

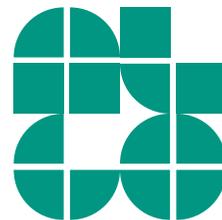
Algorithmen zur Visualisierung von Graphen

Übung 6: Knickminimierung und orthogonales Zeichnen

INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Tamara Mchedlidze · **Benjamin Niedermann**

24.01.2017

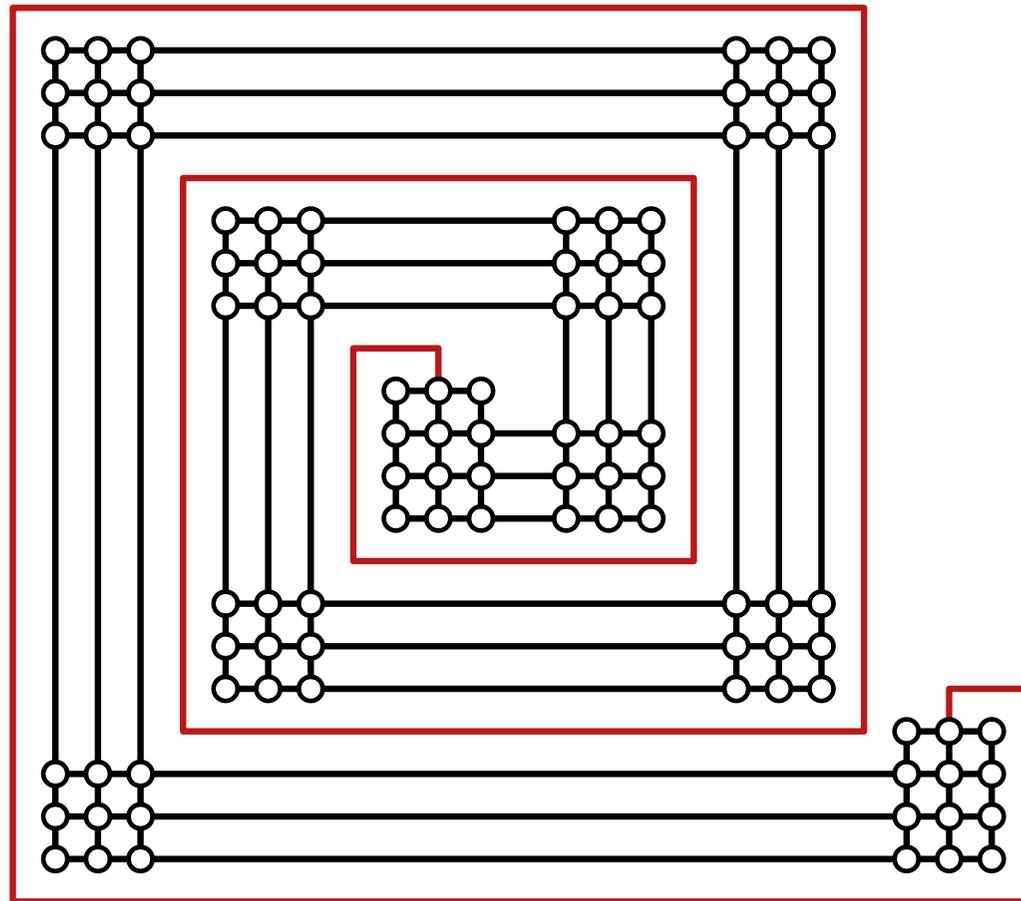


Aufgabe 1 – Fürchterlich viele Knicke

Konstruieren Sie eine Familie von eingebetteten planaren Graphen mit Maximalgrad 4 und mit $\mathcal{O}(n)$ Knoten, für die es in jeder knickminimalen einbettungserhaltenden orthogonalen Zeichnung eine Kante mit $\Omega(n)$ Knicken gibt.

Aufgabe 1 – Fürchterlich viele Knicke

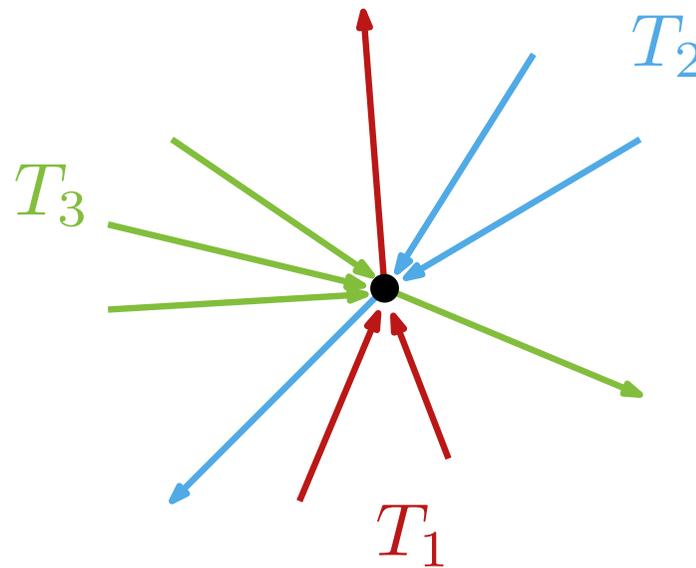
Konstruieren Sie eine Familie von eingebetteten planaren Graphen mit Maximalgrad 4 und mit $\mathcal{O}(n)$ Knoten, für die es in jeder knickminimalen einbettungserhaltenden orthogonalen Zeichnung eine Kante mit $\Omega(n)$ Knicken gibt.



Schnyder Realizer

Sei G ein triangulierter planarer Graph. Ein **Schnyder Realizer** partitioniert die internen Kanten in drei Mengen T_1 , T_2 , T_3 von gerichteten Kanten, so dass

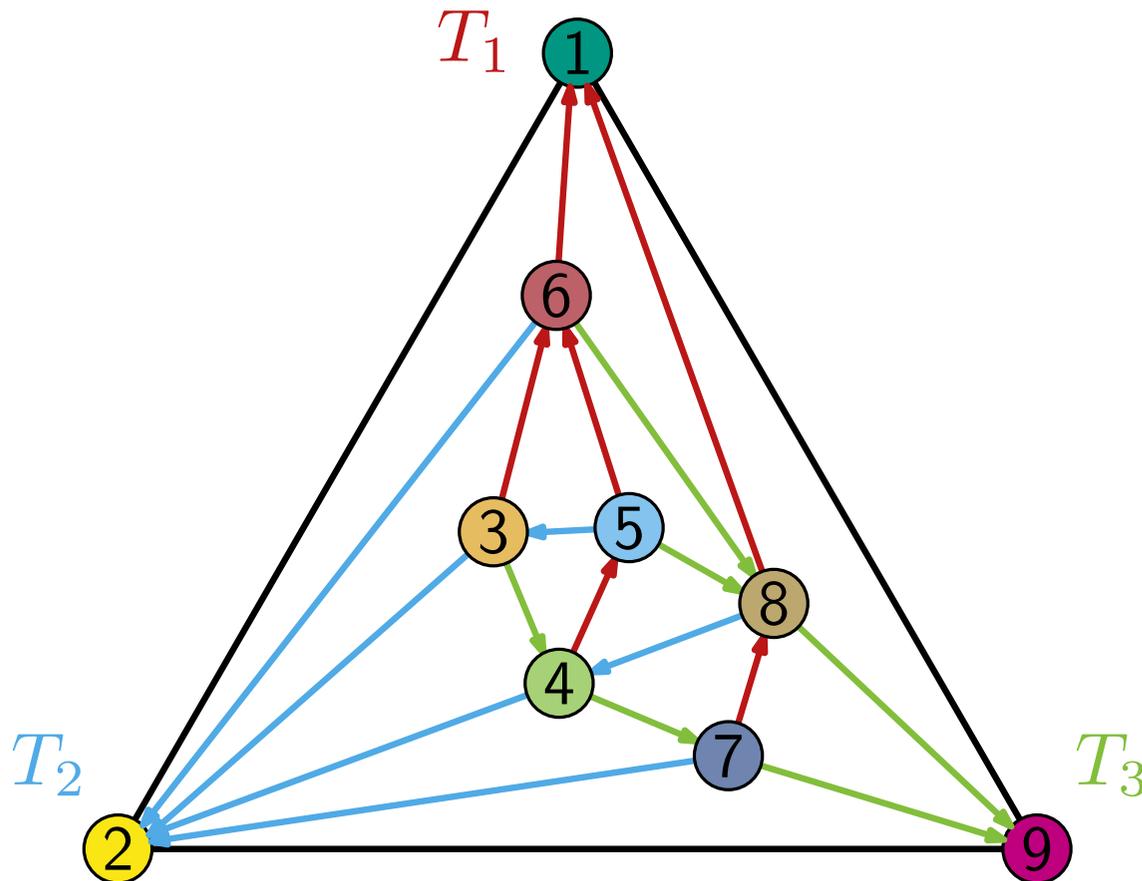
- jeder innere Knoten v hat genau eine Kante in jedem T_i^{out}
- die Ordnung der Kanten um jeden Knoten v im GUZS ist T_1^{in} , T_3^{out} , T_2^{in} , T_1^{out} , T_3^{in} , T_2^{out}



Schnyder Realizer

Satz: Jede Menge T_i ($i = 1, 2, 3$) ist ein Spannbaum aller inneren Knoten und eines äußeren Knotens.

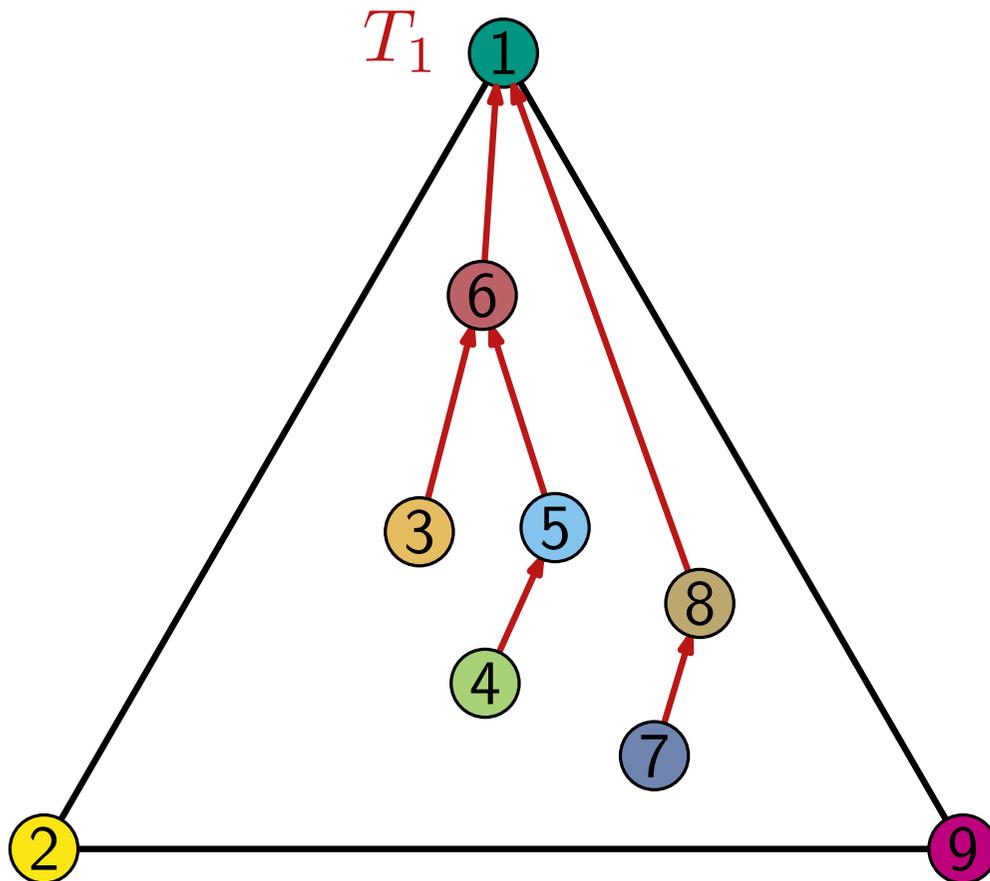
Jeder triangulierte Graph besitzt einen Schnyder Realizer und dieser kann in $O(n)$ Zeit berechnet werden.



Schnyder Realizer

Satz: Jede Menge T_i ($i = 1, 2, 3$) ist ein Spannbaum aller inneren Knoten und eines äußeren Knotens.

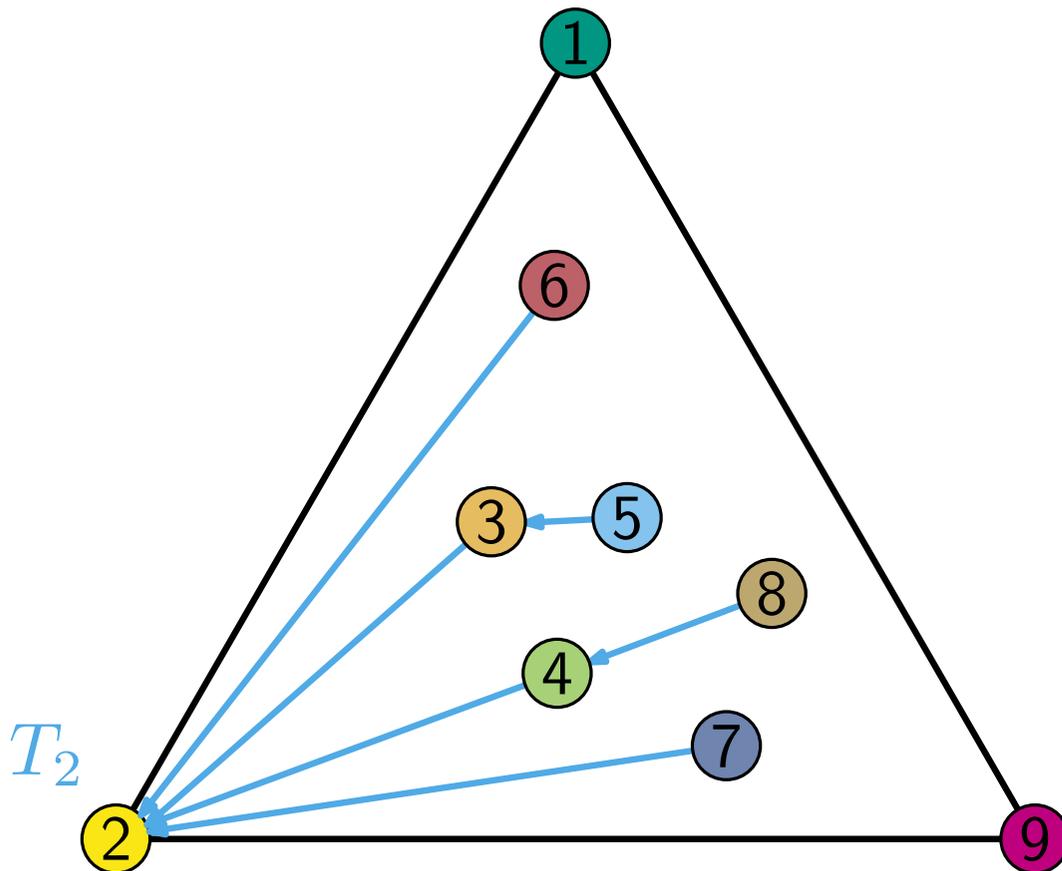
Jeder triangulierte Graph besitzt einen Schnyder Realizer und dieser kann in $O(n)$ Zeit berechnet werden.



Schnyder Realizer

Satz: Jede Menge T_i ($i = 1, 2, 3$) ist ein Spannbaum aller inneren Knoten und eines äußeren Knotens.

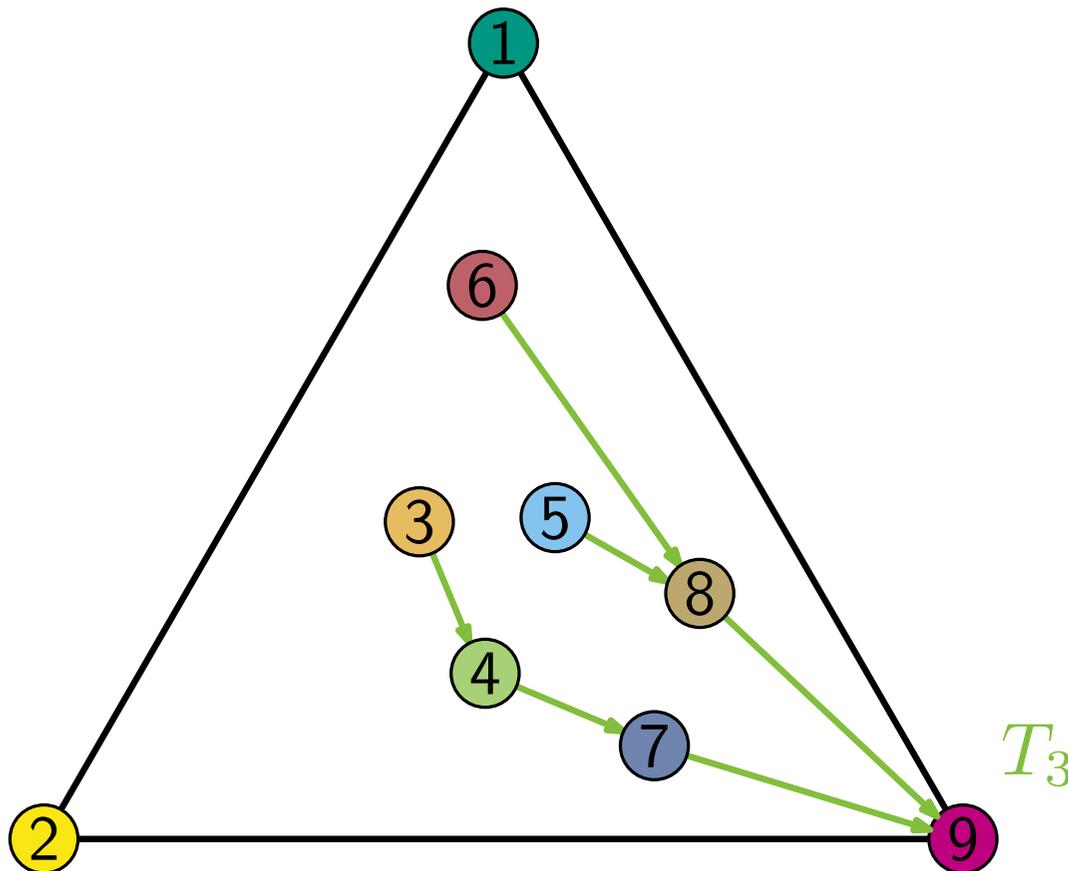
Jeder triangulierte Graph besitzt einen Schnyder Realizer und dieser kann in $O(n)$ Zeit berechnet werden.



Schnyder Realizer

Satz: Jede Menge T_i ($i = 1, 2, 3$) ist ein Spannbaum aller inneren Knoten und eines äußeren Knotens.

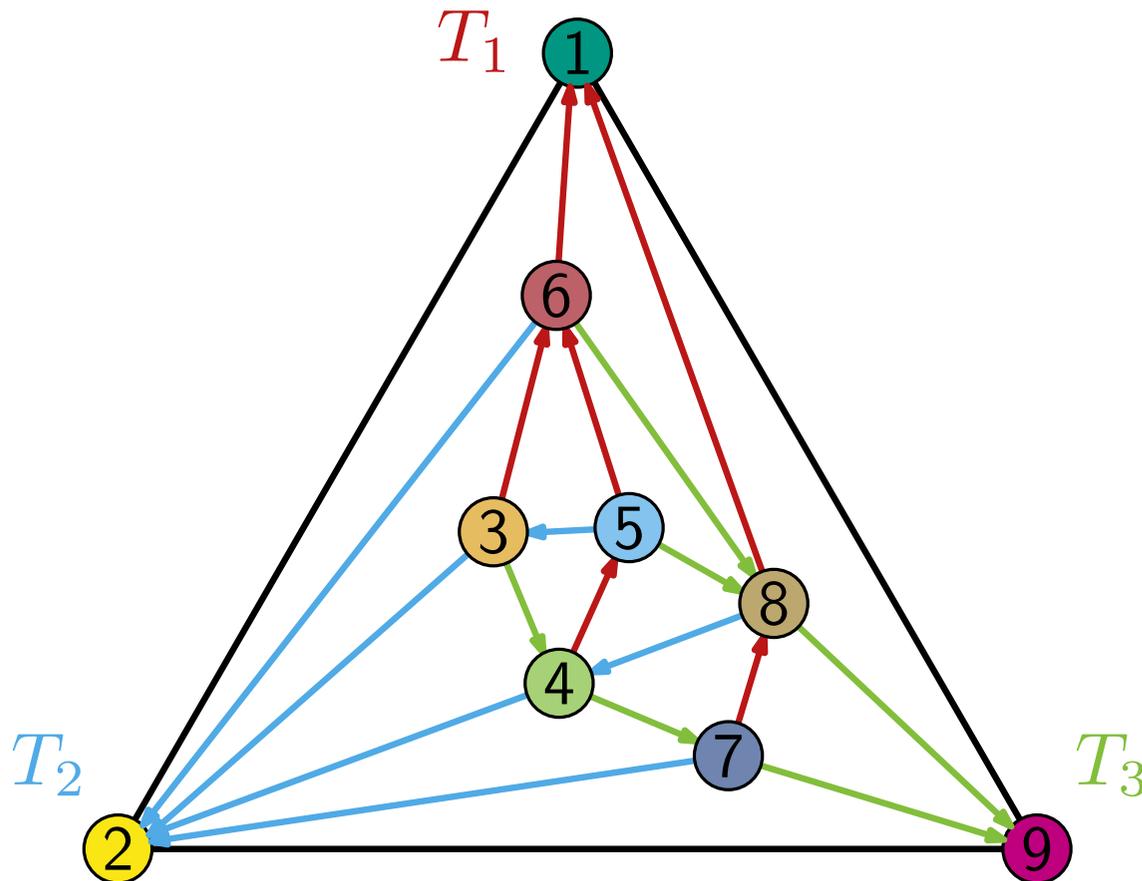
Jeder triangulierte Graph besitzt einen Schnyder Realizer und dieser kann in $O(n)$ Zeit berechnet werden.



Schnyder Realizer

Satz: Jede Menge T_i ($i = 1, 2, 3$) ist ein Spannbaum aller inneren Knoten und eines äußeren Knotens.

Jeder triangulierte Graph besitzt einen Schnyder Realizer und dieser kann in $O(n)$ Zeit berechnet werden.



Kanonische Ordnung

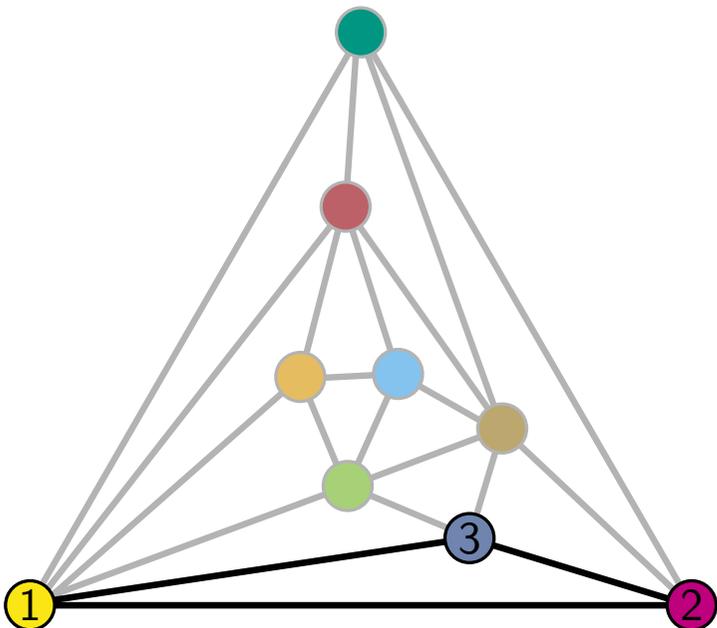
Die **kanonische Ordnung** der Knoten eines triangulierten planaren Graphen G mit den drei äußeren Knoten u, v, w im GUZS ist eine Ordnung $v_1 = u, v_2 = v, v_3, \dots, v_n = w$, so dass für jedes i ($4 \leq i \leq n$) gilt:

- Graph $G_{i-1} \subseteq G$ induziert durch v_1, \dots, v_{i-1} ist 2-fach zshgd. und äußere Facette ist begrenzt durch Kreis C_{i-1} , der v_1v_2 enthält
- Knoten v_i liegt in der äußeren Facette von G_{i-1} und seine Nachbarn in G_{i-1} bilden zusammenhängendes Intervall im Pfad $C_{i-1} \setminus v_1v_2$

Kanonische Ordnung

Die **kanonische Ordnung** der Knoten eines triangulierten planaren Graphen G mit den drei äußeren Knoten u, v, w im GUZS ist eine Ordnung $v_1 = u, v_2 = v, v_3, \dots, v_n = w$, so dass für jedes i ($4 \leq i \leq n$) gilt:

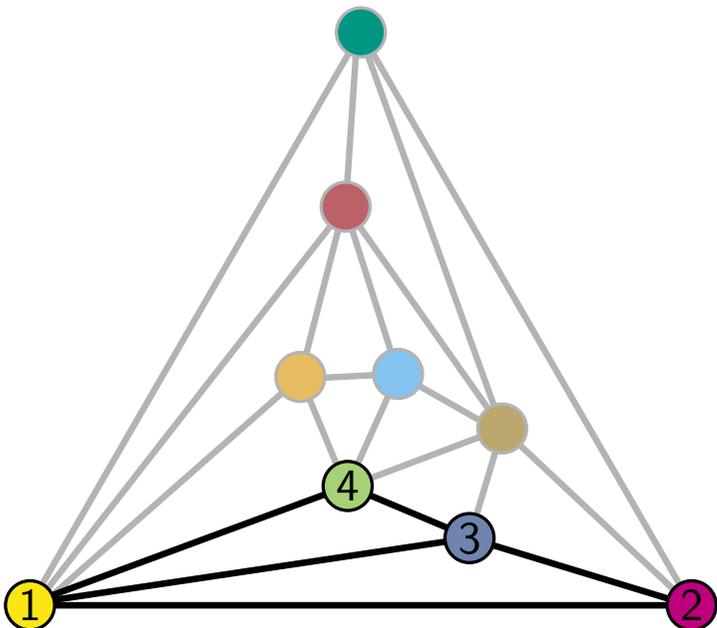
- Graph $G_{i-1} \subseteq G$ induziert durch v_1, \dots, v_{i-1} ist 2-fach zshgd. und äußere Facette ist begrenzt durch Kreis C_{i-1} , der v_1v_2 enthält
- Knoten v_i liegt in der äußeren Facette von G_{i-1} und seine Nachbarn in G_{i-1} bilden zusammenhängendes Intervall im Pfad $C_{i-1} \setminus v_1v_2$



Kanonische Ordnung

Die **kanonische Ordnung** der Knoten eines triangulierten planaren Graphen G mit den drei äußeren Knoten u, v, w im GUZS ist eine Ordnung $v_1 = u, v_2 = v, v_3, \dots, v_n = w$, so dass für jedes i ($4 \leq i \leq n$) gilt:

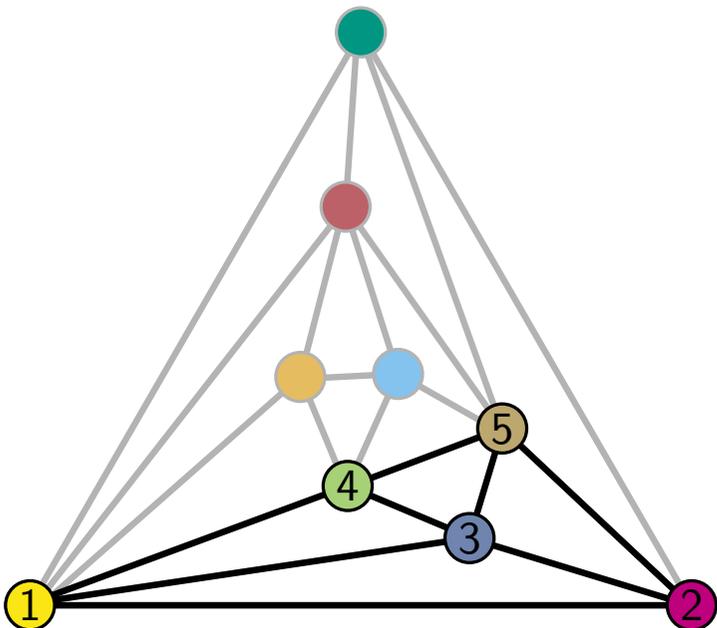
- Graph $G_{i-1} \subseteq G$ induziert durch v_1, \dots, v_{i-1} ist 2-fach zshgd. und äußere Facette ist begrenzt durch Kreis C_{i-1} , der v_1v_2 enthält
- Knoten v_i liegt in der äußeren Facette von G_{i-1} und seine Nachbarn in G_{i-1} bilden zusammenhängendes Intervall im Pfad $C_{i-1} \setminus v_1v_2$



Kanonische Ordnung

Die **kanonische Ordnung** der Knoten eines triangulierten planaren Graphen G mit den drei äußeren Knoten u, v, w im GUZS ist eine Ordnung $v_1 = u, v_2 = v, v_3, \dots, v_n = w$, so dass für jedes i ($4 \leq i \leq n$) gilt:

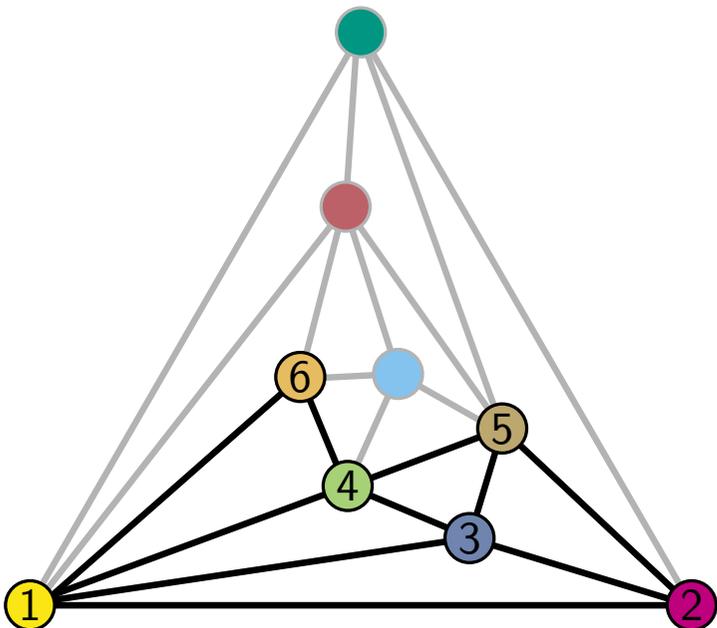
- Graph $G_{i-1} \subseteq G$ induziert durch v_1, \dots, v_{i-1} ist 2-fach zshgd. und äußere Facette ist begrenzt durch Kreis C_{i-1} , der v_1v_2 enthält
- Knoten v_i liegt in der äußeren Facette von G_{i-1} und seine Nachbarn in G_{i-1} bilden zusammenhängendes Intervall im Pfad $C_{i-1} \setminus v_1v_2$



Kanonische Ordnung

Die **kanonische Ordnung** der Knoten eines triangulierten planaren Graphen G mit den drei äußeren Knoten u, v, w im GUZS ist eine Ordnung $v_1 = u, v_2 = v, v_3, \dots, v_n = w$, so dass für jedes i ($4 \leq i \leq n$) gilt:

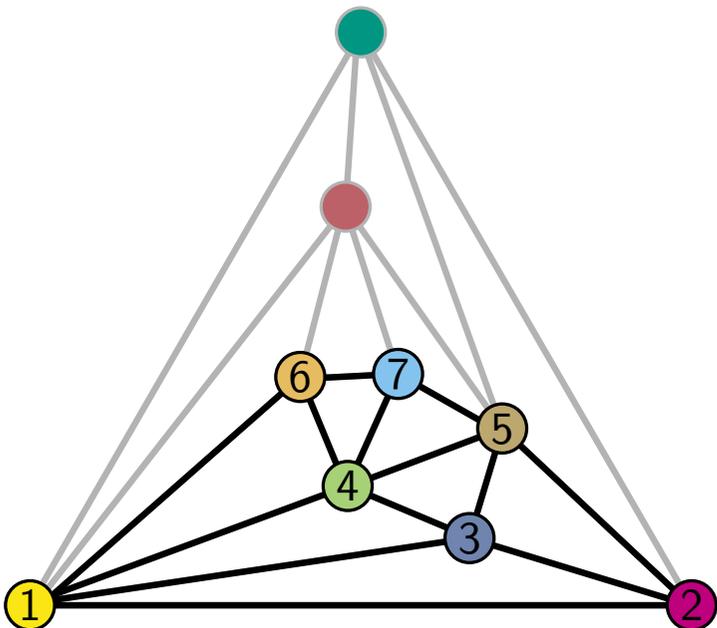
- Graph $G_{i-1} \subseteq G$ induziert durch v_1, \dots, v_{i-1} ist 2-fach zshgd. und äußere Facette ist begrenzt durch Kreis C_{i-1} , der v_1v_2 enthält
- Knoten v_i liegt in der äußeren Facette von G_{i-1} und seine Nachbarn in G_{i-1} bilden zusammenhängendes Intervall im Pfad $C_{i-1} \setminus v_1v_2$



Kanonische Ordnung

Die **kanonische Ordnung** der Knoten eines triangulierten planaren Graphen G mit den drei äußeren Knoten u, v, w im GUZS ist eine Ordnung $v_1 = u, v_2 = v, v_3, \dots, v_n = w$, so dass für jedes i ($4 \leq i \leq n$) gilt:

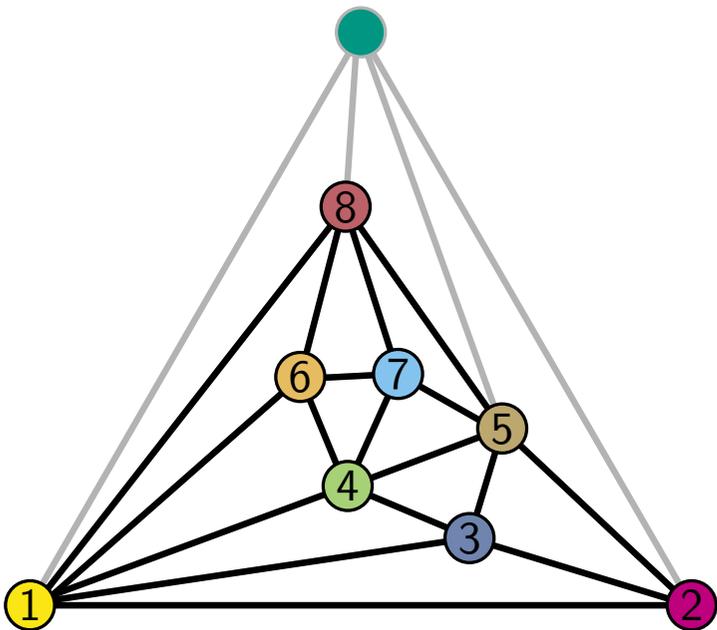
- Graph $G_{i-1} \subseteq G$ induziert durch v_1, \dots, v_{i-1} ist 2-fach zshgd. und äußere Facette ist begrenzt durch Kreis C_{i-1} , der v_1v_2 enthält
- Knoten v_i liegt in der äußeren Facette von G_{i-1} und seine Nachbarn in G_{i-1} bilden zusammenhängendes Intervall im Pfad $C_{i-1} \setminus v_1v_2$



Kanonische Ordnung

Die **kanonische Ordnung** der Knoten eines triangulierten planaren Graphen G mit den drei äußeren Knoten u, v, w im GUZS ist eine Ordnung $v_1 = u, v_2 = v, v_3, \dots, v_n = w$, so dass für jedes i ($4 \leq i \leq n$) gilt:

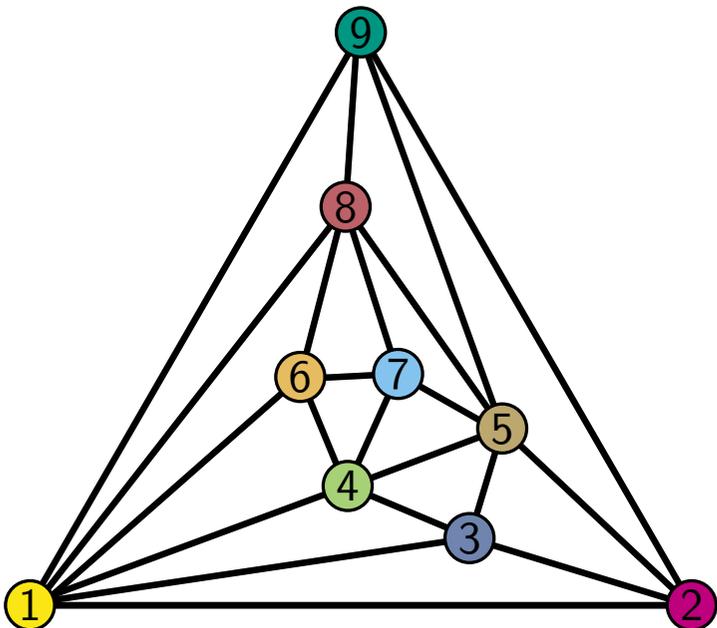
- Graph $G_{i-1} \subseteq G$ induziert durch v_1, \dots, v_{i-1} ist 2-fach zshgd. und äußere Facette ist begrenzt durch Kreis C_{i-1} , der v_1v_2 enthält
- Knoten v_i liegt in der äußeren Facette von G_{i-1} und seine Nachbarn in G_{i-1} bilden zusammenhängendes Intervall im Pfad $C_{i-1} \setminus v_1v_2$



Kanonische Ordnung

Die **kanonische Ordnung** der Knoten eines triangulierten planaren Graphen G mit den drei äußeren Knoten u, v, w im GUZS ist eine Ordnung $v_1 = u, v_2 = v, v_3, \dots, v_n = w$, so dass für jedes i ($4 \leq i \leq n$) gilt:

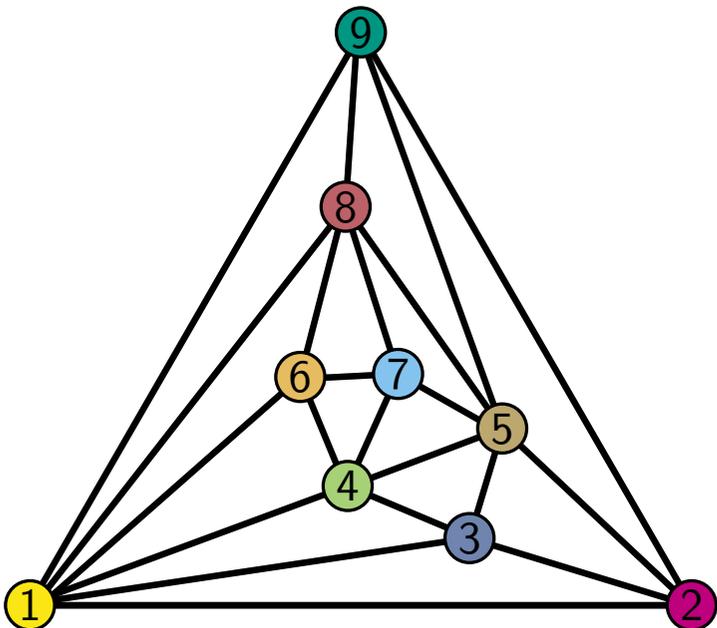
- Graph $G_{i-1} \subseteq G$ induziert durch v_1, \dots, v_{i-1} ist 2-fach zshgd. und äußere Facette ist begrenzt durch Kreis C_{i-1} , der v_1v_2 enthält
- Knoten v_i liegt in der äußeren Facette von G_{i-1} und seine Nachbarn in G_{i-1} bilden zusammenhängendes Intervall im Pfad $C_{i-1} \setminus v_1v_2$



Kanonische Ordnung

Die **kanonische Ordnung** der Knoten eines triangulierten planaren Graphen G mit den drei äußeren Knoten u, v, w im GUZS ist eine Ordnung $v_1 = u, v_2 = v, v_3, \dots, v_n = w$, so dass für jedes i ($4 \leq i \leq n$) gilt:

- Graph $G_{i-1} \subseteq G$ induziert durch v_1, \dots, v_{i-1} ist 2-fach zshgd. und äußere Facette ist begrenzt durch Kreis C_{i-1} , der v_1v_2 enthält
- Knoten v_i liegt in der äußeren Facette von G_{i-1} und seine Nachbarn in G_{i-1} bilden zusammenhängendes Intervall im Pfad $C_{i-1} \setminus v_1v_2$

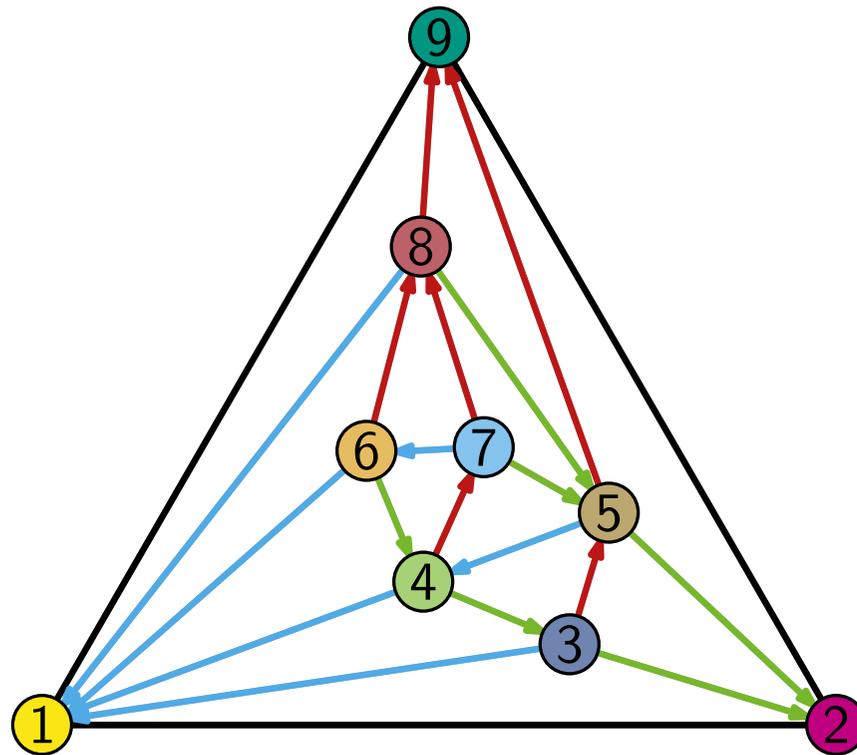


Satz: Jeder triangulierte Graph G besitzt eine kanonische Ordnung; sie kann in $O(n)$ Zeit berechnet werden.

→ mehr dazu in der VL Algorithmen zur Visualisierung von Graphen

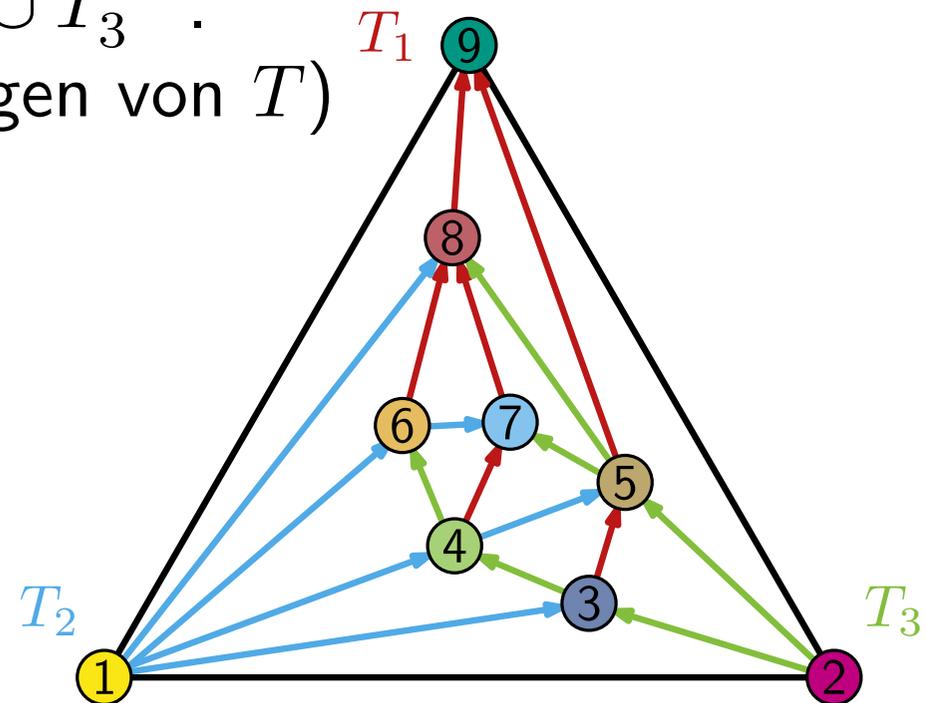
Kanonische Ordnung & Schnyder Realizer

- Eine kanonische Ordnung der Knoten von G definiert einen Schnyder Realizer von G , indem jedem Knoten v_i drei ausgehende Kanten zum ersten und letzten Knoten in C_{i-1} und zum Nachfolger mit höchstem Index zugeordnet werden.



Kanonische Ordnung & Schnyder Realizer

- Eine kanonische Ordnung der Knoten von G definiert einen Schnyder Realizer von G , indem jedem Knoten v_i drei ausgehende Kanten zum ersten und letzten Knoten in C_{i-1} und zum Nachfolger mit höchstem Index zugeordnet werden.
- Ein Schnyder Realizer mit Bäumen T_1, T_2, T_3 definiert eine kanonische Ordnung als topologische Ordnung des azyklischen Graphen $T_1 \cup T_2^{-1} \cup T_3^{-1}$.
(T^{-1} : invertiere Kantenrichtungen von T)



8-seitige rektilineare Kartogramme [Alam et al. '11]

Algorithmus hat drei Phasen:

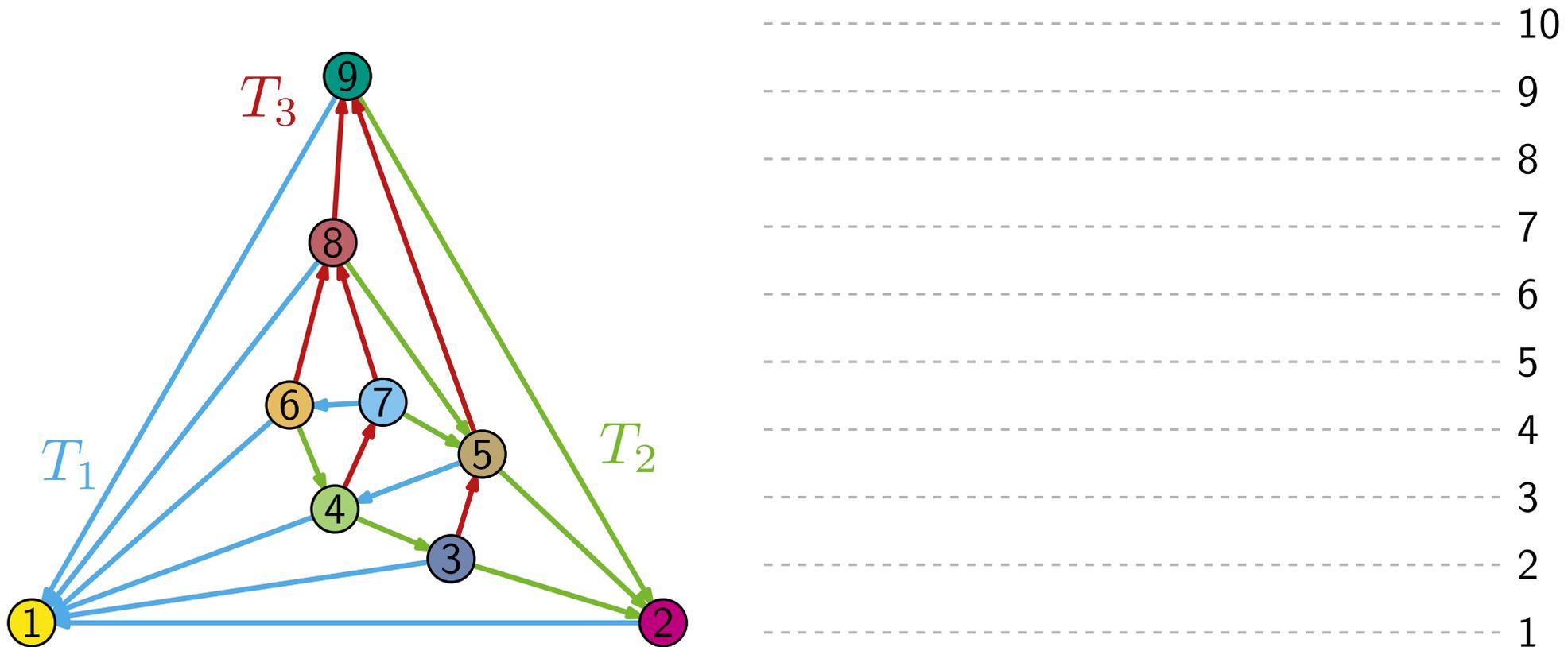
- erzeuge T-Kontaktrepräsentation
- wandle jedes T in T-förmiges Polygon um
- weise verbleibende Löcher den T-Polygonen zu

8-seitige rektilineare Kartogramme [Alam et al. '11]

Algorithmus hat drei Phasen:

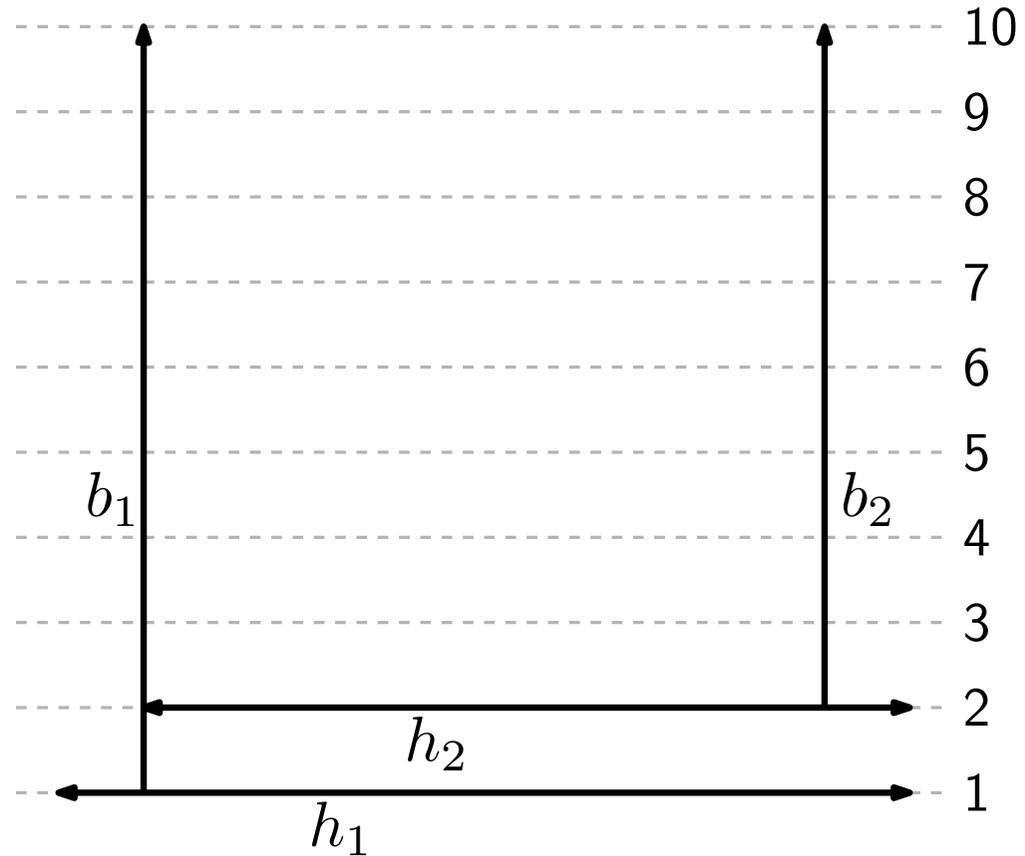
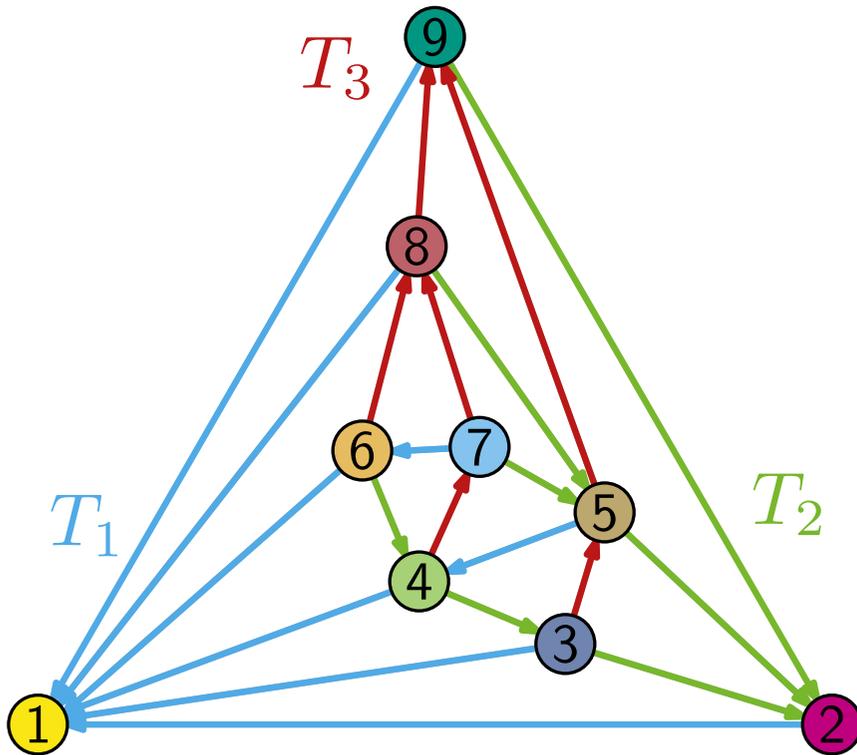
- erzeuge T-Kontaktrepräsentation
- wandle jedes T in T-förmiges Polygon um
- weise verbleibende Löcher den T-Polygonen zu

Phase 1: T-Kontaktrepräsentation



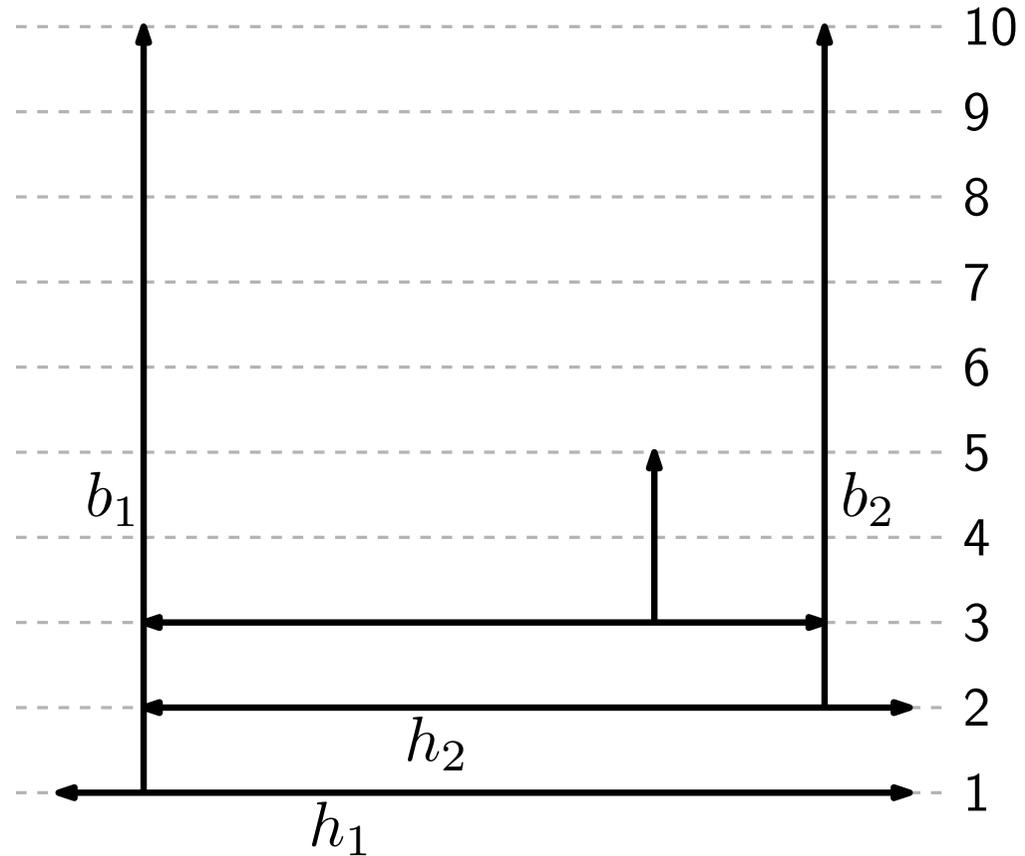
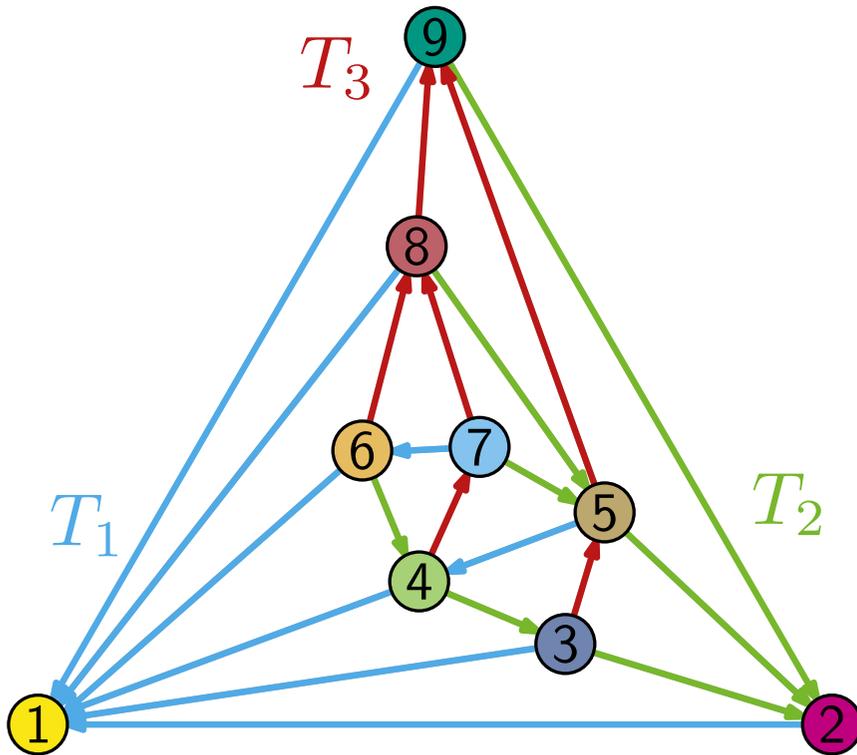
- betrachte Knoten in kanonischer Ordnung und zug. Schnyder Realizer
- $\Phi_i(k)$ sei Vater in T_i des Knotens v_k

Phase 1: T-Kontaktrepräsentation



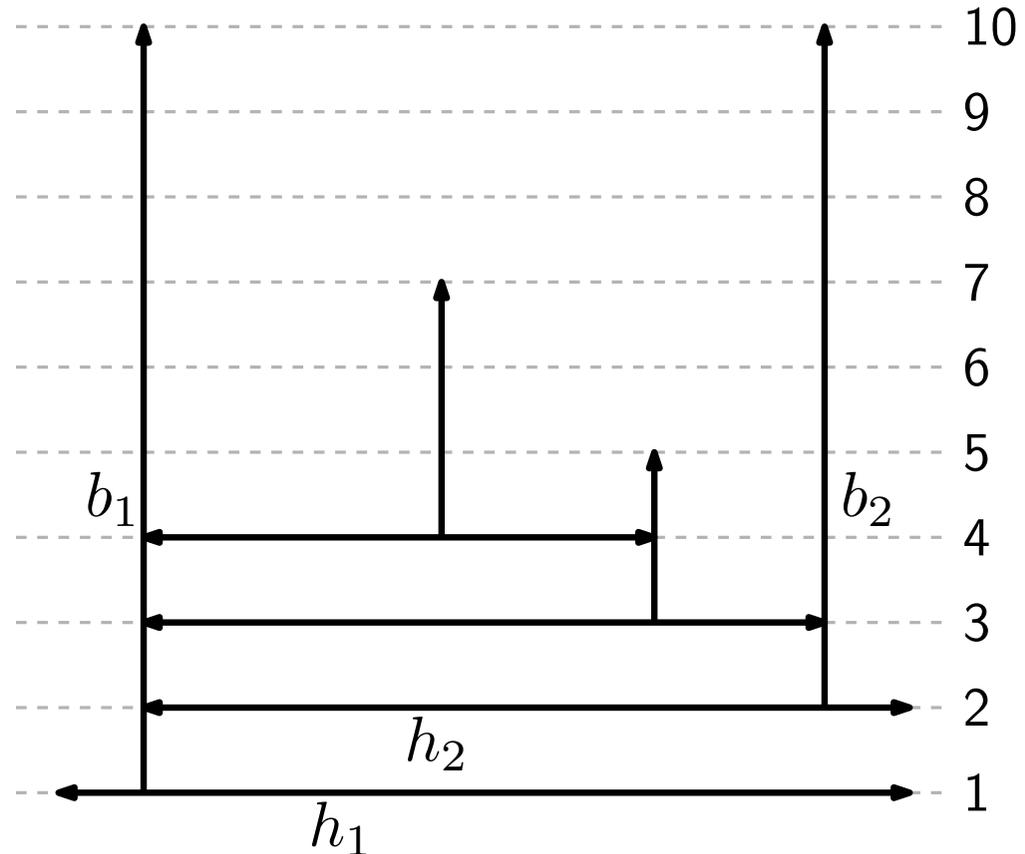
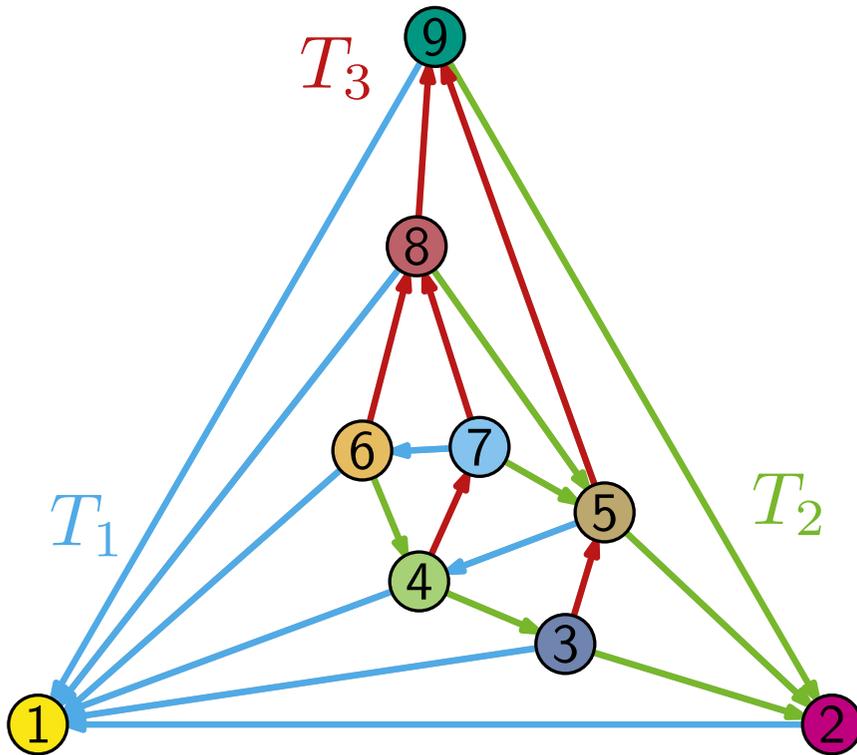
- betrachte Knoten in kanonischer Ordnung und zug. Schnyder Realizer
- $\Phi_i(k)$ sei Vater in T_i des Knotens v_k
- setze T's für v_1 und v_2 auf $y = 1$ und $y = 2$

Phase 1: T-Kontaktrepräsentation



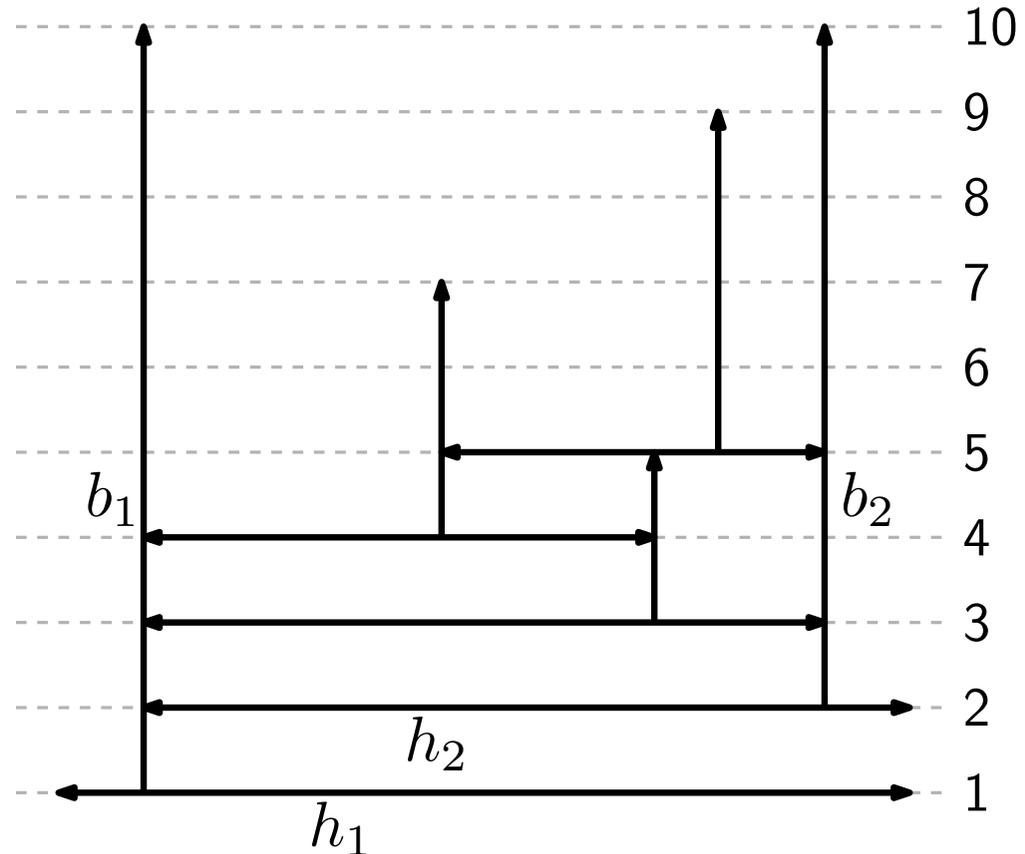
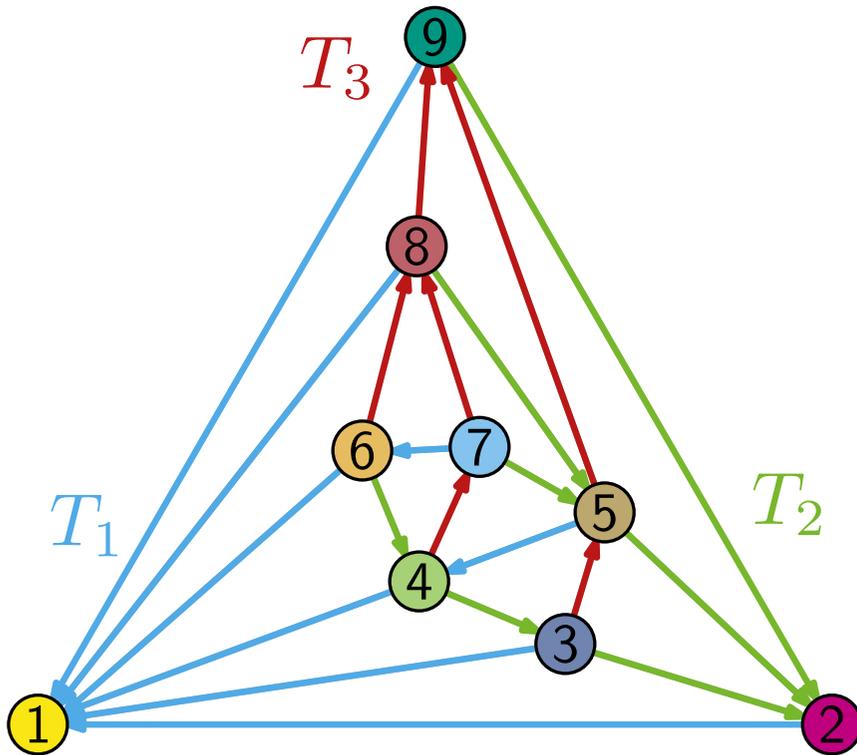
- betrachte Knoten in kanonischer Ordnung und zug. Schnyder Realizer
- $\Phi_i(k)$ sei Vater in T_i des Knotens v_k
- setze T 's für v_1 und v_2 auf $y = 1$ und $y = 2$
- setze h_i auf $y = i$ mit linker und rechter Position von $\Phi_1(i)$ und $\Phi_2(i)$ und b_i von $y = i$ bis zum Index von $\Phi_3(i)$

Phase 1: T-Kontaktrepräsentation



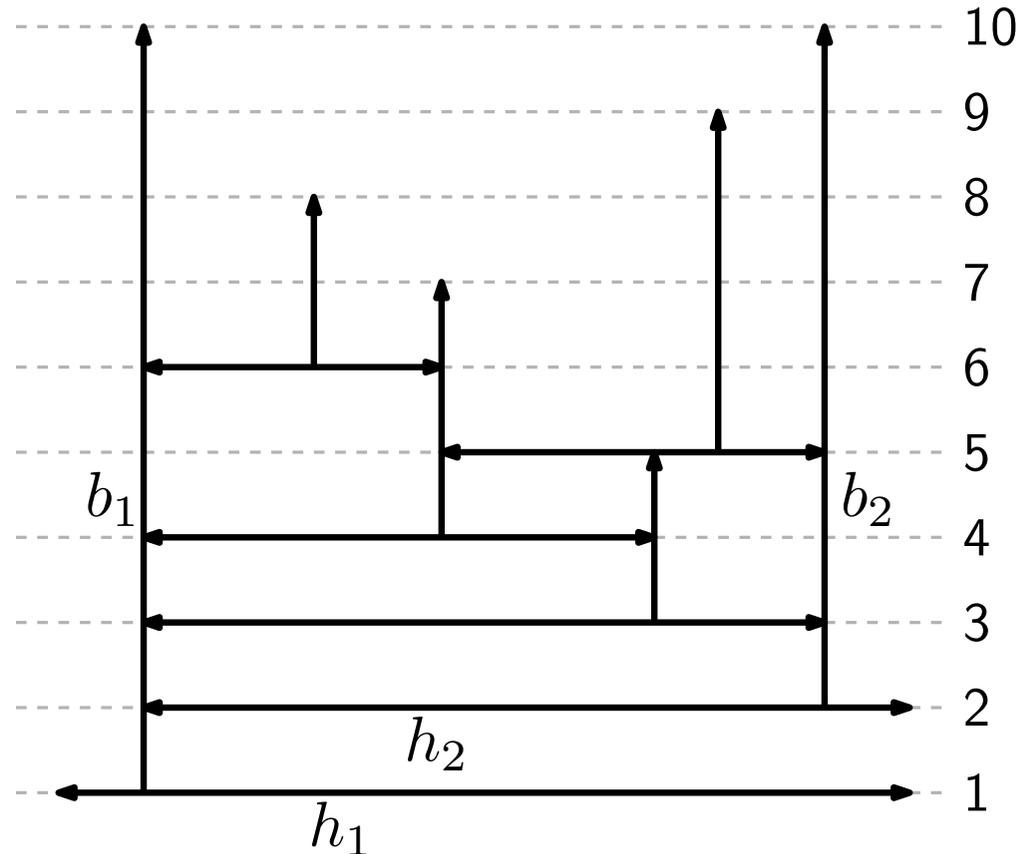
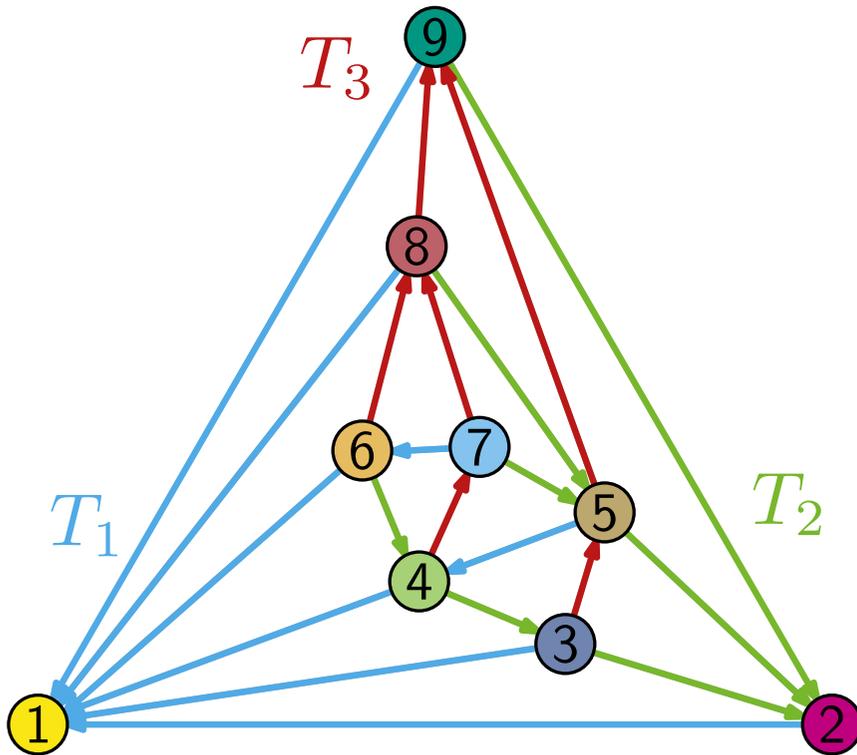
- betrachte Knoten in kanonischer Ordnung und zug. Schnyder Realizer
- $\Phi_i(k)$ sei Vater in T_i des Knotens v_k
- setze T 's für v_1 und v_2 auf $y = 1$ und $y = 2$
- setze h_i auf $y = i$ mit linker und rechter Position von $\Phi_1(i)$ und $\Phi_2(i)$ und b_i von $y = i$ bis zum Index von $\Phi_3(i)$

Phase 1: T-Kontaktrepräsentation



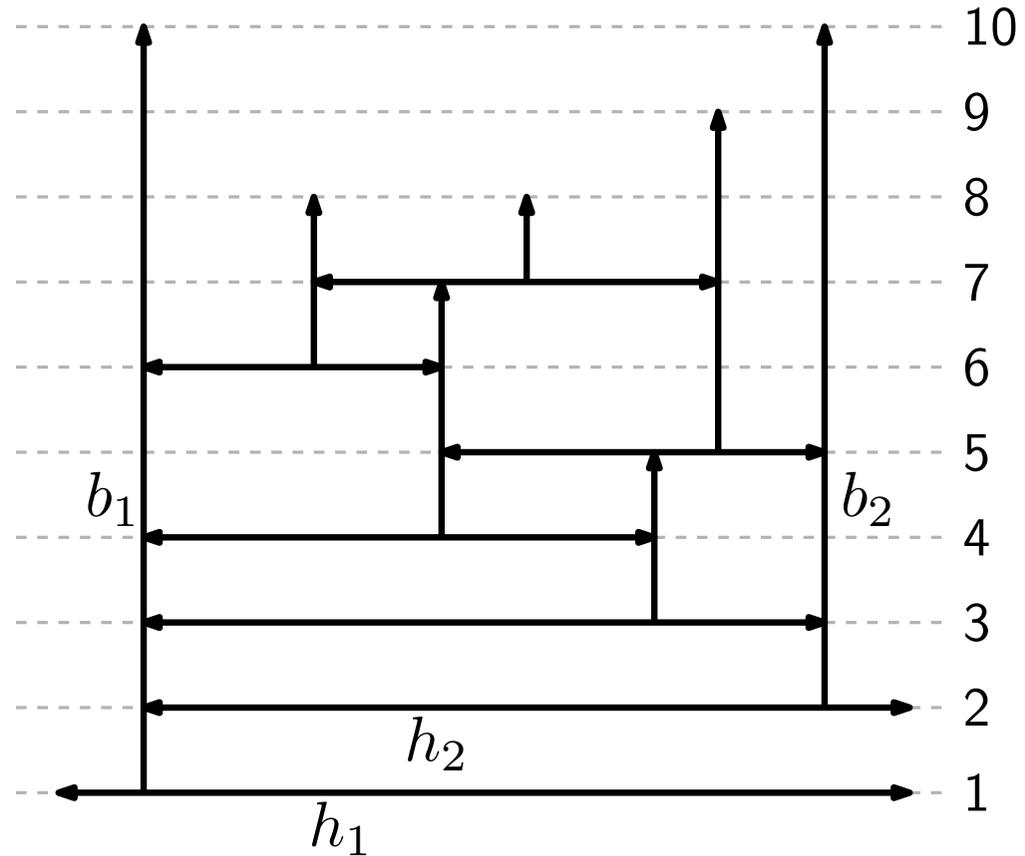
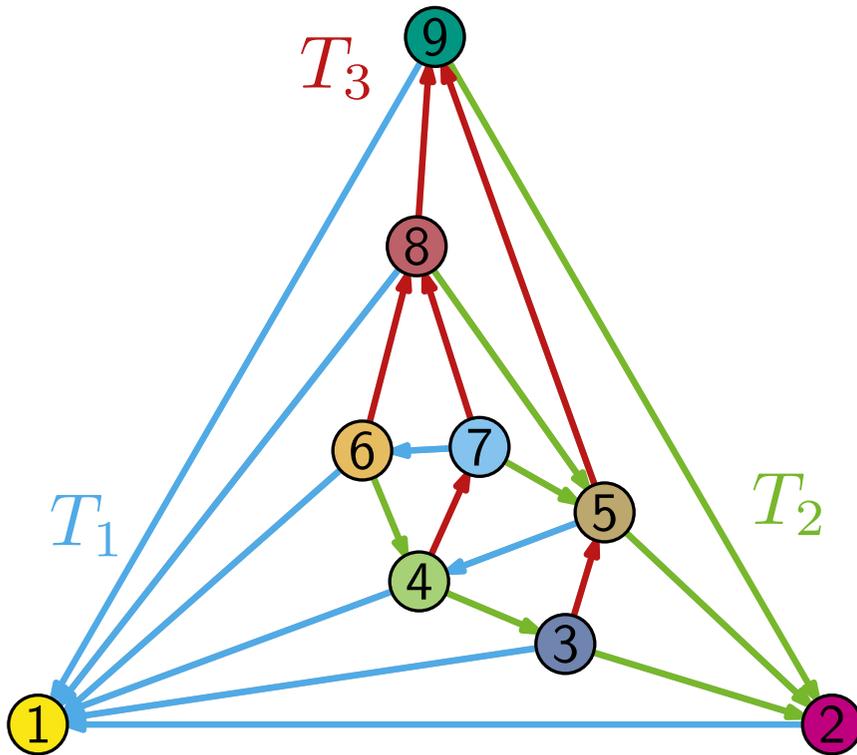
- betrachte Knoten in kanonischer Ordnung und zug. Schnyder Realizer
- $\Phi_i(k)$ sei Vater in T_i des Knotens v_k
- setze T 's für v_1 und v_2 auf $y = 1$ und $y = 2$
- setze h_i auf $y = i$ mit linker und rechter Position von $\Phi_1(i)$ und $\Phi_2(i)$ und b_i von $y = i$ bis zum Index von $\Phi_3(i)$

Phase 1: T-Kontaktrepräsentation



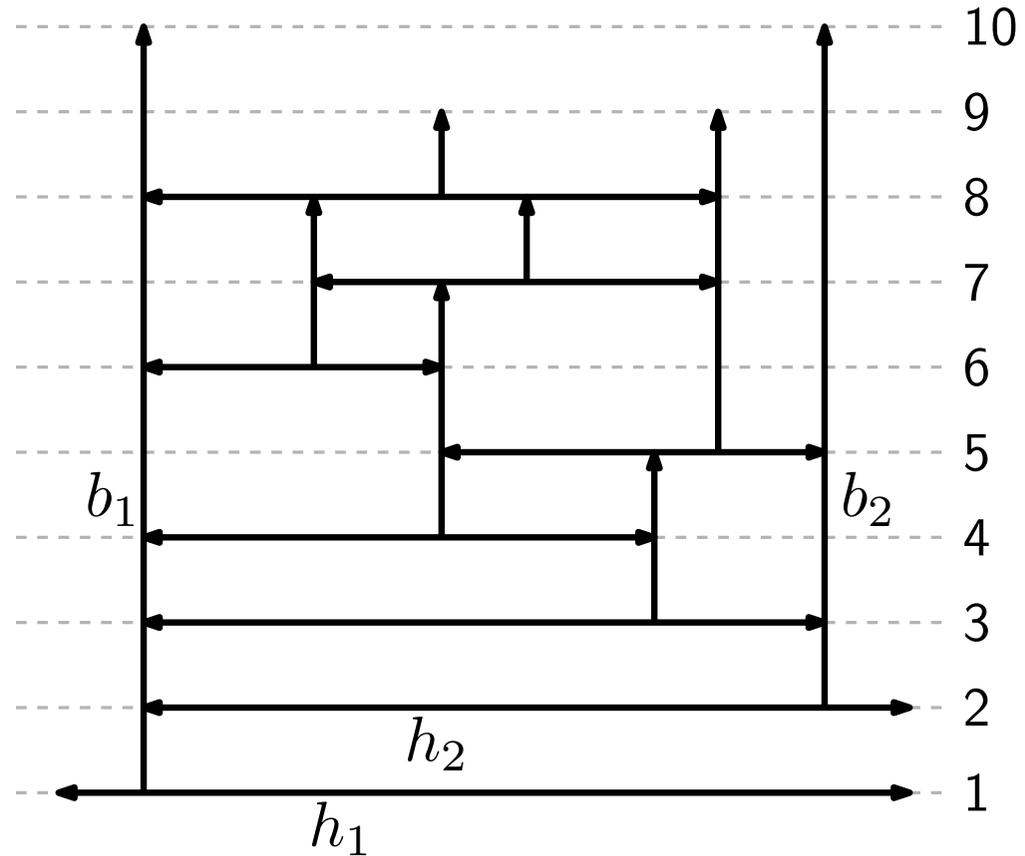
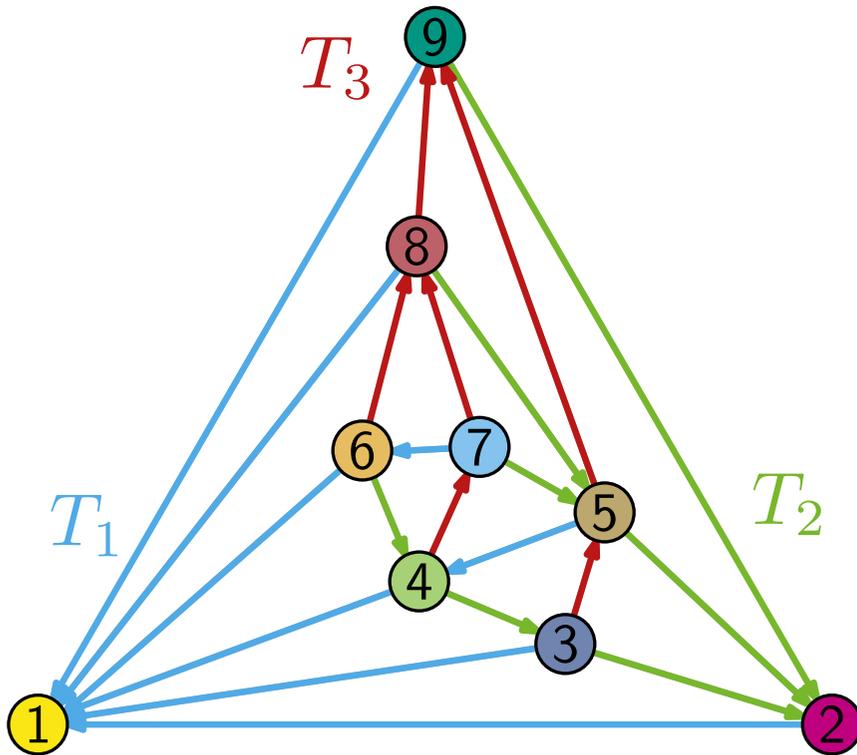
- betrachte Knoten in kanonischer Ordnung und zug. Schnyder Realizer
- $\Phi_i(k)$ sei Vater in T_i des Knotens v_k
- setze T 's für v_1 und v_2 auf $y = 1$ und $y = 2$
- setze h_i auf $y = i$ mit linker und rechter Position von $\Phi_1(i)$ und $\Phi_2(i)$ und b_i von $y = i$ bis zum Index von $\Phi_3(i)$

Phase 1: T-Kontaktrepräsentation



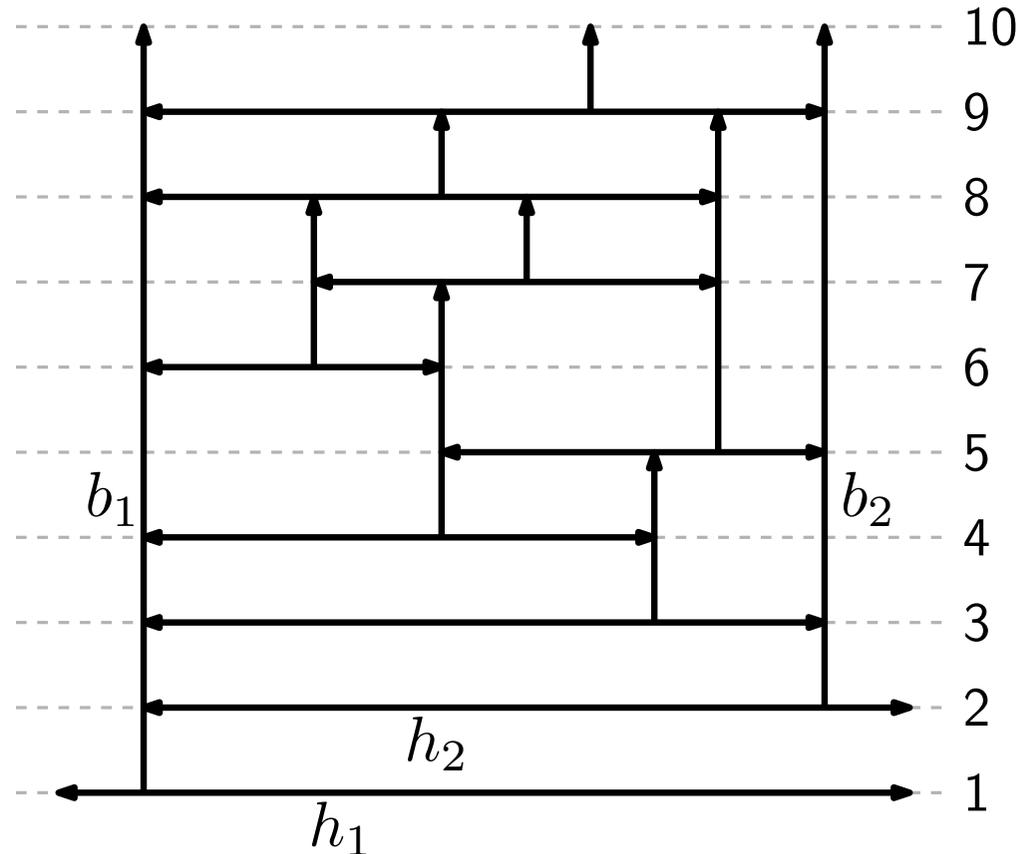
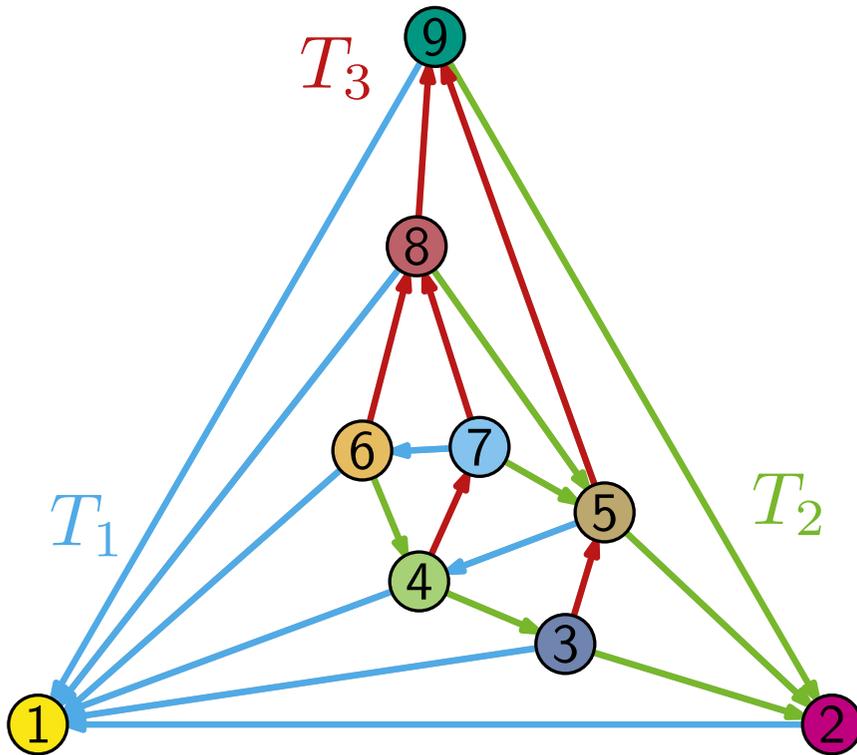
- betrachte Knoten in kanonischer Ordnung und zug. Schnyder Realizer
- $\Phi_i(k)$ sei Vater in T_i des Knotens v_k
- setze T 's für v_1 und v_2 auf $y = 1$ und $y = 2$
- setze h_i auf $y = i$ mit linker und rechter Position von $\Phi_1(i)$ und $\Phi_2(i)$ und b_i von $y = i$ bis zum Index von $\Phi_3(i)$

Phase 1: T-Kontaktrepräsentation



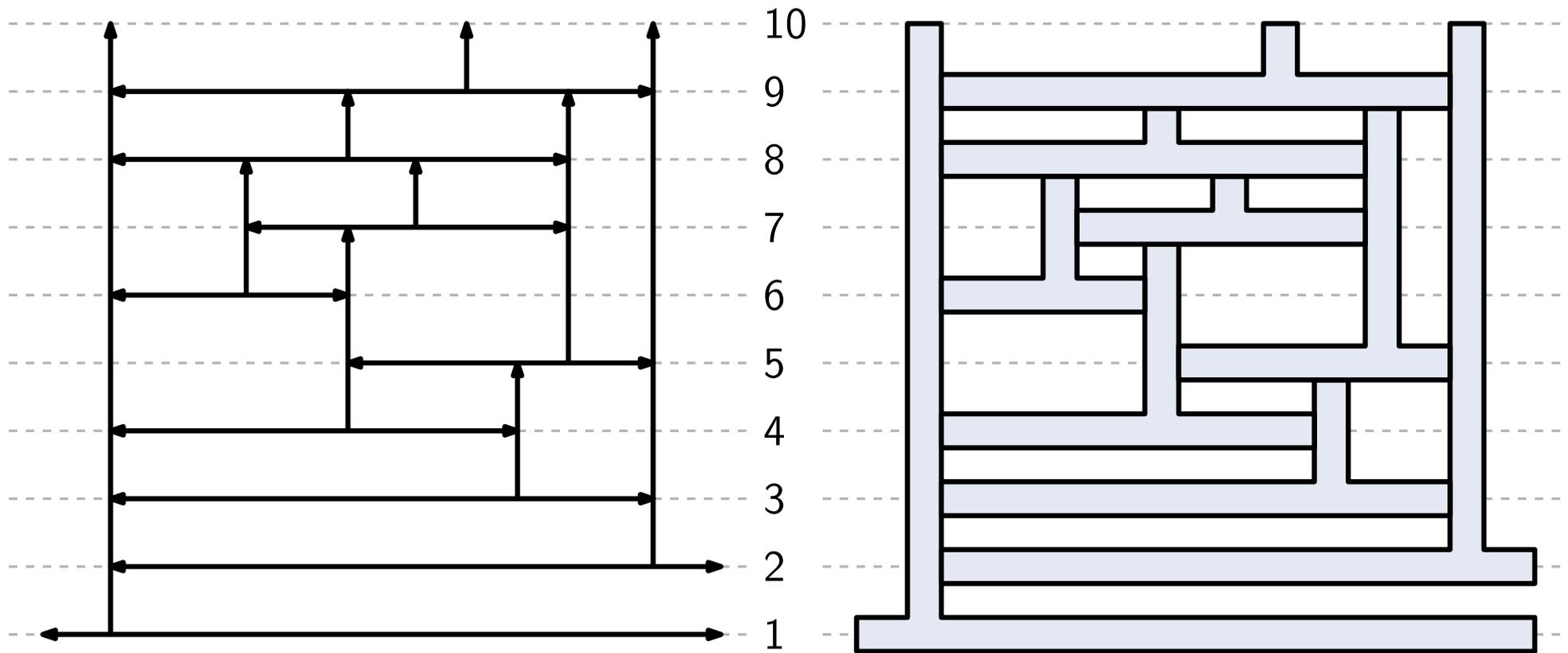
- betrachte Knoten in kanonischer Ordnung und zug. Schnyder Realizer
- $\Phi_i(k)$ sei Vater in T_i des Knotens v_k
- setze T's für v_1 und v_2 auf $y = 1$ und $y = 2$
- setze h_i auf $y = i$ mit linker und rechter Position von $\Phi_1(i)$ und $\Phi_2(i)$ und b_i von $y = i$ bis zum Index von $\Phi_3(i)$

Phase 1: T-Kontaktrepräsentation



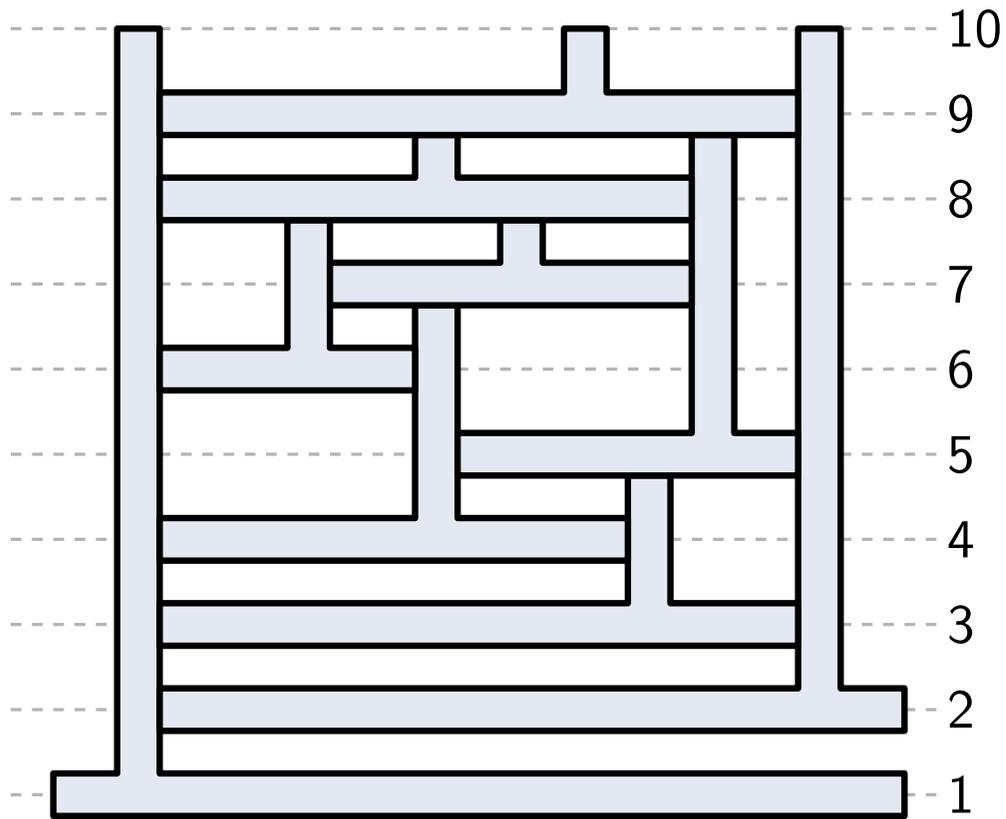
- betrachte Knoten in kanonischer Ordnung und zug. Schnyder Realizer
- $\Phi_i(k)$ sei Vater in T_i des Knotens v_k
- setze T's für v_1 und v_2 auf $y = 1$ und $y = 2$
- setze h_i auf $y = i$ mit linker und rechter Position von $\Phi_1(i)$ und $\Phi_2(i)$ und b_i von $y = i$ bis zum Index von $\Phi_3(i)$

Phase 2: λ -Verdickung

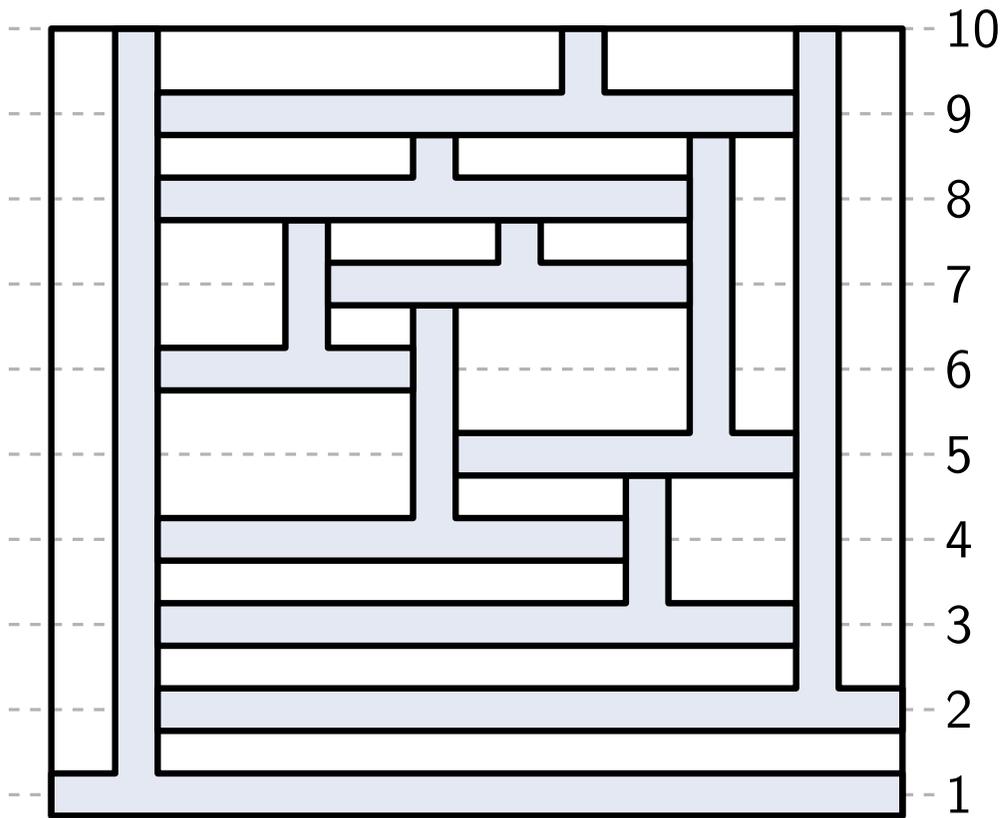


Ersetze jedes T durch ein T-förmiges Polygon mit Dicke λ für $0 < \lambda < 1/2$

Phase 3: Löcher entfernen

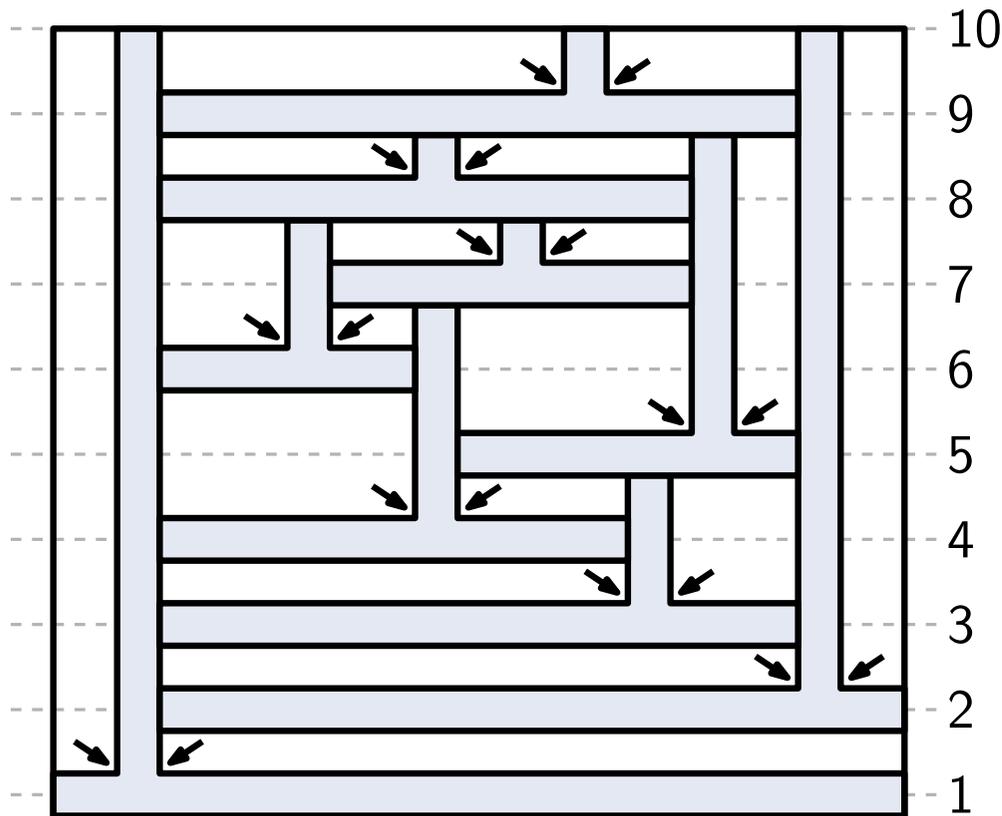


Phase 3: Löcher entfernen



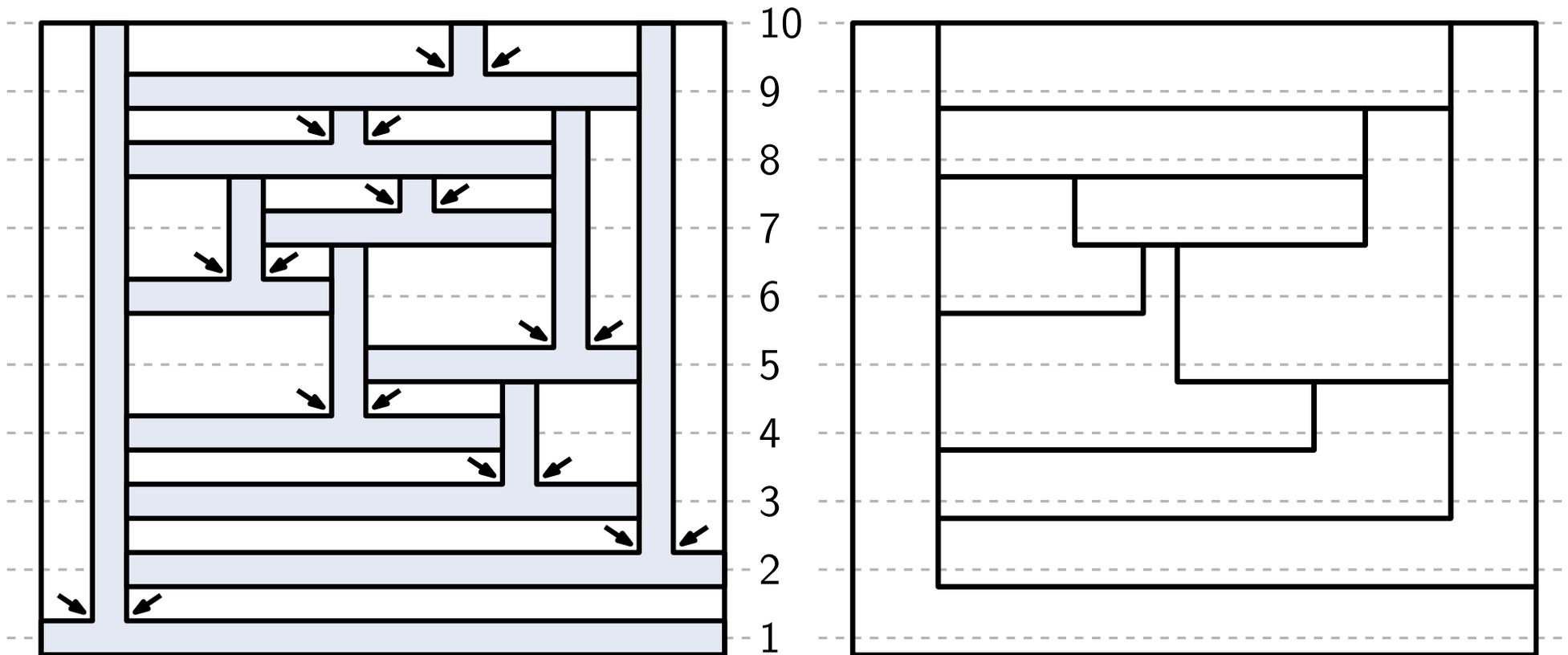
- setze Rahmen um die T-Polygone

Phase 3: Löcher entfernen



- setze Rahmen um die T-Polygone
- jedes (innere) Loch stammt von Dreiecks-Facette
- weise Löcher den eindeutigen konkaven Ecken zu

Phase 3: Löcher entfernen



- setze Rahmen um die T-Polygone
- jedes (innere) Loch stammt von Dreiecks-Facette
- weise Löcher den eindeutigen konkaven Ecken zu
- es entsteht ein rektilineares Dual mit maximal 8-seitigen Polygonen

(Planar) Orthogonal Drawings

Three-step approach: *Topology – Shape – Metrics*

[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

(Planar) Orthogonal Drawings

Three-step approach: *Topology – Shape – Metrics*

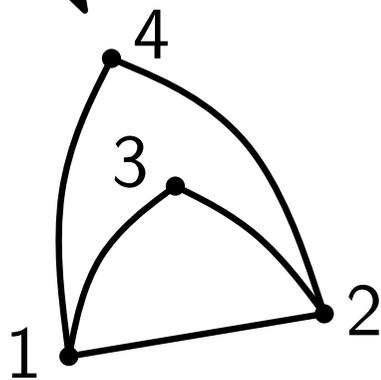
[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

Reduce Crossings

combinatorial
embedding/
planarization



(Planar) Orthogonal Drawings

Three-step approach: *Topology – Shape – Metrics*

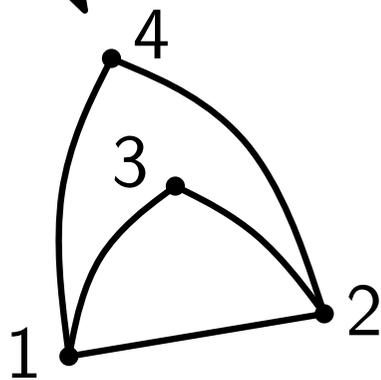
[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

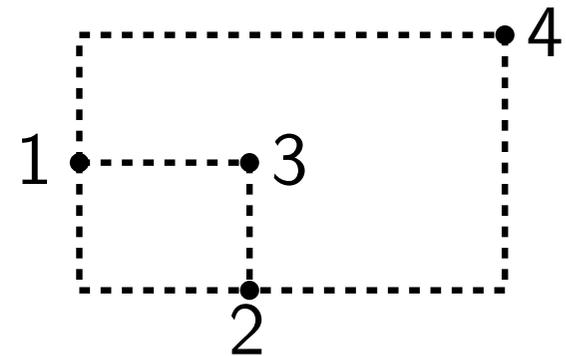
Reduce Crossings

combinatorial
embedding/
planarization



Bend Minimization

orthogonal
representation



(Planar) Orthogonal Drawings

Three-step approach: *Topology – Shape – Metrics*

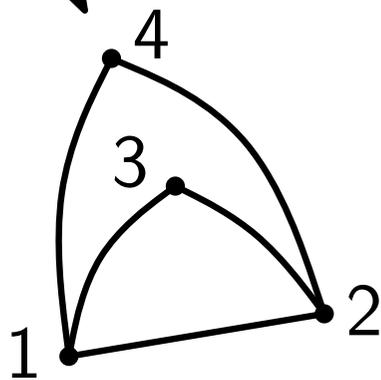
[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

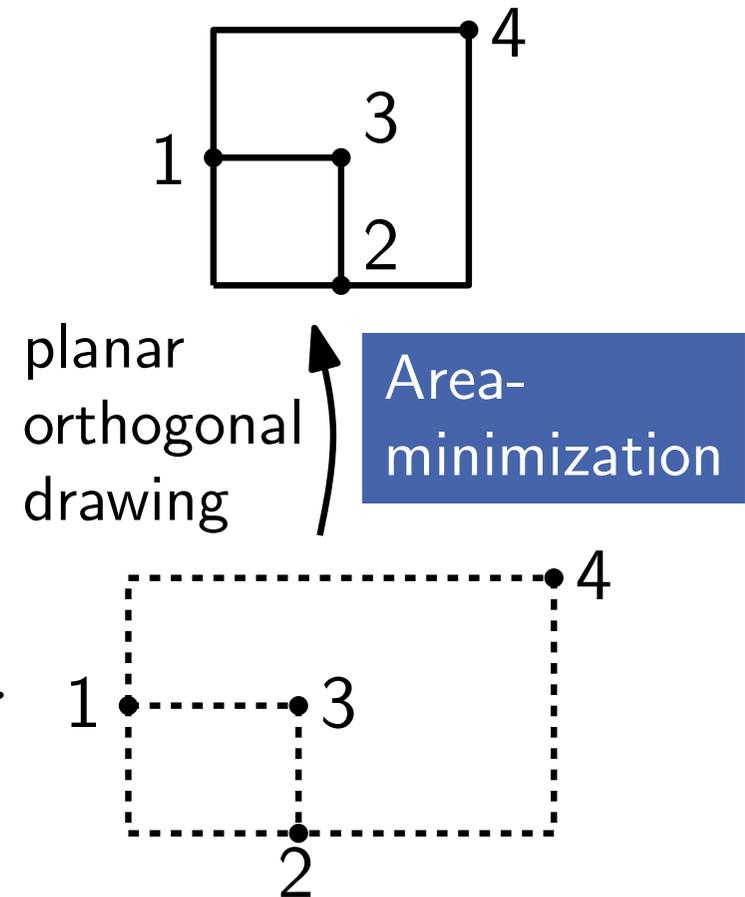
Reduce Crossings

combinatorial
embedding/
planarization



Bend Minimization

orthogonal
representation



planar
orthogonal
drawing

Area-
minimization

(Planar) Orthogonal Drawings

Three-step approach: *Topology – Shape – Metrics*

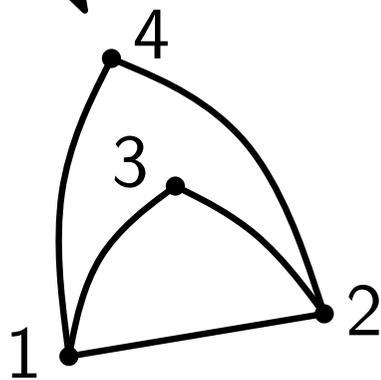
[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

Reduce Crossings

combinatorial
embedding/
planarization

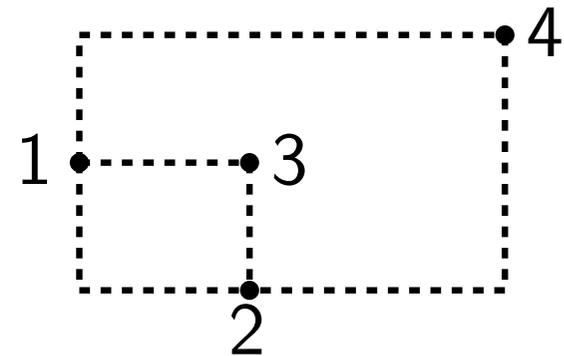
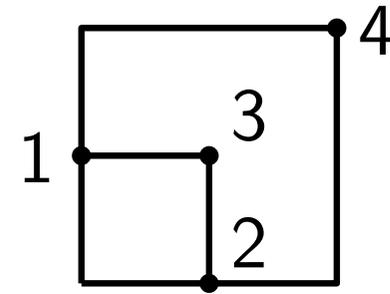


Bend Minimization

orthogonal
representation

planar
orthogonal
drawing

Area-
minimization



Combinatorial Bend Minimization

Problem Combinatorial Bend Minimization

Given: ■ Graph $G = (V, E)$ with maximum degree 4
■ combinatorial embedding \mathcal{F} and outer face f_0

Find: **orthogonal representation** $H(G)$ with minimum number of bends

Combinatorial Bend Minimization

Problem Combinatorial Bend Minimization

Given: ■ Graph $G = (V, E)$ with maximum degree 4
■ combinatorial embedding \mathcal{F} and outer face f_0

Find: **orthogonal representation** $H(G)$ with minimum number of bends

Idea: formulate as a network flow problem

- a unit of flow represents an angle $\pi/2$
- flow from vertices to faces represents the angles at the vertices
- flow between adjacent faces represent the bends at the edges

Flow Network for Bend Minimization

Flow Network $N(G) = ((V \cup \mathcal{F}, A); \ell; c; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ incident to } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid f, g \text{ adjacent through edge } e\}$

Flow Network for Bend Minimization

Flow Network $N(G) = ((V \cup \mathcal{F}, A); \ell; c; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ incident to } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid f, g \text{ adjacent through edge } e\}$
- $b(v) = 4 \quad \forall v \in V$

Flow Network for Bend Minimization

Flow Network $N(G) = ((V \cup \mathcal{F}, A); \ell; c; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ incident to } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid f, g \text{ adjacent through edge } e\}$
- $b(v) = 4 \quad \forall v \in V$
- $b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\}$
- $b(f_0) = -2(d_G(f_0) + 2)$

Flow Network for Bend Minimization

Flow Network $N(G) = ((V \cup \mathcal{F}, A); \ell; c; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ incident to } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid f, g \text{ adjacent through edge } e\}$
 - $b(v) = 4 \quad \forall v \in V$
 - $b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\}$
 - $b(f_0) = -2(d_G(f_0) + 2)$
- } $\Rightarrow \sum_w b(w) \stackrel{?}{=} 0$

Flow Network for Bend Minimization

Flow Network $N(G) = ((V \cup \mathcal{F}, A); \ell; c; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ incident to } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid f, g \text{ adjacent through edge } e\}$
 - $b(v) = 4 \quad \forall v \in V$
 - $b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\}$
 - $b(f_0) = -2(d_G(f_0) + 2)$
- } $\Rightarrow \sum_w b(w) = 0$
(Euler)

Flow Network for Bend Minimization

Flow Network $N(G) = ((V \cup \mathcal{F}, A); \ell; c; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ incident to } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid f, g \text{ adjacent through edge } e\}$
 - $b(v) = 4 \quad \forall v \in V$
 - $b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\}$
 - $b(f_0) = -2(d_G(f_0) + 2)$
- $\left. \vphantom{\begin{matrix} \text{■ } b(v) = 4 \quad \forall v \in V \\ \text{■ } b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\} \\ \text{■ } b(f_0) = -2(d_G(f_0) + 2) \end{matrix}} \right\} \Rightarrow \sum_w b(w) = 0$
(Euler)

$$\forall (f, g) \in A, f, g \in \mathcal{F}$$

$$\ell(f, g) := 0 \leq X(f, g) \leq \infty =: c(f, g)$$

$$\text{cost}(f, g) = 1$$

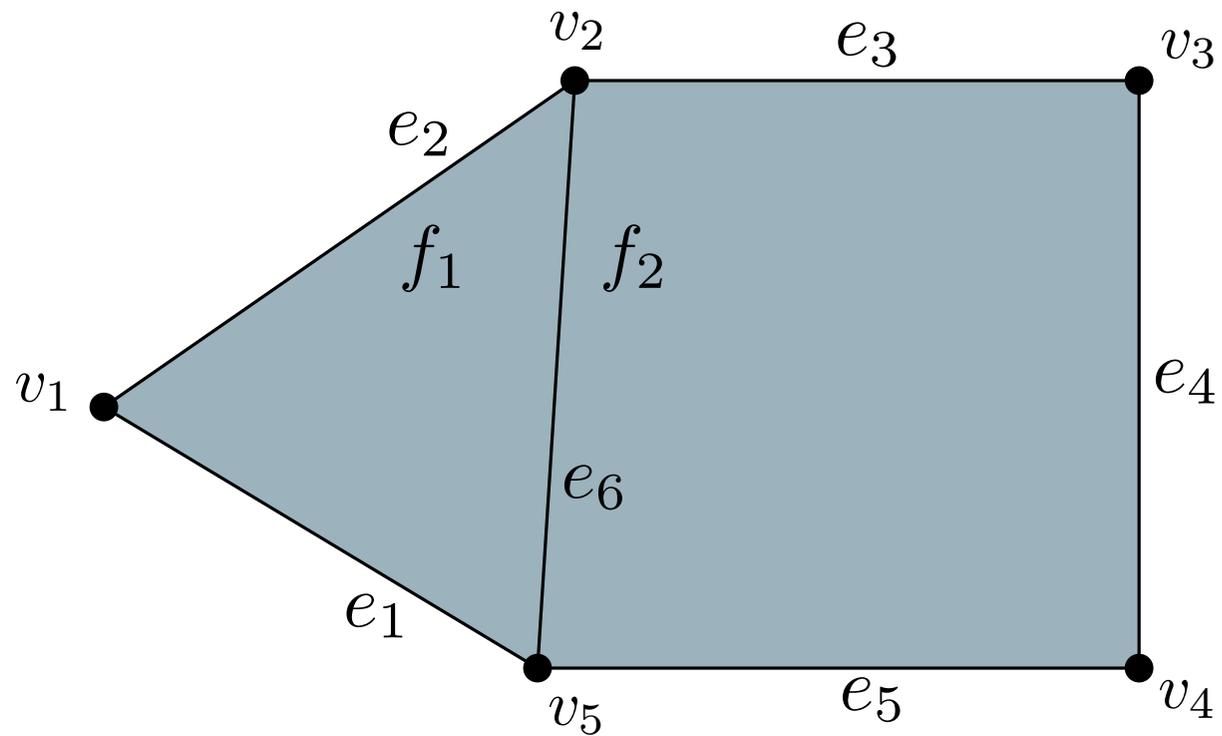
$$\forall (v, f) \in A, v \in V, f \in \mathcal{F}$$

$$\ell(v, f) := 1 \leq X(v, f) \leq 4 =: c(v, f)$$

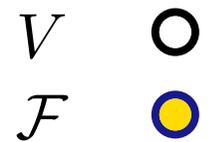
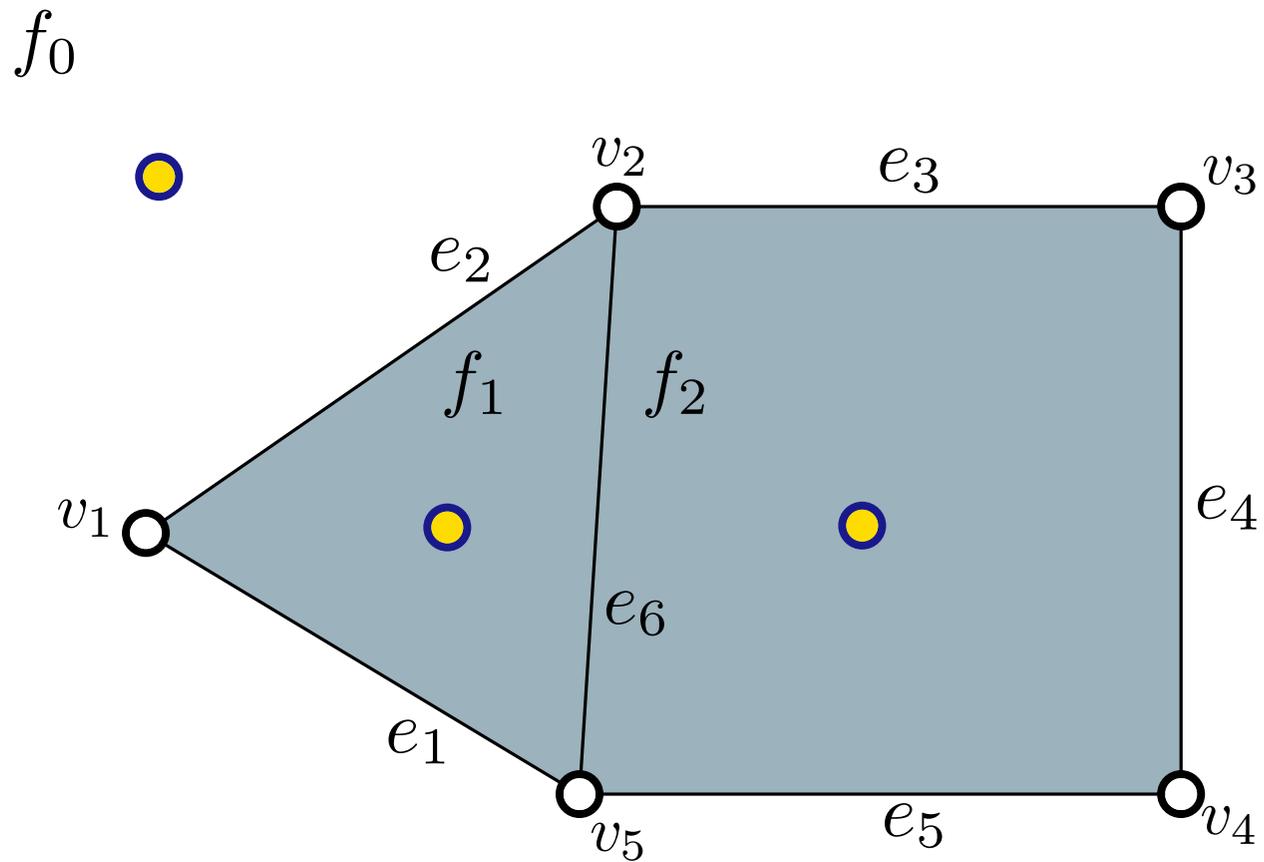
$$\text{cost}(v, f) = 0$$

Example Flow Network

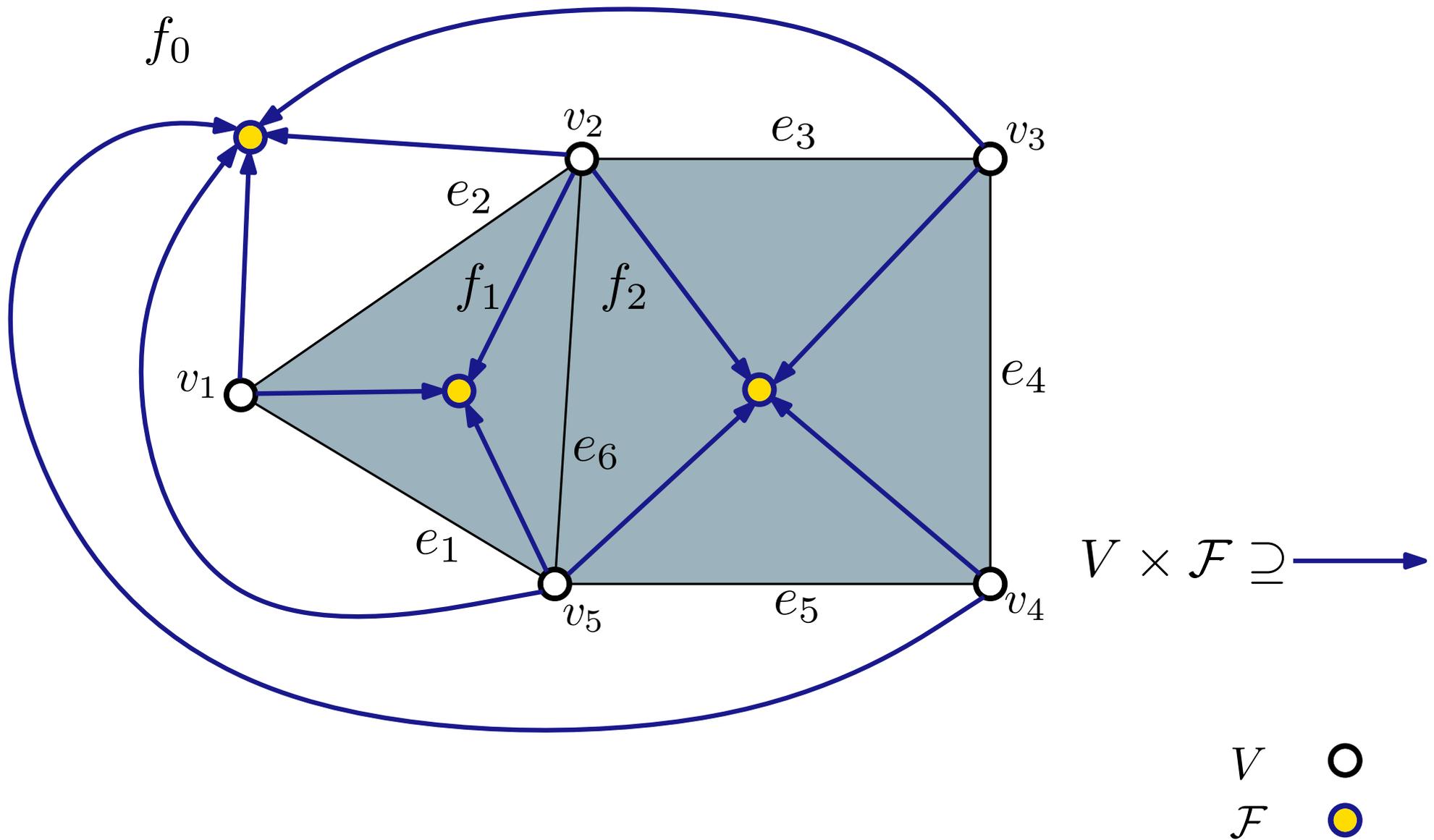
f_0



Example Flow Network

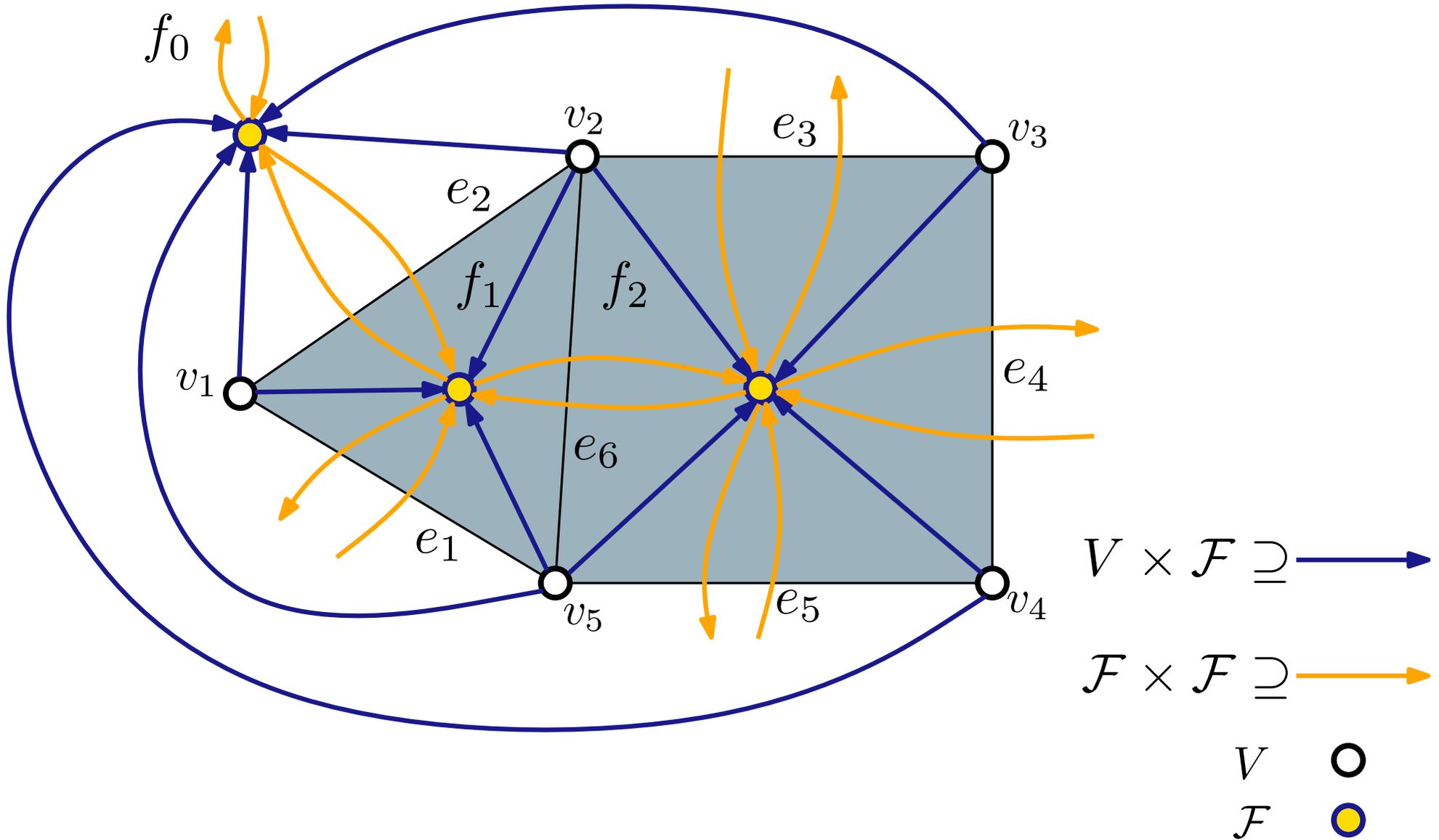


Example Flow Network

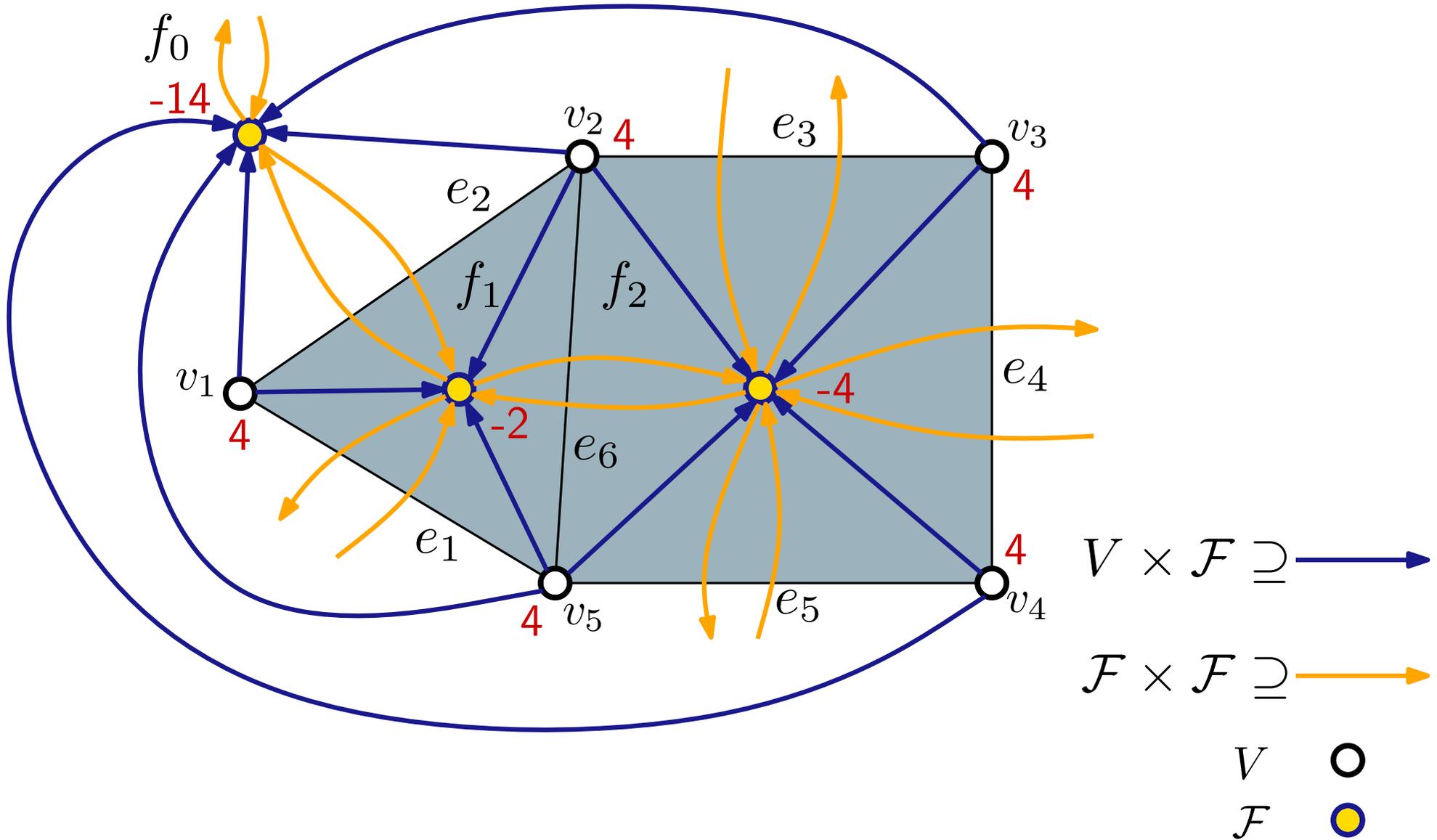


V ○
 F ●

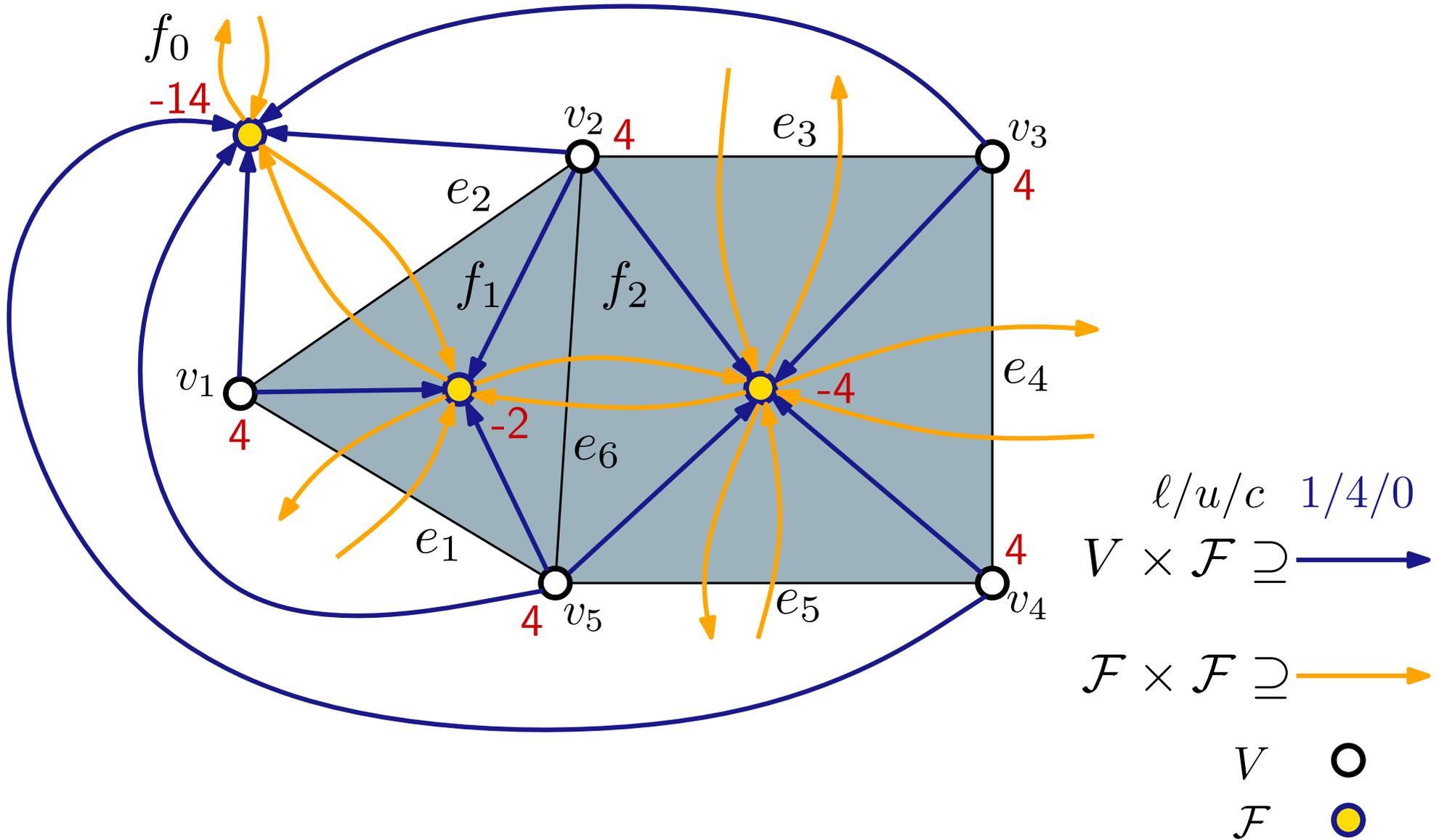
Example Flow Network



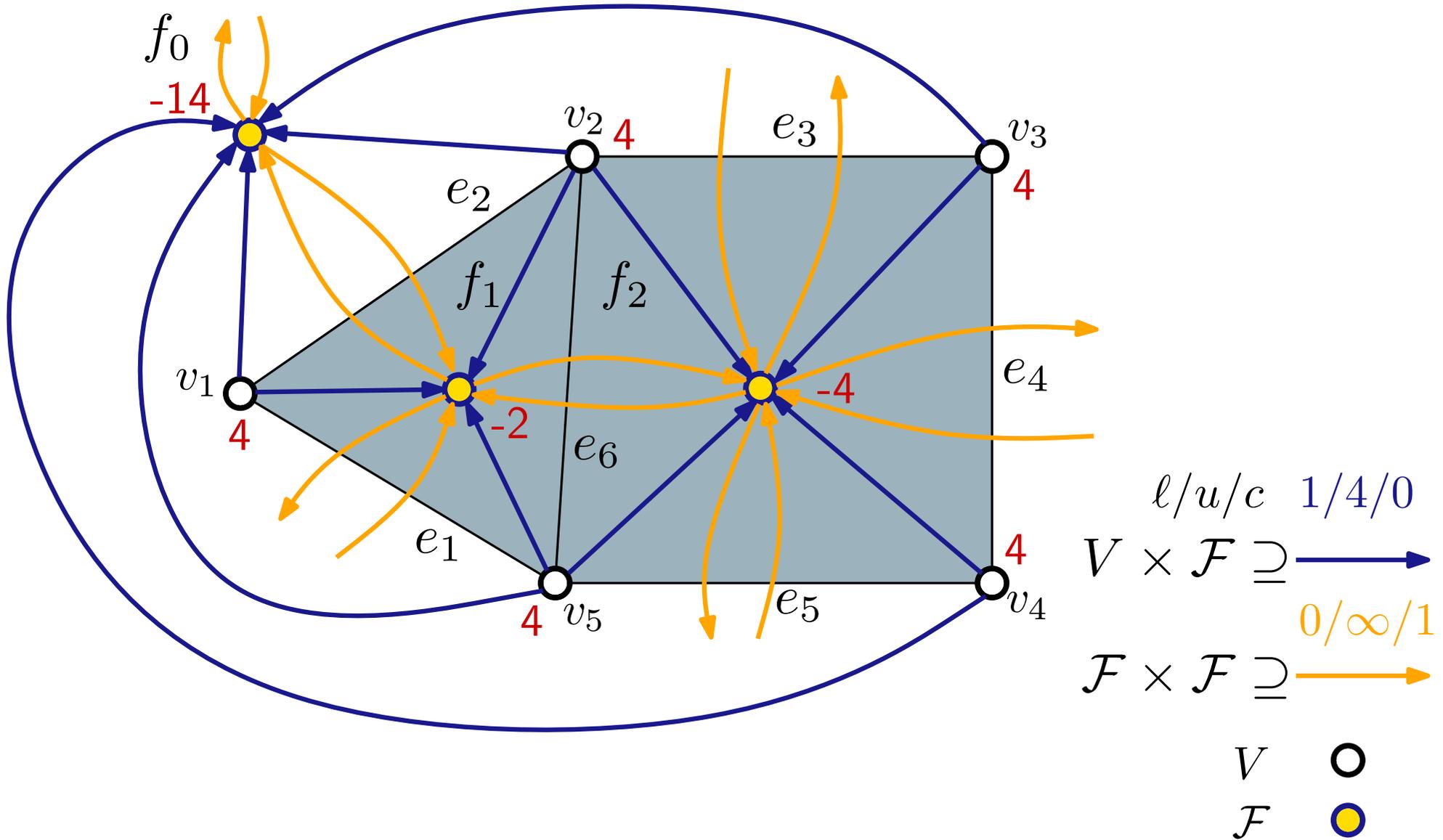
Example Flow Network



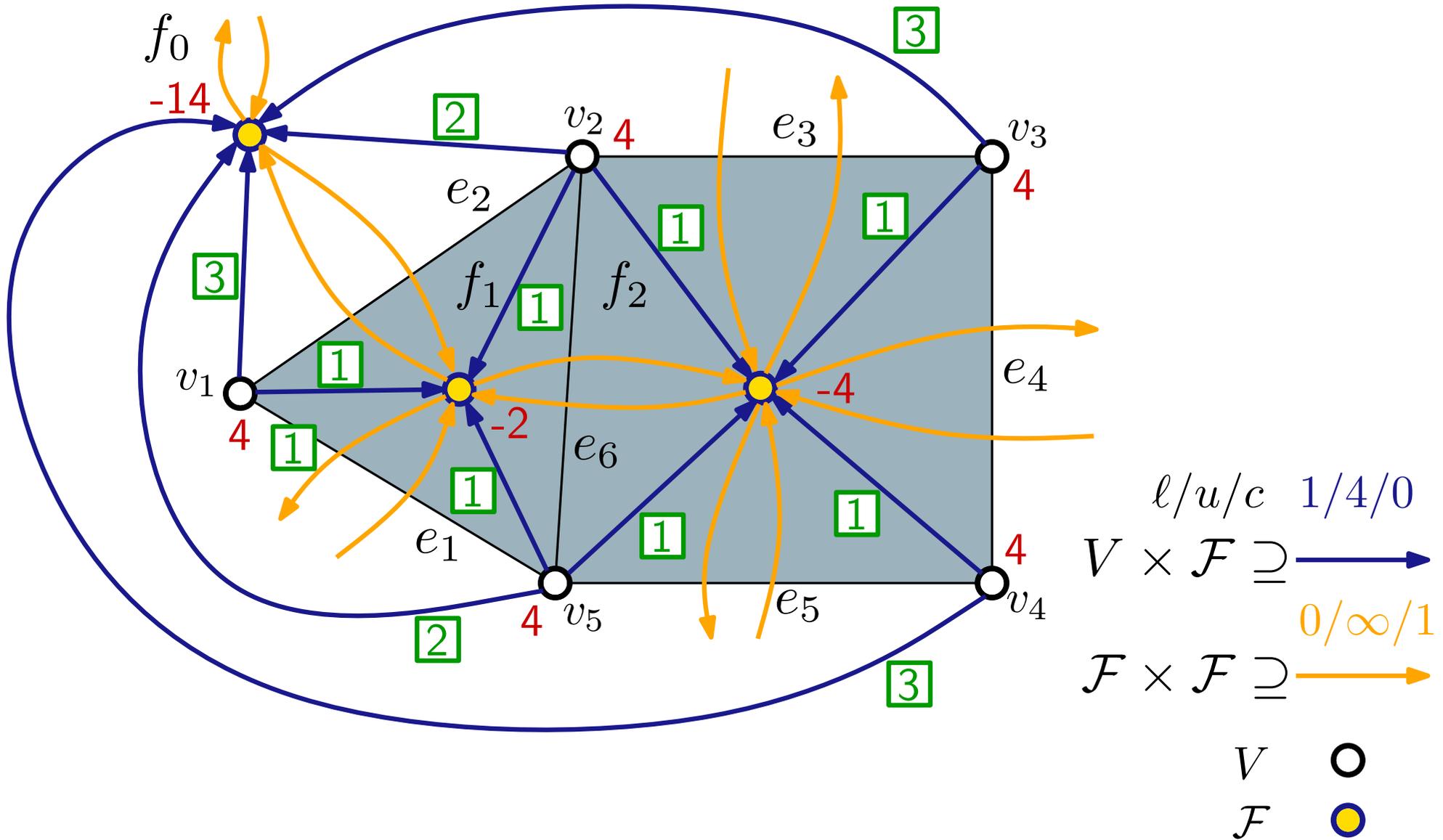
Example Flow Network



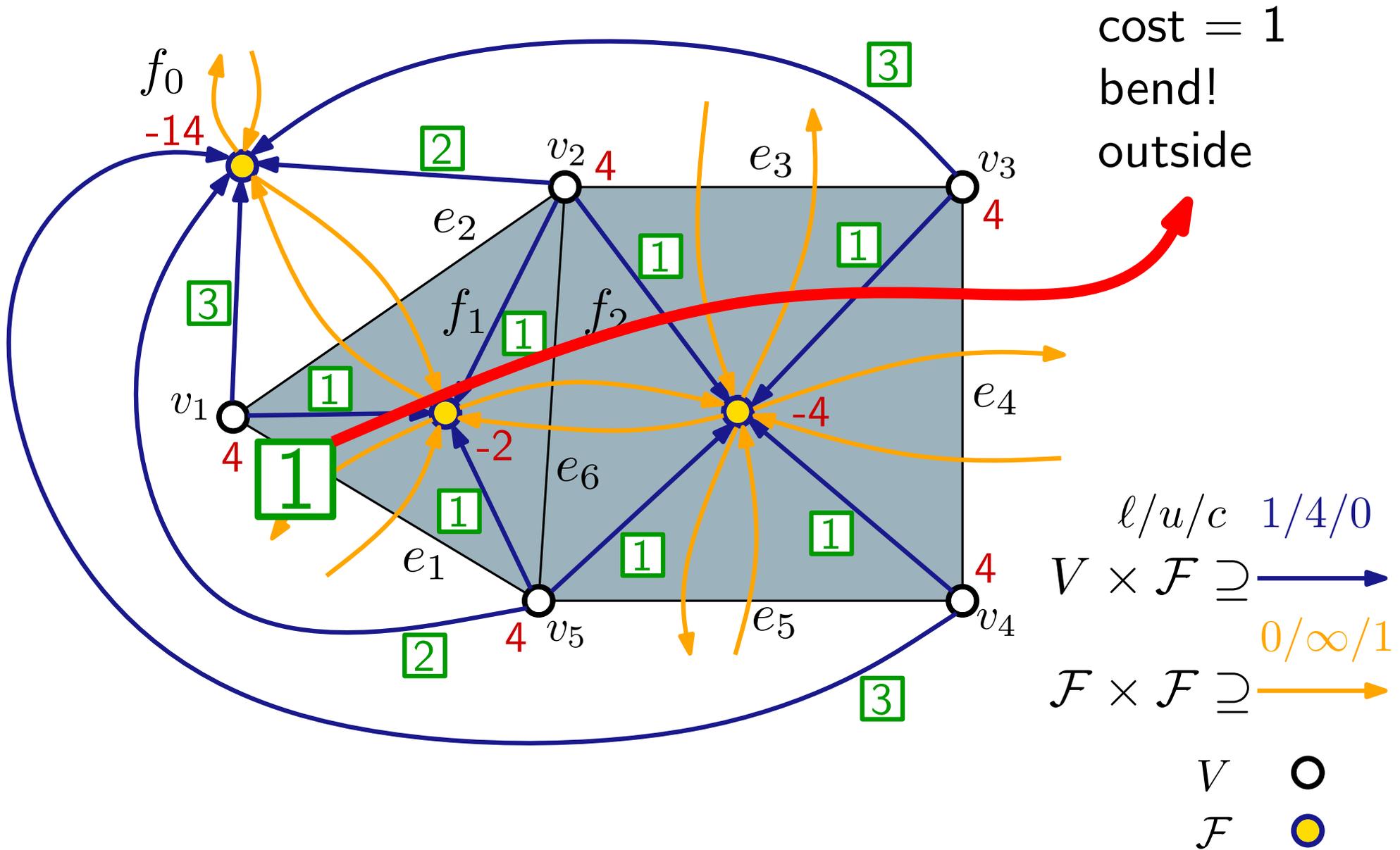
Example Flow Network



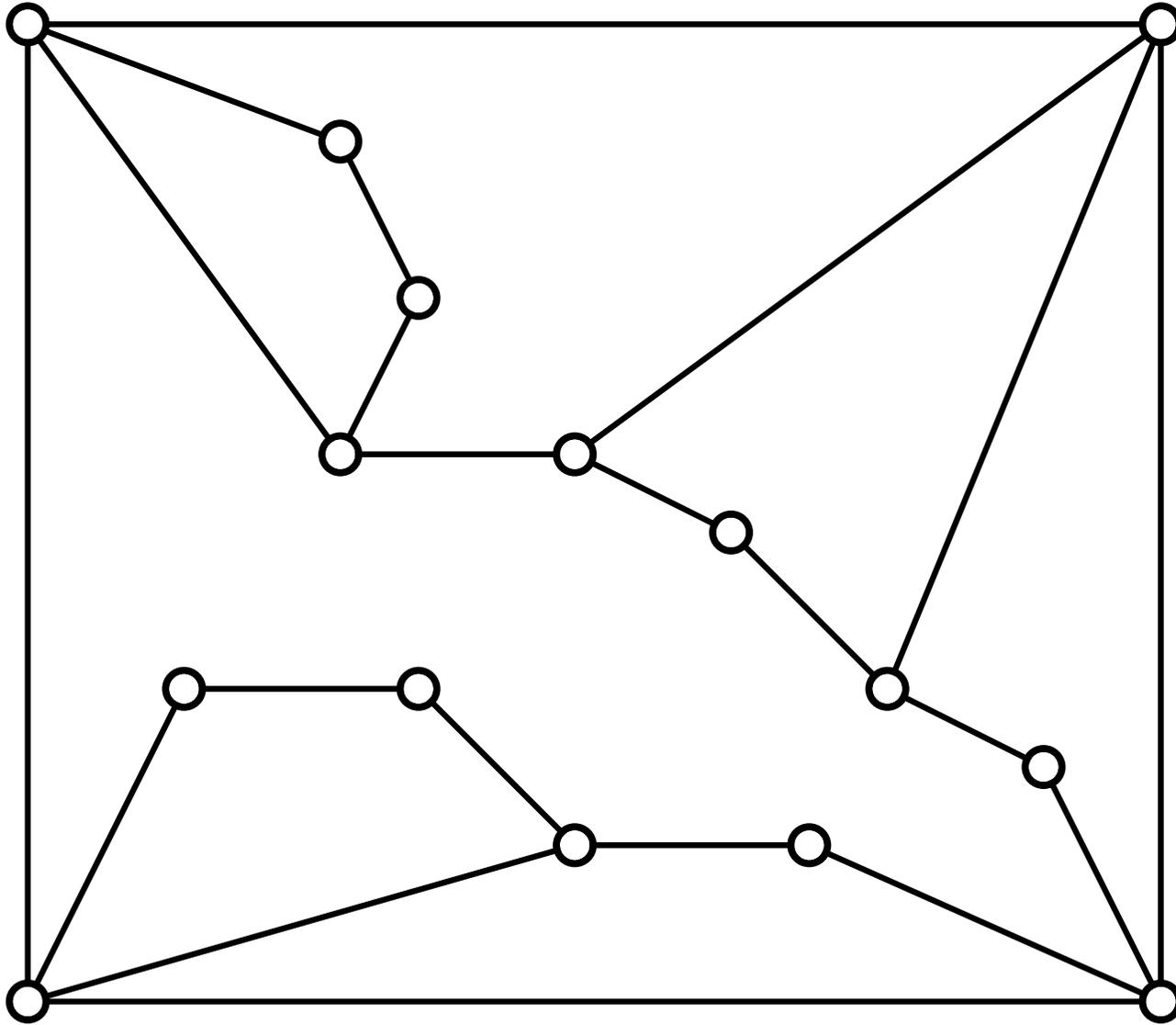
Example Flow Network



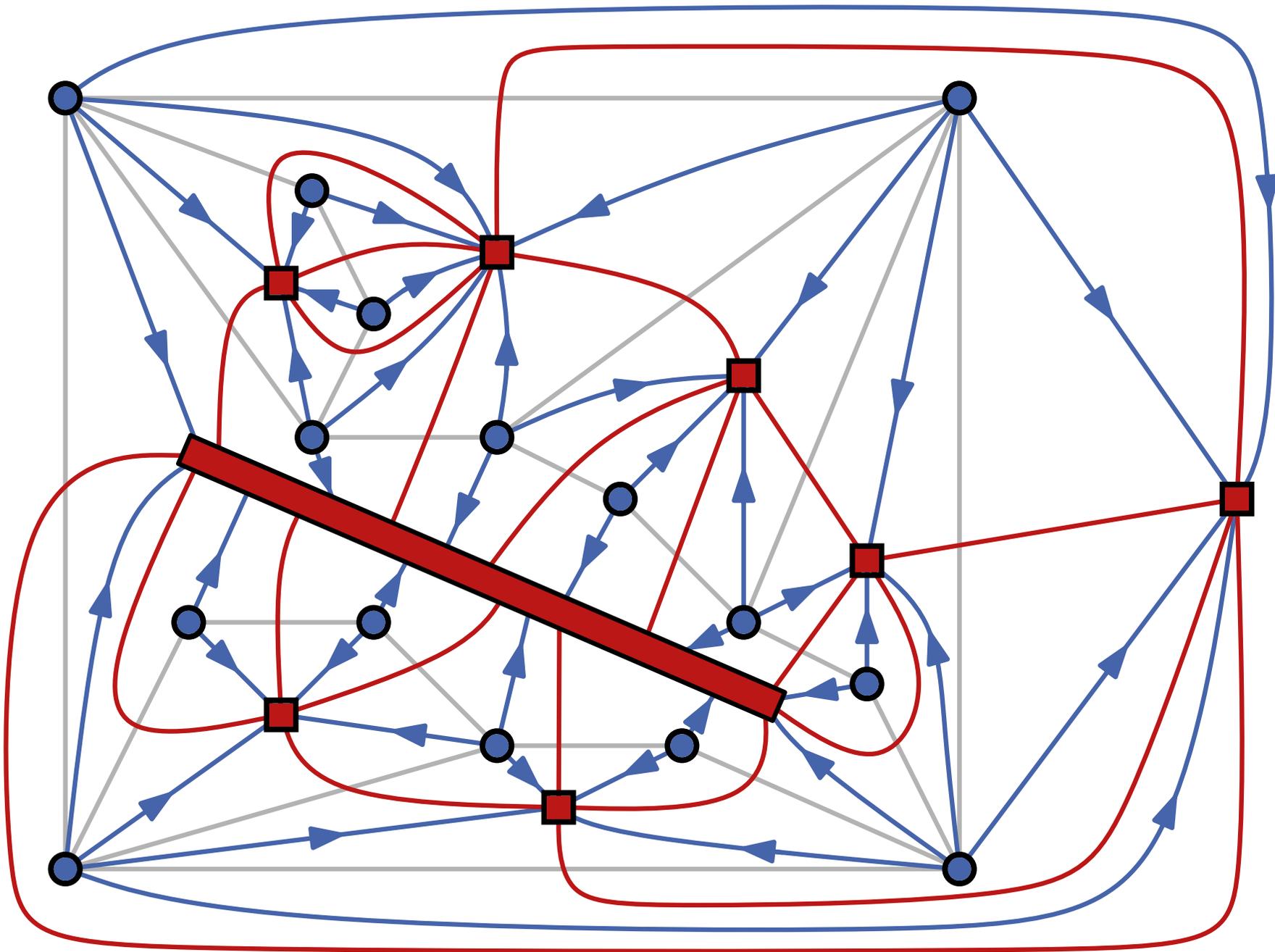
Example Flow Network



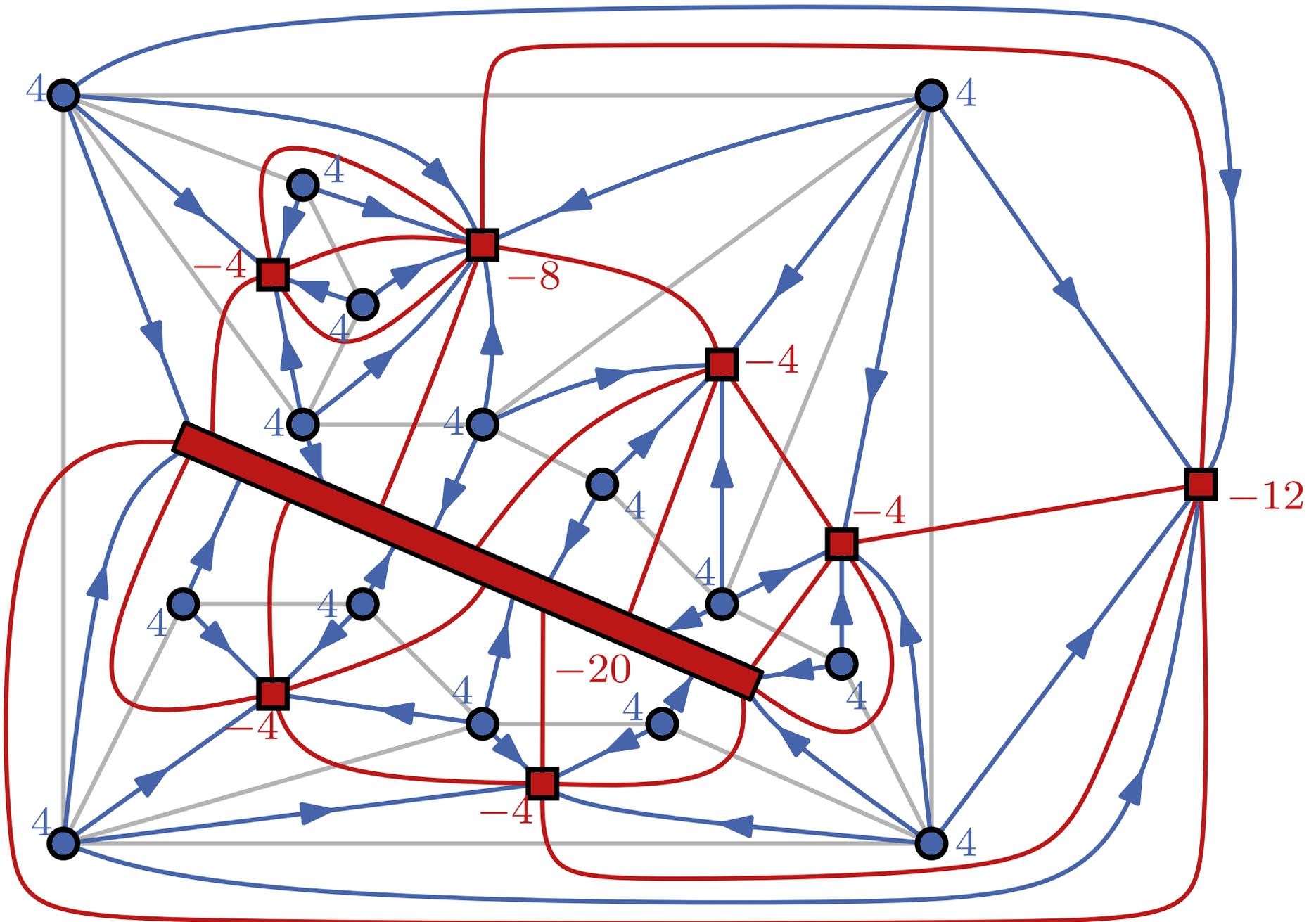
Aufgabe 3 – Knickminimierung



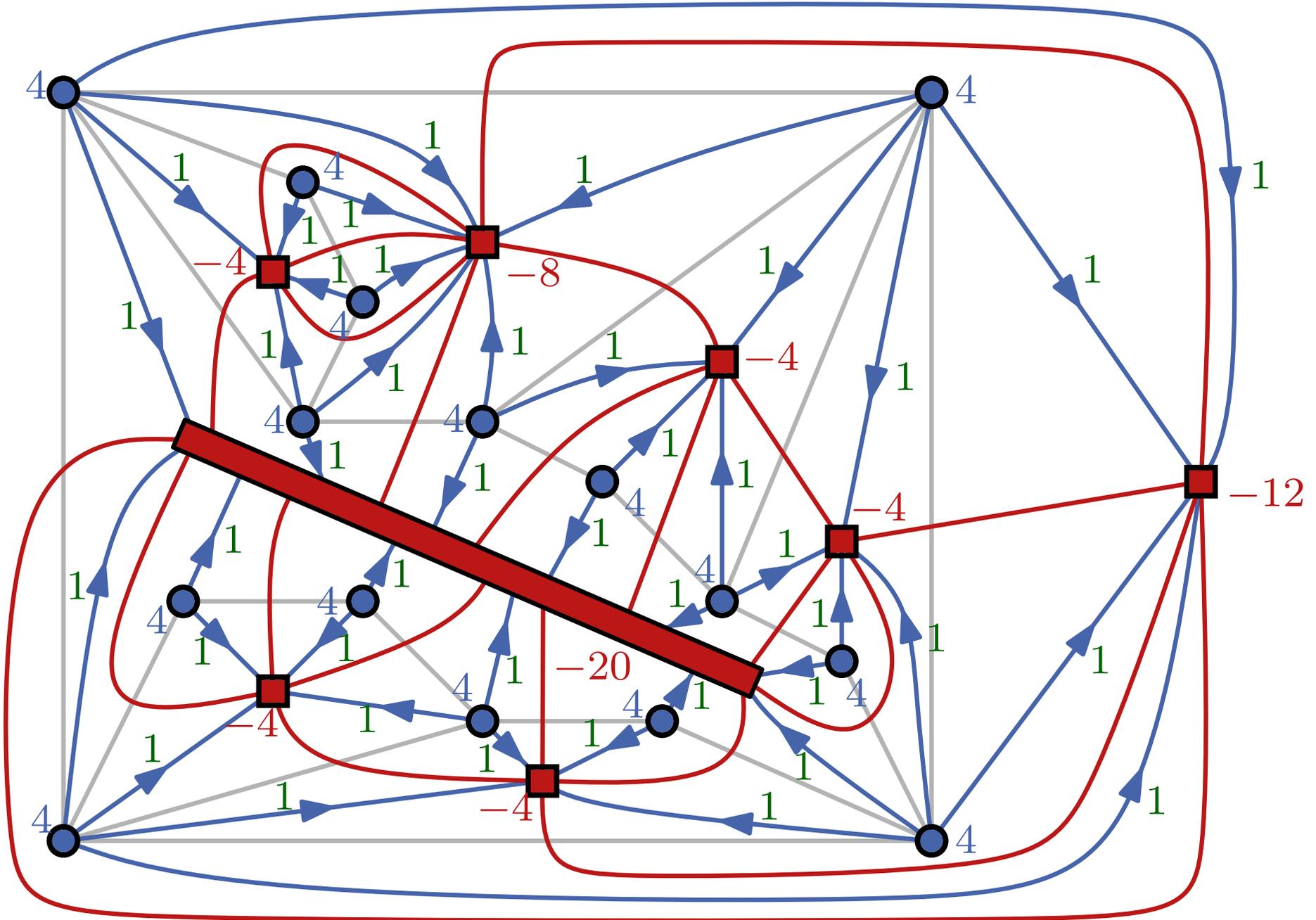
Aufgabe 3 – Knickminimierung



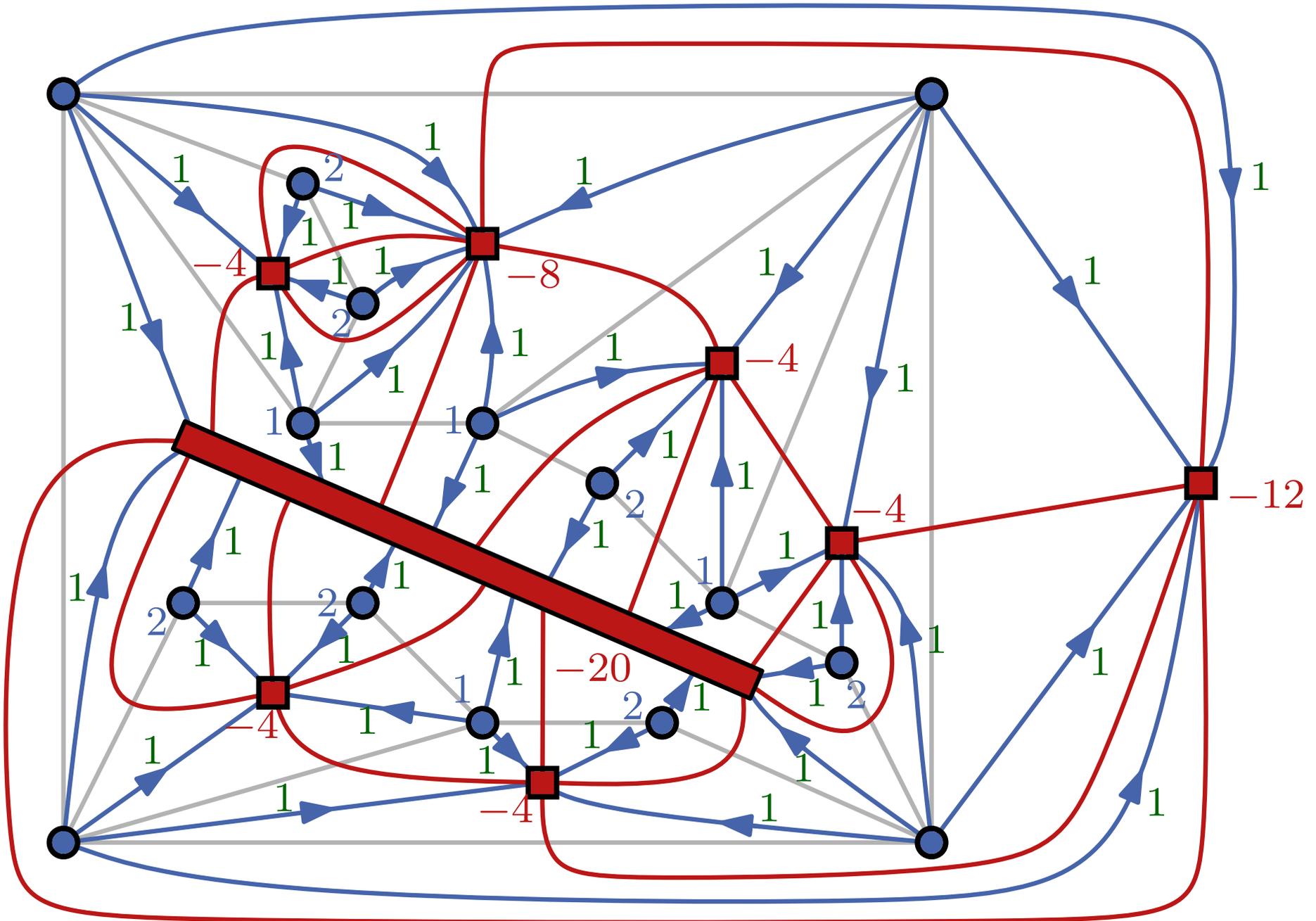
Aufgabe 3 – Knickminimierung



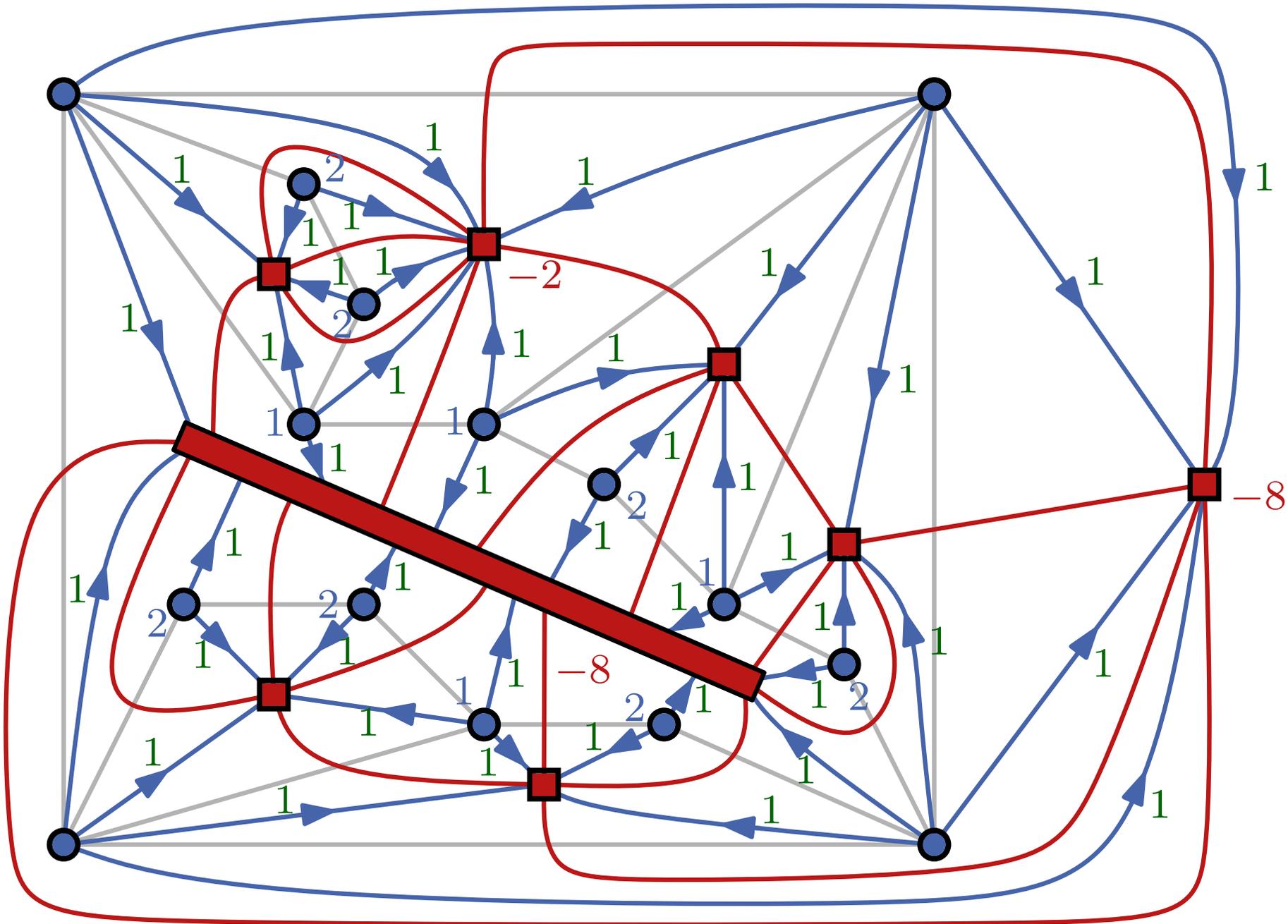
Aufgabe 3 – Knickminimierung



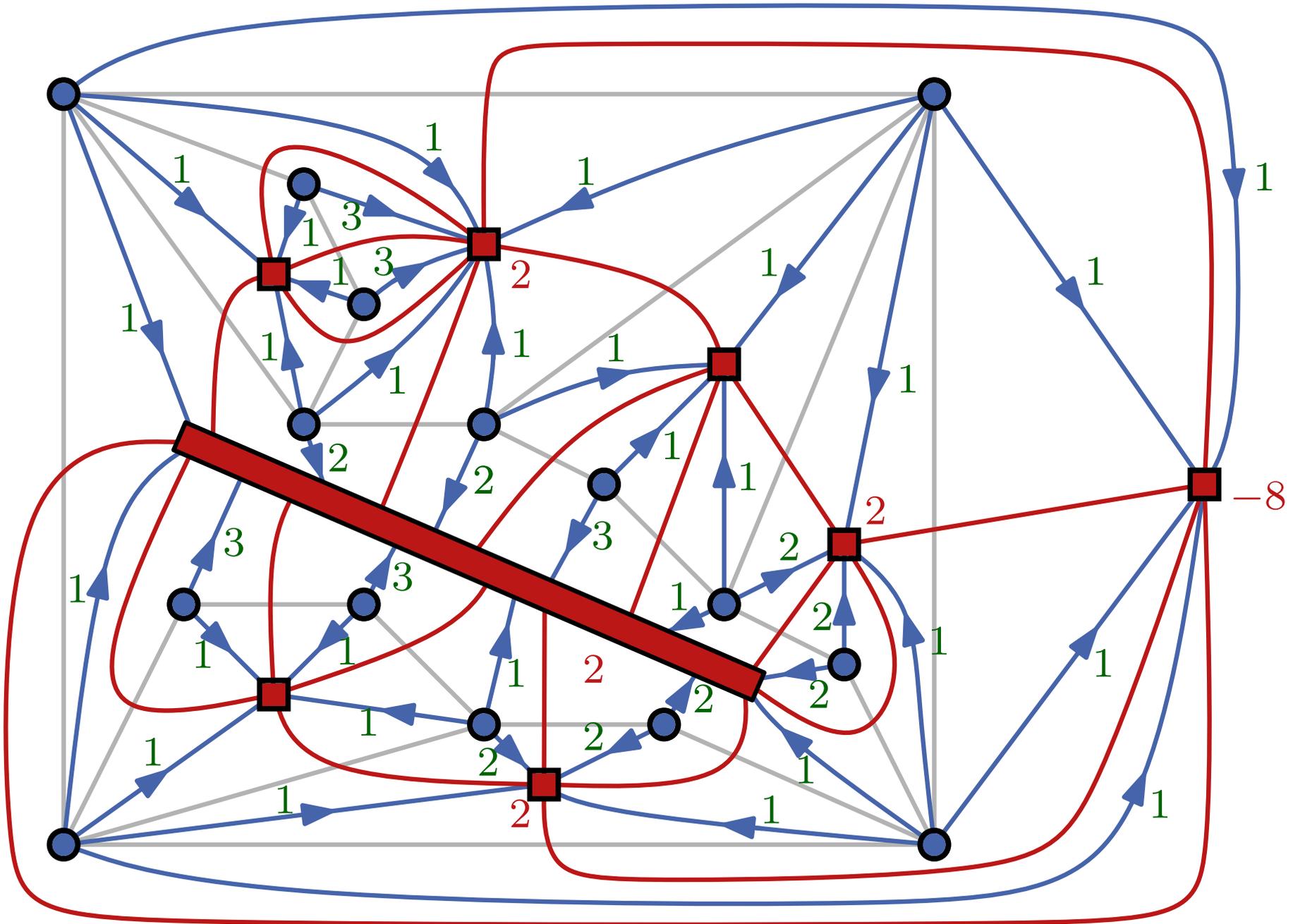
Aufgabe 3 – Knickminimierung



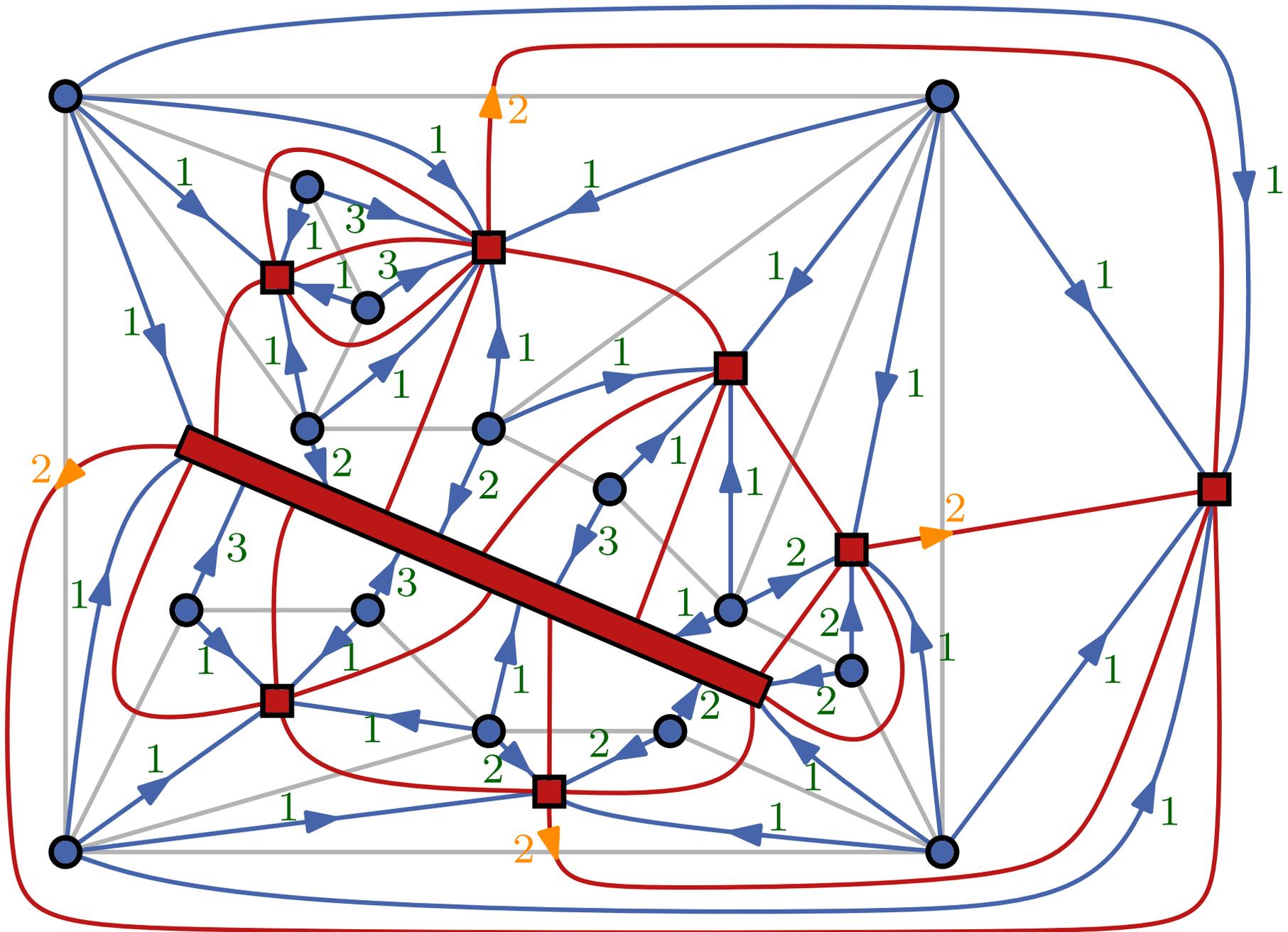
Aufgabe 3 – Knickminimierung



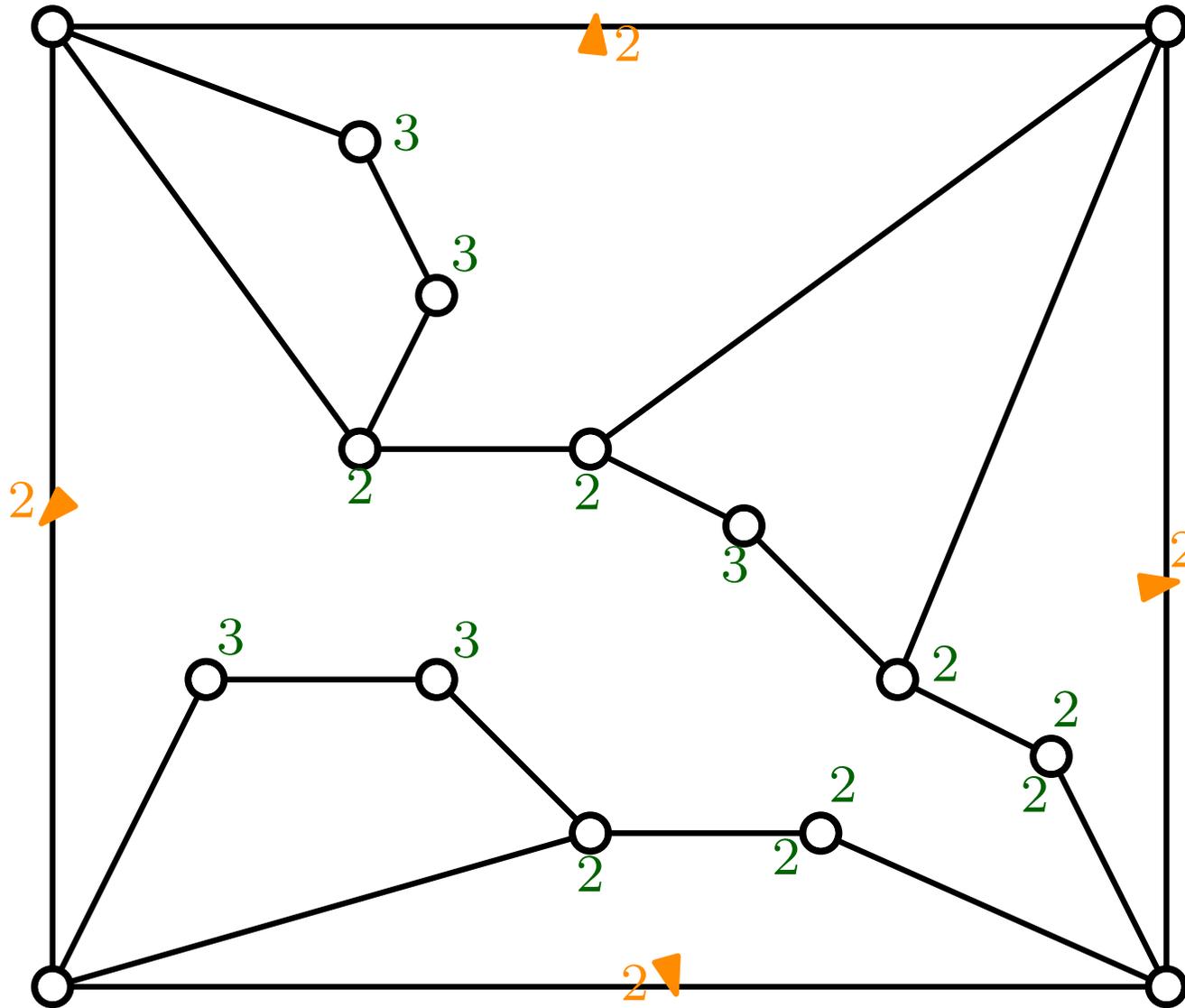
Aufgabe 3 – Knickminimierung



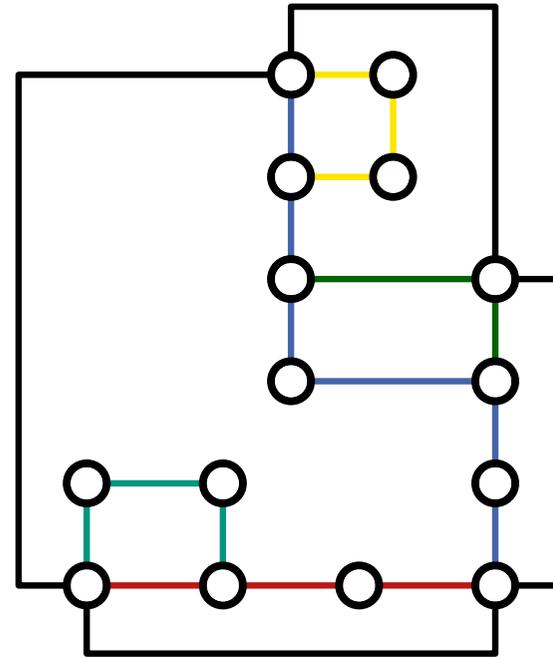
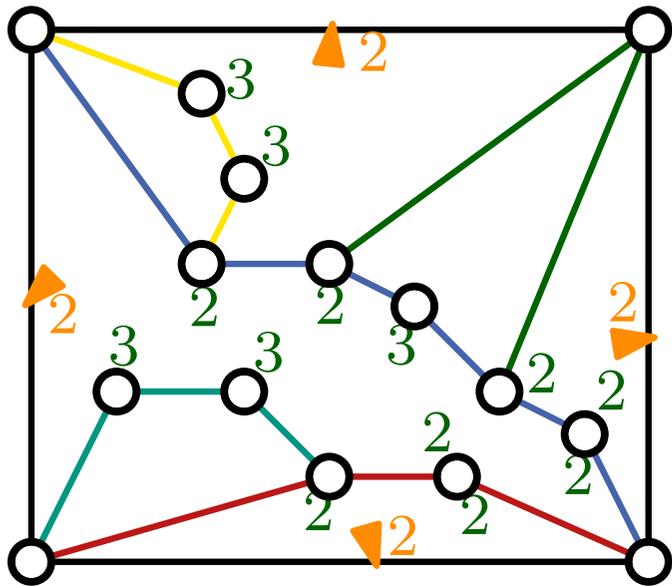
Aufgabe 3 – Knickminimierung



Aufgabe 3 – Knickminimierung



Aufgabe 3 – Knickminimierung



Aufgabe 3 – Knickminimierung

