$O(m^4)$	NP-hard
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

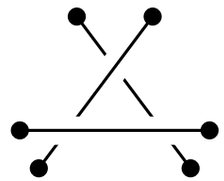
Tractability

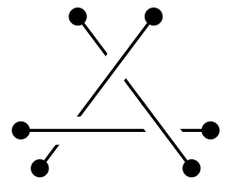
	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	VC(χ_G)	*open*
MinSwitchEdges	tw(K_{δ_G})	tw(K_{δ_G})	tw(K_{δ_G})

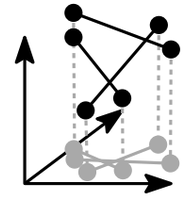


- k-switch-planar graphs
- MILP implementation

Outlook





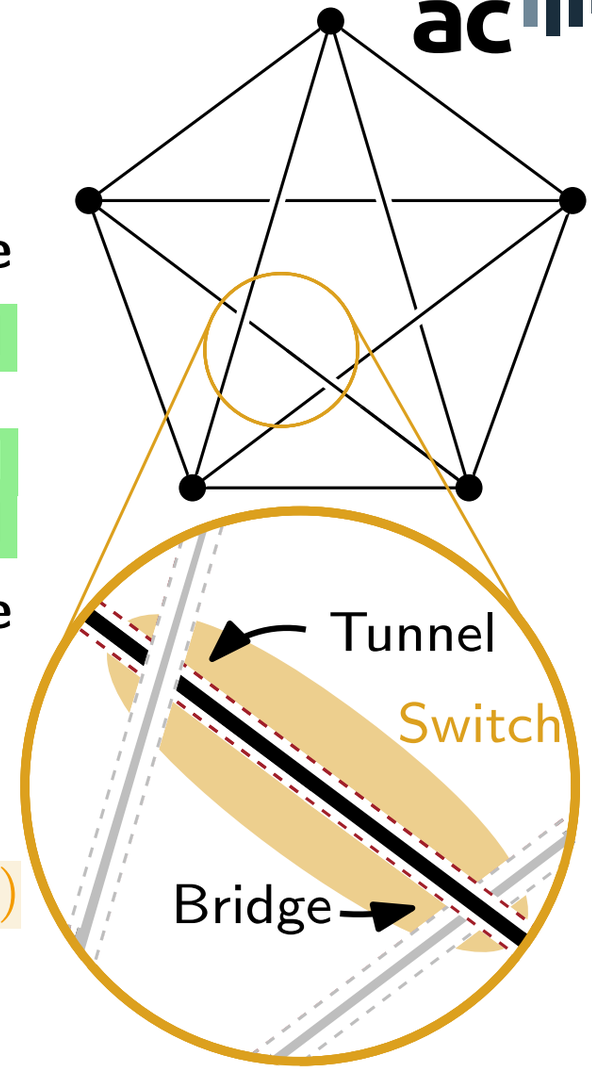


Hardness

	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	$O(m^4)$	NP-hard
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

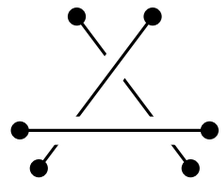
Tractability

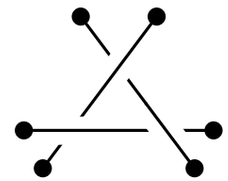
	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	VC(χ_G)	*open*
MinSwitchEdges	tw(K_{δ_G})	tw(K_{δ_G})	tw(K_{δ_G})

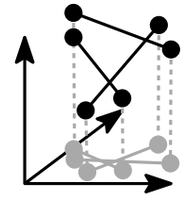


- k-switch-planar graphs Full characterization?
- MILP implementation

Outlook





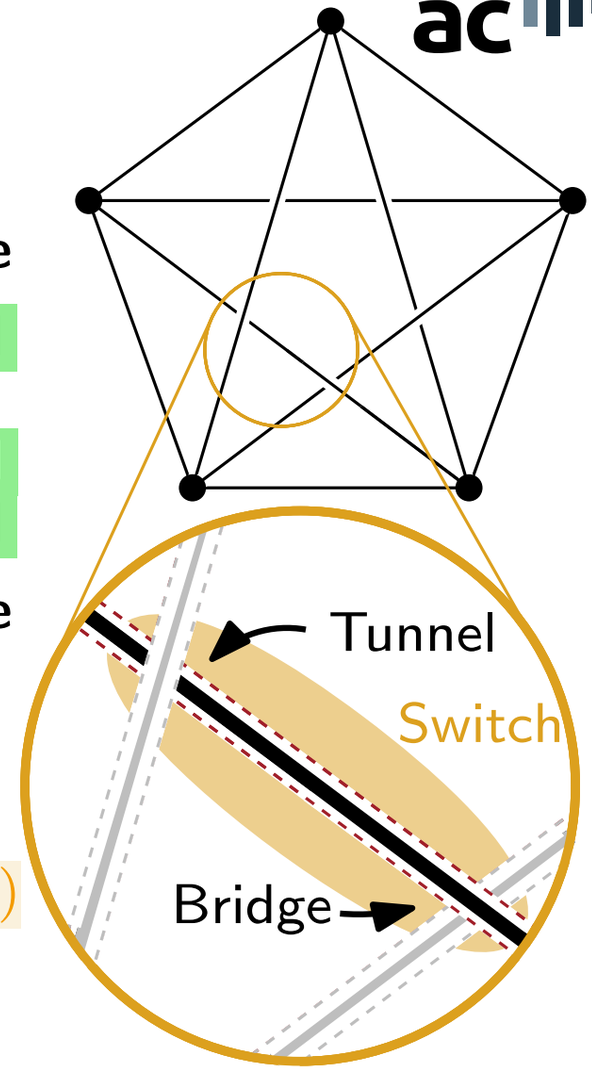


Hardness

	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	$O(m^4)$	NP-hard
MaxTotalSwitches	*open*	$O(m^4)$	*open*
MinMaxSwitches	*open*	NP-hard	NP-hard
MinSwitchEdges	NP-hard	NP-hard	NP-hard

Tractability

	Stacking	Weaving	Realizable
MinTotalSwitches	*open*	*n/a*	*open*
MaxTotalSwitches	*open*	*n/a*	*open*
MinMaxSwitches	*open*	VC(χ_G)	*open*
MinSwitchEdges	tw(K_{δ_G})	tw(K_{δ_G})	tw(K_{δ_G})



- k-switch-planar graphs — Full characterization?
- MILP implementation — Basis for user studies?