Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Chair for
INFORMATICS I
Efficient Algorithms and
Knowledge-Based Systems

Institute for Informatics

Approximationsalgorithmen

SteinerTree
und
Mehrwegeschnitte

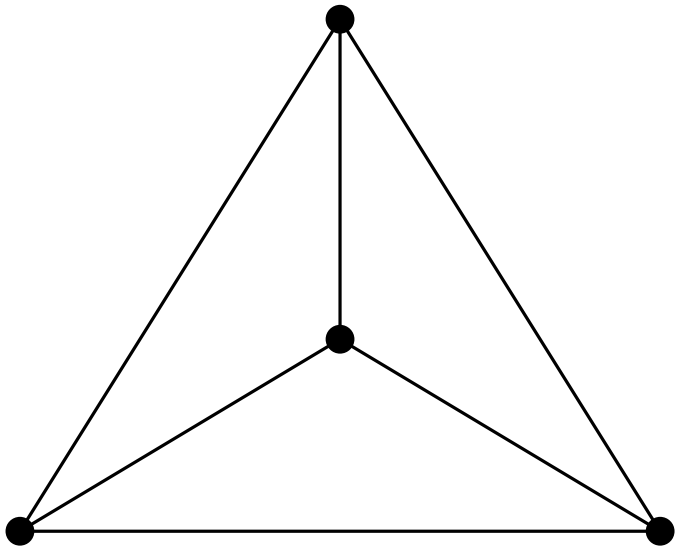
3. Vorlesung

Philipp Kindermann

Wintersemester 2019/20

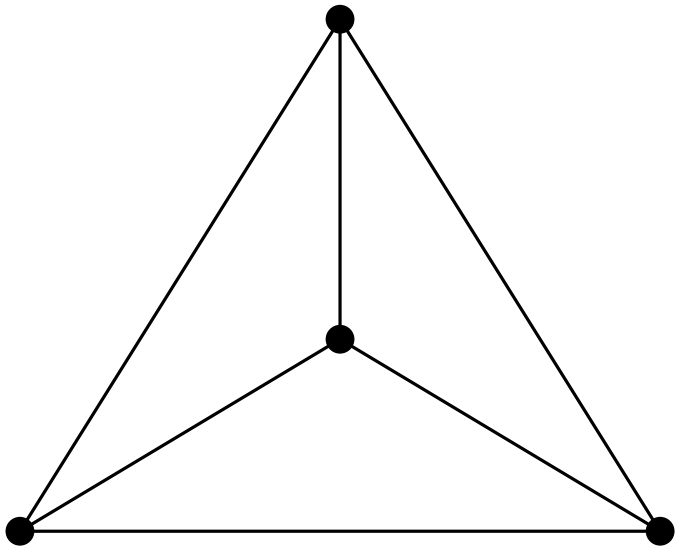
SteinerTree

Gegeben sei ein Graph $G = (V, E)$



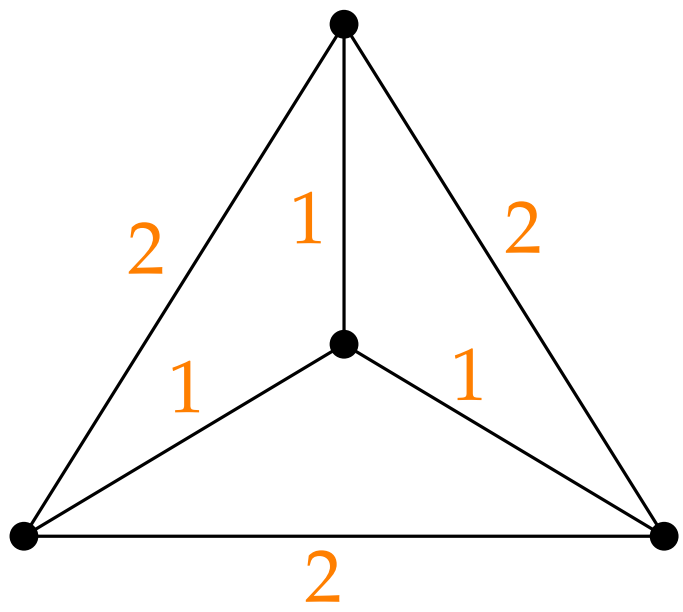
SteinerTree

Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$



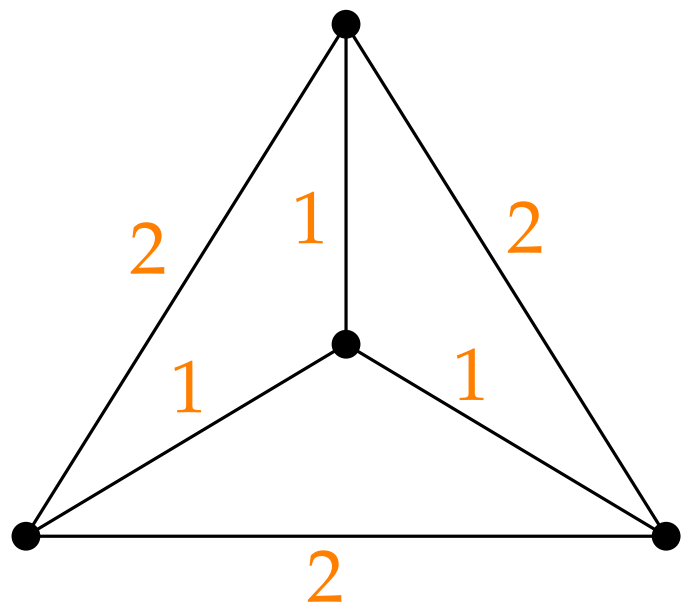
SteinerTree

Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$



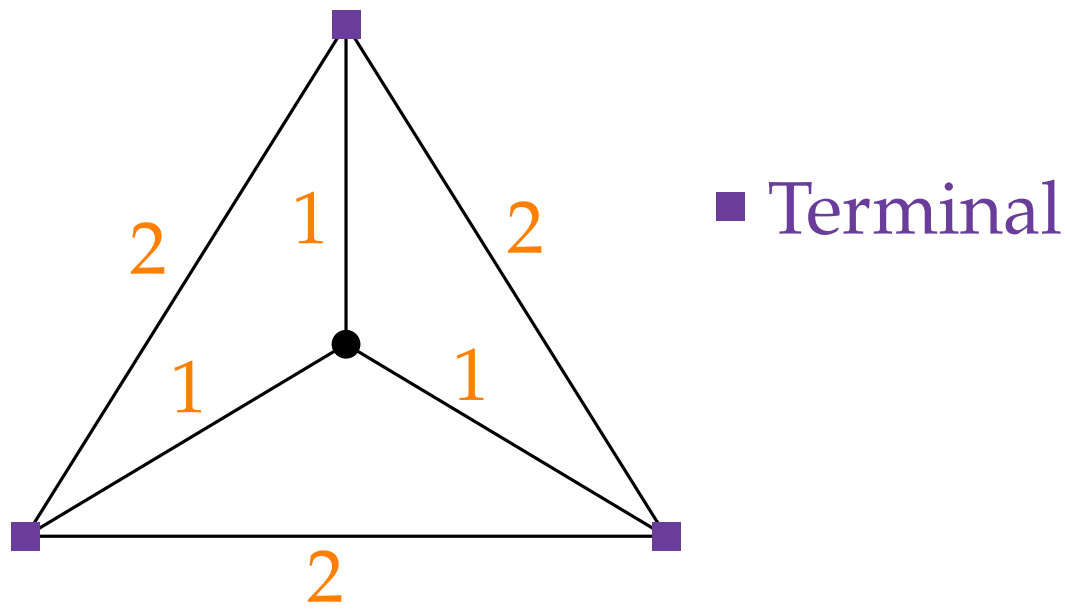
SteinerTree

Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$ sowie eine Partitionierung von V in eine Menge T von **Terminalen** und eine Menge S von **Steinerknoten**.



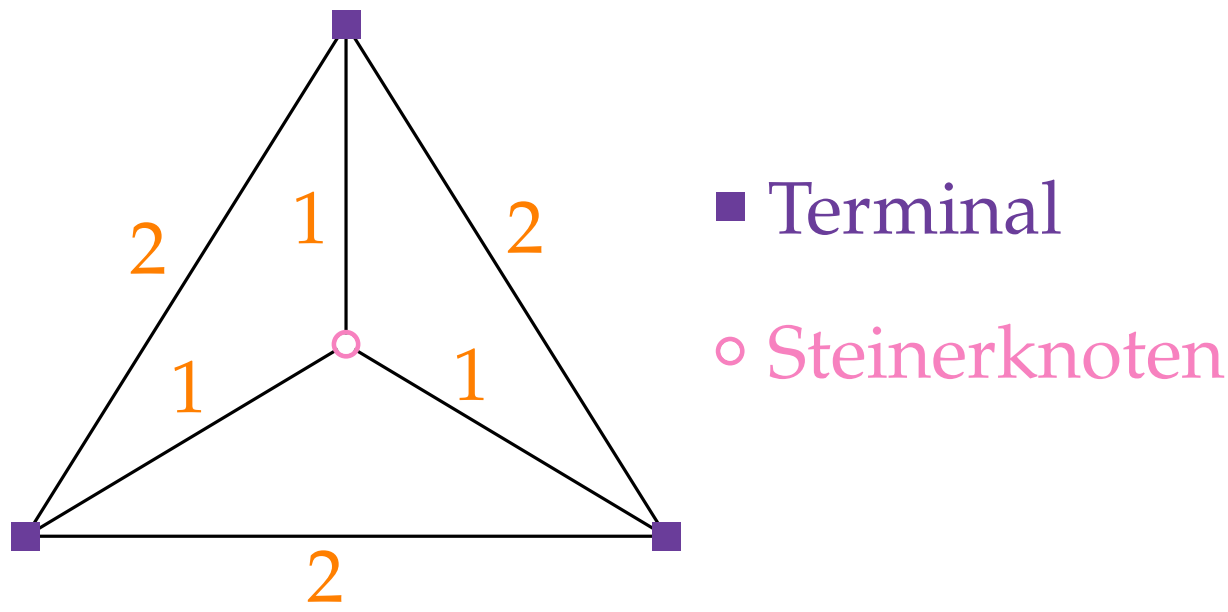
SteinerTree

Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$ sowie eine Partitionierung von V in eine Menge T von **Terminalen** und eine Menge S von **Steinerknoten**.



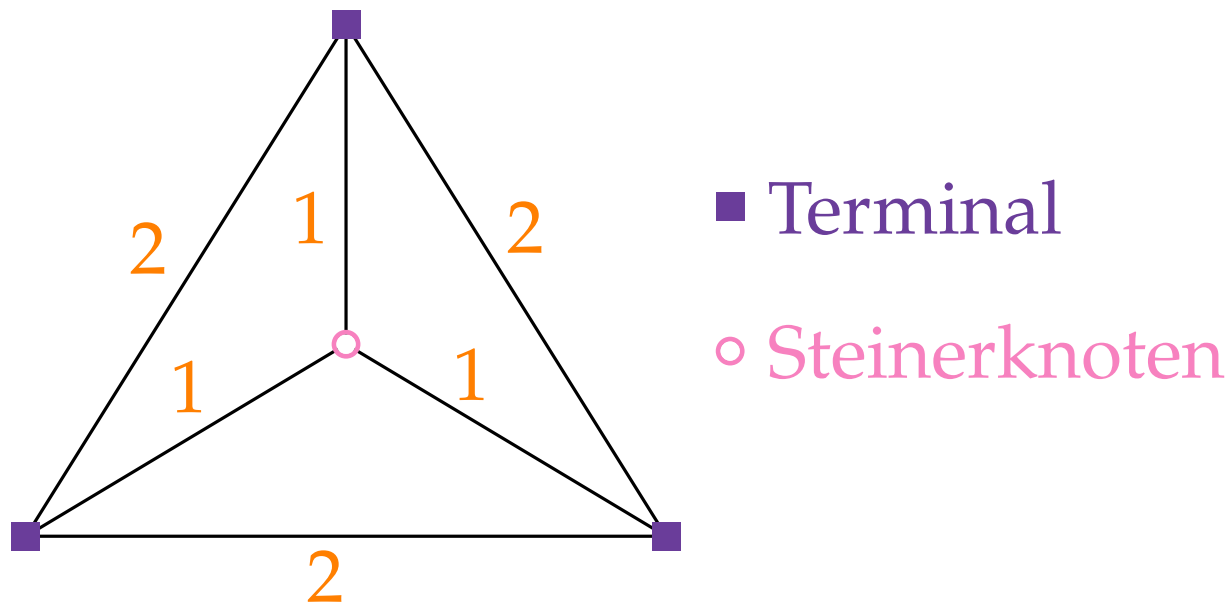
SteinerTree

Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$ sowie eine Partitionierung von V in eine Menge T von **Terminalen** und eine Menge S von **Steinerknoten**.



SteinerTree

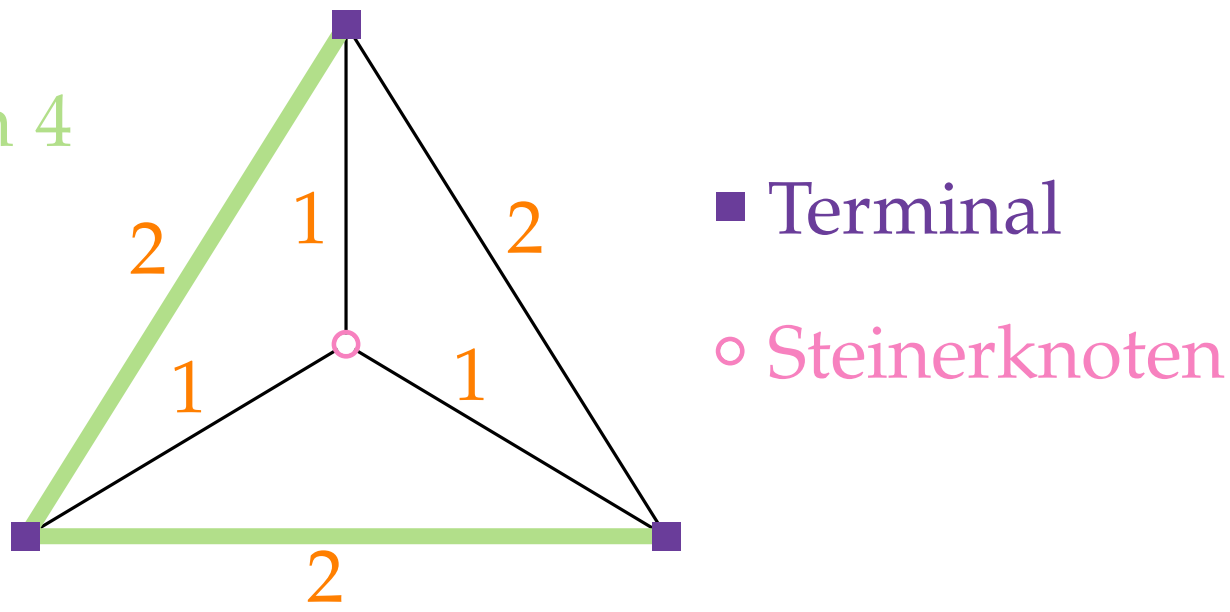
Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$ sowie eine Partitionierung von V in eine Menge T von **Terminalen** und eine Menge S von **Steinerknoten**. Gesucht ist ein Teilbaum $B = (V', E')$ von G, der alle Terminale enthält, d.h. $T \subseteq V'$



SteinerTree

Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$ sowie eine Partitionierung von V in eine Menge T von **Terminalen** und eine Menge S von **Steinerknoten**. Gesucht ist ein Teilbaum $B = (V', E')$ von G, der alle Terminale enthält, d.h. $T \subseteq V'$

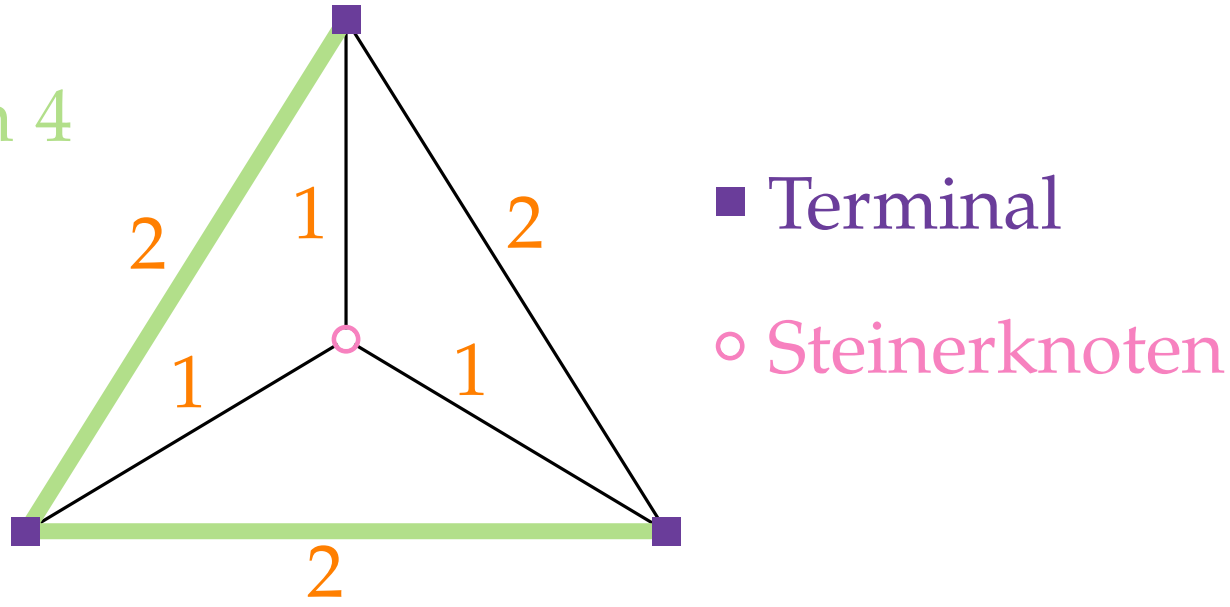
zul. Lösung mit Kosten 4



SteinerTree

Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$ sowie eine Partitionierung von V in eine Menge T von **Terminalen** und eine Menge S von **Steinerknoten**. Gesucht ist ein Teilbaum $B = (V', E')$ von G, der alle Terminale enthält, d.h. $T \subseteq V'$, und der minimale Kosten $c(E') := \sum_{e \in E'} c(e)$ unter allen Teilbäumen mit dieser Eigenschaft hat.

zul. Lösung mit Kosten 4

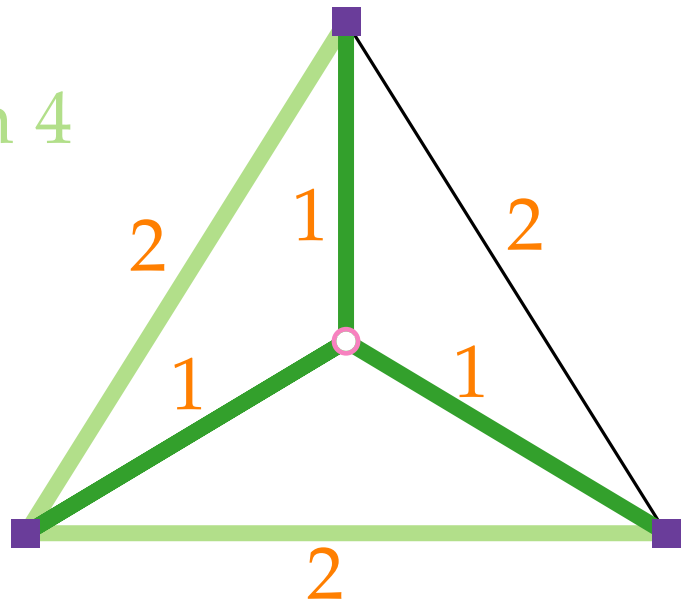


SteinerTree

Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$ sowie eine Partitionierung von V in eine Menge T von **Terminalen** und eine Menge S von **Steinerknoten**. Gesucht ist ein Teilbaum $B = (V', E')$ von G, der alle Terminale enthält, d.h. $T \subseteq V'$, und der minimale Kosten $c(E') := \sum_{e \in E'} c(e)$ unter allen Teilbäumen mit dieser Eigenschaft hat.

zul. Lösung mit Kosten 4

optimale Lösung mit Kosten 3



■ Terminal

○ Steinerknoten

Metrisches SteinerTree

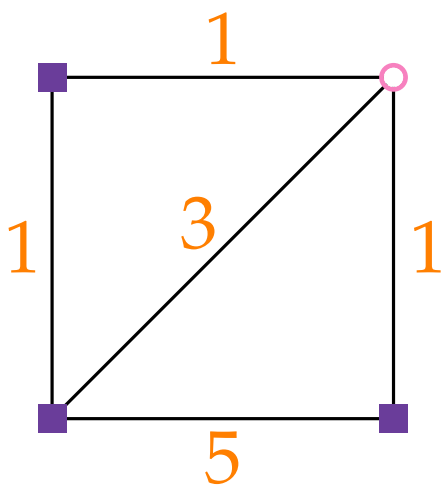
Einschränkung von SteinerTree, bei der der Graph G vollständig und metrisch sein muss, d.h.

Metrisches SteinerTree

Einschränkung von SteinerTree, bei der der Graph G vollständig und metrisch sein muss, d.h. für jedes Tripel u, v, w von Knoten gilt $c(u,w) \leq c(u,v) + c(v,w)$.

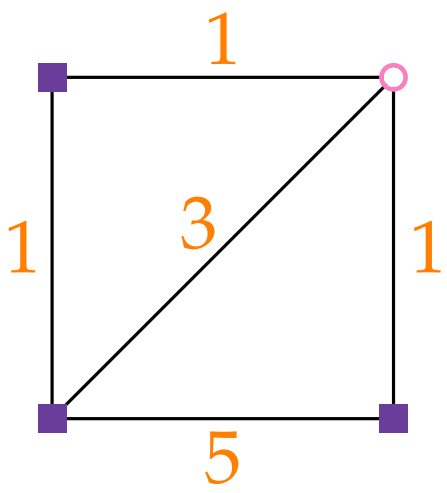
Metrisches SteinerTree

Einschränkung von SteinerTree, bei der der Graph G vollständig und metrisch sein muss, d.h. für jedes Tripel u, v, w von Knoten gilt $c(u,w) \leq c(u,v) + c(v,w)$.



Metrisches SteinerTree

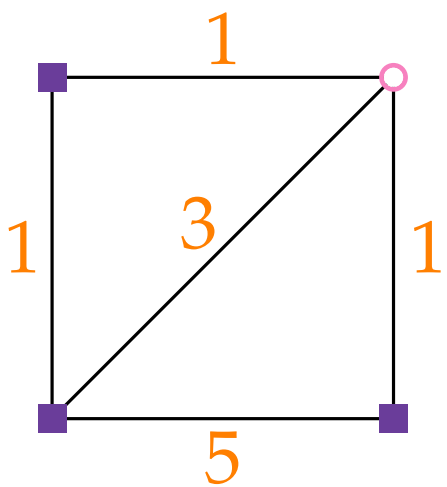
Einschränkung von SteinerTree, bei der der Graph G vollständig und metrisch sein muss, d.h. für jedes Tripel u, v, w von Knoten gilt $c(u,w) \leq c(u,v) + c(v,w)$.



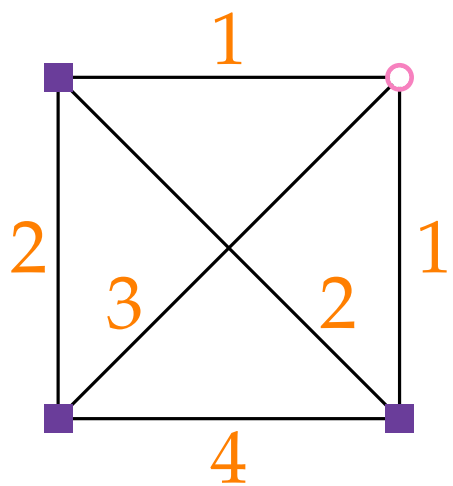
nicht vollständig
nicht metrisch

Metrisches SteinerTree

Einschränkung von SteinerTree, bei der der Graph G vollständig und metrisch sein muss, d.h. für jedes Tripel u, v, w von Knoten gilt $c(u,w) \leq c(u,v) + c(v,w)$.



nicht vollständig
nicht metrisch



vollständig
metrisch

Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme.

Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme.

Probleme Π_1 Π_2

Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme. Eine **approximationserhaltende Reduktion** von Π_1 auf Π_2 ist ein Paar (f, g) effizient berechenbarer Funktionen mit folgenden Eigenschaften:

Probleme Π_1 Π_2

Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme. Eine **approximationserhaltende Reduktion** von Π_1 auf Π_2 ist ein Paar (f, g) effizient berechenbarer Funktionen mit folgenden Eigenschaften:

(i) Für jede Instanz I_1 für Π_1 ist $I_2 := f(I_1)$ eine Instanz für Π_2 mit $\mathrm{OPT}_{\Pi_2}(I_2) \leq \mathrm{OPT}_{\Pi_1}(I_1)$.

Probleme Π_1 Π_2

Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme. Eine **approximationserhaltende Reduktion** von Π_1 auf Π_2 ist ein Paar (f, g) effizient berechenbarer Funktionen mit folgenden Eigenschaften:

(i) Für jede Instanz I_1 für Π_1 ist $I_2 := f(I_1)$ eine Instanz für Π_2 mit $\mathrm{OPT}_{\Pi_2}(I_2) \leq \mathrm{OPT}_{\Pi_1}(I_1)$.

Probleme	Π_1	Π_2
Instanzen	I_1	

Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme. Eine **approximationserhaltende Reduktion** von Π_1 auf Π_2 ist ein Paar (f, g) effizient berechenbarer Funktionen mit folgenden Eigenschaften:

(i) Für jede Instanz I_1 für Π_1 ist $I_2 := f(I_1)$ eine Instanz für Π_2 mit $\mathrm{OPT}_{\Pi_2}(I_2) \leq \mathrm{OPT}_{\Pi_1}(I_1)$.

Probleme	Π_1		Π_2
Instanzen	I_1	$\xrightarrow{f}$	I_2

Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme. Eine **approximationserhaltende Reduktion** von Π_1 auf Π_2 ist ein Paar (f, g) effizient berechenbarer Funktionen mit folgenden Eigenschaften:

(i) Für jede Instanz I_1 für Π_1 ist $I_2 := f(I_1)$ eine Instanz für Π_2 mit $\mathrm{OPT}_{\Pi_2}(I_2) \leq \mathrm{OPT}_{\Pi_1}(I_1)$.

(ii) Für jede Lösung t von I_2 ist $s := g(I_1, t)$ eine Lösung für I_1 mit $\mathrm{obj}_{\Pi_1}(I_1, s) \leq \mathrm{obj}_{\Pi_2}(I_2, t)$.

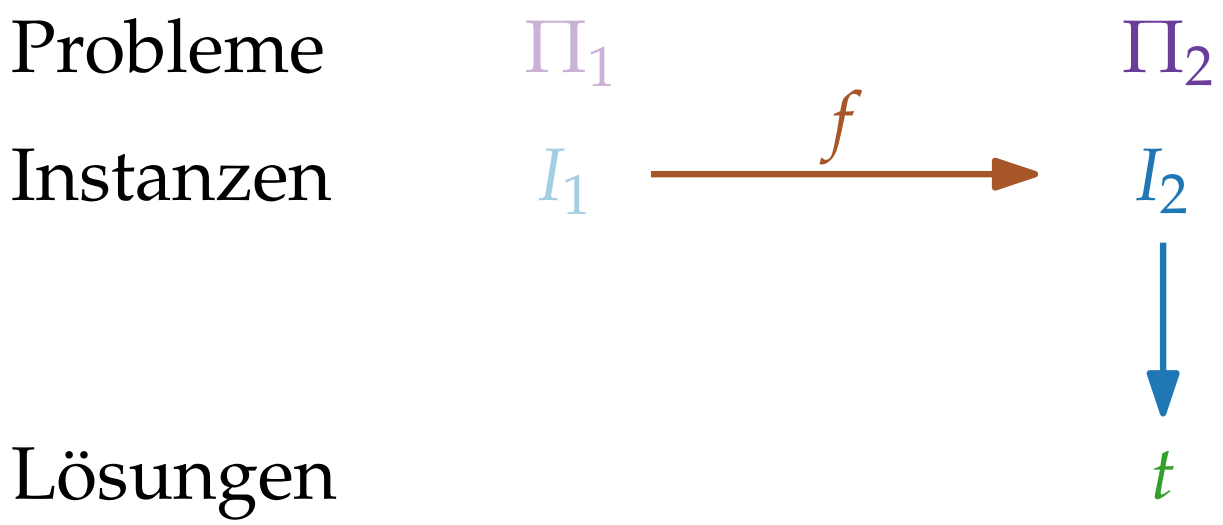
Probleme	Π_1		Π_2
Instanzen	I_1	$\xrightarrow{f}$	I_2

Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme. Eine **approximationserhaltende Reduktion** von Π_1 auf Π_2 ist ein Paar (f, g) effizient berechenbarer Funktionen mit folgenden Eigenschaften:

(i) Für jede Instanz I_1 für Π_1 ist $I_2 := f(I_1)$ eine Instanz für Π_2 mit $\mathrm{OPT}_{\Pi_2}(I_2) \leq \mathrm{OPT}_{\Pi_1}(I_1)$.

(ii) Für jede Lösung t von I_2 ist $s := g(I_1, t)$ eine Lösung für I_1 mit $\mathrm{obj}_{\Pi_1}(I_1, s) \leq \mathrm{obj}_{\Pi_2}(I_2, t)$.

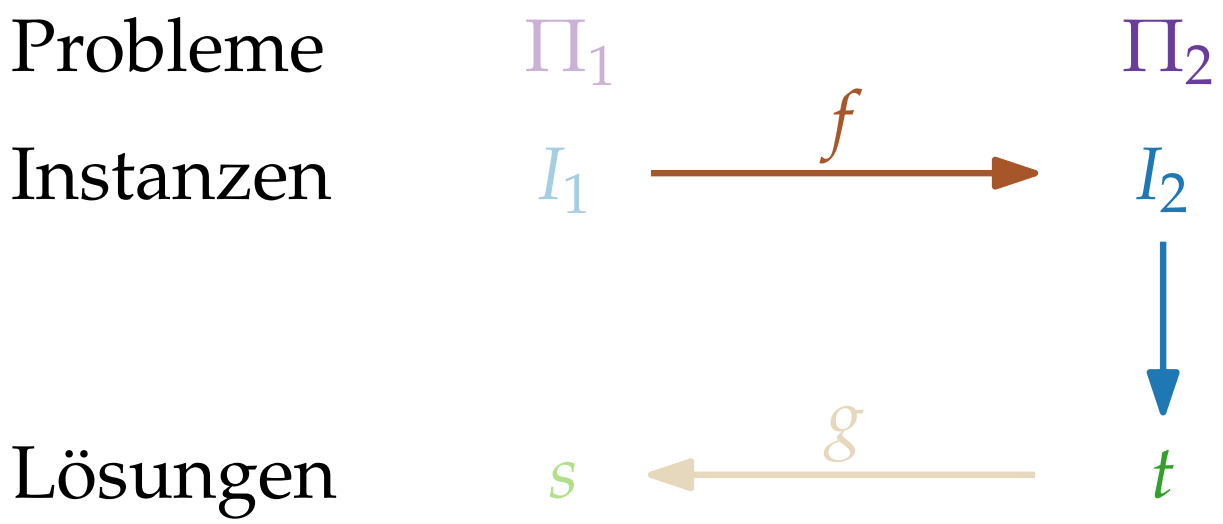


Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme. Eine **approximationserhaltende Reduktion** von Π_1 auf Π_2 ist ein Paar (f, g) effizient berechenbarer Funktionen mit folgenden Eigenschaften:

(i) Für jede Instanz I_1 für Π_1 ist $I_2 := f(I_1)$ eine Instanz für Π_2 mit $\mathrm{OPT}_{\Pi_2}(I_2) \leq \mathrm{OPT}_{\Pi_1}(I_1)$.

(ii) Für jede Lösung t von I_2 ist $s := g(I_1, t)$ eine Lösung für I_1 mit $\mathrm{obj}_{\Pi_1}(I_1, s) \leq \mathrm{obj}_{\Pi_2}(I_2, t)$.

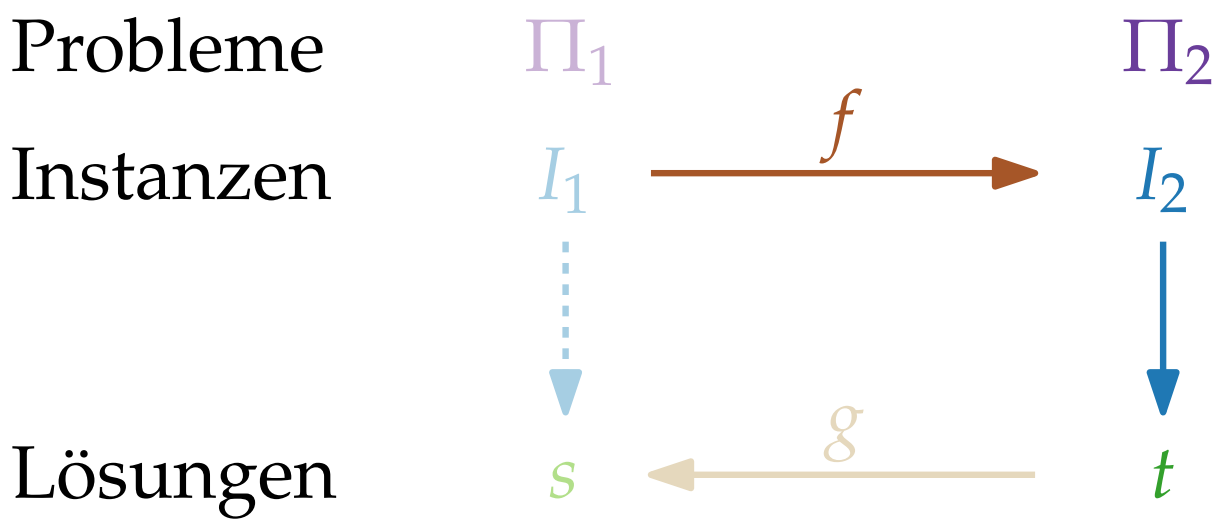


Approximationserhaltende Reduktion

Seien Π_1, Π_2 Minimierungsprobleme. Eine **approximationserhaltende Reduktion** von Π_1 auf Π_2 ist ein Paar (f, g) effizient berechenbarer Funktionen mit folgenden Eigenschaften:

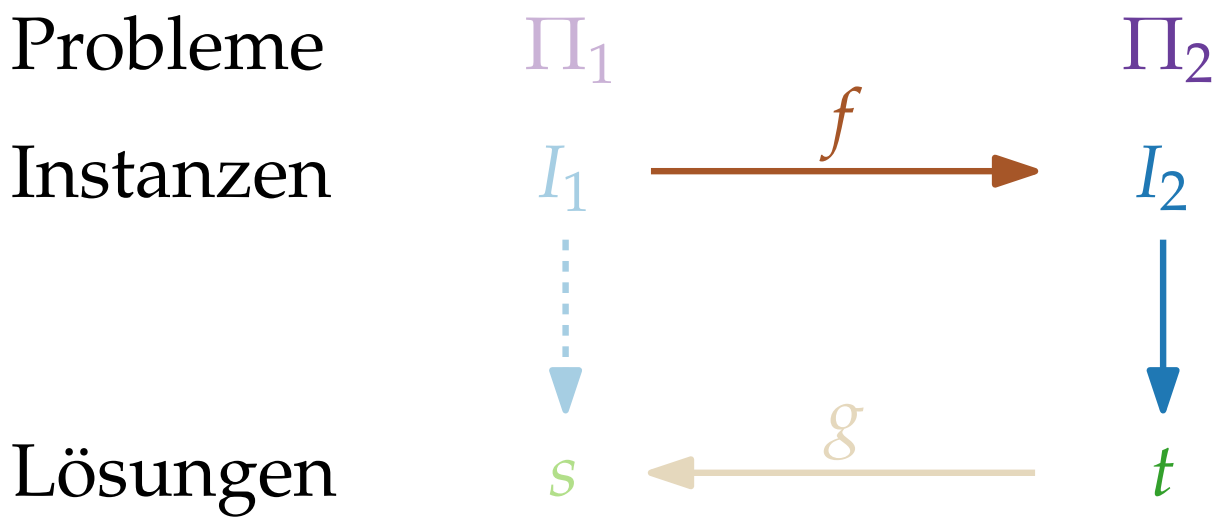
(i) Für jede Instanz I_1 für Π_1 ist $I_2 := f(I_1)$ eine Instanz für Π_2 mit $\mathrm{OPT}_{\Pi_2}(I_2) \leq \mathrm{OPT}_{\Pi_1}(I_1)$.

(ii) Für jede Lösung t von I_2 ist $s := g(I_1, t)$ eine Lösung für I_1 mit $\mathrm{obj}_{\Pi_1}(I_1, s) \leq \mathrm{obj}_{\Pi_2}(I_2, t)$.



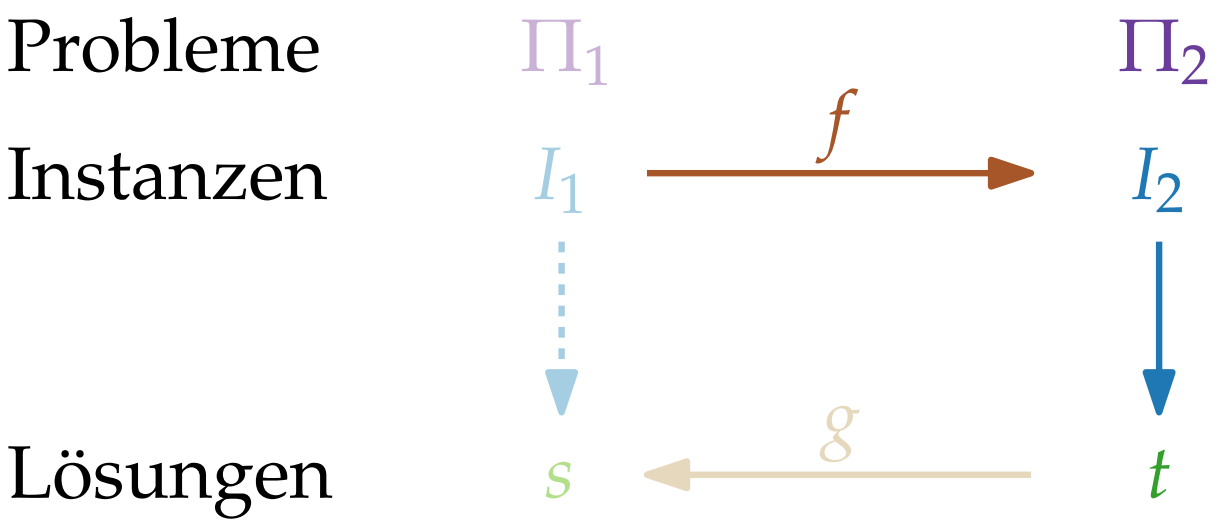
Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt.



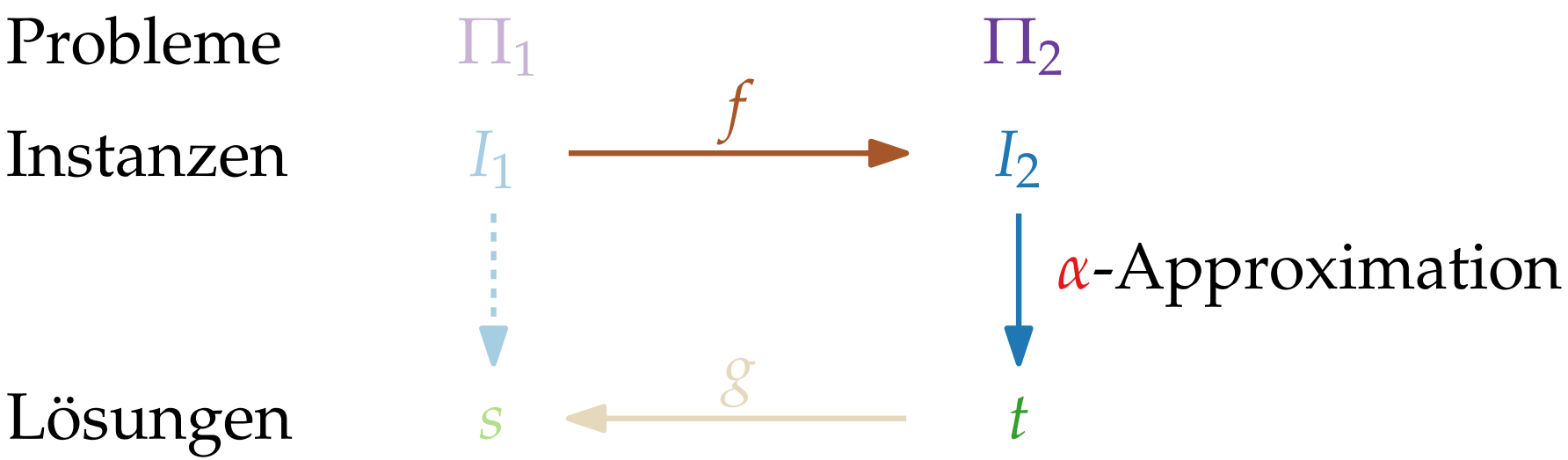
Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor ? Approximationsalgorithmus für Π_1.



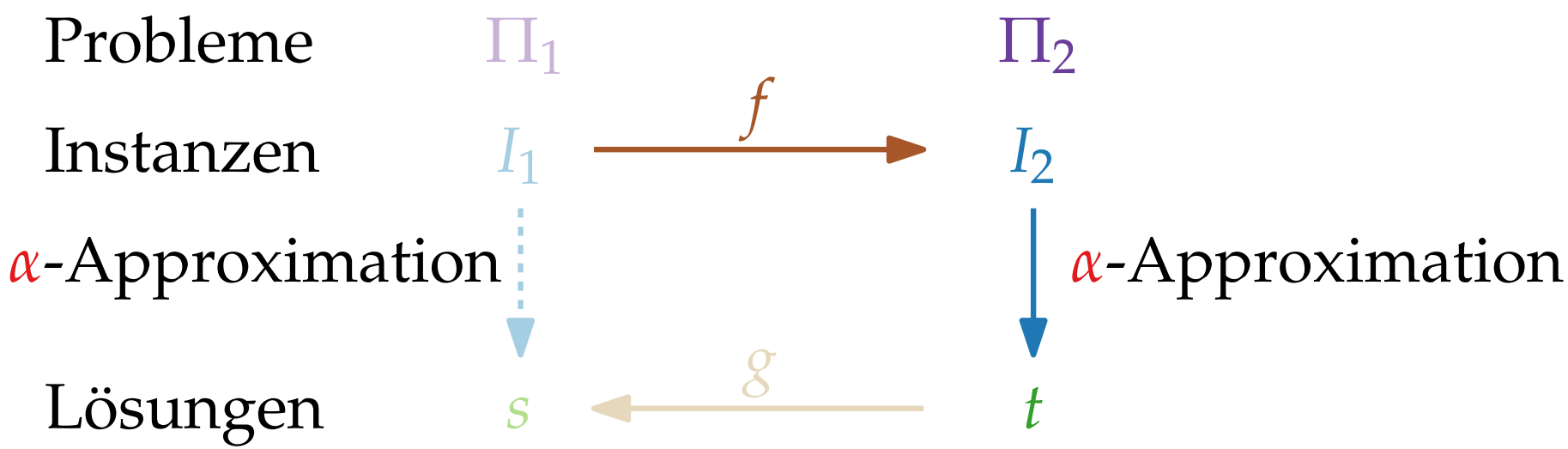
Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor ? Approximationsalgorithmus für Π_1.



Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

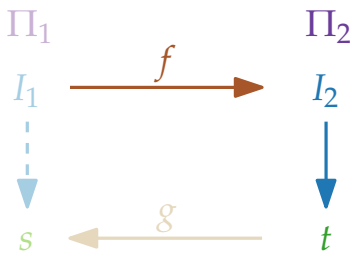


Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.

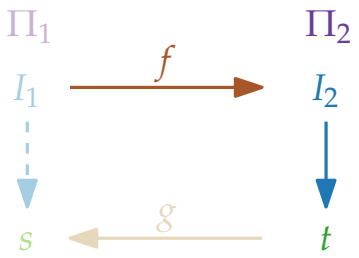
Sei A ein Faktor-α-Approx.-Alg. für Π_2.



Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.



Sei A ein Faktor-α-Approx.-Alg. für Π_2.

Sei I_1 eine Instanz von Π_1.

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.

$$\begin{tikzcd} \Pi_1 & & \Pi_2 \\ I_1 \arrow[rr, "f"] \arrow[d, dashed] & & I_2 \arrow[d] \\ s & & t \arrow[ll, "g"'] \end{tikzcd}$$

Sei A ein Faktor-α-Approx.-Alg. für Π_2.

Sei I_1 eine Instanz von Π_1.

Setze $I_2 :=$, $t :=$ und $s :=$.

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.

$$\begin{tikzcd} \Pi_1 & & \Pi_2 \\ I_1 \arrow[r, "f"] \arrow[d, dashed] & & I_2 \arrow[d] \\ s & & t \arrow[l, "g"'] \end{tikzcd}$$

Sei A ein Faktor-α-Approx.-Alg. für Π_2.

Sei I_1 eine Instanz von Π_1.

Setze $I_2 := f(I_1)$, $t :=$ und $s :=$.

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.

$$\begin{tikzcd} \Pi_1 & \Pi_2 \\ I_1 \arrow[r, "f"] \arrow[d, dashed] & I_2 \arrow[d, "A"] \\ s & t \arrow[l, "g"'] \end{tikzcd}$$

Sei A ein Faktor-α-Approx.-Alg. für Π_2.

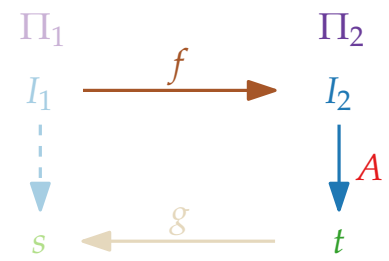
Sei I_1 eine Instanz von Π_1.

Setze $I_2 := f(I_1)$, $t := A(I_2)$ und $s :=$.

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.



Sei A ein Faktor-α-Approx.-Alg. für Π_2.

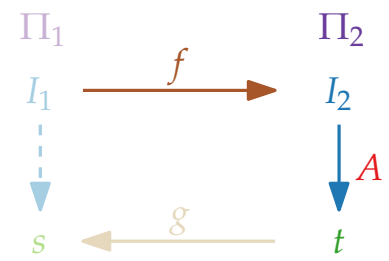
Sei I_1 eine Instanz von Π_1.

Setze $I_2 := f(I_1)$, $t := A(I_2)$ und $s := g(I_1, t)$.

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.



Sei A ein Faktor-α-Approx.-Alg. für Π_2.

Sei I_1 eine Instanz von Π_1.

Setze $I_2 := f(I_1)$, $t := A(I_2)$ und $s := g(I_1, t)$.

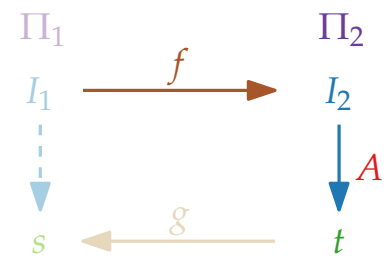
Dann gilt:

$\mathrm{obj}_{\Pi_1}(I_1, s)$

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.



Sei A ein Faktor-α-Approx.-Alg. für Π_2.

Sei I_1 eine Instanz von Π_1.

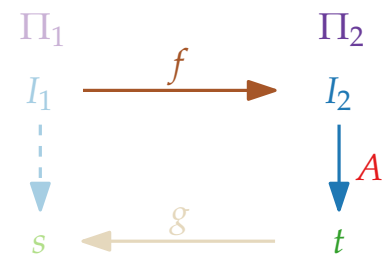
Setze $I_2 := f(I_1)$, $t := A(I_2)$ und $s := g(I_1, t)$.

Dann gilt:
$\mathrm{obj}_{\Pi_1}(I_1, s) \leq \mathrm{obj}_{\Pi_2}(I_2, t)$

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.



Sei A ein Faktor-α-Approx.-Alg. für Π_2.

Sei I_1 eine Instanz von Π_1.

Setze $I_2 := f(I_1)$, $t := A(I_2)$ und $s := g(I_1, t)$.

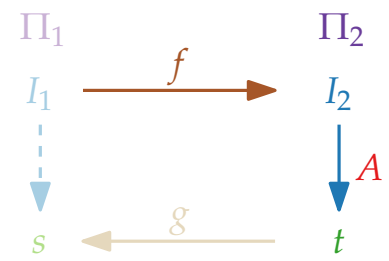
Dann gilt:

$$\mathrm{obj}_{\Pi_1}(I_1, s) \leq \mathrm{obj}_{\Pi_2}(I_2, t) \leq \alpha \cdot \mathrm{OPT}_{\Pi_2}(I_2)$$

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion (f, g) von Π_1 auf Π_2 gibt. Dann gibt es für jeden Faktor-α-Approximationsalgorithmus für Π_2 auch einen Faktor-α-Approximationsalgorithmus für Π_1.

Beweis.



Sei A ein Faktor-α-Approx.-Alg. für Π_2.

Sei I_1 eine Instanz von Π_1.

Setze $I_2 := f(I_1)$, $t := A(I_2)$ und $s := g(I_1, t)$.

Dann gilt:

$$\mathrm{obj}_{\Pi_1}(I_1, s) \leq \mathrm{obj}_{\Pi_2}(I_2, t) \leq \alpha \cdot \mathrm{OPT}_{\Pi_2}(I_2) \leq \alpha \cdot \mathrm{OPT}_{\Pi_1}(I_1)$$

Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $I_1 \xrightarrow{f} I_2$

Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

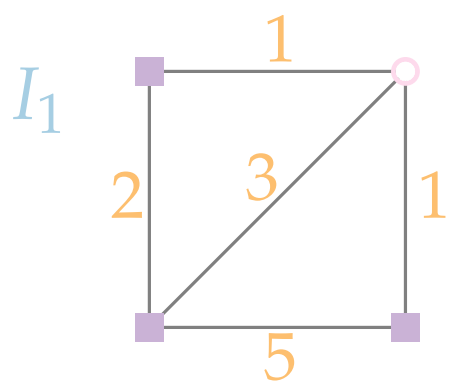
Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$



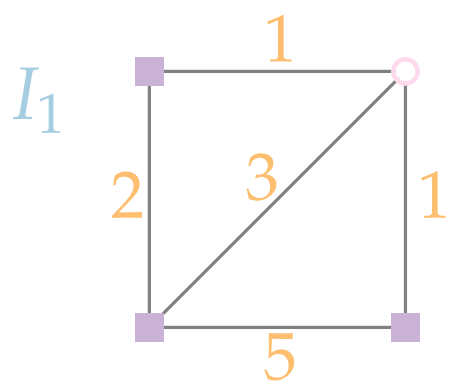
Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1



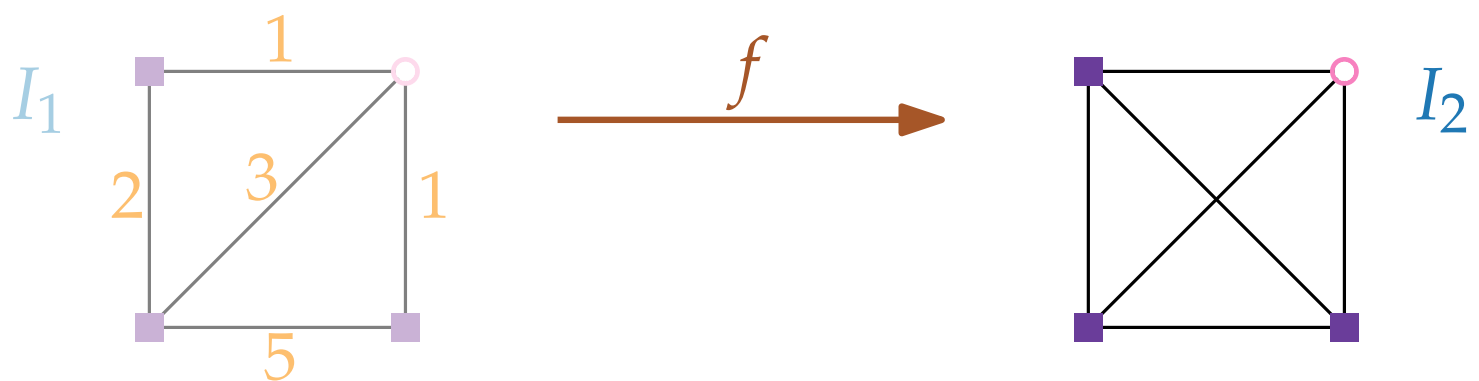
Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1



Metrisches SteinerTree

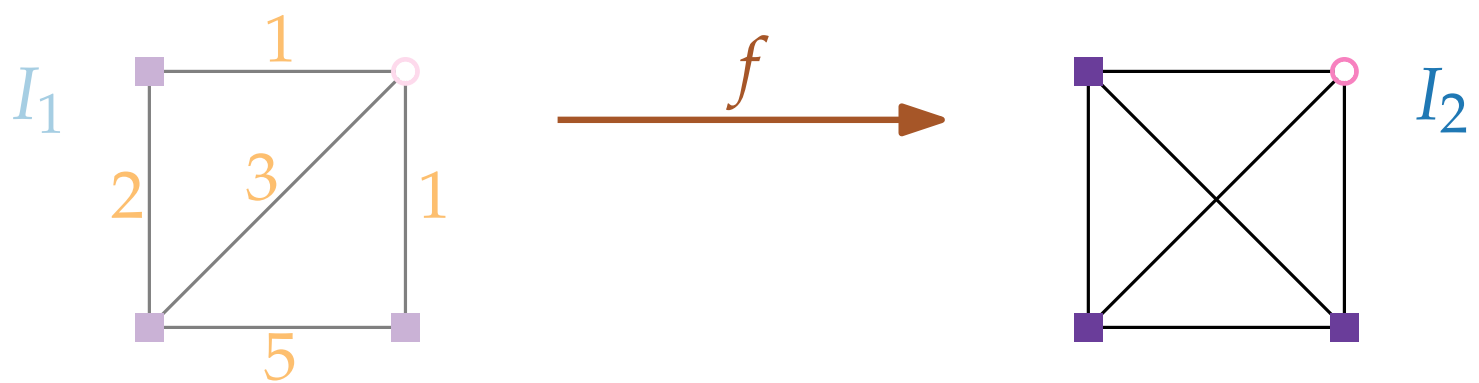
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

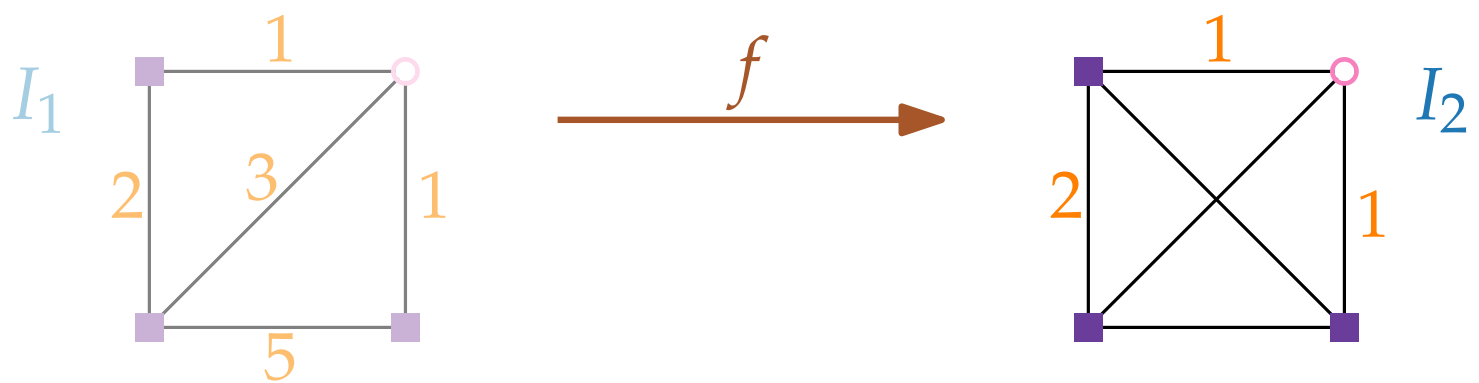
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

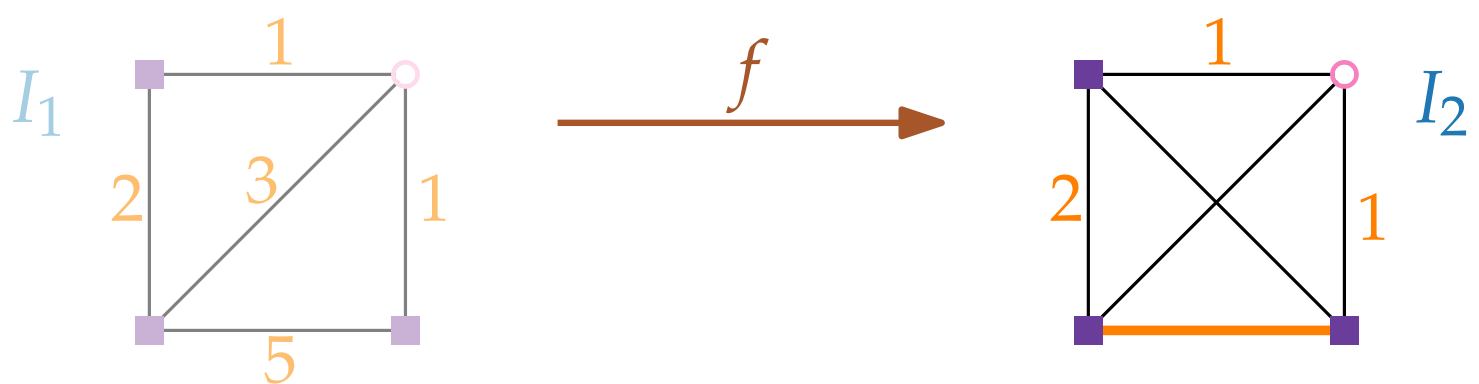
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

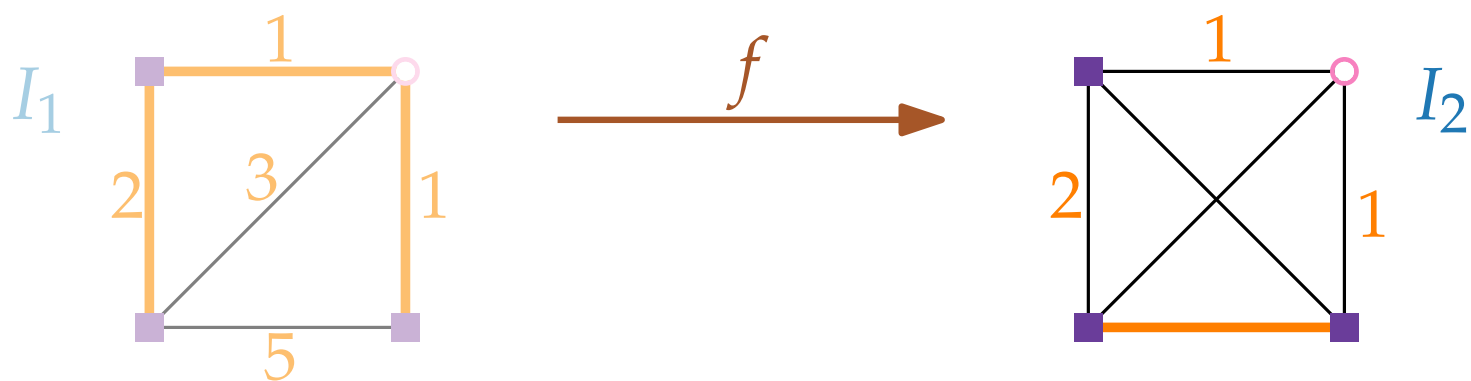
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

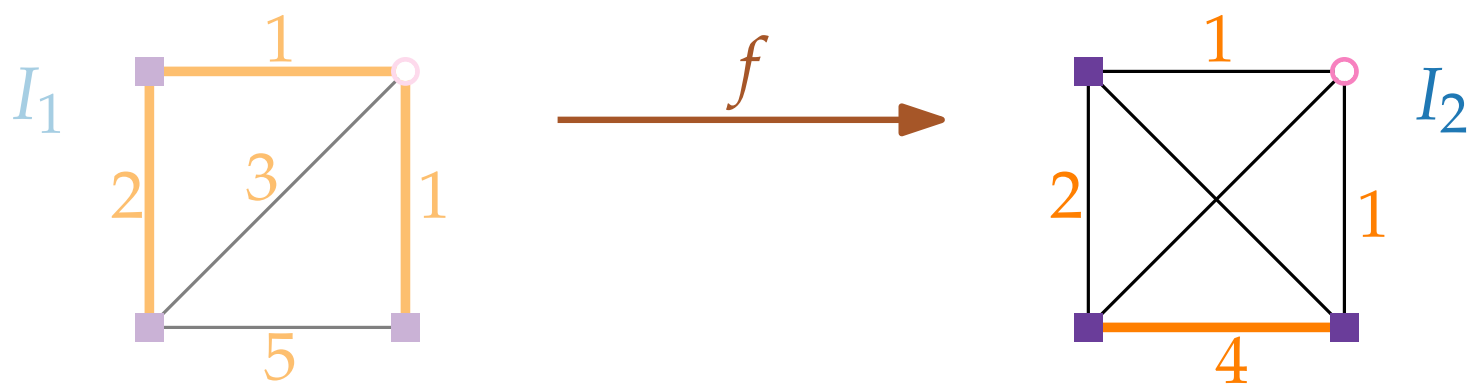
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

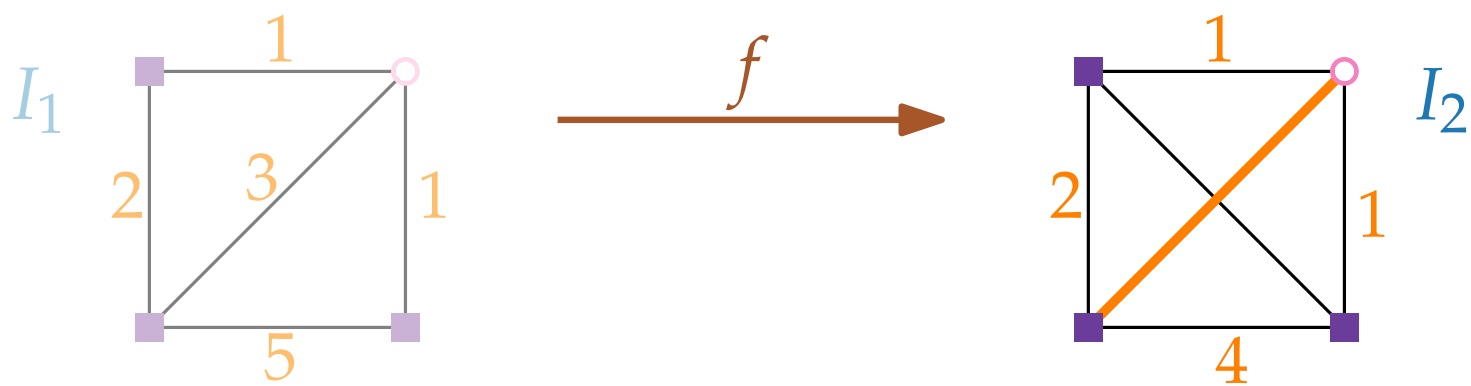
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

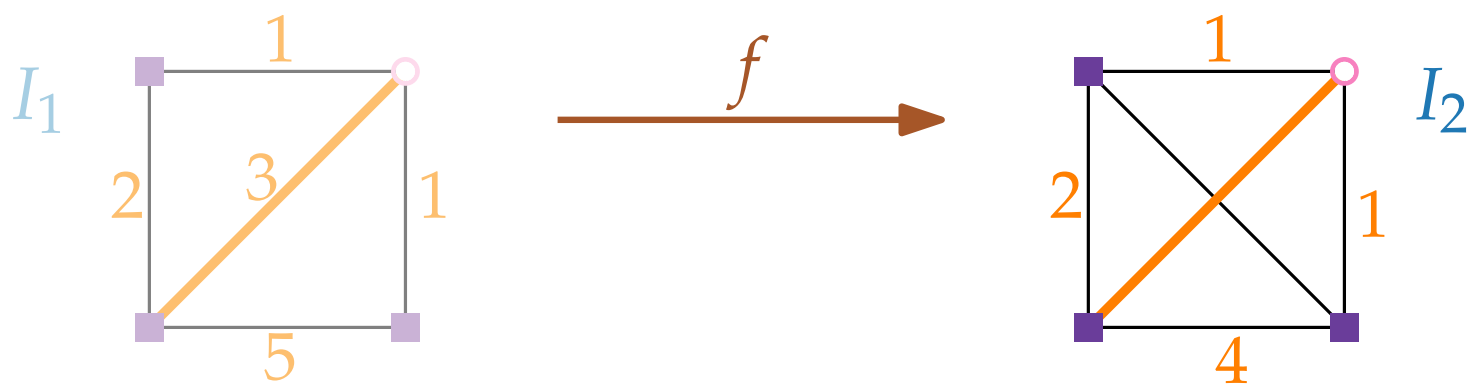
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

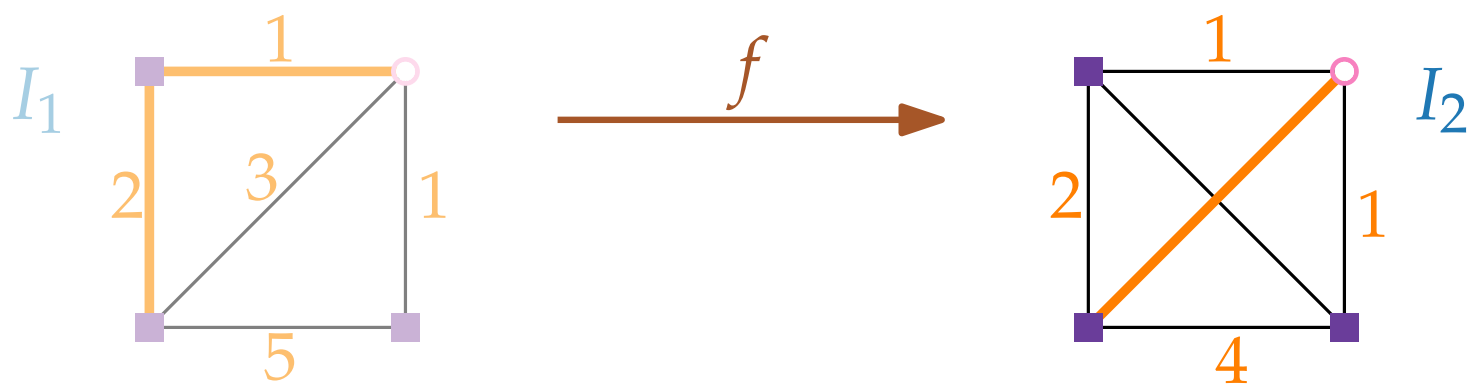
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

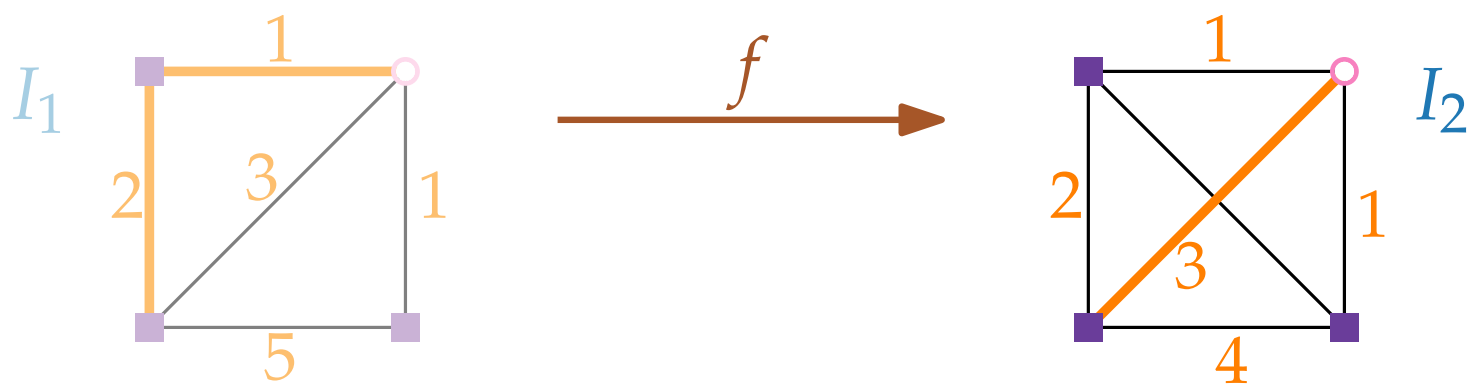
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

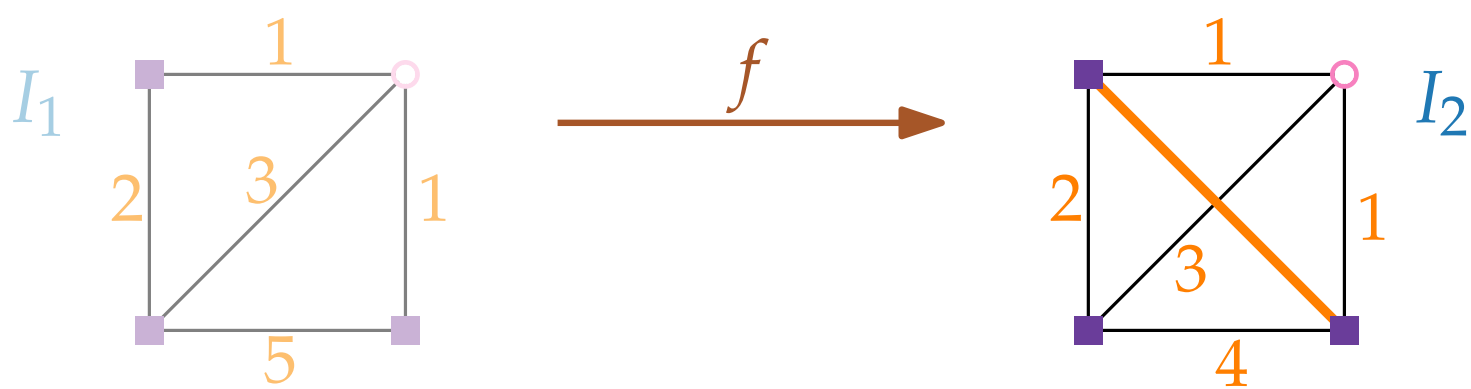
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

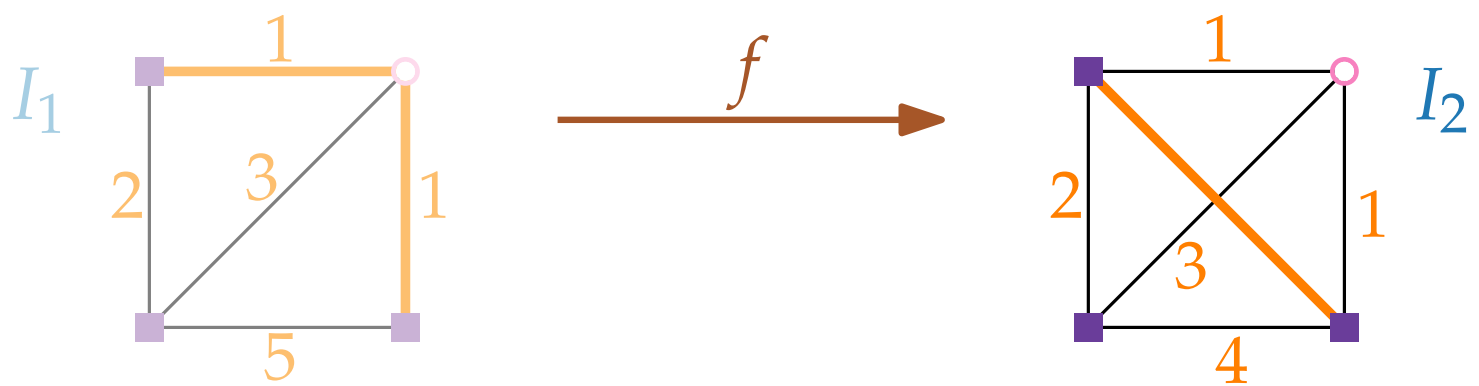
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

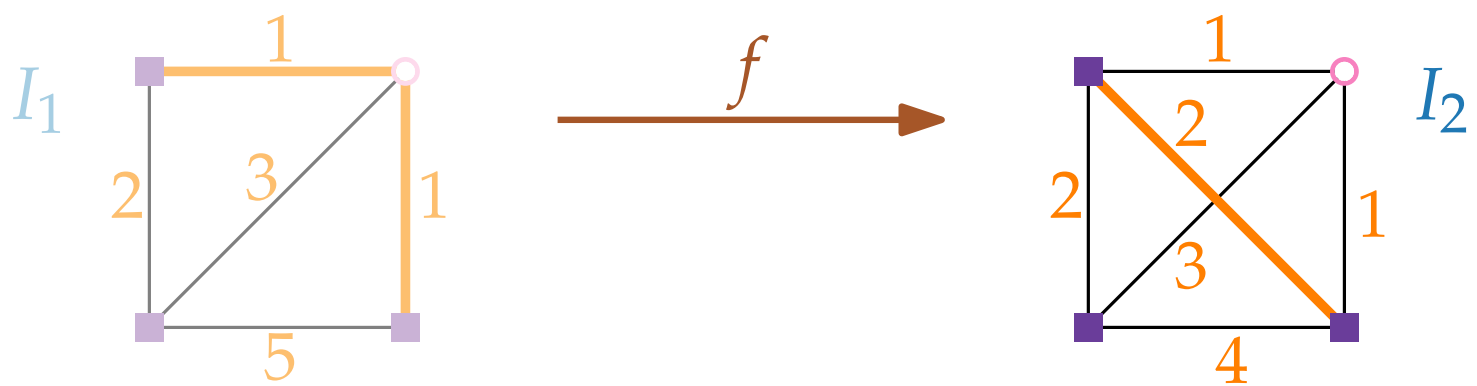
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

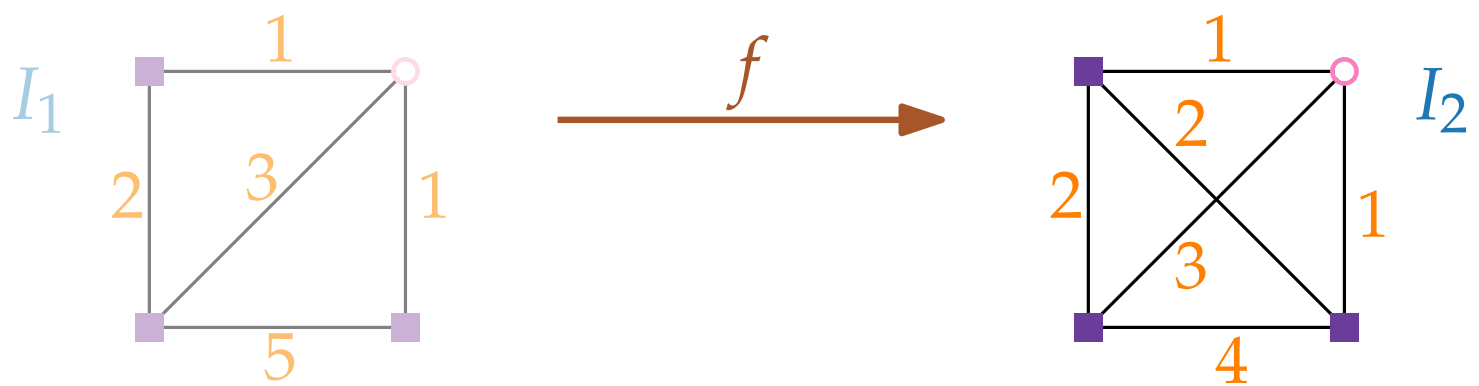
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1



Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

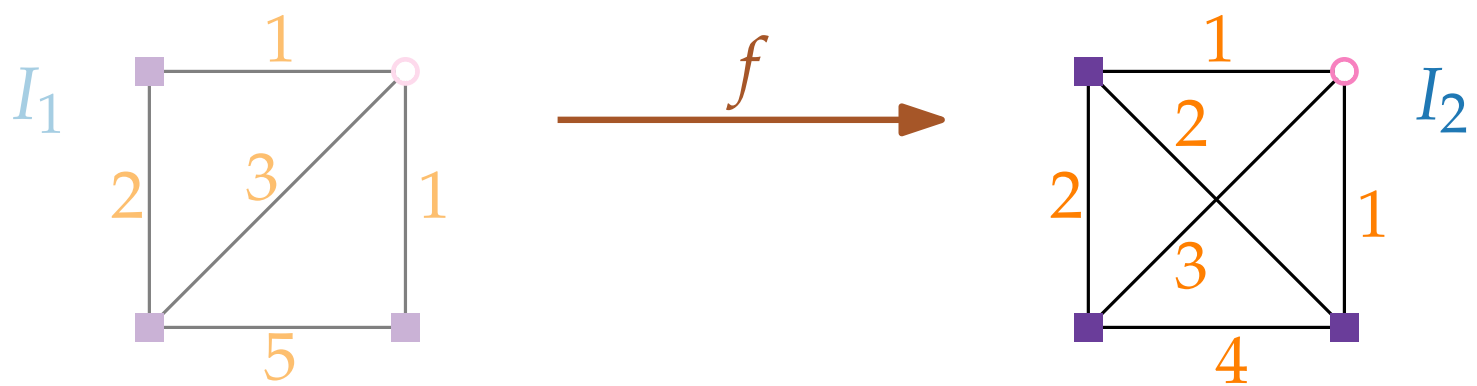
Beweis. (1) Abbildung f $\quad I_1 \xrightarrow{f} I_2$

Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = T \mathbin{\dot\cup} S$

Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch T, S wie bei I_1

$c_2(u,v) :=$ Länge eines kürzesten u–v-Weges in G_1

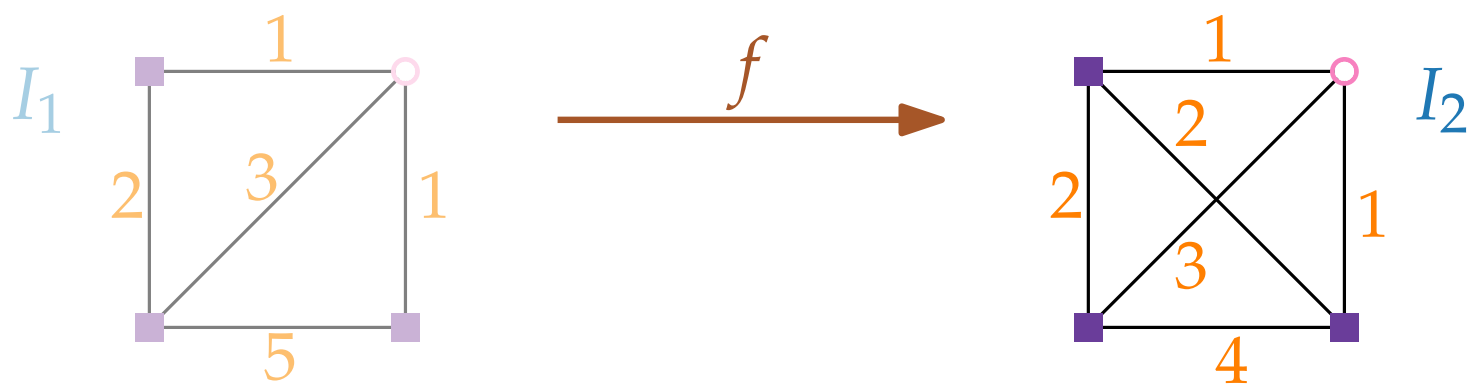
$c_2(u,v) \leq c_1(u,v)$ für alle $(u,v) \in E$



Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (2) $\mathrm{OPT}(I_2) \leq \mathrm{OPT}(I_1)$

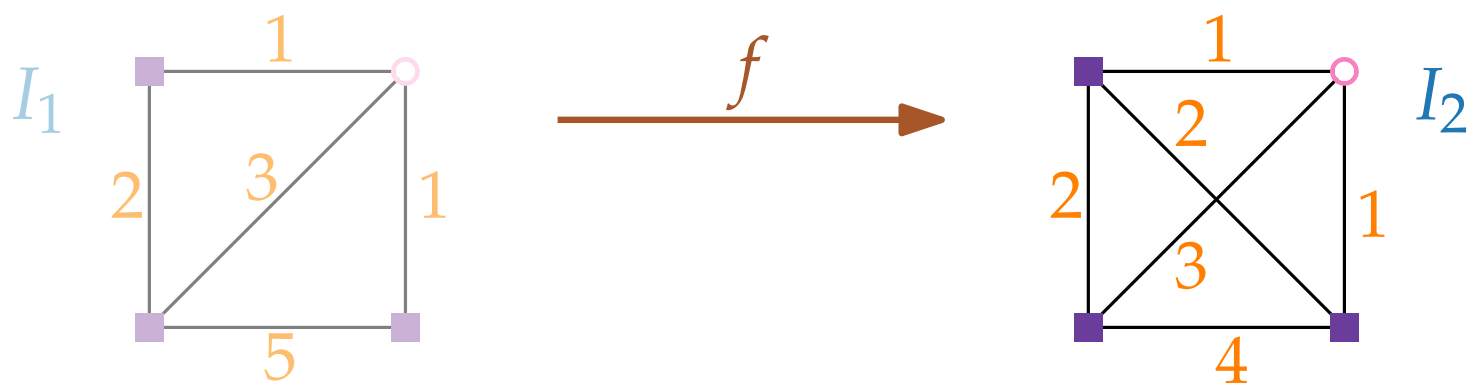


Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (2) $\mathrm{OPT}(I_2) \leq \mathrm{OPT}(I_1)$

Sei B^* optimaler Steiner Baum für I_1

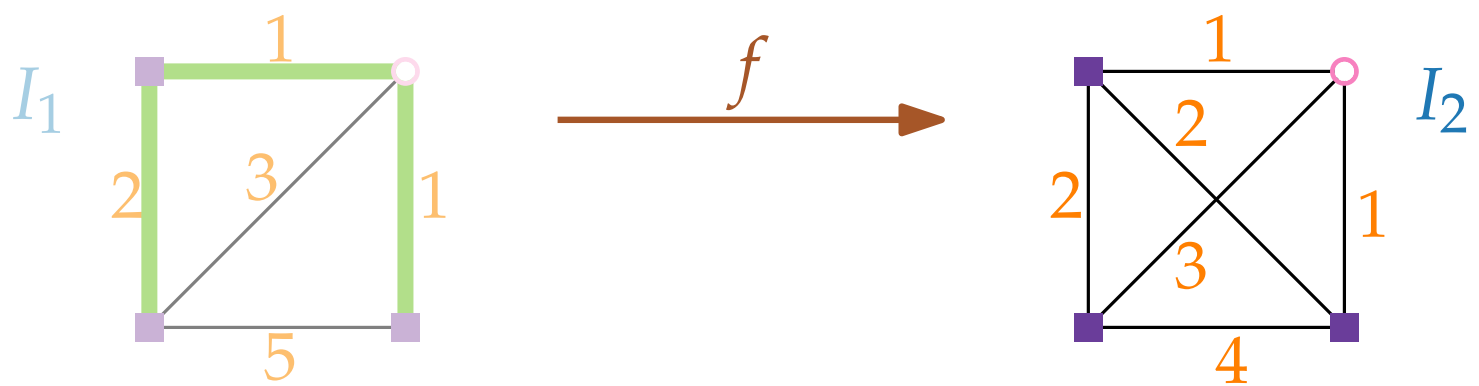


Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (2) $\mathrm{OPT}(I_2) \leq \mathrm{OPT}(I_1)$

Sei B^* optimaler Steiner Baum für I_1



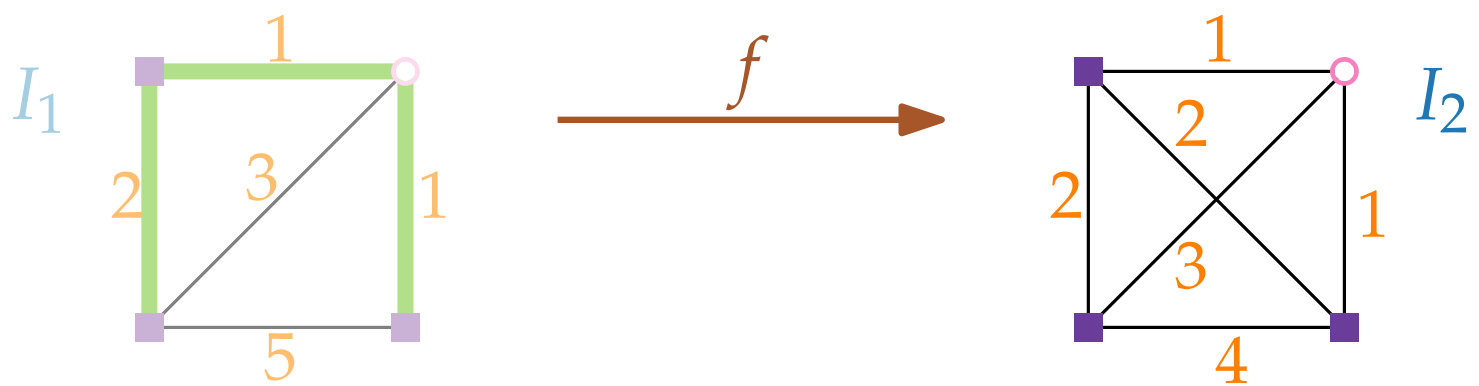
Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (2) $\mathrm{OPT}(I_2) \leq \mathrm{OPT}(I_1)$

Sei B^* optimaler Steiner Baum für I_1

B^* ist auch zulässige Lösung für I_2, da $E_1 \subseteq E_2$ und Knotenmengen V, T, S gleich



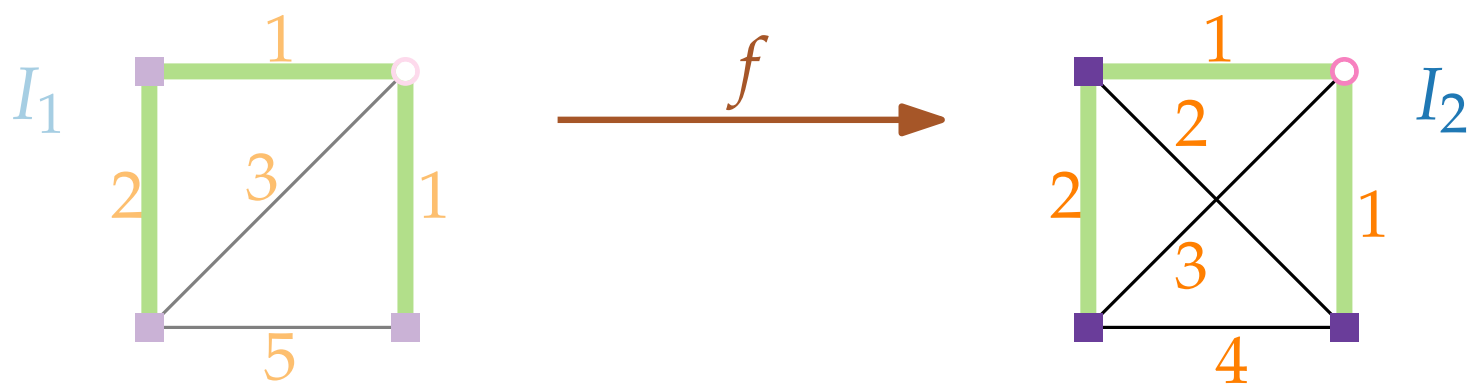
Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (2) $\mathrm{OPT}(I_2) \leq \mathrm{OPT}(I_1)$

Sei B^* optimaler Steiner Baum für I_1

B^* ist auch zulässige Lösung für I_2, da $E_1 \subseteq E_2$ und Knotenmengen V, T, S gleich



Metrisches SteinerTree

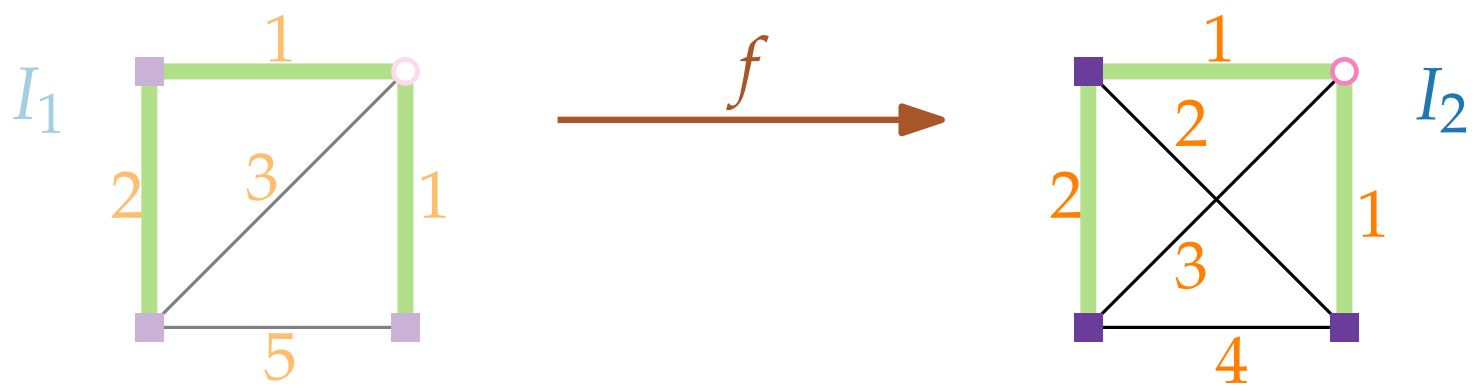
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (2) $\mathrm{OPT}(I_2) \leq \mathrm{OPT}(I_1)$

Sei B^* optimaler Steiner Baum für I_1

B^* ist auch zulässige Lösung für I_2, da $E_1 \subseteq E_2$ und Knotenmengen V, T, S gleich

$\mathrm{OPT}(I_2)$



Metrisches SteinerTree

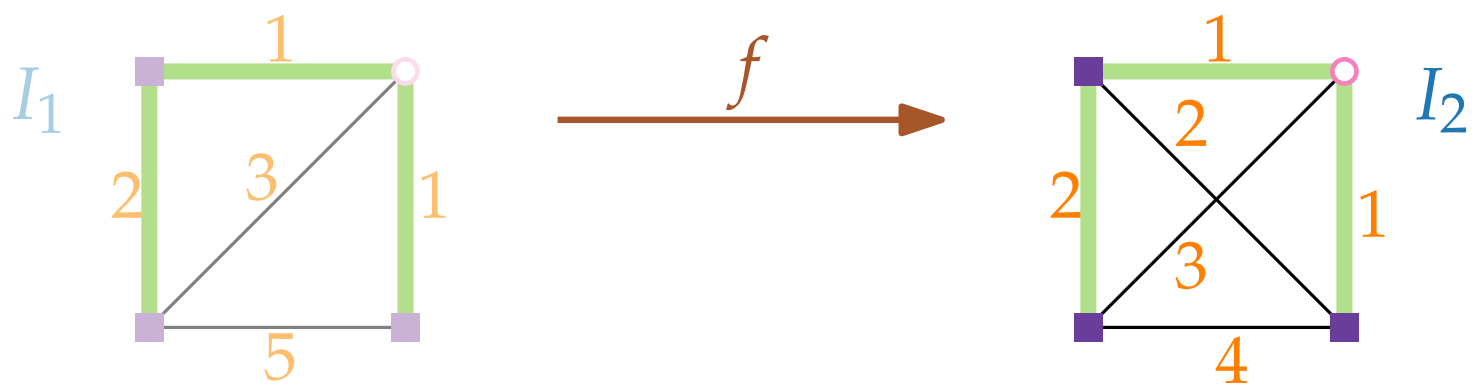
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (2) $\mathrm{OPT}(I_2) \leq \mathrm{OPT}(I_1)$

Sei B^* optimaler Steiner Baum für I_1

B^* ist auch zulässige Lösung für I_2, da $E_1 \subseteq E_2$ und Knotenmengen V, T, S gleich

$\mathrm{OPT}(I_2) \leq c_2(B^*)$



Metrisches SteinerTree

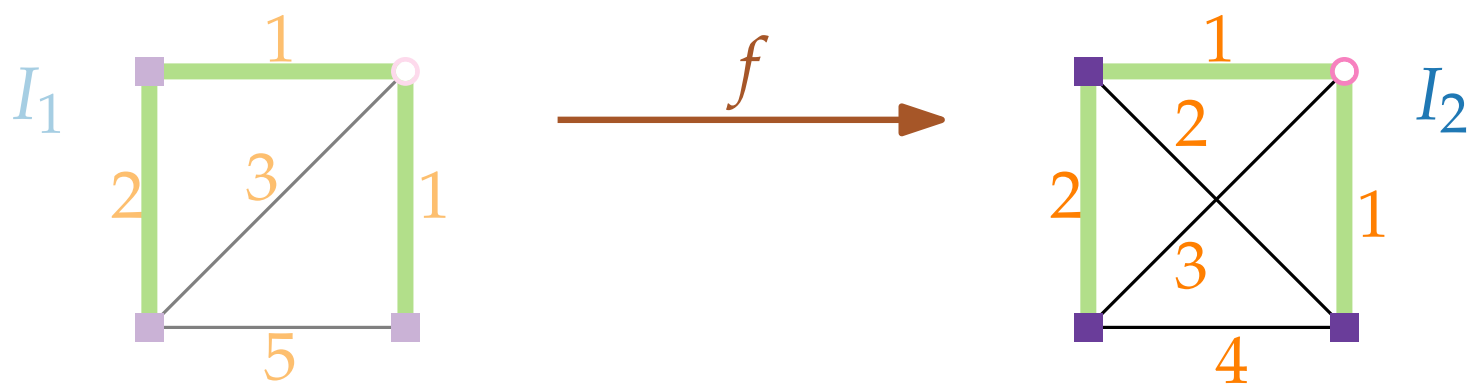
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (2) $\mathrm{OPT}(I_2) \leq \mathrm{OPT}(I_1)$

Sei B^* optimaler Steiner Baum für I_1

B^* ist auch zulässige Lösung für I_2, da $E_1 \subseteq E_2$ und Knotenmengen V, T, S gleich

$\mathrm{OPT}(I_2) \leq c_2(B^*) \leq c_1(B^*)$



Metrisches SteinerTree

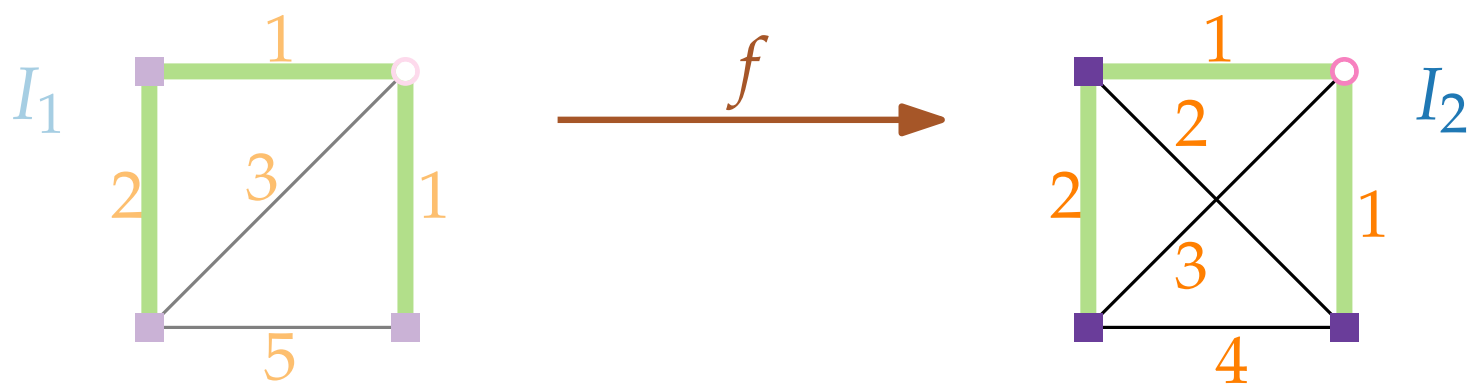
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (2) $\mathrm{OPT}(I_2) \leq \mathrm{OPT}(I_1)$

Sei B^* optimaler Steiner Baum für I_1

B^* ist auch zulässige Lösung für I_2, da $E_1 \subseteq E_2$ und Knotenmengen V, T, S gleich

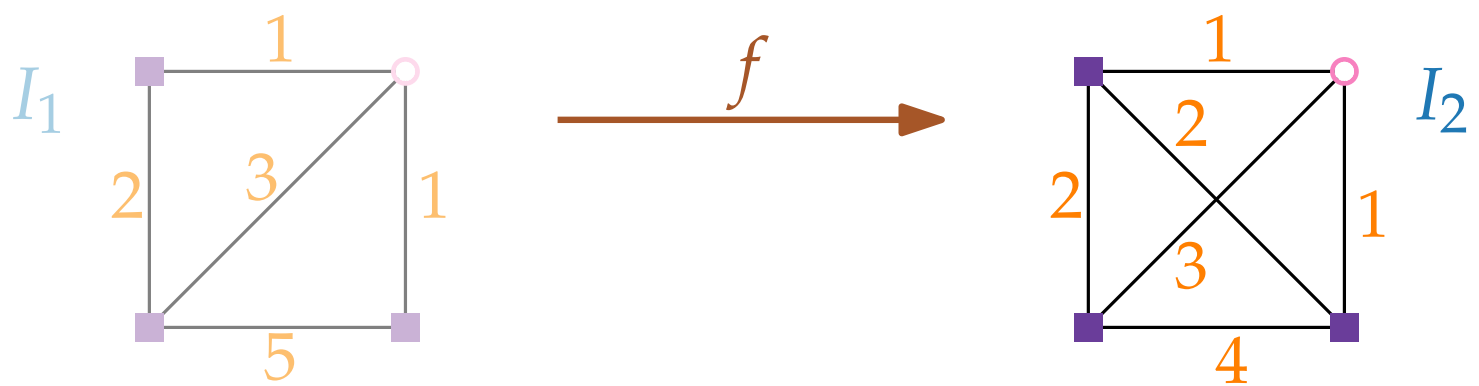
$\mathrm{OPT}(I_2) \leq c_2(B^*) \leq c_1(B^*) = \mathrm{OPT}(I_1)$



Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $\quad s \xleftarrow{g} t$

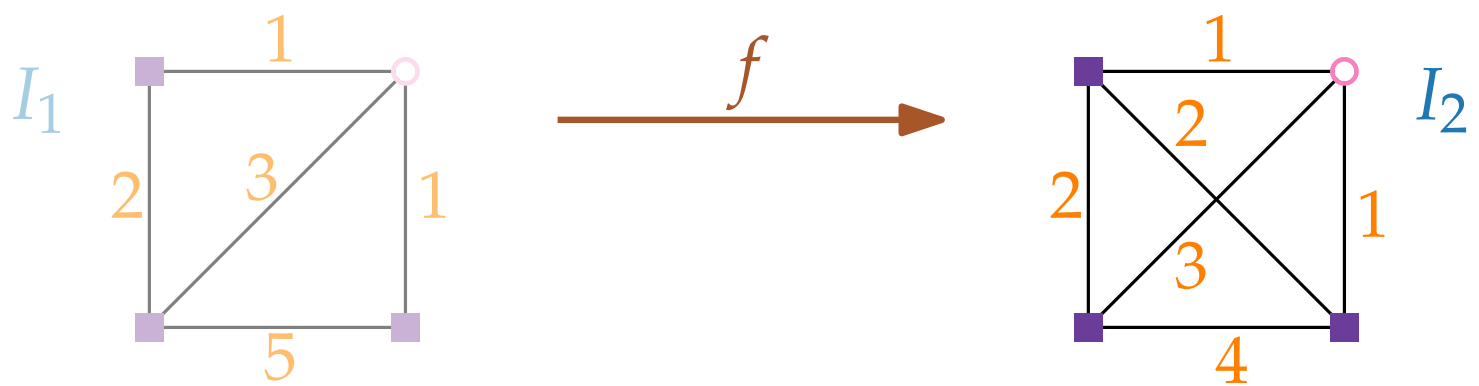


Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $\quad s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

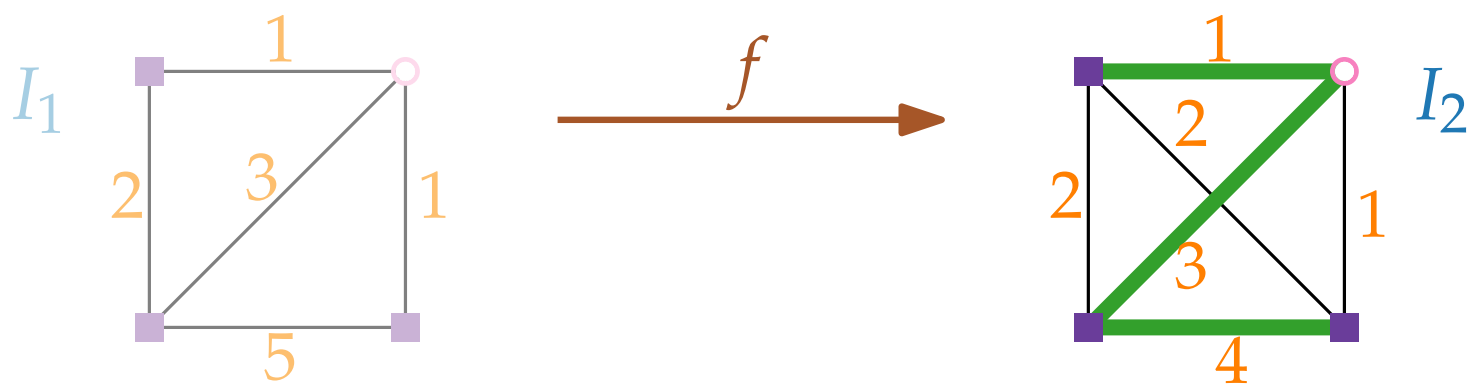


Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $\quad s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2



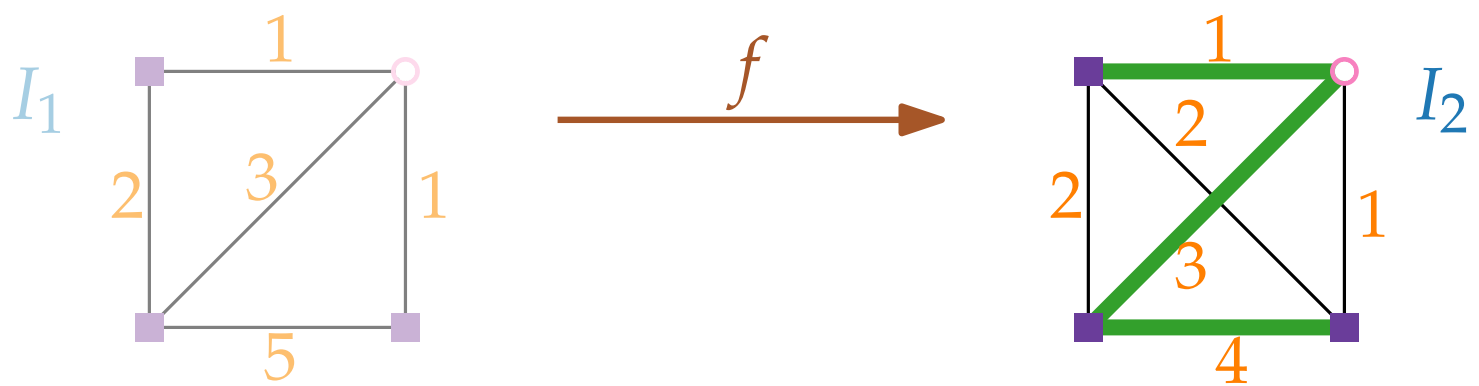
Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $\quad s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt



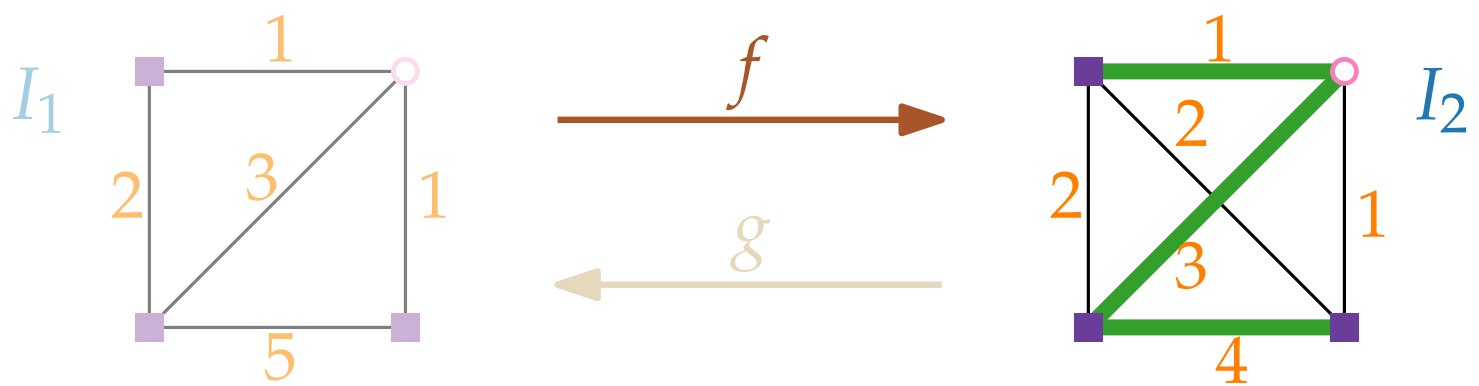
Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt



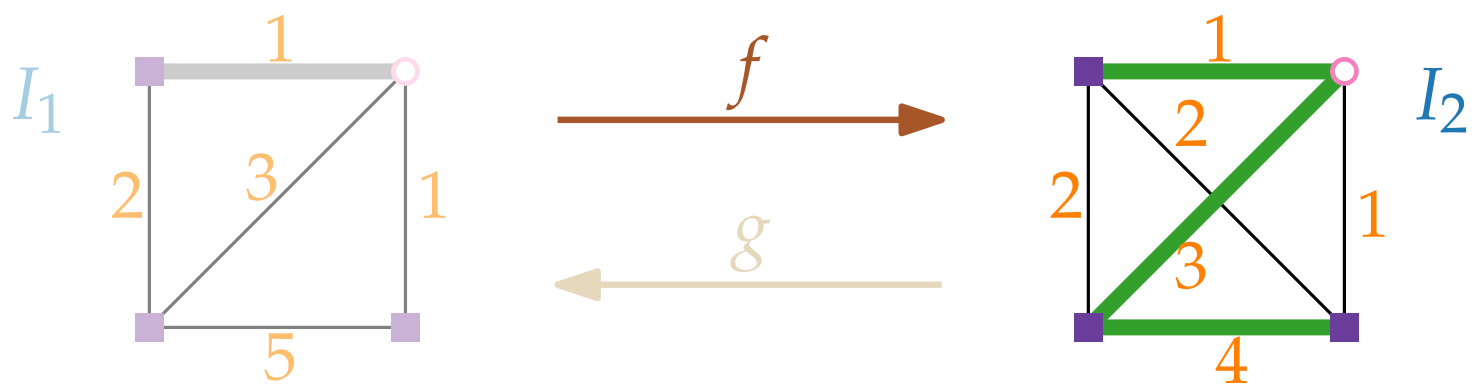
Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt



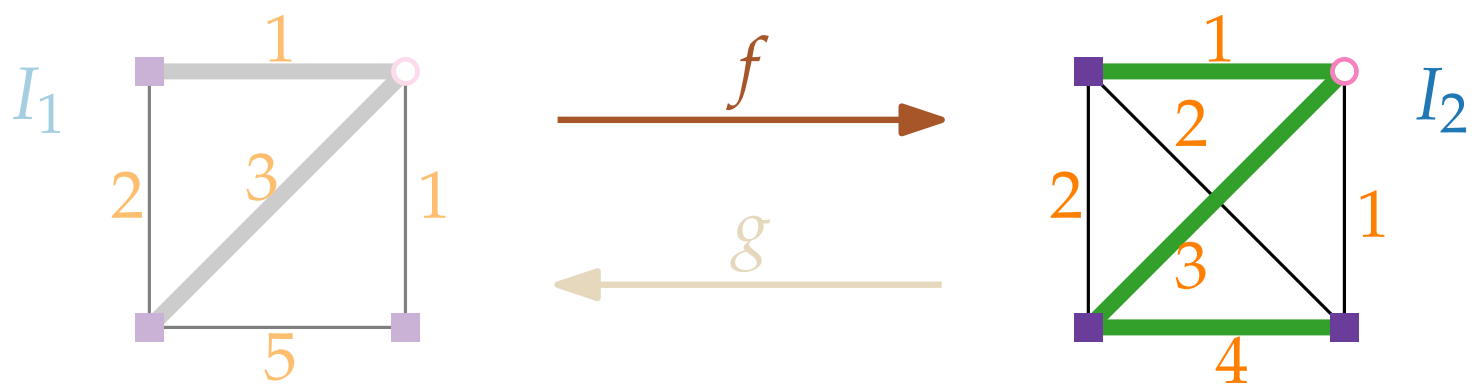
Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt



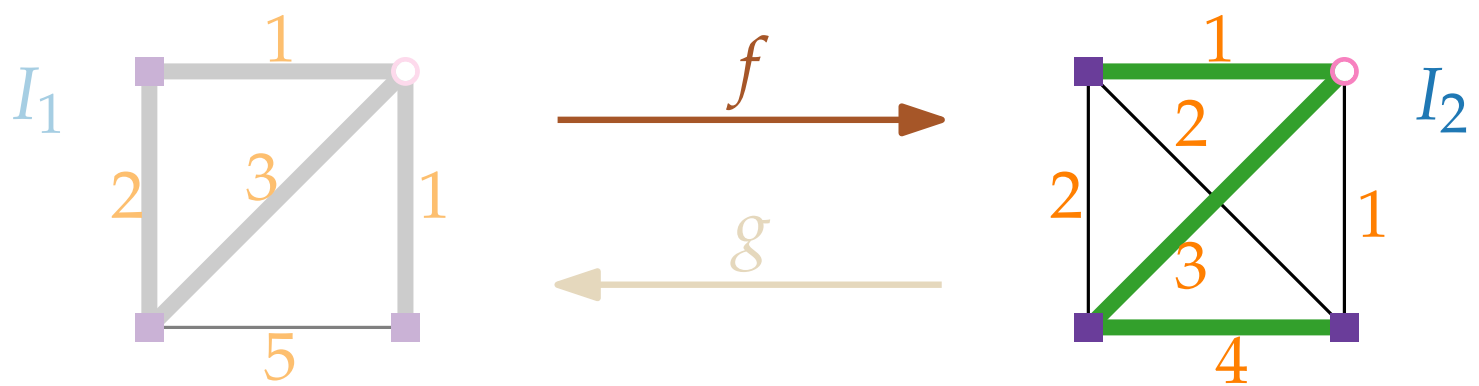
Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt



Metrisches SteinerTree

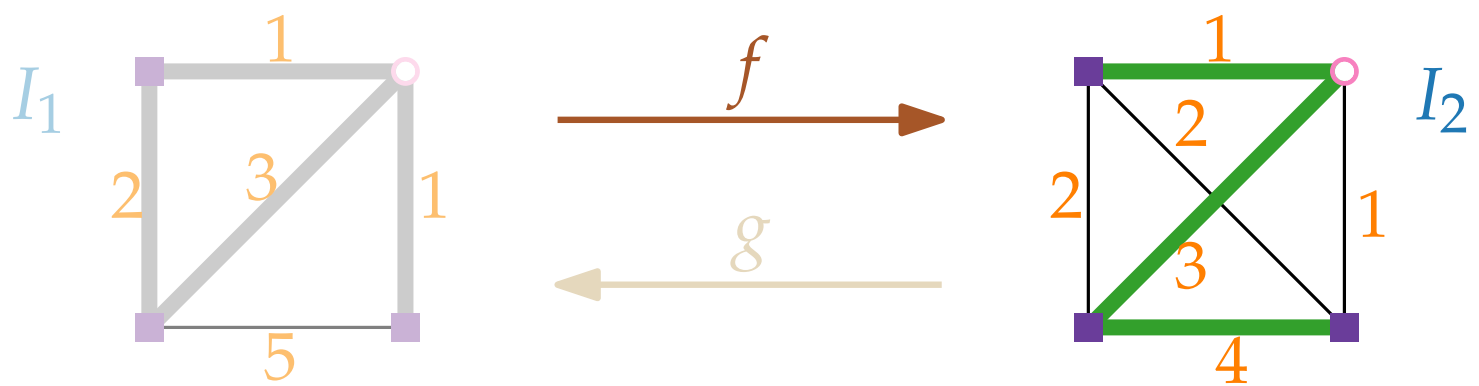
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt

$c_1(G_1') \leq c_2(B_2)$



Metrisches SteinerTree

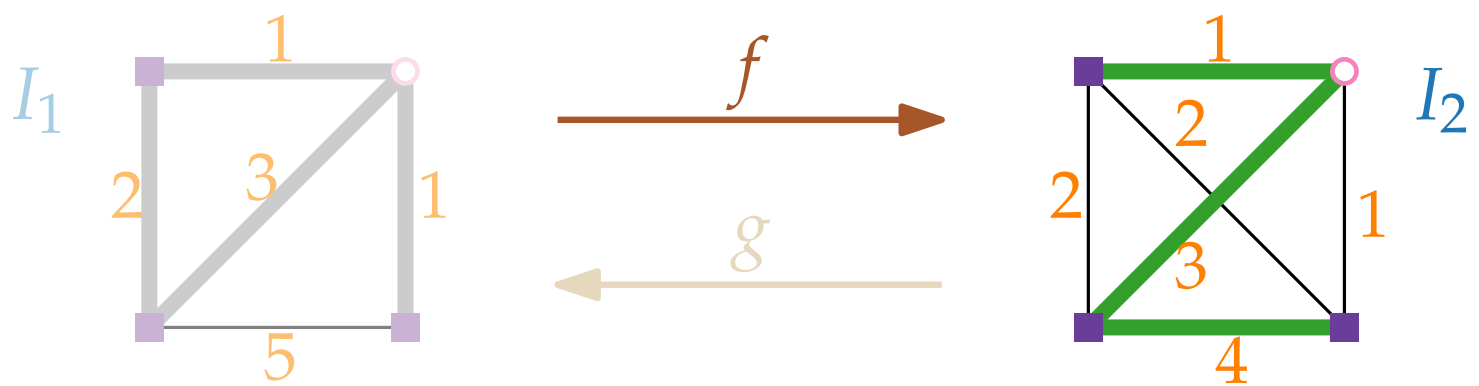
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt

$c_1(G_1') \leq c_2(B_2)$; G_1' verbindet alle Terminale



Metrisches SteinerTree

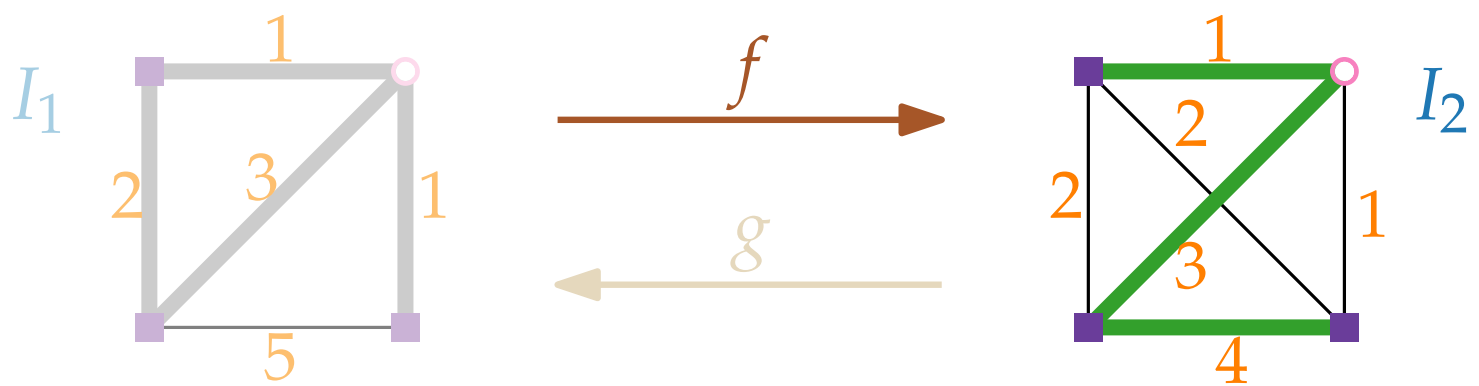
Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt

$c_1(G_1') \leq c_2(B_2)$; G_1' verbindet alle Terminale; evtl. kein Baum



Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

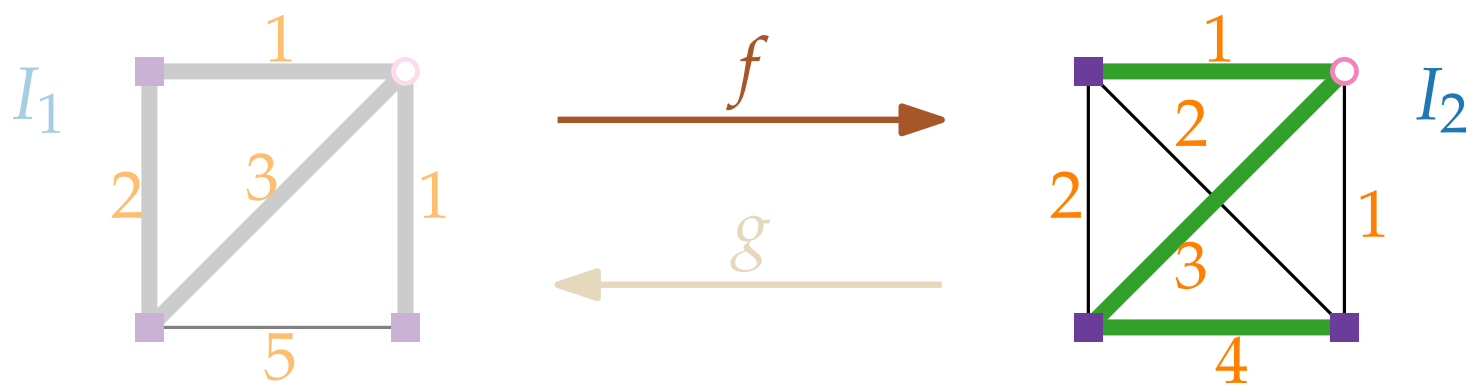
Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt

$c_1(G_1') \leq c_2(B_2)$; G_1' verbindet alle Terminale; evtl. kein Baum

Betrachte Spannbaum B_1 von G_1'



Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

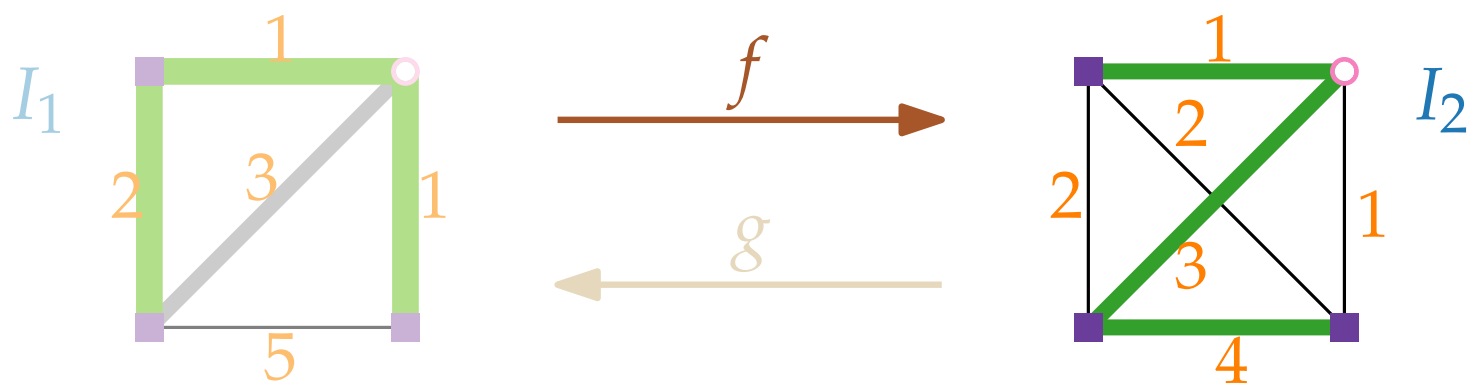
Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt

$c_1(G_1') \leq c_2(B_2)$; G_1' verbindet alle Terminale; evtl. kein Baum

Betrachte Spannbaum B_1 von G_1'



Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

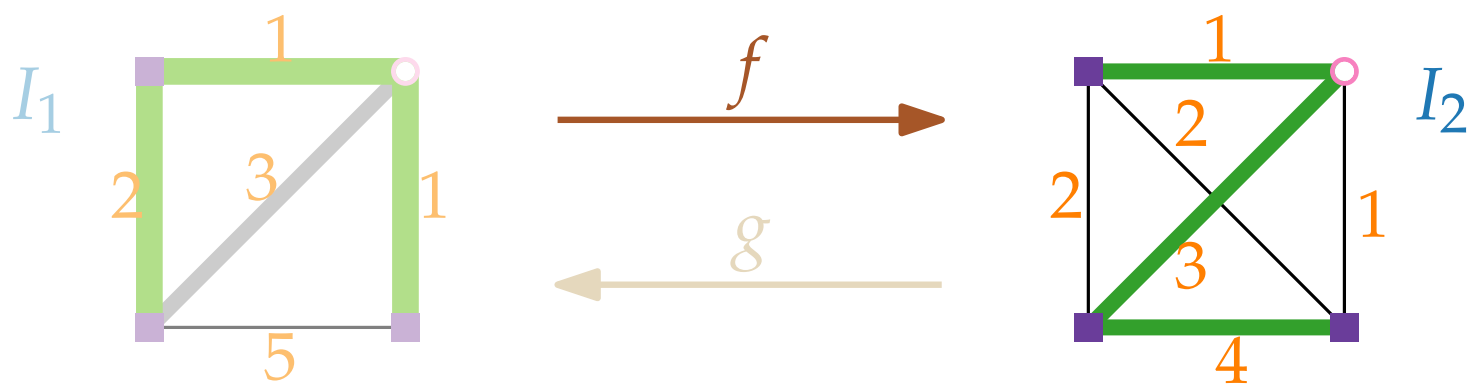
Beweis. (3) Abbildung g $s \xleftarrow{g} t$

Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt

$c_1(G_1') \leq c_2(B_2)$; G_1' verbindet alle Terminale; evtl. kein Baum

Betrachte Spannbaum B_1 von $G_1' \rightsquigarrow$ Steinerbaum B_1 für G_1



Metrisches SteinerTree

Satz. Es gibt eine approximationserhaltende Reduktion von SteinerTree auf metrisches SteinerTree.

Beweis. (3) Abbildung g $\quad s \xleftarrow{g} t$

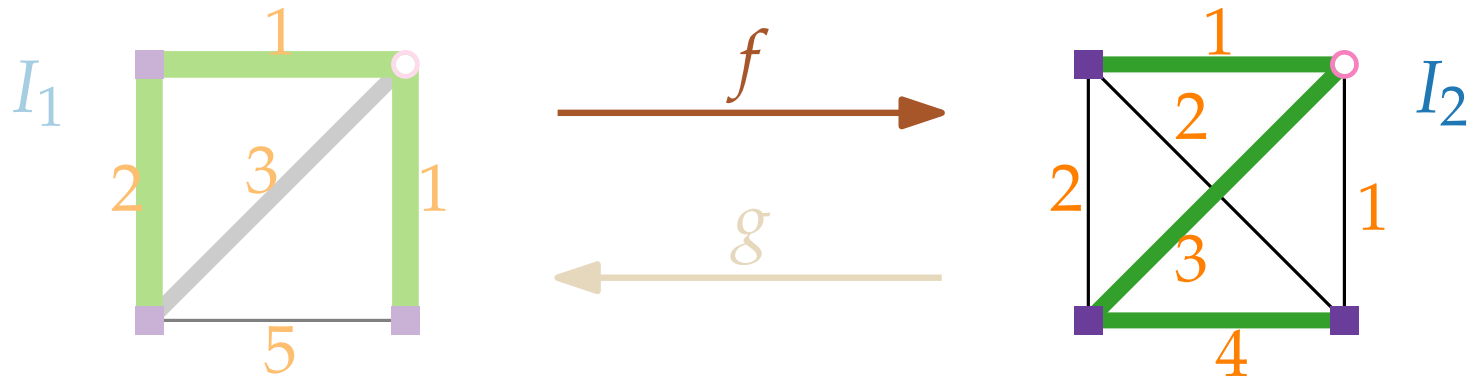
Sei B_2 Steinerbaum für G_2

$G_1' \subseteq G_1$ entstehe aus B_2 indem man jede Kante (u, v) aus B_2 durch einen kürzesten u–v-Pfad aus G_1 ersetzt

$c_1(G_1') \leq c_2(B_2)$; G_1' verbindet alle Terminale; evtl. kein Baum

Betrachte Spannbaum B_1 von $G_1' \rightsquigarrow$ Steinerbaum B_1 für G_1

$c_1(B_1) \leq c_1(G_1') \leq c_2(B_2)$



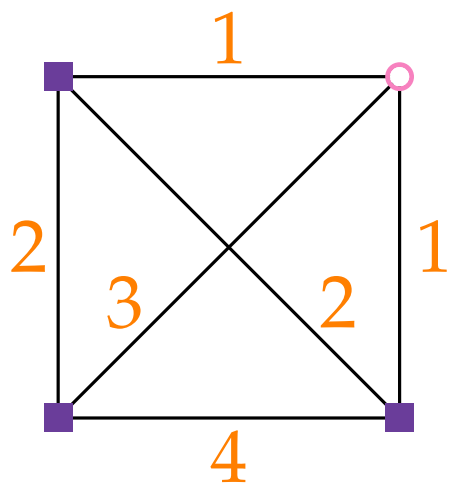
2-Approximation für SteinerTree

2-Approximation für SteinerTree

Satz. Gegeben sei eine Instanz für metrisches SteinerTree. Sei B ein minimaler Spannbaum (MSB) für den durch die Terminalmenge T induzierten Subgraphen $G[T]$.
Dann gilt $c(B) \leq 2 \cdot \mathrm{OPT}$.

2-Approximation für SteinerTree

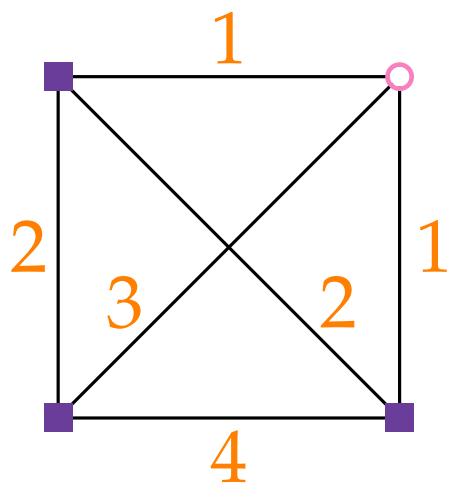
Satz. Gegeben sei eine Instanz für metrisches SteinerTree. Sei B ein minimaler Spannbaum (MSB) für den durch die Terminalmenge T induzierten Subgraphen $G[T]$.
Dann gilt $c(B) \leq 2 \cdot \mathrm{OPT}$.



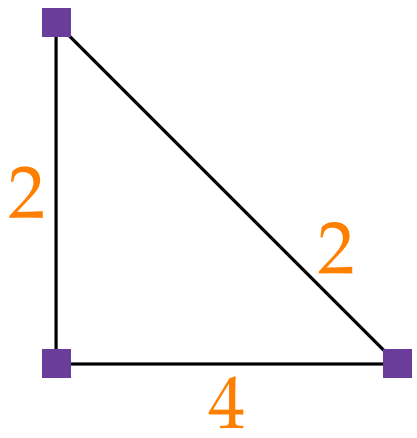
G

2-Approximation für SteinerTree

Satz. Gegeben sei eine Instanz für metrisches SteinerTree. Sei B ein minimaler Spannbaum (MSB) für den durch die Terminalmenge T induzierten Subgraphen $G[T]$.
Dann gilt $c(B) \leq 2 \cdot \mathrm{OPT}$.



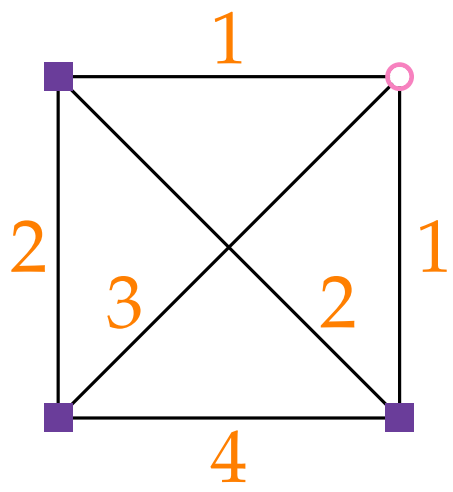
G



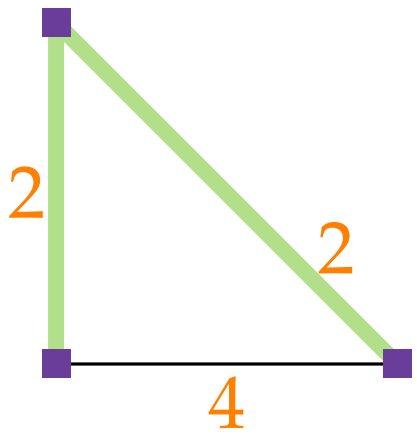
$G[T]$

2-Approximation für SteinerTree

Satz. Gegeben sei eine Instanz für metrisches SteinerTree. Sei B ein minimaler Spannbaum (MSB) für den durch die Terminalmenge T induzierten Subgraphen $G[T]$.
Dann gilt $c(B) \leq 2 \cdot \mathrm{OPT}$.



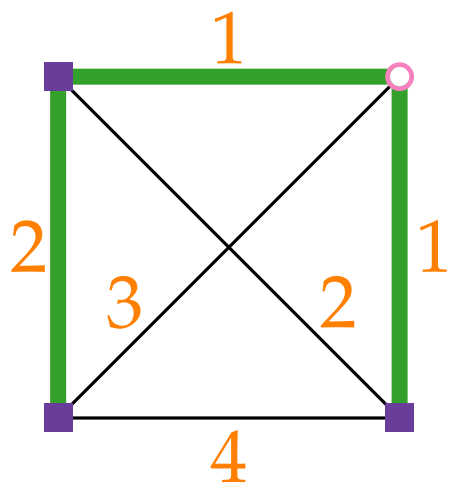
G



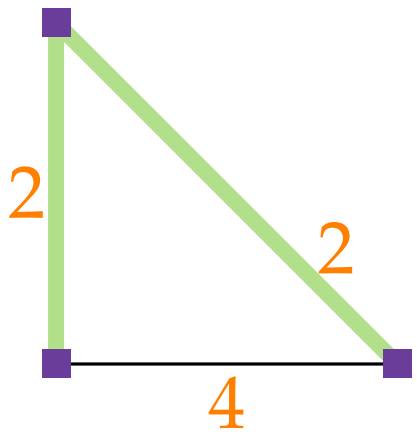
$G[T]$

2-Approximation für SteinerTree

Satz. Gegeben sei eine Instanz für metrisches SteinerTree. Sei B ein minimaler Spannbaum (MSB) für den durch die Terminalmenge T induzierten Subgraphen $G[T]$.
Dann gilt $c(B) \leq 2 \cdot \mathrm{OPT}$.



G



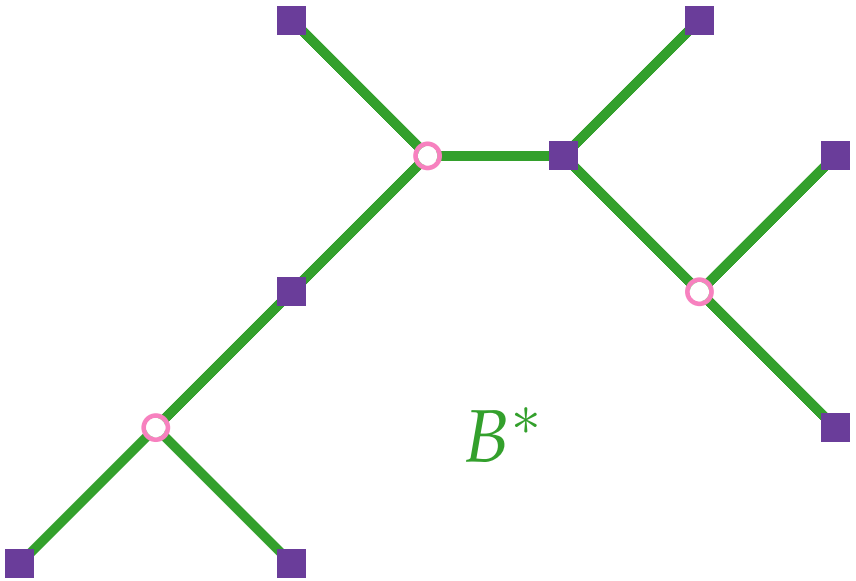
$G[T]$

Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*

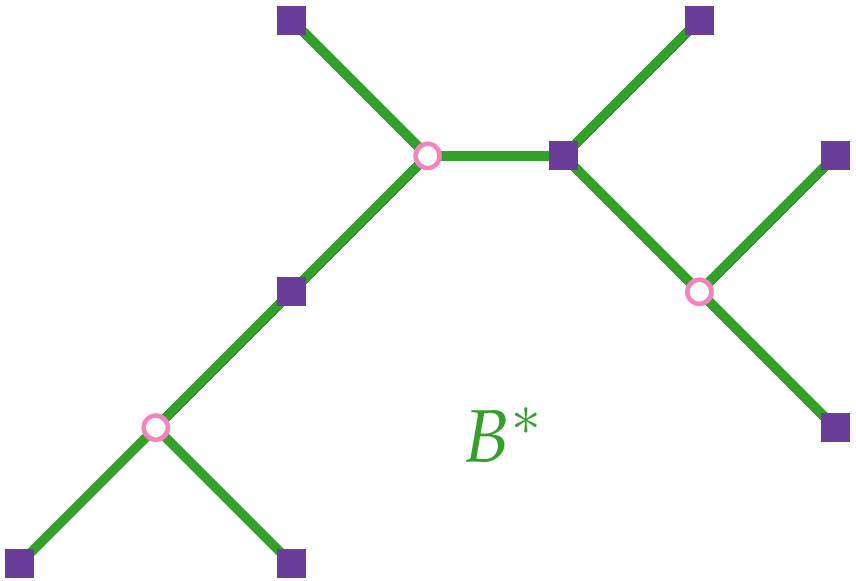
Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*



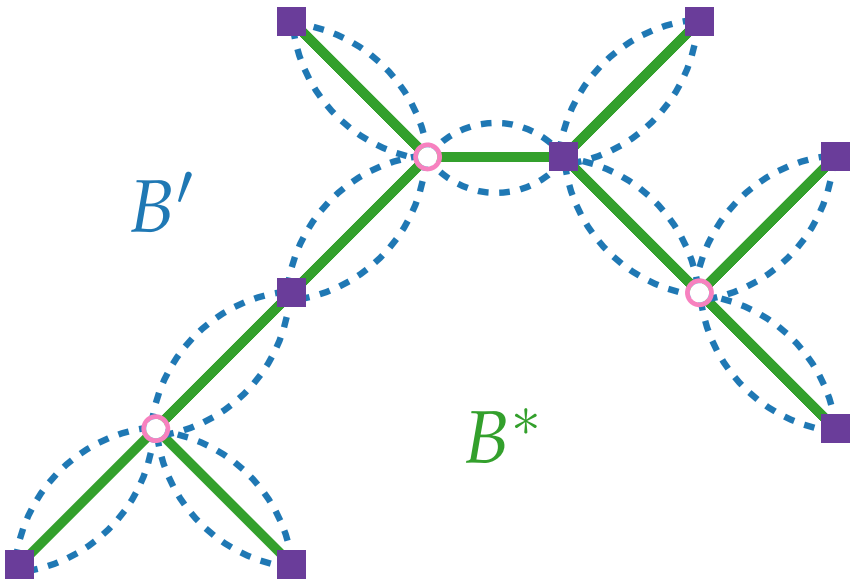
Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*
Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$



Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*
Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$

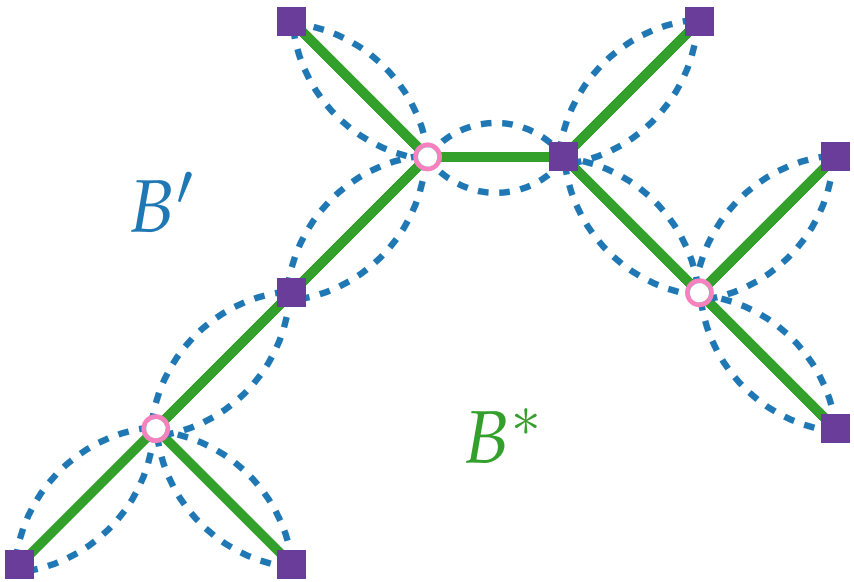


Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*

Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$

Finde eine Eulertour T' in B' $\rightsquigarrow$ $c(T') = c(B') = 2 \cdot \mathrm{OPT}$

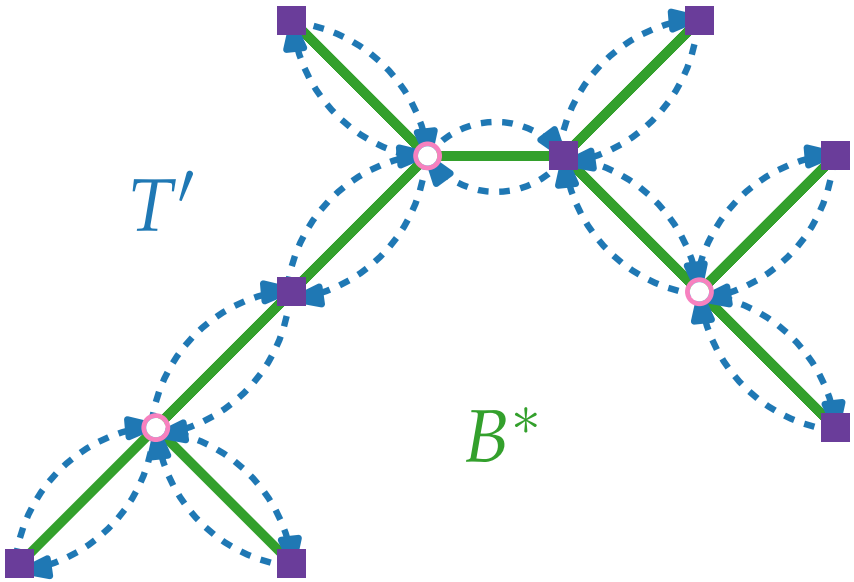


Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*

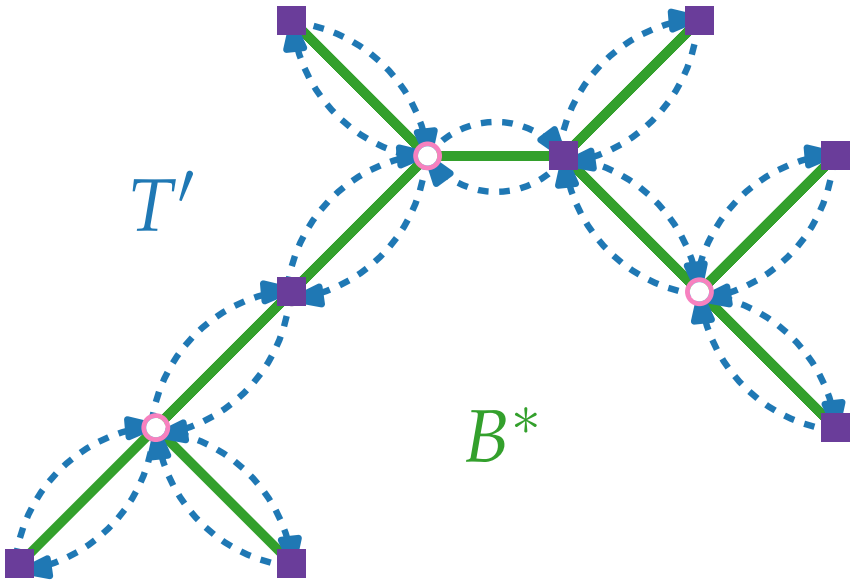
Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$

Finde eine Eulertour T' in B' $\rightsquigarrow$ $c(T') = c(B') = 2 \cdot \mathrm{OPT}$



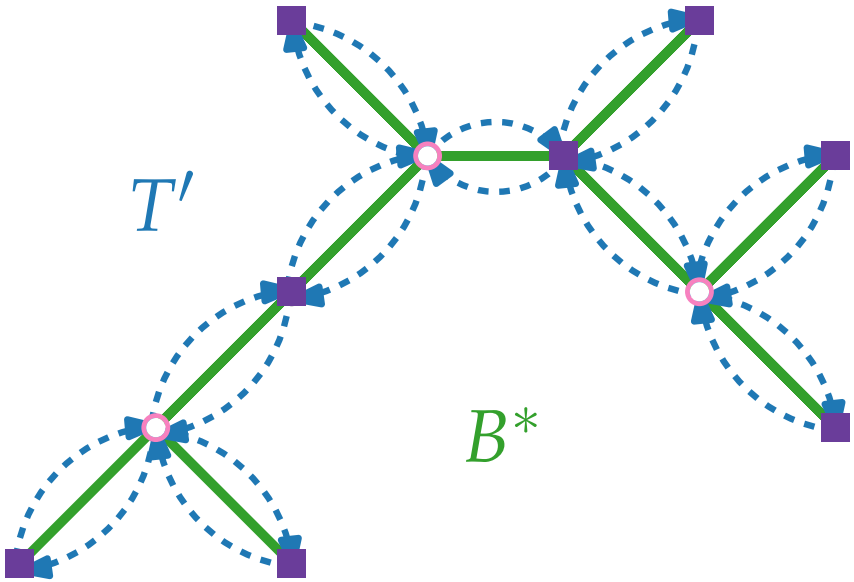
Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*
Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$
Finde eine Eulertour T' in B' $\rightsquigarrow$ $c(T') = c(B') = 2 \cdot \mathrm{OPT}$
Hamiltonpfad H für $G[T]$ durch „Abkürzen“ von Steinerknoten und bereits besuchten Terminalen



Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*
Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$
Finde eine Eulertour T' in B' $\rightsquigarrow$ $c(T') = c(B') = 2 \cdot \mathrm{OPT}$
Hamiltonpfad H für $G[T]$ durch „Abkürzen“ von Steinerknoten und bereits besuchten Terminalen
$\rightsquigarrow$ $c(H) \leq c(T') = 2 \cdot \mathrm{OPT}$, da G metrisch



Beweis der Approximationsgüte

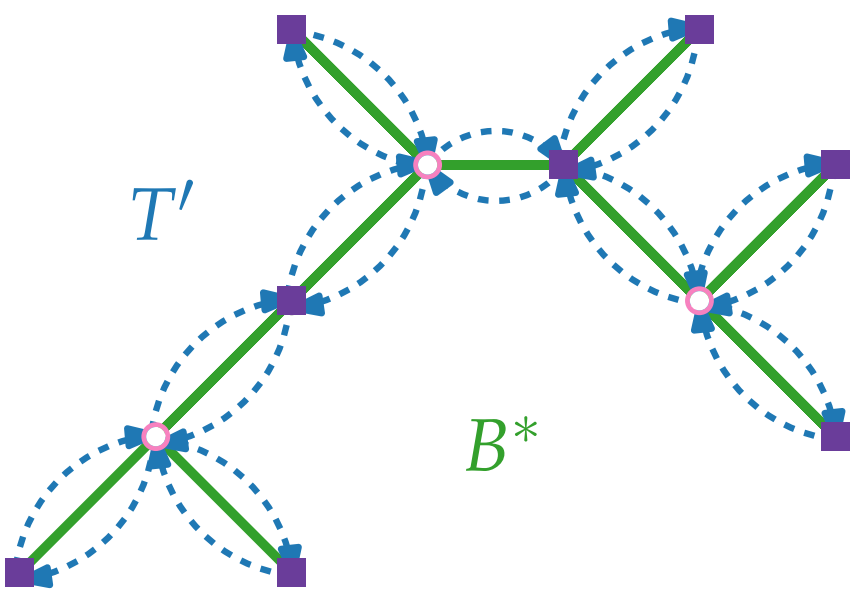
Betrachte optimalen Steinerbaum B^*

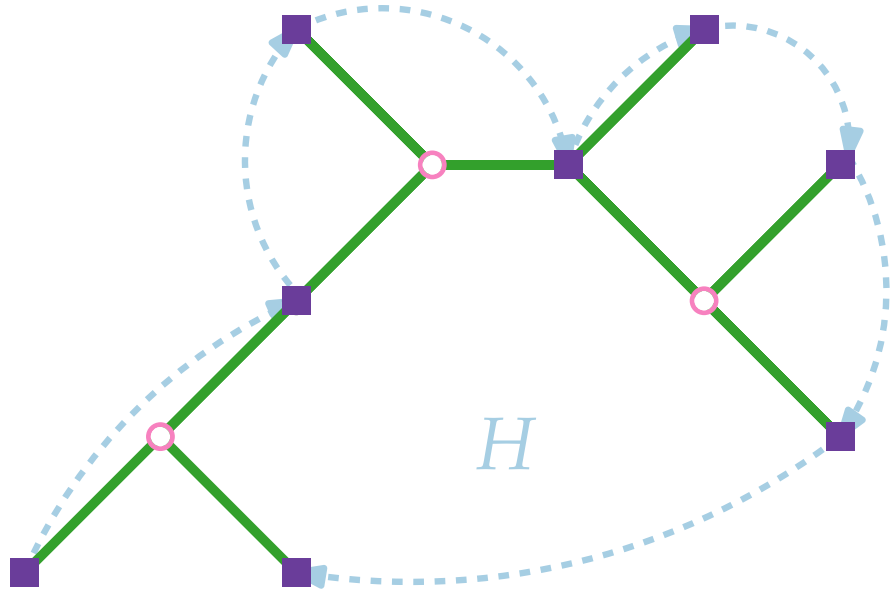
Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$

Finde eine Eulertour T' in B' $\rightsquigarrow$ $c(T') = c(B') = 2 \cdot \mathrm{OPT}$

Hamiltonpfad H für $G[T]$ durch „Abkürzen“ von Steinerknoten und bereits besuchten Terminalen

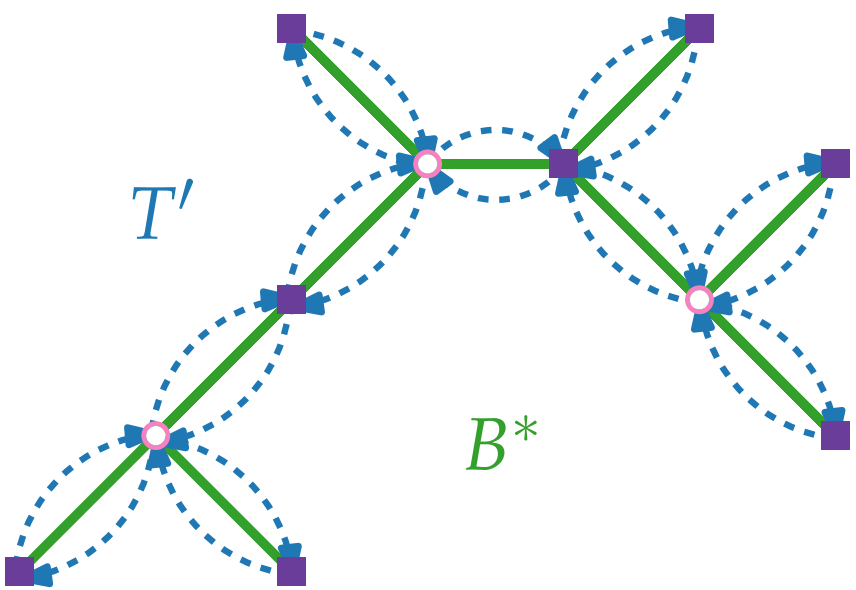
$\rightsquigarrow$ $c(H) \leq c(T') = 2 \cdot \mathrm{OPT}$, da G metrisch

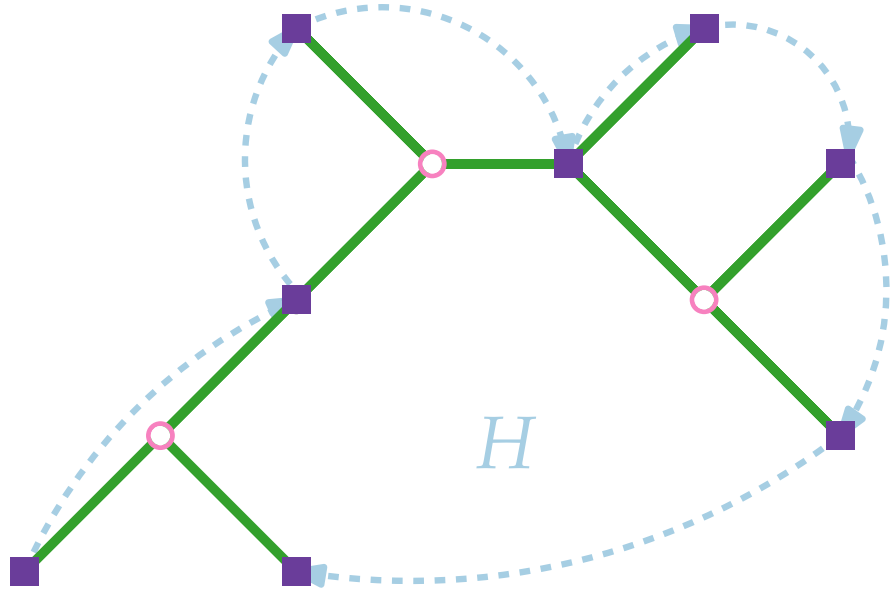




Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*
Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$
Finde eine Eulertour T' in B' $\rightsquigarrow$ $c(T') = c(B') = 2 \cdot \mathrm{OPT}$
Hamiltonpfad H für $G[T]$ durch „Abkürzen“ von Steinerknoten und bereits besuchten Terminalen
$\rightsquigarrow$ $c(H) \leq c(T') = 2 \cdot \mathrm{OPT}$, da G metrisch
MST B für $G[T]$ erfüllt $c(B) \leq c(H) \leq 2 \cdot \mathrm{OPT}$,
da





Beweis der Approximationsgüte

Betrachte optimalen Steinerbaum B^*

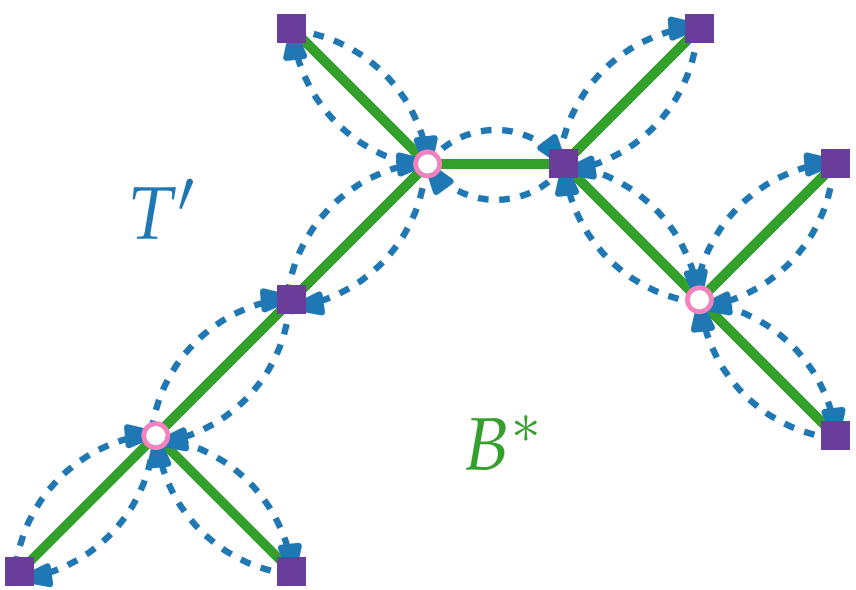
Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$

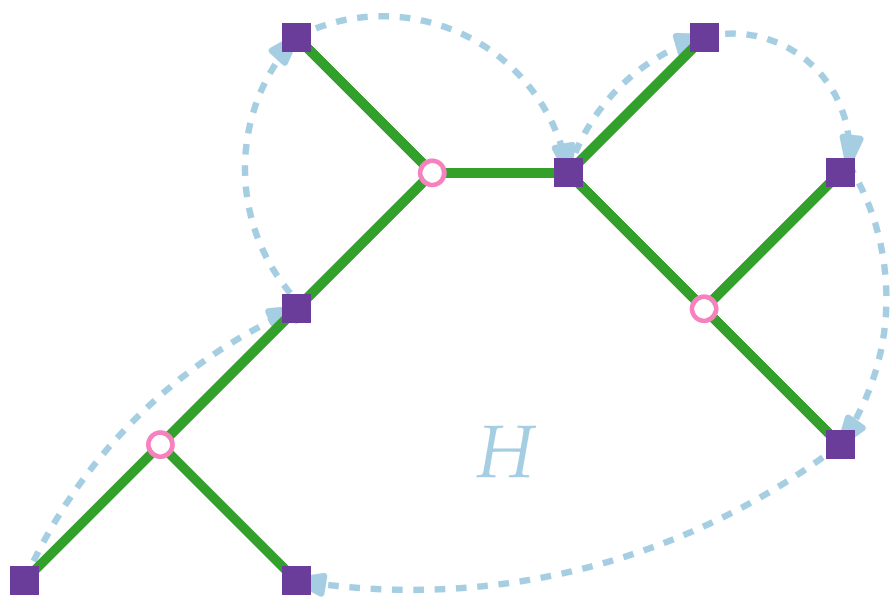
Finde eine Eulertour T' in B' $\rightsquigarrow$ $c(T') = c(B') = 2 \cdot \mathrm{OPT}$

Hamiltonpfad H für $G[T]$ durch „Abkürzen“ von Steinerknoten und bereits besuchten Terminalen

$\rightsquigarrow$ $c(H) \leq c(T') = 2 \cdot \mathrm{OPT}$, da G metrisch

MST B für $G[T]$ erfüllt $c(B) \leq c(H) \leq 2 \cdot \mathrm{OPT}$, da H Spannbaum für $G[T]$



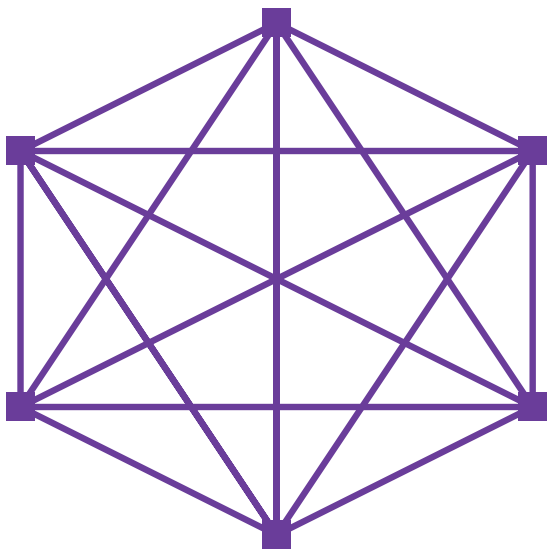


Analyse scharf?

Analyse scharf?

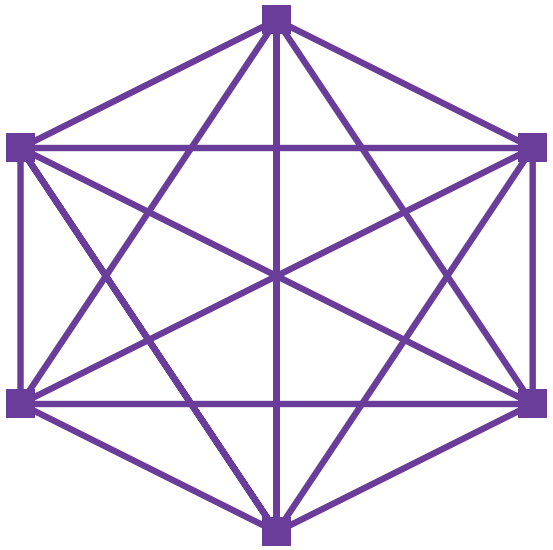
- Terminal

K_n



Analyse scharf?

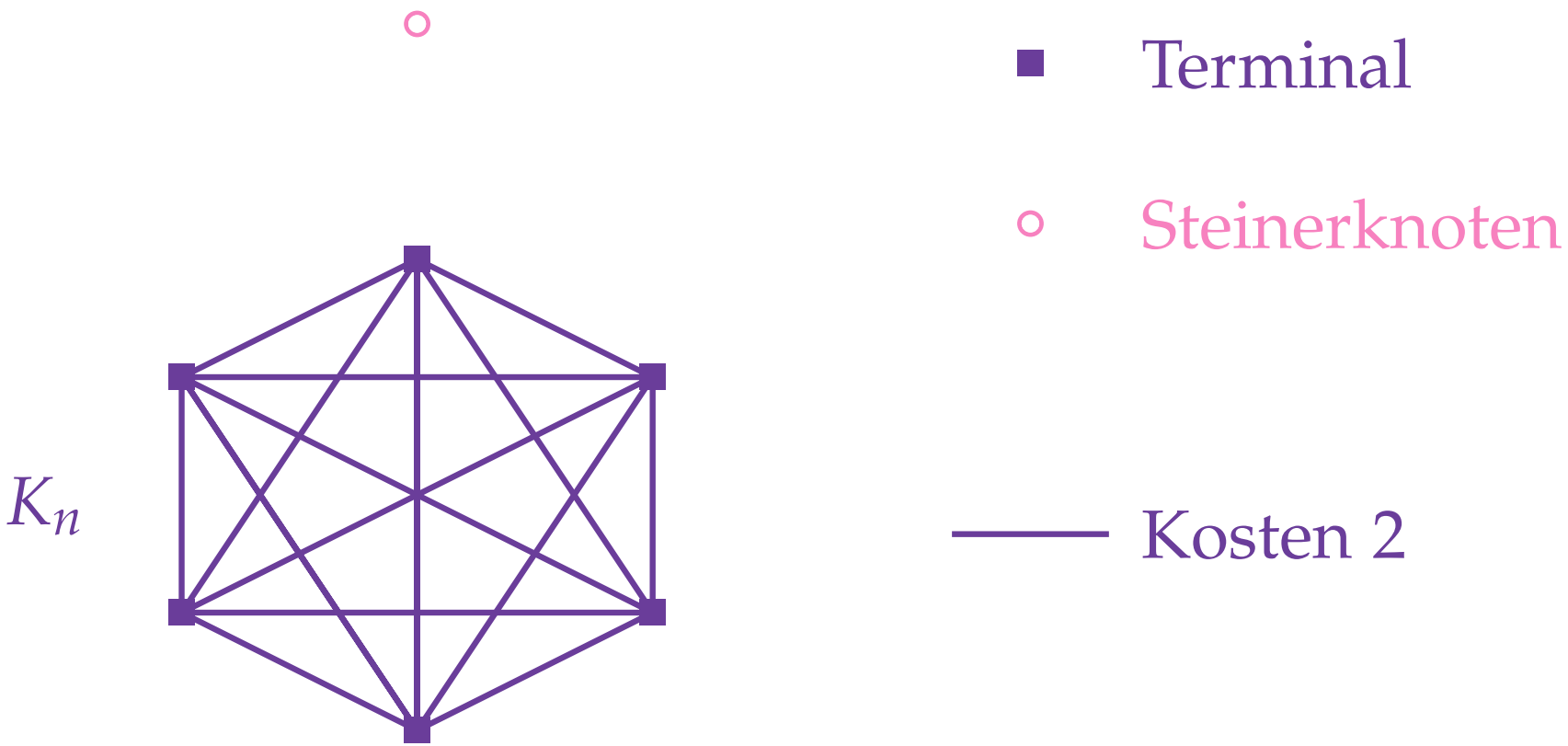
K_n



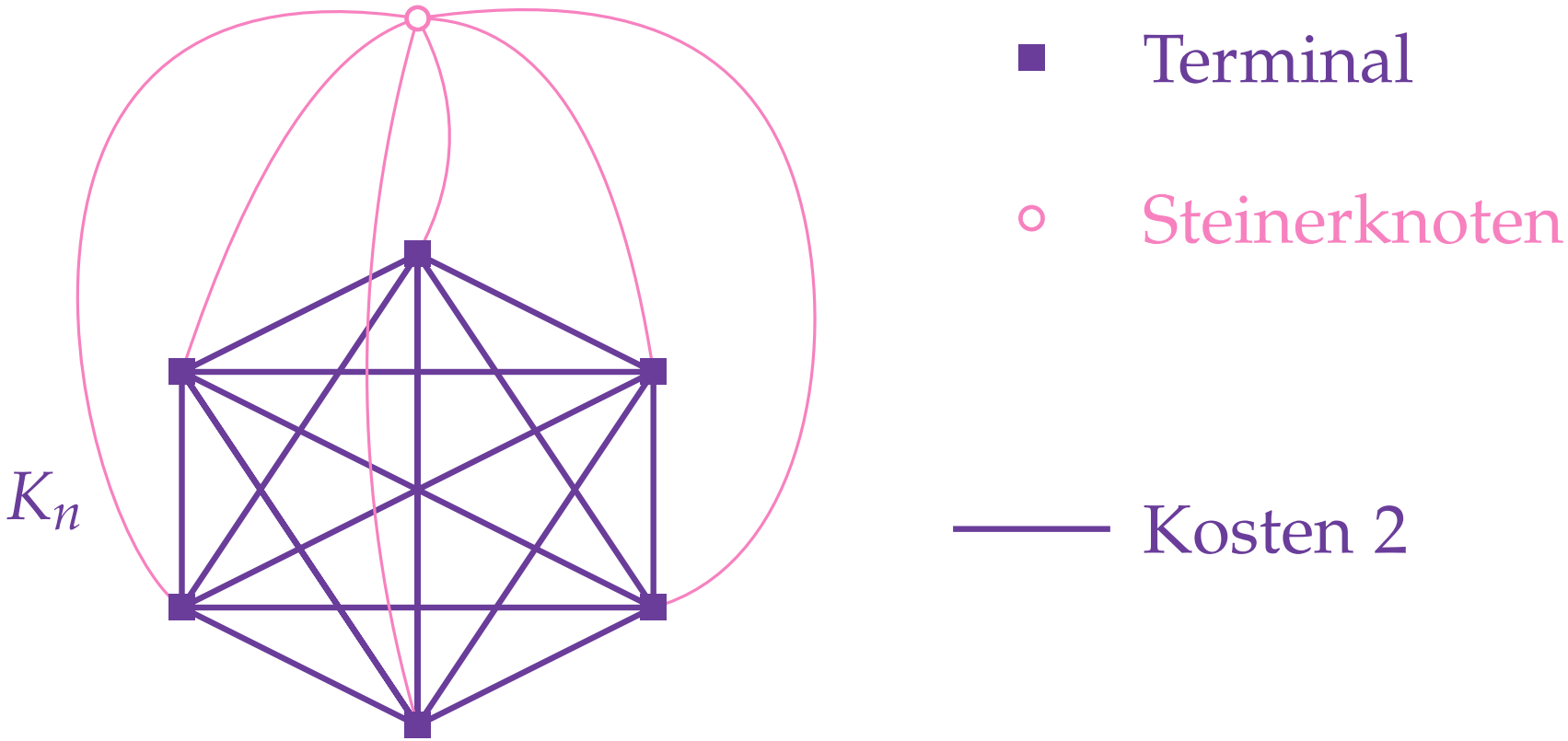
Terminal

Kosten 2

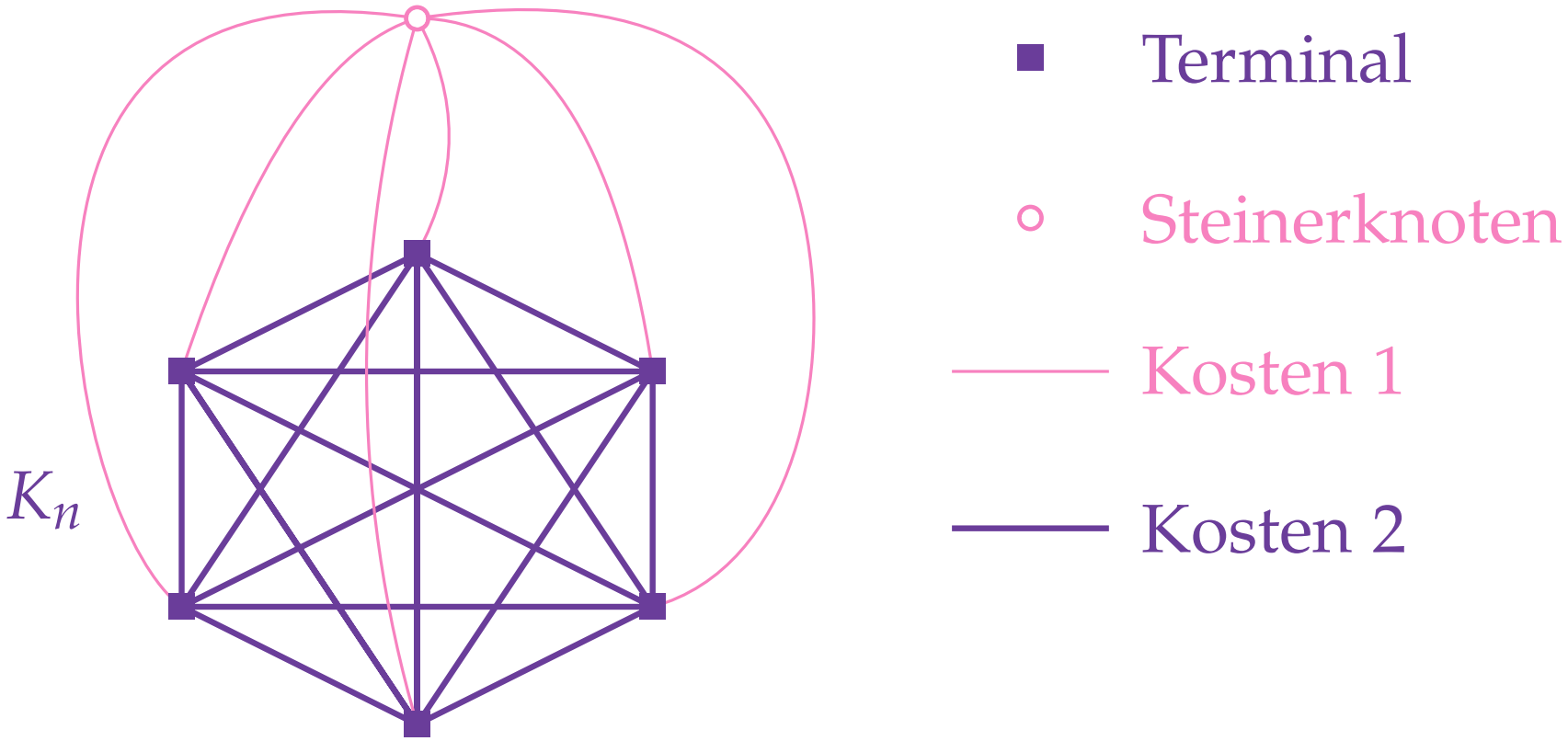
Analyse scharf?



Analyse scharf?

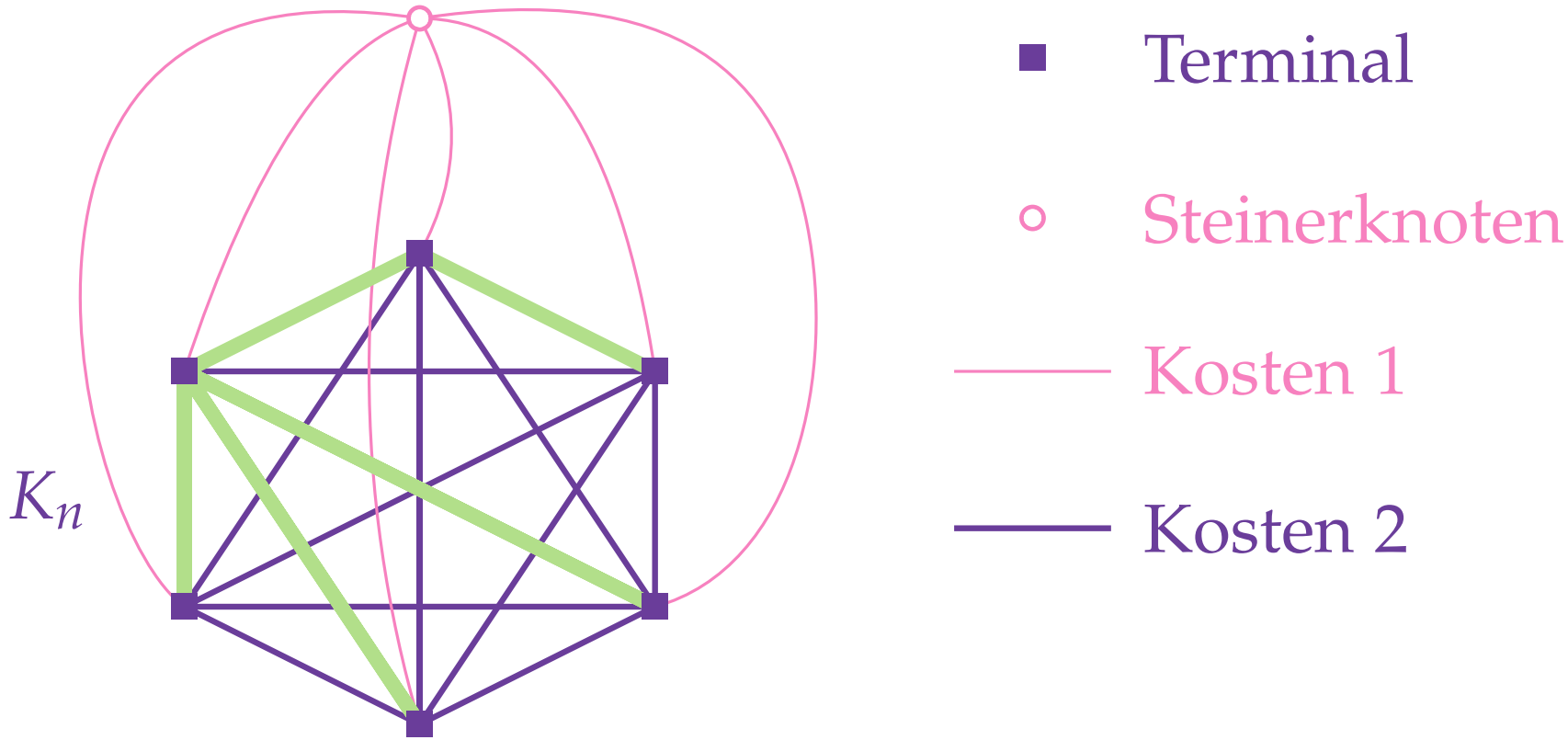


Analyse scharf?



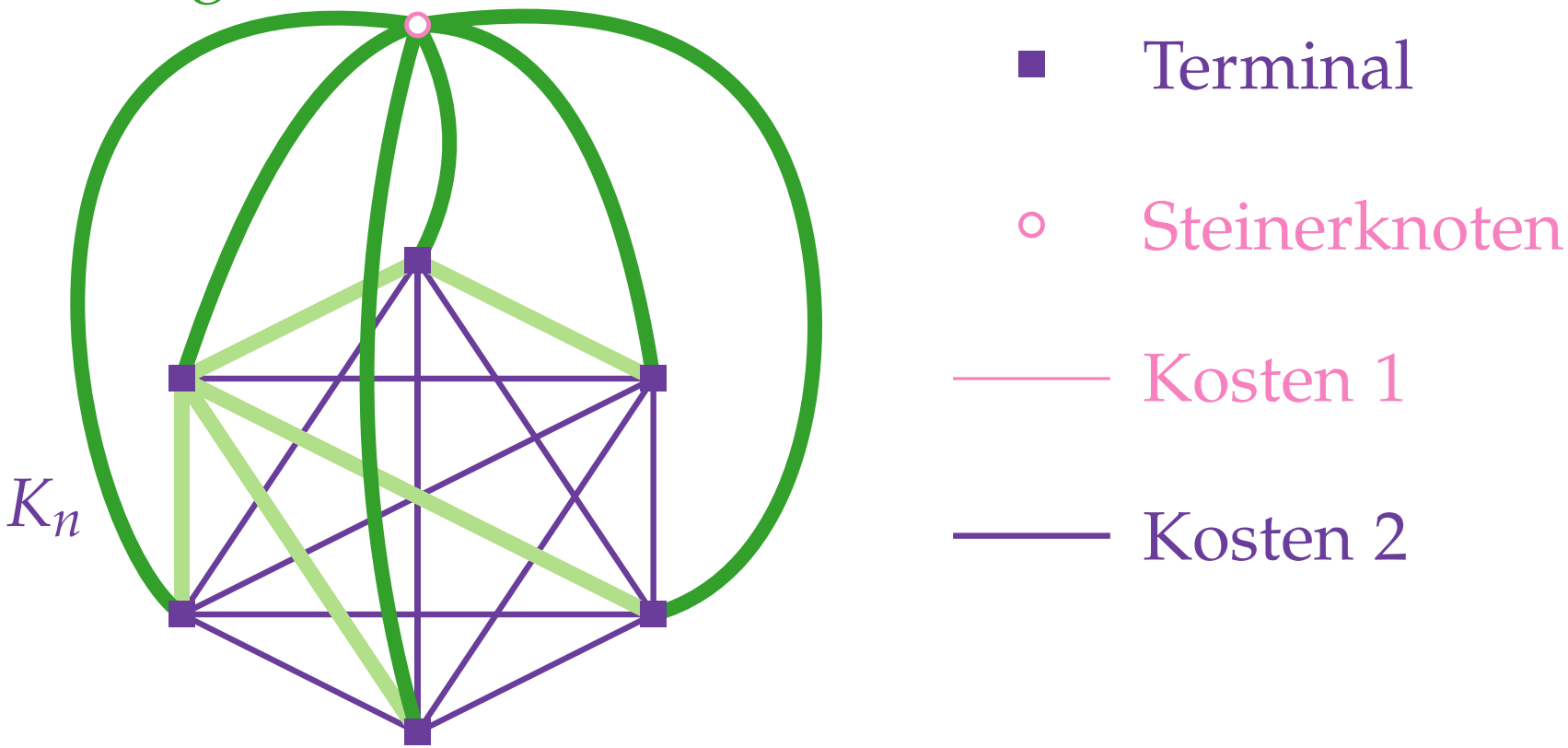
Analyse scharf?

MST von $G[T]$ mit Kosten $2(n-1)$



Analyse scharf?

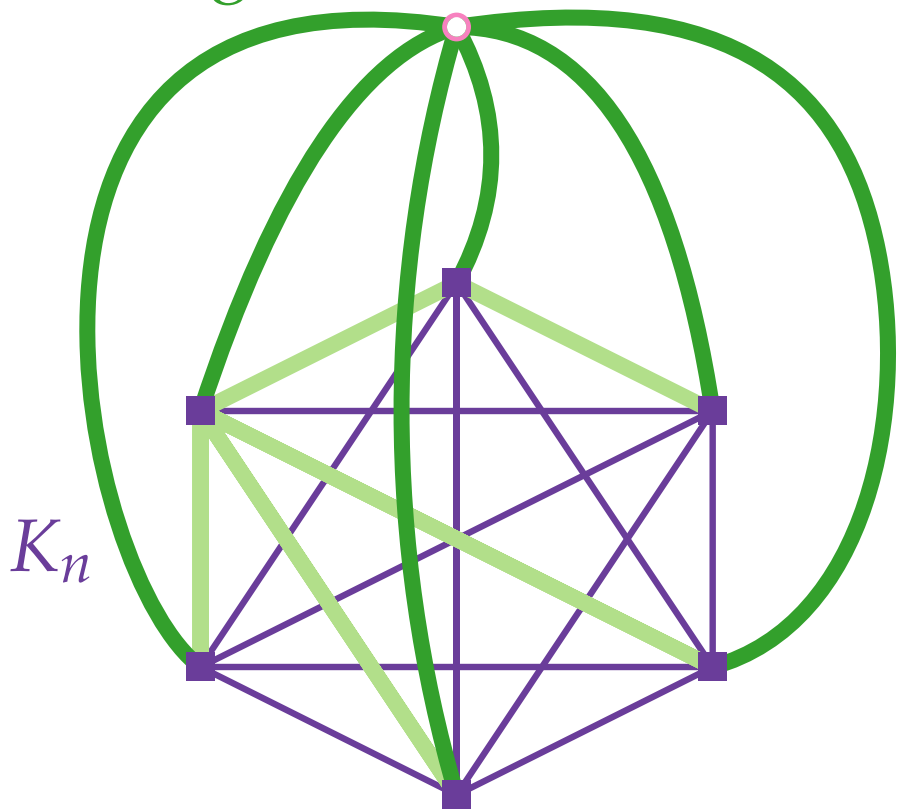
MST von $G[T]$ mit Kosten $2(n-1)$
Optimale Lösung mit Kosten n



Analyse scharf?

MST von $G[T]$ mit Kosten $2(n-1)$
Optimale Lösung mit Kosten n

$$\frac{2(n-1)}{n} \to 2$$



Terminal

Steinerknoten

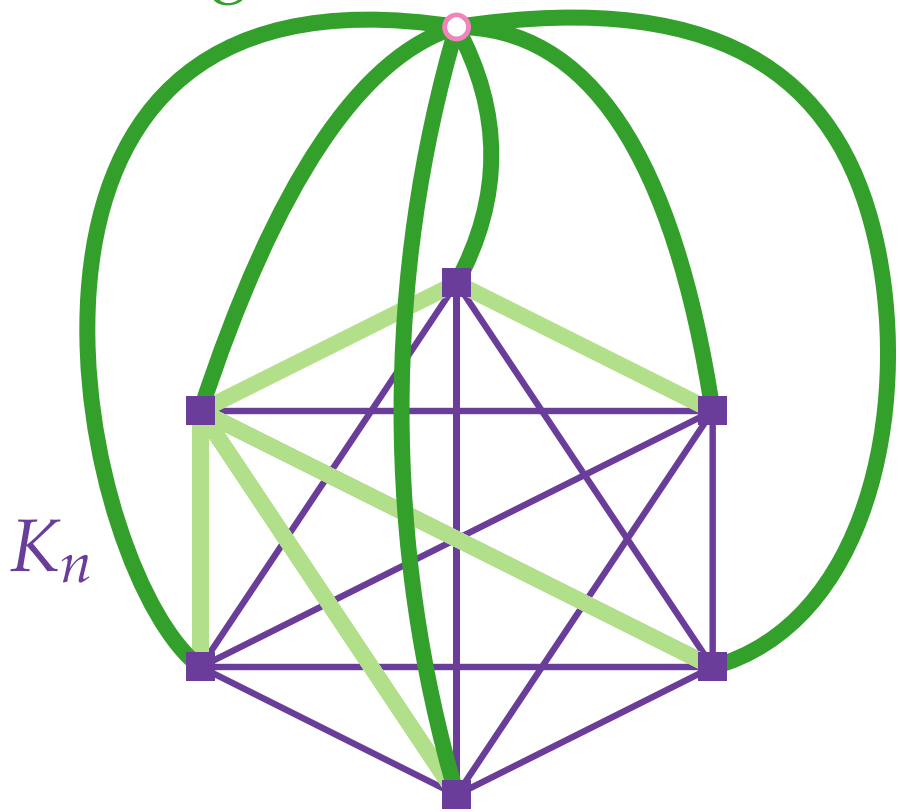
Kosten 1

Kosten 2

Analyse scharf?

MST von $G[T]$ mit Kosten $2(n-1)$
Optimale Lösung mit Kosten n

$$\frac{2(n-1)}{n} \to 2$$



Terminal

Steinerknoten

Kosten 1

Kosten 2

besser?

Analyse scharf?

MST von $G[T]$ mit Kosten $2(n-1)$
Optimale Lösung mit Kosten n

$$\frac{2(n-1)}{n} \to 2$$

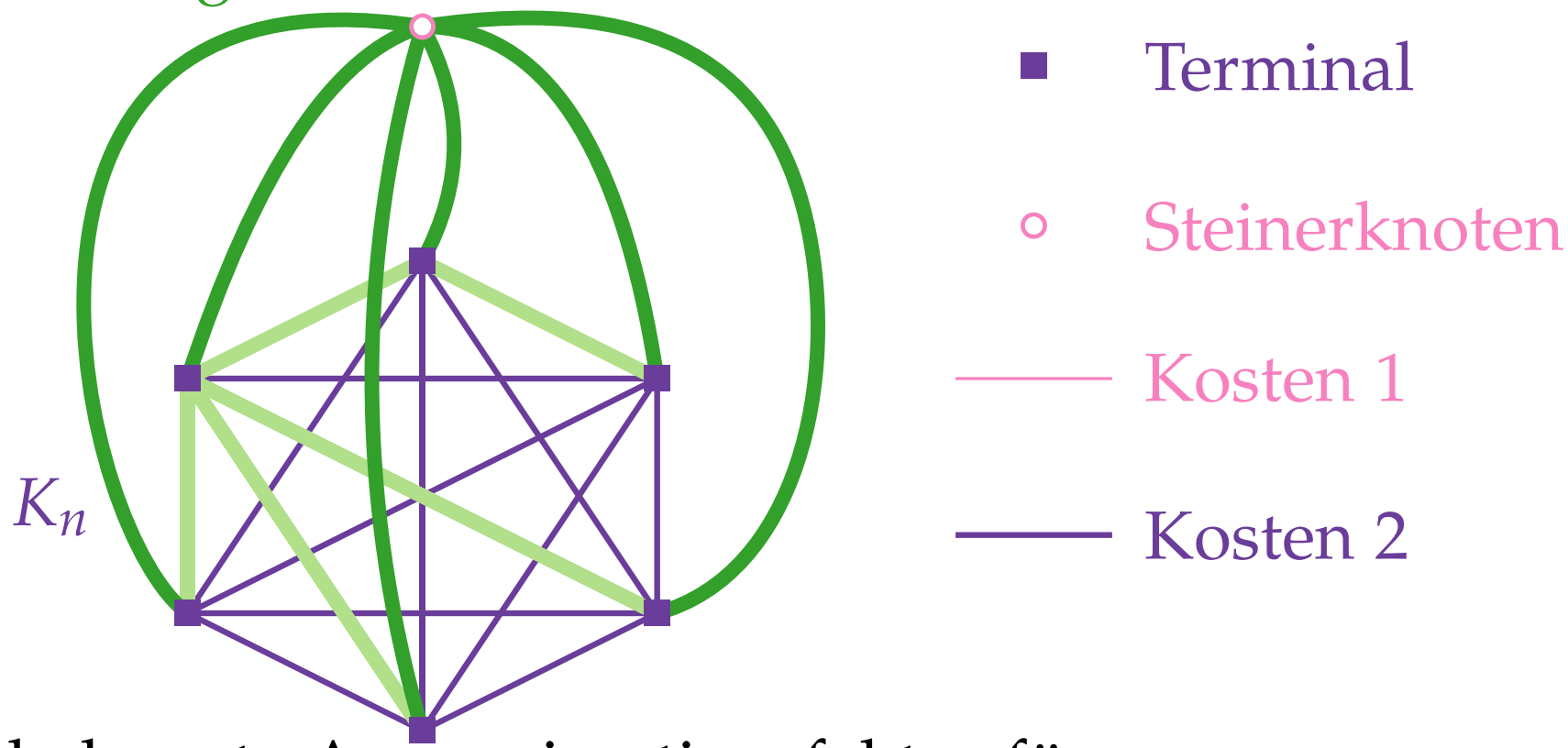
besser?
Der beste bekannte Approximationsfaktor für
STEINERTREE ist $\ln(4) + \varepsilon \approx 1{,}39$

[Byrka, Grandoni, Rothvoß & Sanita '10]

Analyse scharf?

MST von $G[T]$ mit Kosten $2(n-1)$
Optimale Lösung mit Kosten n

$$\frac{2(n-1)}{n} \to 2$$



besser?
Der beste bekannte Approximationsfaktor für
StEINERTREE ist $\ln(4) + \varepsilon \approx 1{,}39$ [Byrka, Grandoni, Rothvoß & Sanita '10]

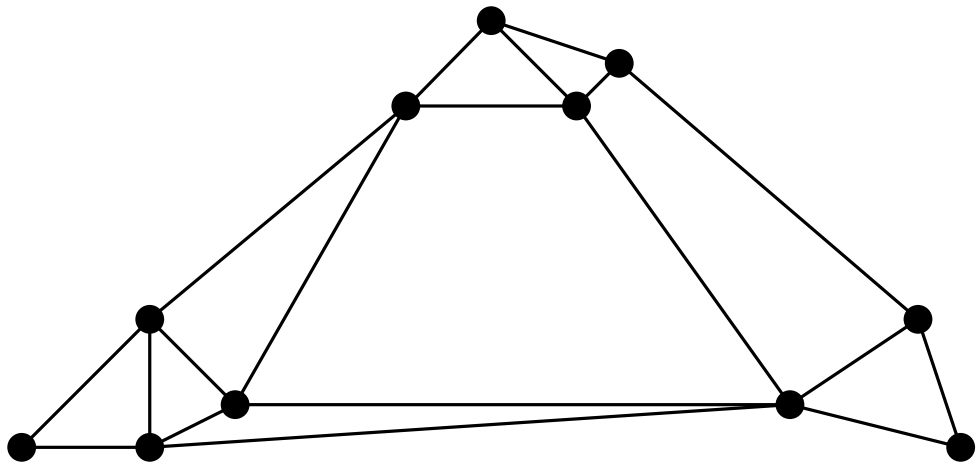
StEINERTREE lässt sich nicht mit Faktor $\frac{96}{95} \approx 1{,}0105$ approximieren (außer P=NP) [Chlebik & Chlebikova '08]

MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$

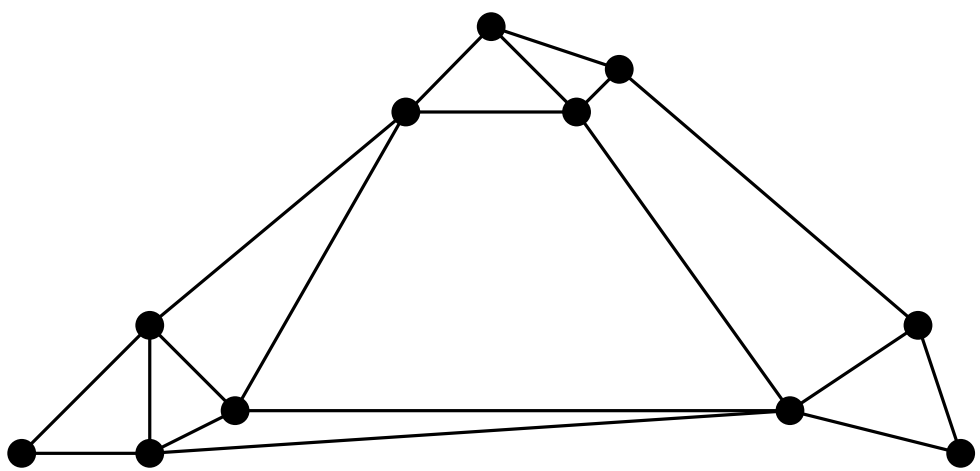
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$



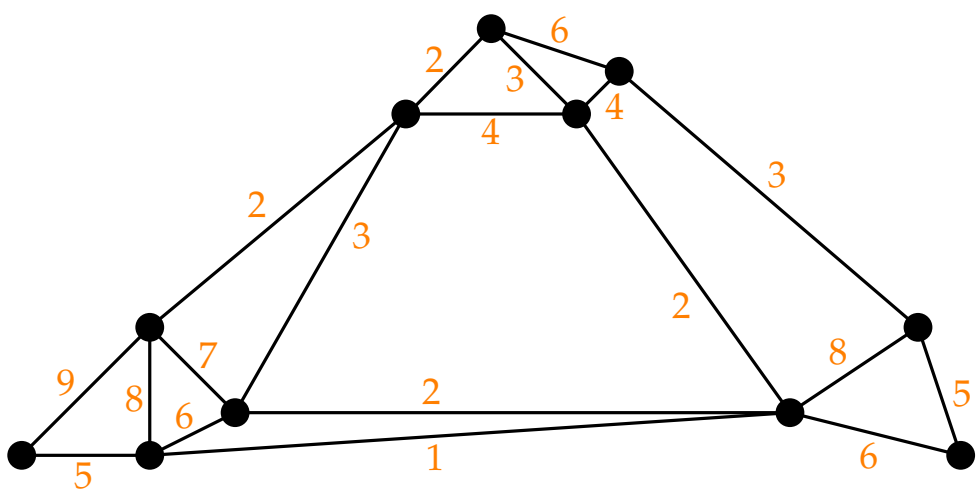
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$



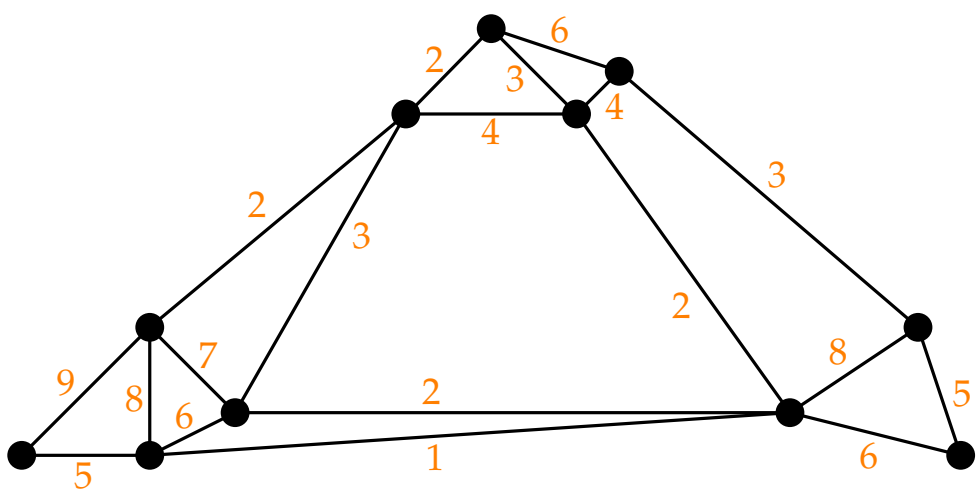
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$



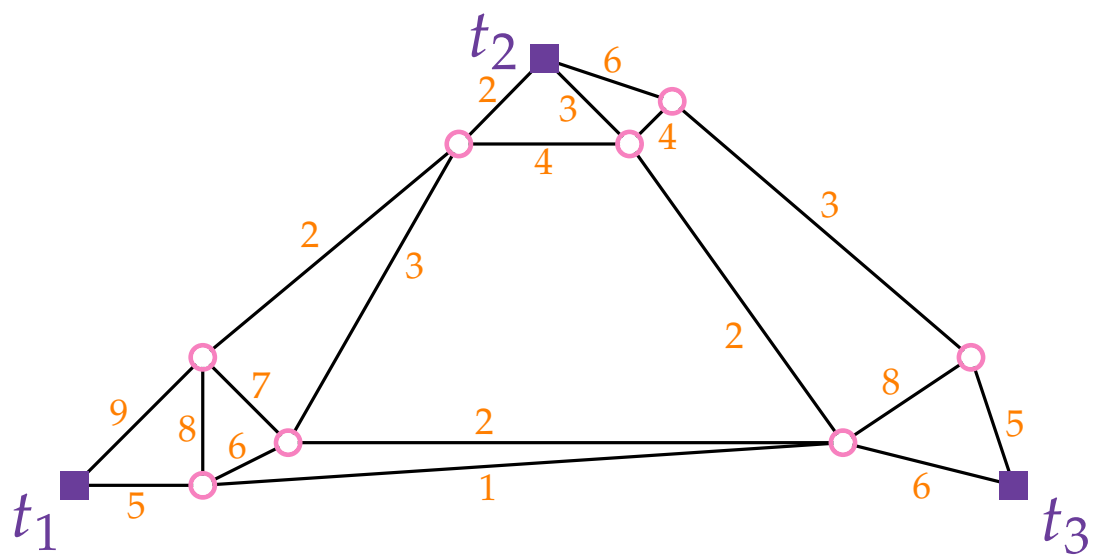
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$ und eine Menge $T = \{t_1, \ldots, t_k\} \subseteq V$ von Terminalen.



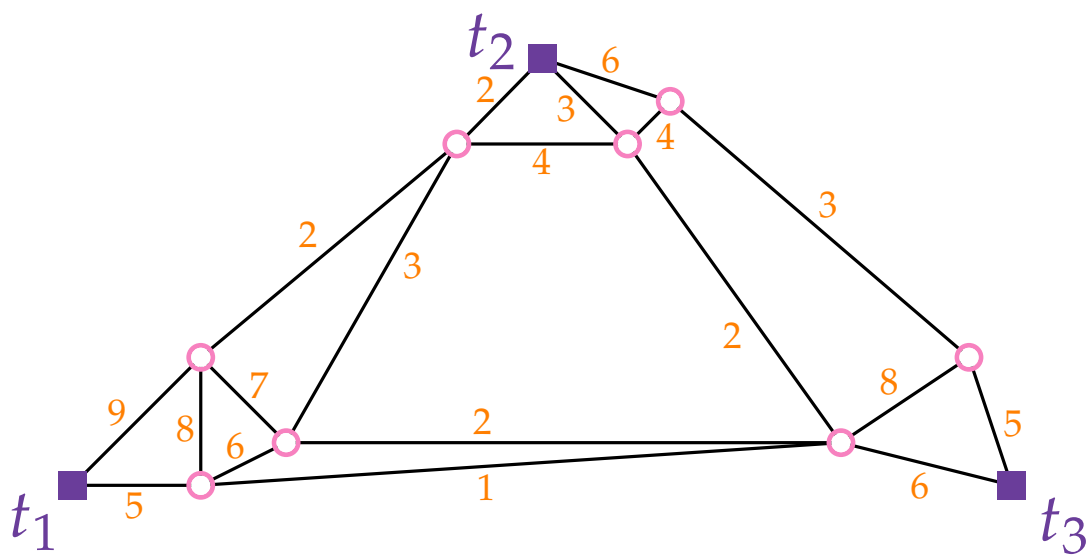
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$ und eine Menge $T = \{t_1, \ldots, t_k\} \subseteq V$ von Terminalen.



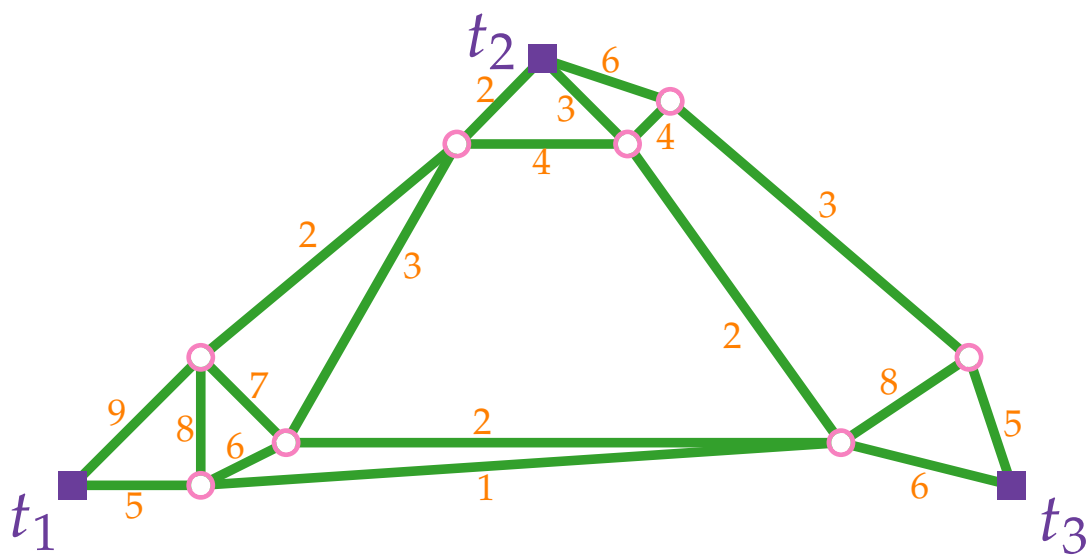
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$ und eine Menge $T = \{t_1, \ldots, t_k\} \subseteq V$ von Terminalen. Ein **Mehrwegeschnitt** für T ist eine Menge E' von Kanten, so dass keine zwei Terminale im Graphen $(V, E - E')$ zusammenhängend sind.



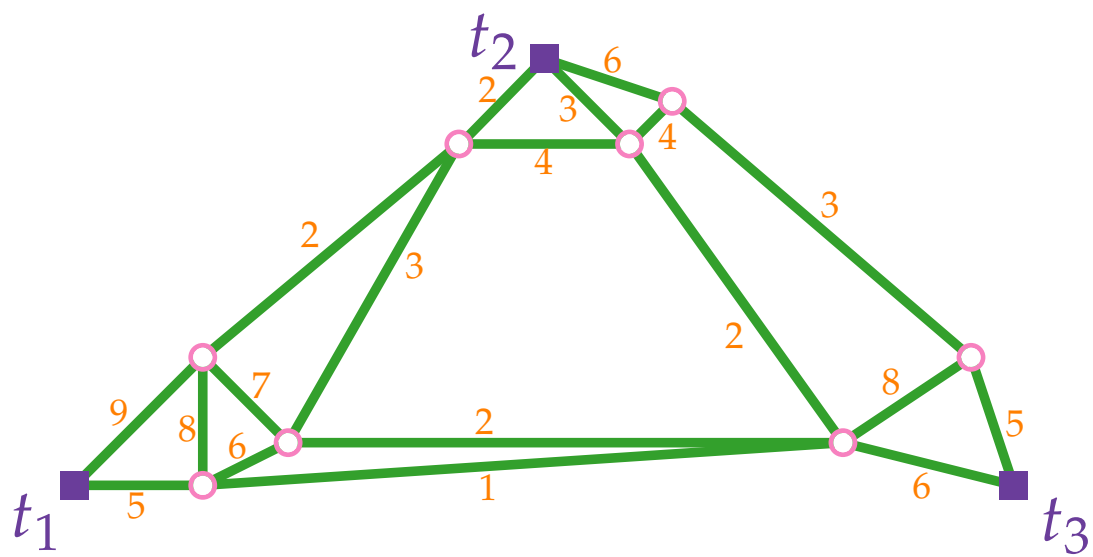
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$ und eine Menge $T = \{t_1, \ldots, t_k\} \subseteq V$ von Terminalen. Ein **Mehrwegeschnitt** für T ist eine Menge E' von Kanten, so dass keine zwei Terminale im Graphen $(V, E - E')$ zusammenhängend sind.



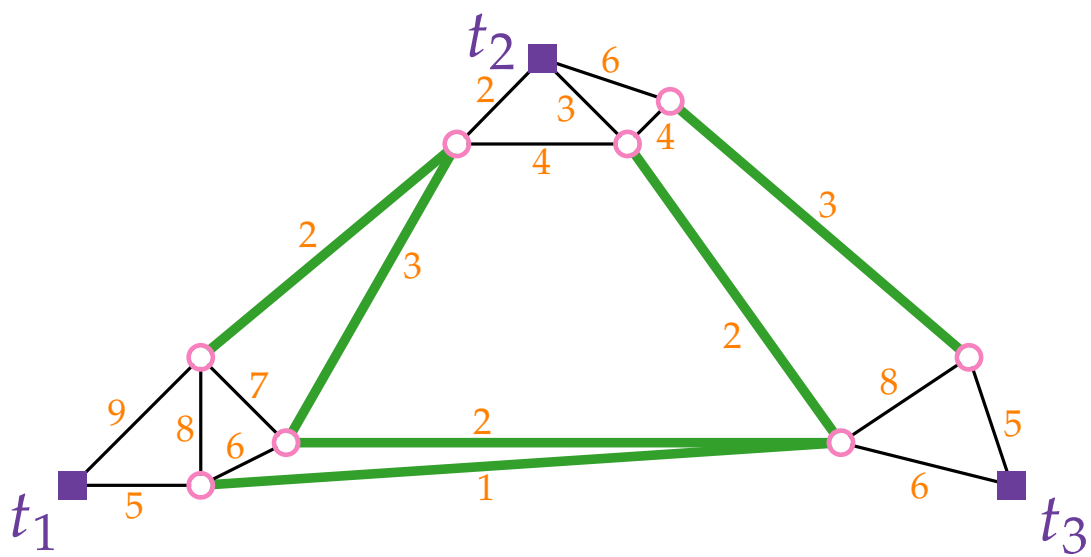
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$ und eine Menge $T = \{t_1, \ldots, t_k\} \subseteq V$ von Terminalen. Ein **Mehrwegeschnitt** für T ist eine Menge E' von Kanten, so dass keine zwei Terminale im Graphen $(V, E - E')$ zusammenhängend sind.
Das Problem MehrwegeSchnitt fragt nach einem kostengünstigsten Mehrwegeschnitt für T.



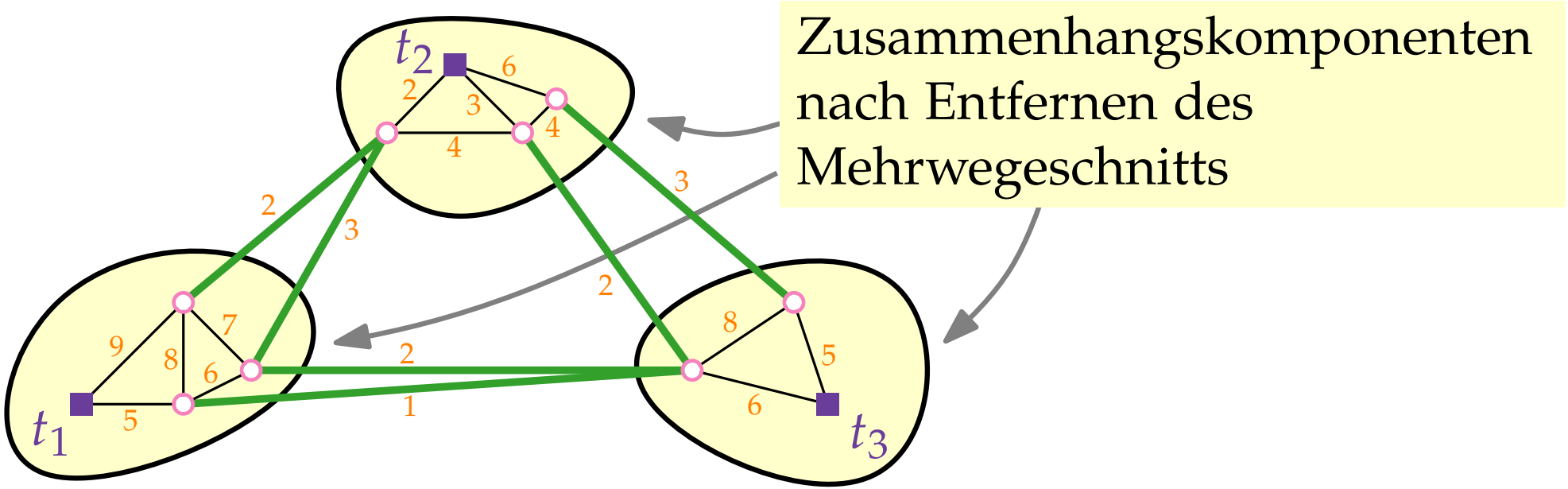
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$ und eine Menge $T = \{t_1, \ldots, t_k\} \subseteq V$ von Terminalen. Ein **Mehrwegeschnitt** für T ist eine Menge E' von Kanten, so dass keine zwei Terminale im Graphen $(V, E - E')$ zusammenhängend sind.
Das Problem MehrwegeSchnitt fragt nach einem kostengünstigsten Mehrwegeschnitt für T.



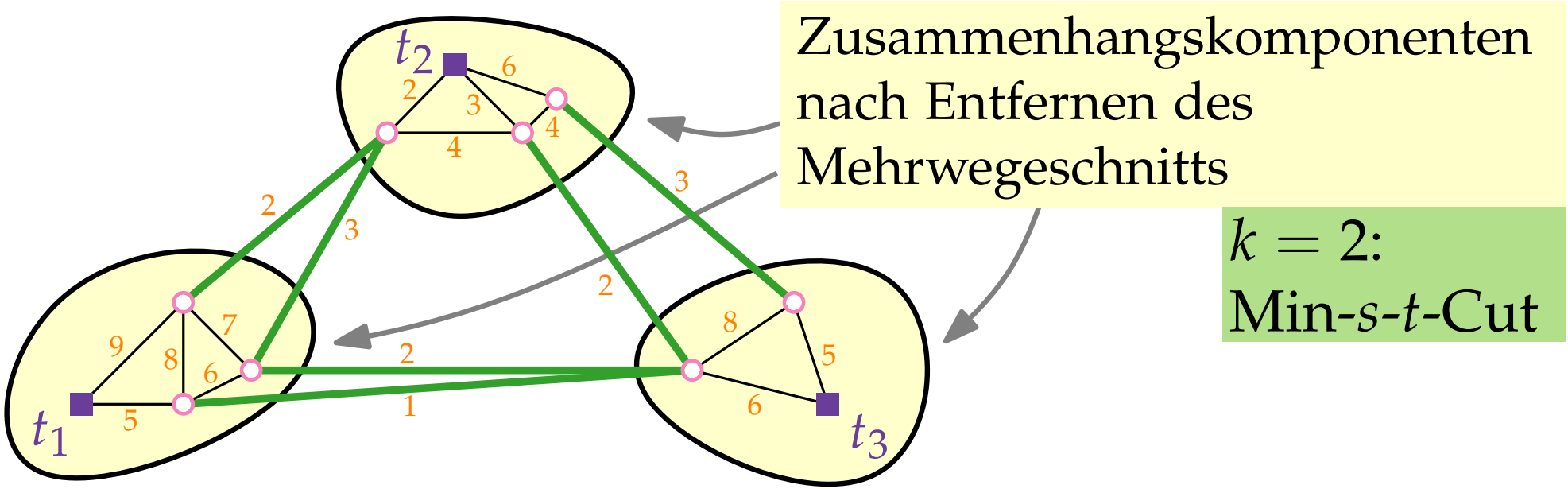
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$ und eine Menge $T = \{t_1, \ldots, t_k\} \subseteq V$ von Terminalen. Ein **Mehrwegeschnitt** für T ist eine Menge E' von Kanten, so dass keine zwei Terminale im Graphen $(V, E - E')$ zusammenhängend sind.
Das Problem MehrwegeSchnitt fragt nach einem kostengünstigsten Mehrwegeschnitt für T.



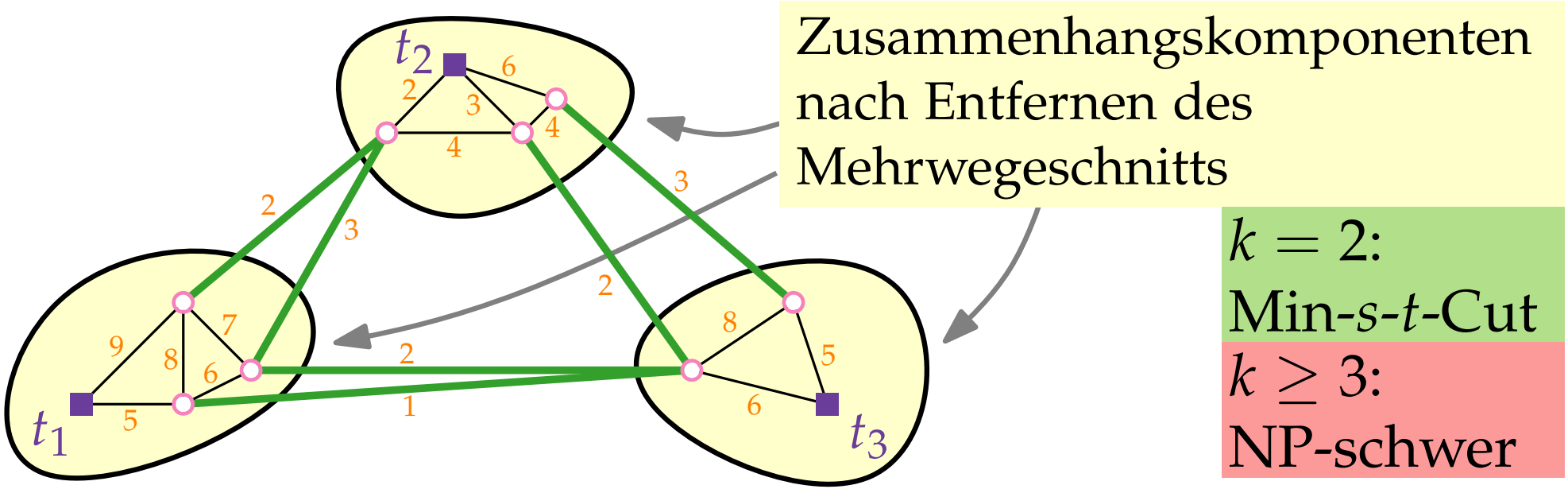
MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$ und eine Menge $T = \{t_1, \ldots, t_k\} \subseteq V$ von Terminalen. Ein **Mehrwegeschnitt** für T ist eine Menge E' von Kanten, so dass keine zwei Terminale im Graphen $(V, E - E')$ zusammenhängend sind.
Das Problem MehrwegeSchnitt fragt nach einem kostengünstigsten Mehrwegeschnitt für T.



MehrwegeSchnitt

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ mit Kantenkosten $c\colon E \to \mathbb{Q}_+$ und eine Menge $T = \{t_1, \ldots, t_k\} \subseteq V$ von Terminalen. Ein **Mehrwegeschnitt** für T ist eine Menge E' von Kanten, so dass keine zwei Terminale im Graphen $(V, E - E')$ zusammenhängend sind.
Das Problem MehrwegeSchnitt fragt nach einem kostengünstigsten Mehrwegeschnitt für T.

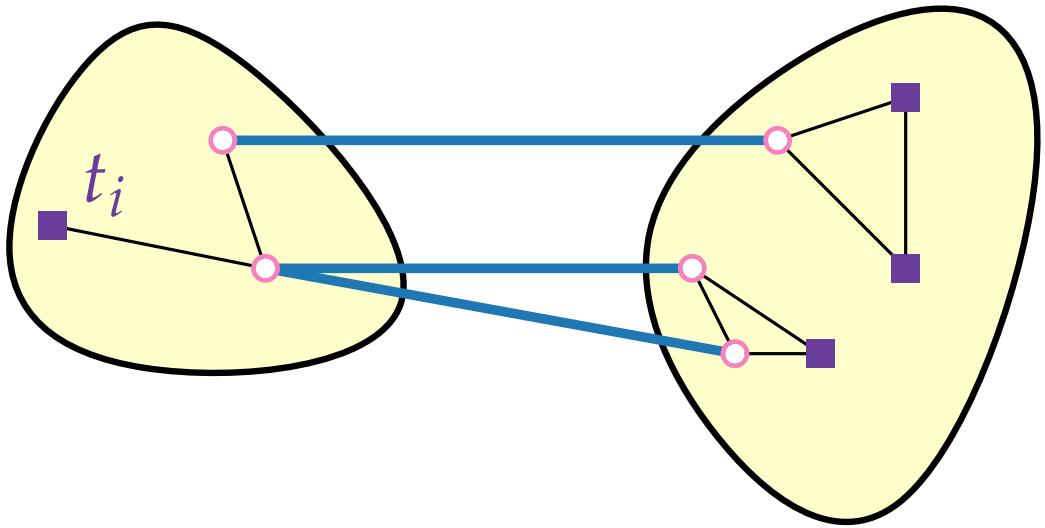


Isolierender Schnitt

Ein **isolierender Schnitt** für Terminal t_i ist eine Menge von Kanten, die t_i von den restlichen Terminalen separiert.

Isolierender Schnitt

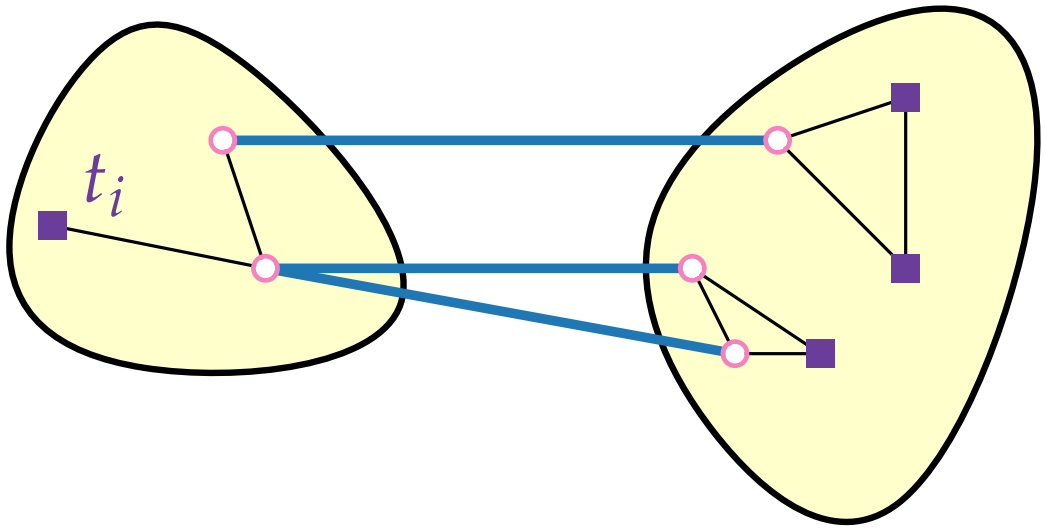
Ein **isolierender Schnitt** für Terminal t_i ist eine Menge von Kanten, die t_i von den restlichen Terminalen separiert.



Isolierender Schnitt

Ein **isolierender Schnitt** für Terminal t_i ist eine Menge von Kanten, die t_i von den restlichen Terminalen separiert.

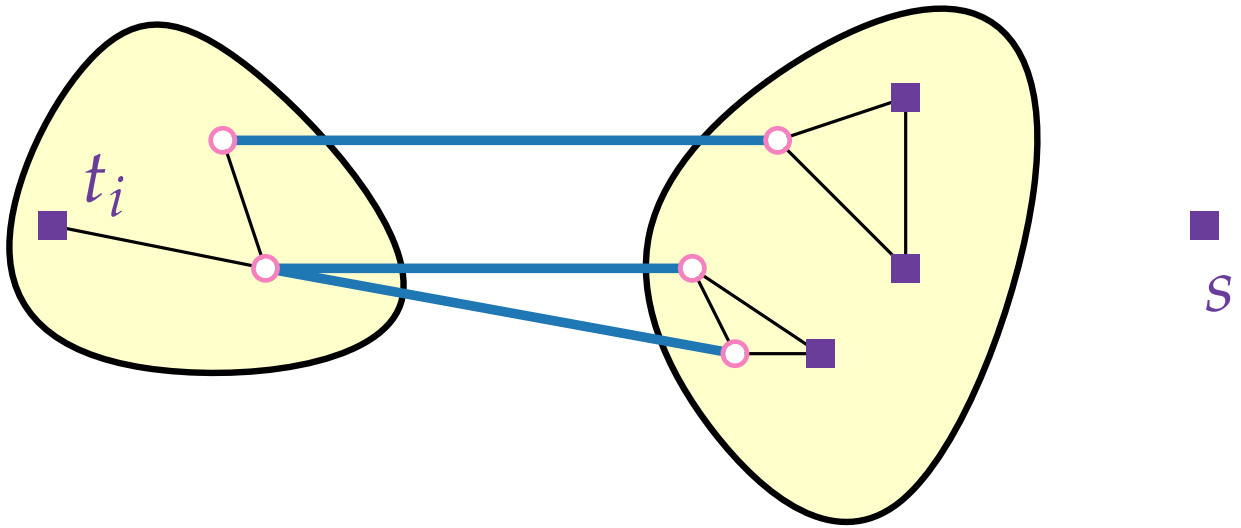
Günstigster isolierender Schnitt effizient berechenbar!



Isolierender Schnitt

Ein **isolierender Schnitt** für Terminal t_i ist eine Menge von Kanten, die t_i von den restlichen Terminalen separiert.

Günstigster isolierender Schnitt effizient berechenbar!

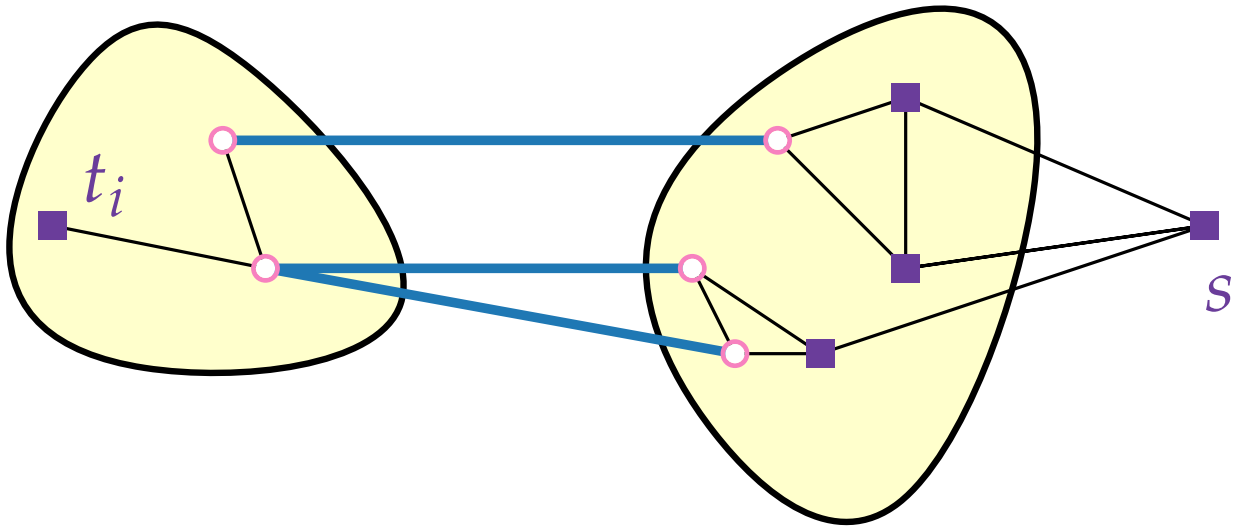


Füge Hilfsterminal s hinzu

Isolierender Schnitt

Ein **isolierender Schnitt** für Terminal t_i ist eine Menge von Kanten, die t_i von den restlichen Terminalen separiert.

Günstigster isolierender Schnitt effizient berechenbar!

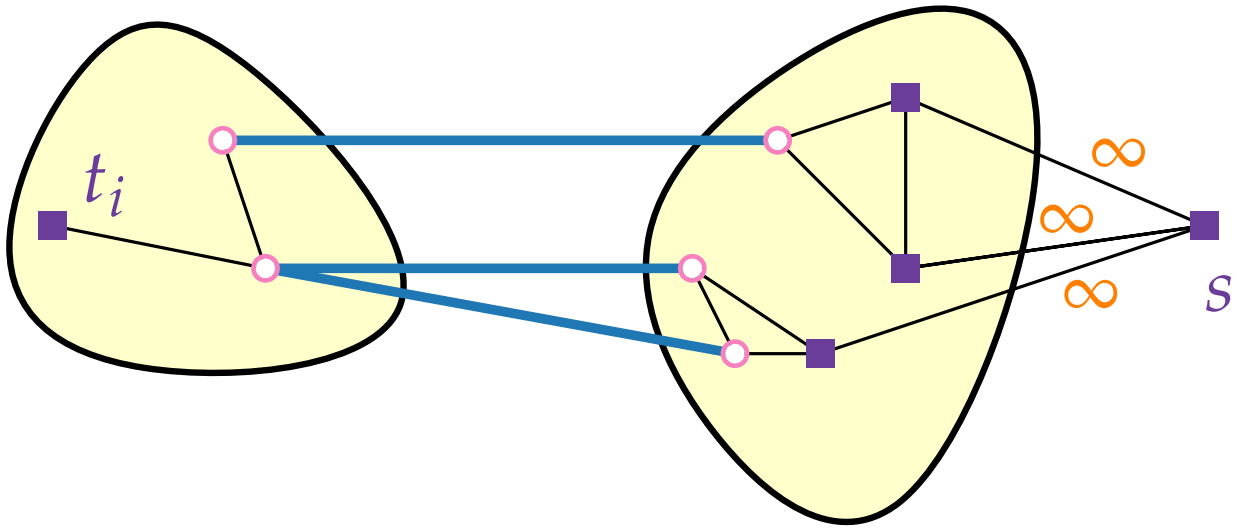


Füge Hilfsterminal s hinzu

Isolierender Schnitt

Ein **isolierender Schnitt** für Terminal t_i ist eine Menge von Kanten, die t_i von den restlichen Terminalen separiert.

Günstigster isolierender Schnitt effizient berechenbar!

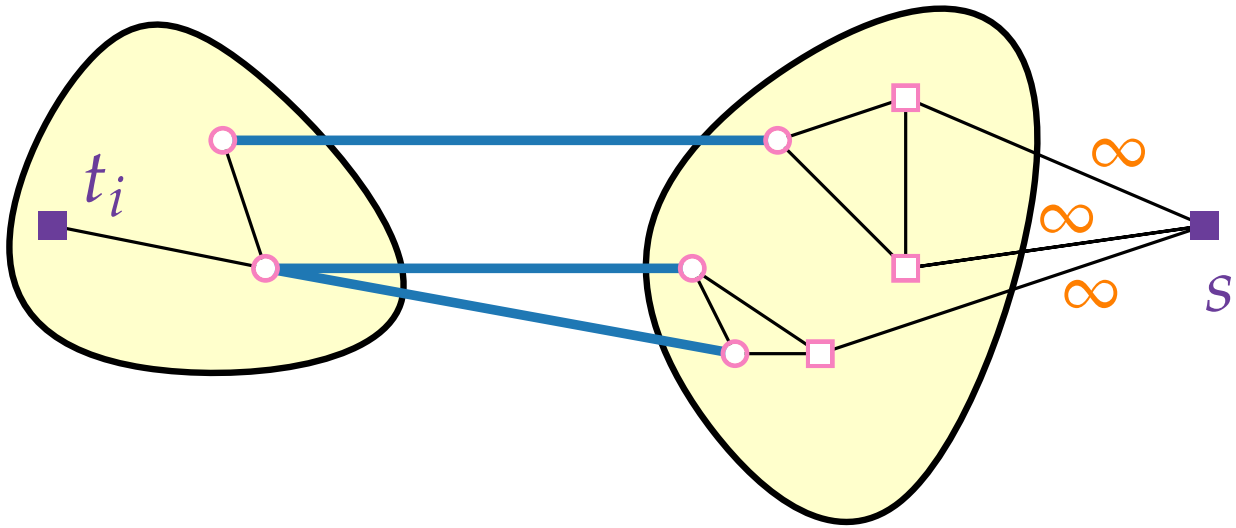


Füge Hilfsterminal s hinzu

Isolierender Schnitt

Ein **isolierender Schnitt** für Terminal t_i ist eine Menge von Kanten, die t_i von den restlichen Terminalen separiert.

Günstigster isolierender Schnitt effizient berechenbar!

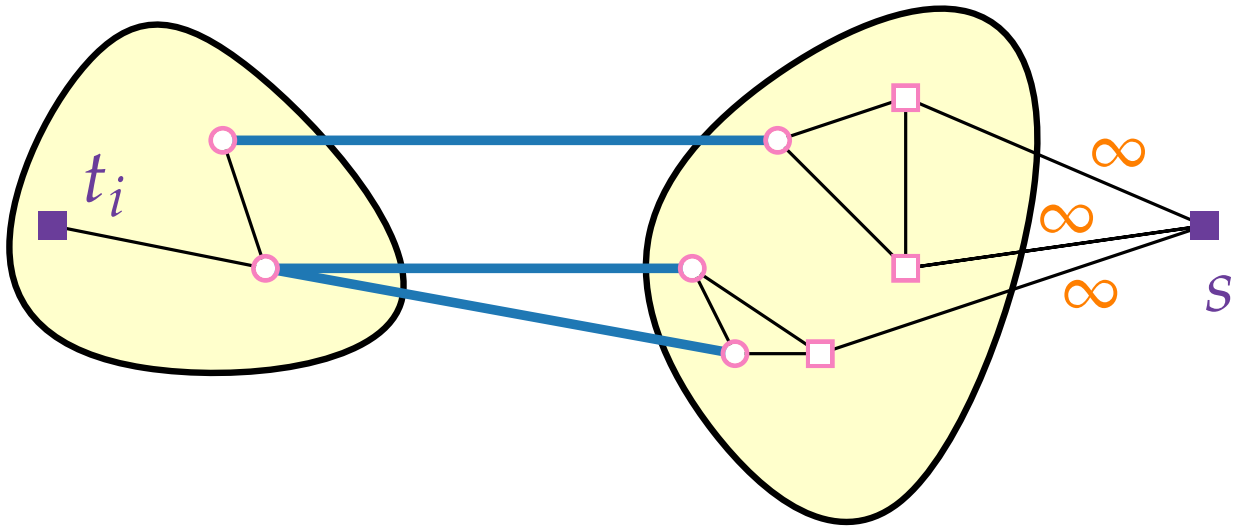


Füge Hilfsterminal s hinzu

Isolierender Schnitt

Ein **isolierender Schnitt** für Terminal t_i ist eine Menge von Kanten, die t_i von den restlichen Terminalen separiert.

Günstigster isolierender Schnitt effizient berechenbar!



Füge Hilfsterminal s hinzu und ermittle günstigsten s-t_i-Schnitt (siehe Vorlesung AGT).

Algorithmus MehrwegeSchnitt

Für $i = 1, \ldots, k$:
Berechne für t_i einen günstigsten isolierenden Schnitt C_i.

Algorithmus MehrwegeSchnitt

Für $i = 1, \ldots, k$:
Berechne für t_i einen günstigsten isolierenden Schnitt C_i.

Gib die Vereinigung $\mathcal{C}$ der $k - 1$ günstigsten dieser Schnitte zurück.

Algorithmus MehrwegeSchnitt

Für $i = 1, \ldots, k$:
Berechne für t_i einen günstigsten isolierenden Schnitt C_i.

Gib die Vereinigung $\mathcal{C}$ der $k - 1$ günstigsten dieser Schnitte zurück.

Mit anderen Worten:
Ignoriere den teuersten der isolierenden Schnitte $C_1, \ldots, C_k$.

Algorithmus MehrwegeSchnitt

Für $i = 1, \ldots, k$:
Berechne für t_i einen günstigsten isolierenden Schnitt C_i.

Gib die Vereinigung $\mathcal{C}$ der $k-1$ günstigsten dieser Schnitte zurück.

Mit anderen Worten:
Ignoriere den teuersten der isolierenden Schnitte $C_1, \ldots, C_k$.

$$\Rightarrow c(\mathcal{C}) \quad ? \quad \sum_{i=1}^{k} c(C_i)$$

Algorithmus MehrwegeSchnitt

Für $i = 1, \ldots, k$:
Berechne für t_i einen günstigsten isolierenden Schnitt C_i.

Gib die Vereinigung $\mathcal{C}$ der $k-1$ günstigsten dieser Schnitte zurück.

Mit anderen Worten:
Ignoriere den teuersten der isolierenden Schnitte $C_1, \ldots, C_k$.

$$\Rightarrow c(\mathcal{C}) \leq \sum_{i=1}^{k} c(C_i)$$

Algorithmus MehrwegeSchnitt

Für $i = 1, \ldots, k$:
Berechne für t_i einen günstigsten isolierenden Schnitt C_i.

Gib die Vereinigung $\mathcal{C}$ der $k-1$ günstigsten dieser Schnitte zurück.

Mit anderen Worten:
Ignoriere den teuersten der isolierenden Schnitte $C_1, \ldots, C_k$.

$$\Rightarrow c(\mathcal{C}) \leq \frac{k-1}{k} \sum_{i=1}^{k} c(C_i)$$

Algorithmus MehrwegeSchnitt

Für $i = 1, \ldots, k$:
Berechne für t_i einen günstigsten isolierenden Schnitt C_i.

Gib die Vereinigung $\mathcal{C}$ der $k-1$ günstigsten dieser Schnitte zurück.

Mit anderen Worten:
Ignoriere den teuersten der isolierenden Schnitte $C_1, \ldots, C_k$.

$\Rightarrow c(\mathcal{C}) \leq \frac{k-1}{k} \sum_{i=1}^{k} c(C_i)$, wegen des Schubfachprinzips:

Algorithmus MehrwegeSchnitt

Für $i = 1, \ldots, k$:
Berechne für t_i einen günstigsten isolierenden Schnitt C_i.

Gib die Vereinigung $\mathcal{C}$ der $k-1$ günstigsten dieser Schnitte zurück.

Mit anderen Worten:
Ignoriere den teuersten der isolierenden Schnitte $C_1, \ldots, C_k$.

$\Rightarrow c(\mathcal{C}) \leq \frac{k-1}{k} \sum_{i=1}^{k} c(C_i)$, wegen des Schubfachprinzips:

denn für den teuersten der Schnitte $C_1, \ldots, C_k$, sagen wir C_1, gilt

$$c(C_1) \geq$$

Algorithmus MehrwegeSchnitt

Für $i = 1, \ldots, k$:
Berechne für t_i einen günstigsten isolierenden Schnitt C_i.

Gib die Vereinigung $\mathcal{C}$ der $k-1$ günstigsten dieser Schnitte zurück.

Mit anderen Worten:
Ignoriere den teuersten der isolierenden Schnitte $C_1, \ldots, C_k$.

$\Rightarrow c(\mathcal{C}) \leq \frac{k-1}{k} \sum_{i=1}^{k} c(C_i)$, wegen des Schubfachprinzips:

denn für den teuersten der Schnitte $C_1, \ldots, C_k$, sagen wir C_1, gilt

$$c(C_1) \geq \frac{1}{k} \sum_{i=1}^{k} c(C_i).$$

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-( )-Approximationsalgorithmus für MehrwegeSchnitt.

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Approximationsgüte

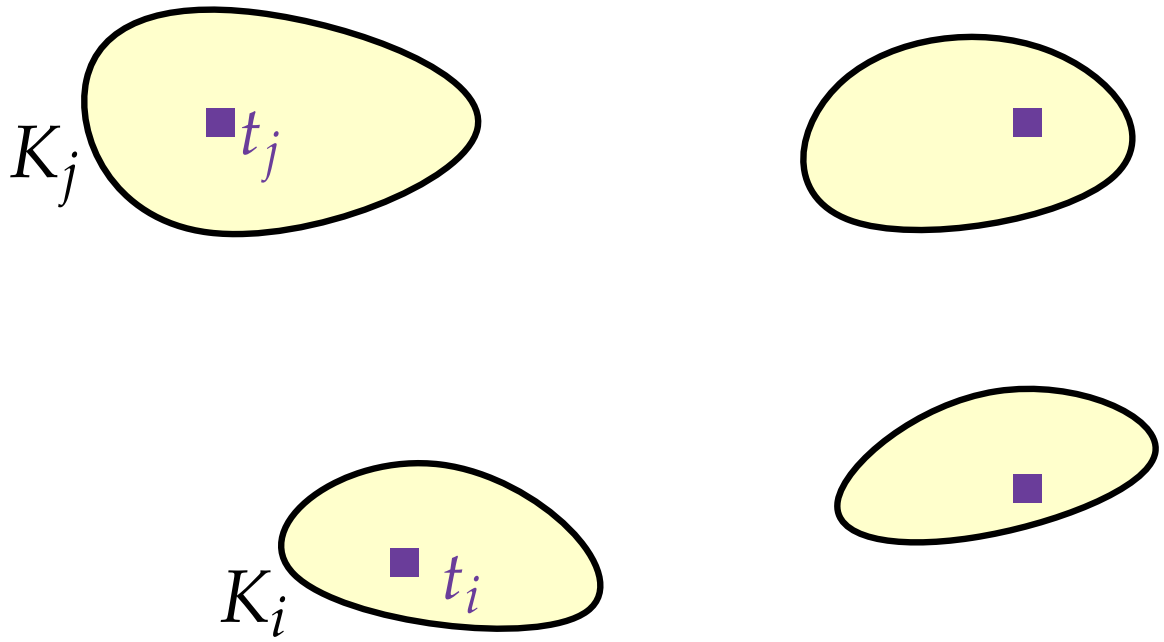
Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

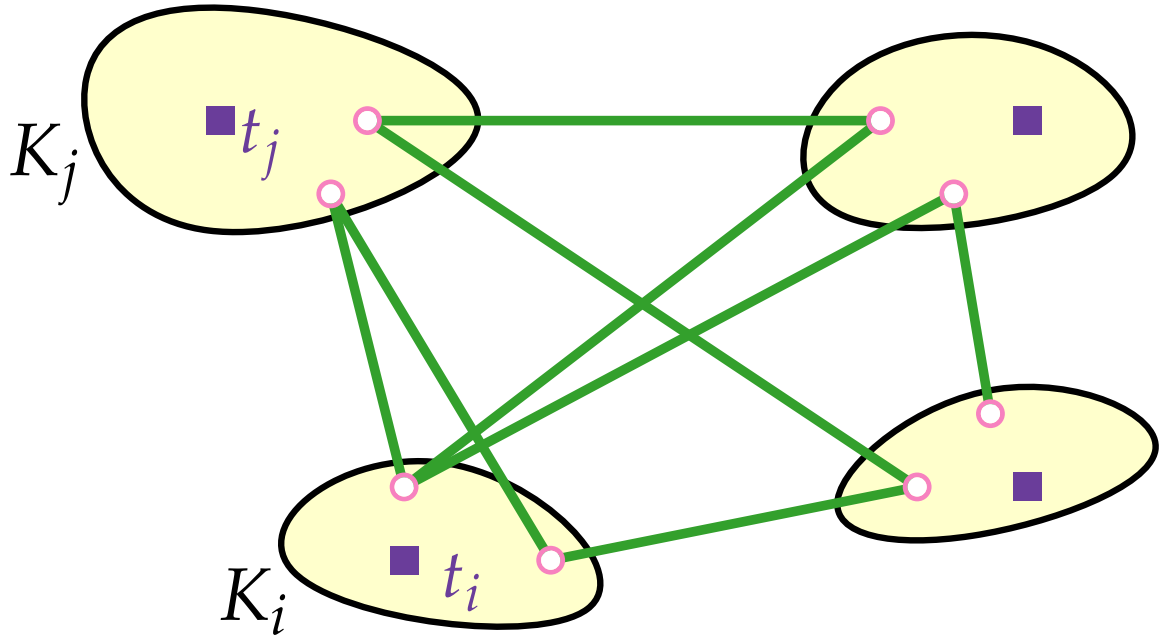
Beweis. Betrachte optimalen Mehrwegeschnitt A:



Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

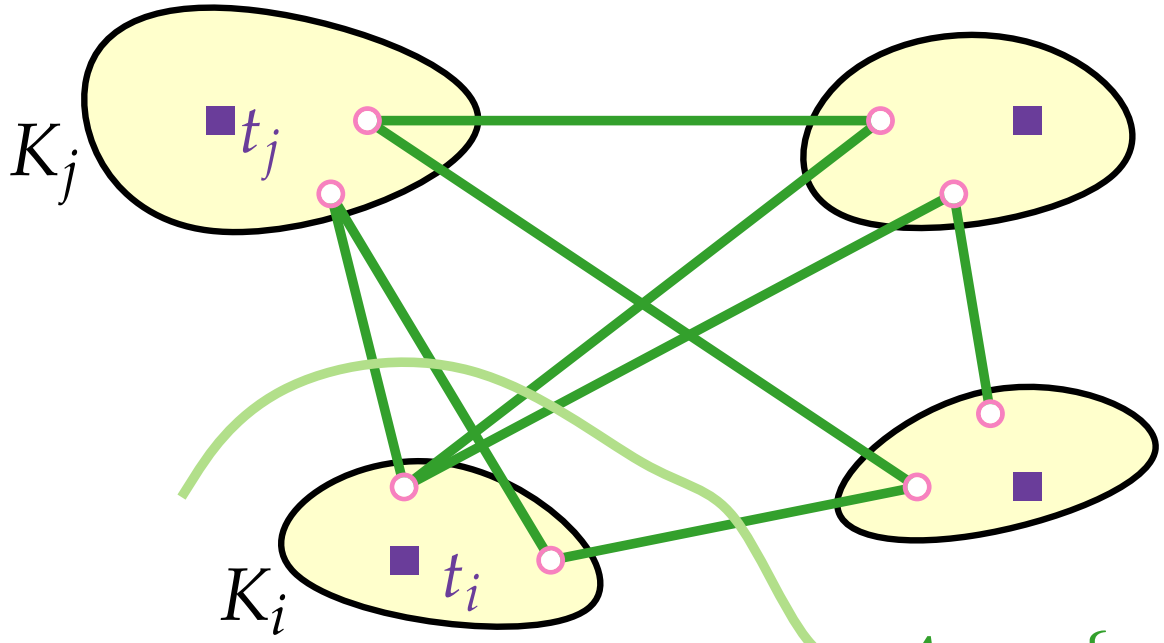
Beweis. Betrachte optimalen Mehrwegeschnitt A:



Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:

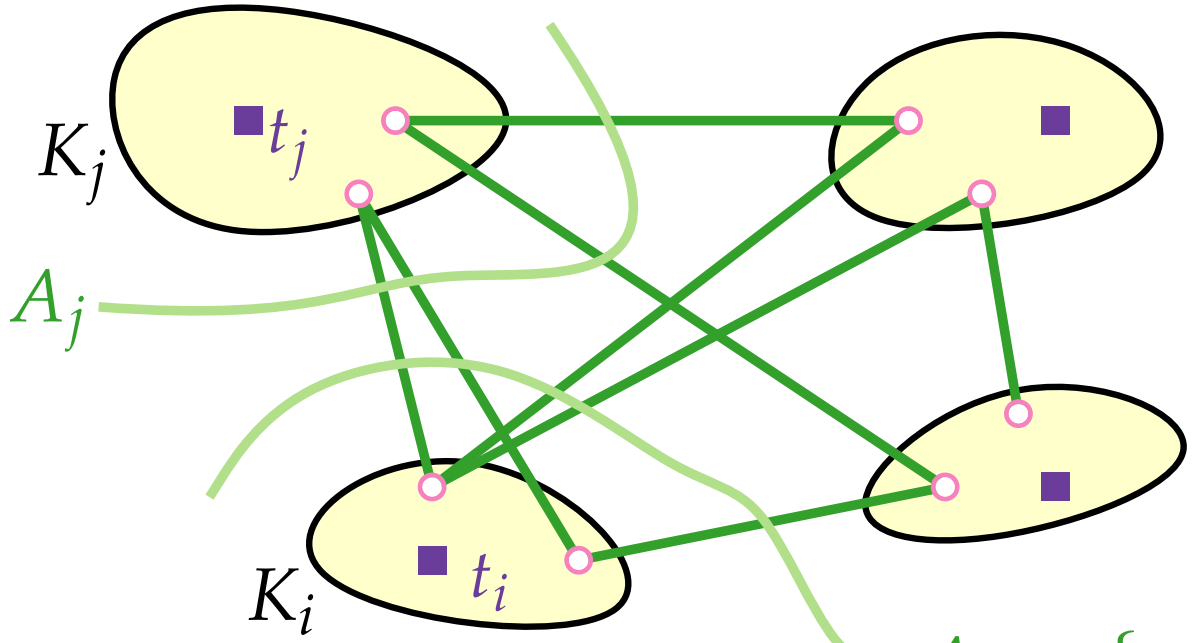


$A_i = \{uv \in A \colon u \in K_i, v \notin K_i\}$

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:

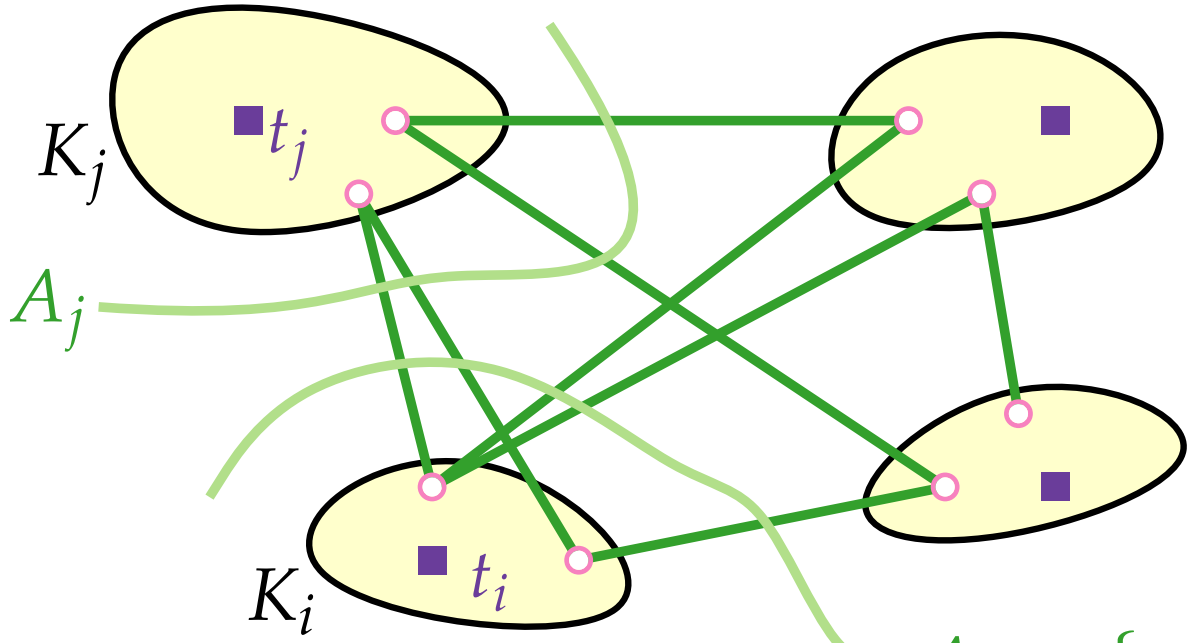


$A_i = \{uv \in A : u \in K_i, v \notin K_i\}$

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:



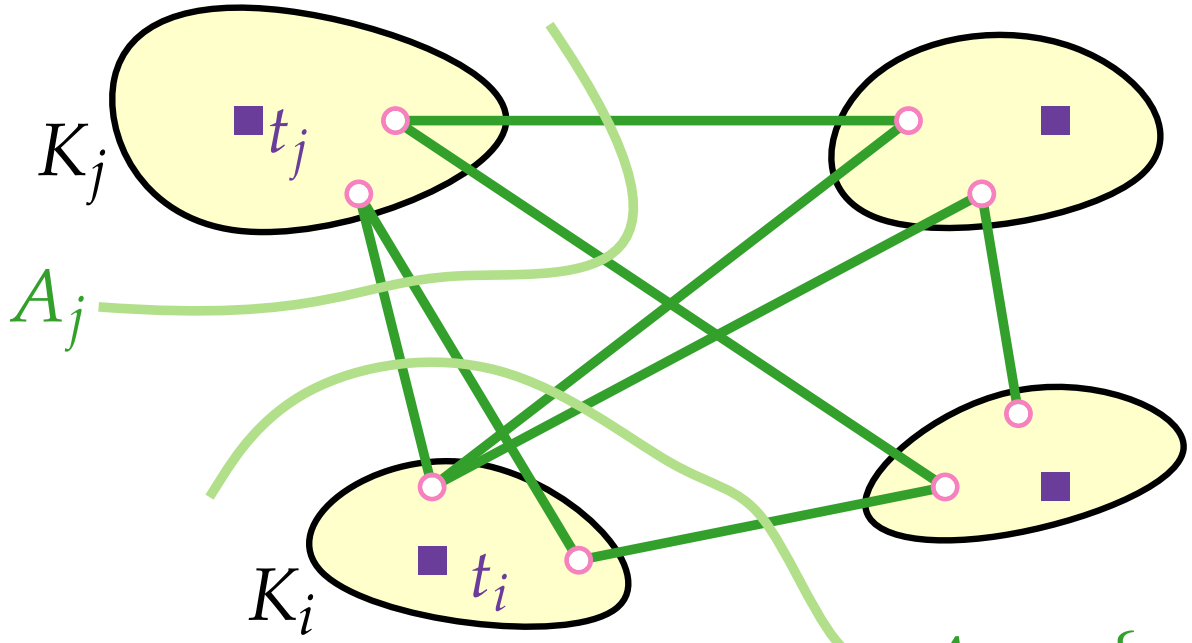
$A_i = \{uv \in A : u \in K_i, v \notin K_i\}$

Beobachtung. $A =$

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:



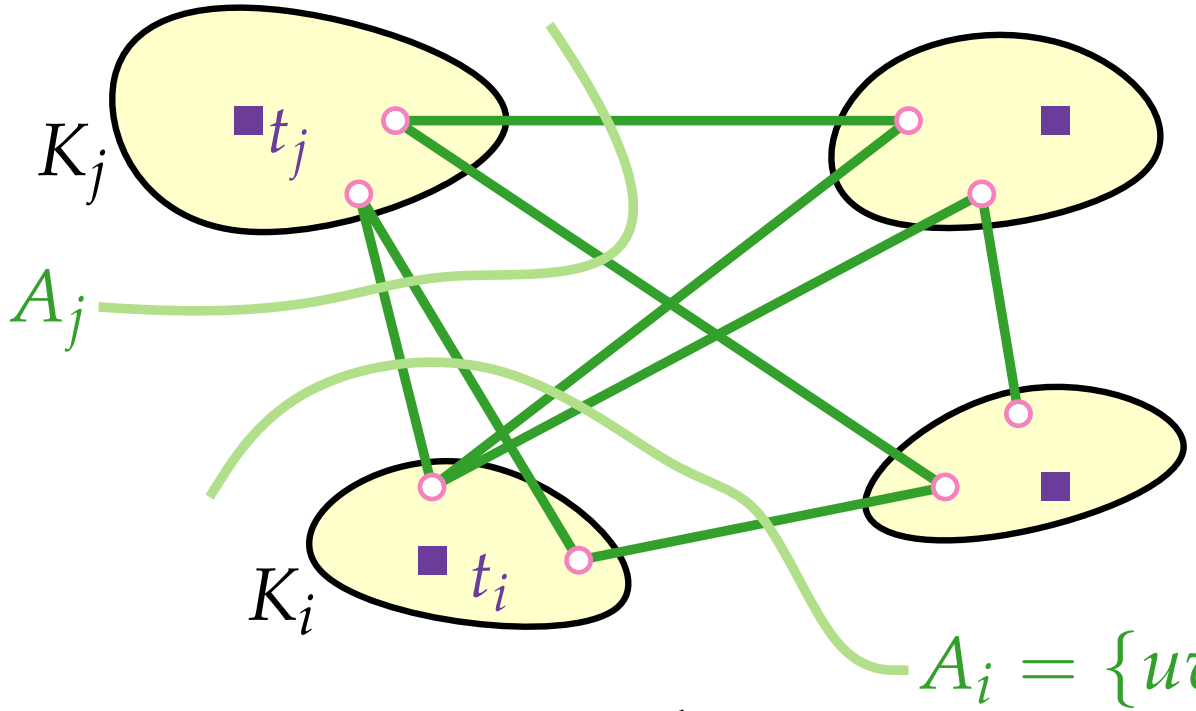
$A_i = \{uv \in A : u \in K_i, v \notin K_i\}$

Beobachtung. $A = \bigcup_{i=1}^{k} A_i$

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:



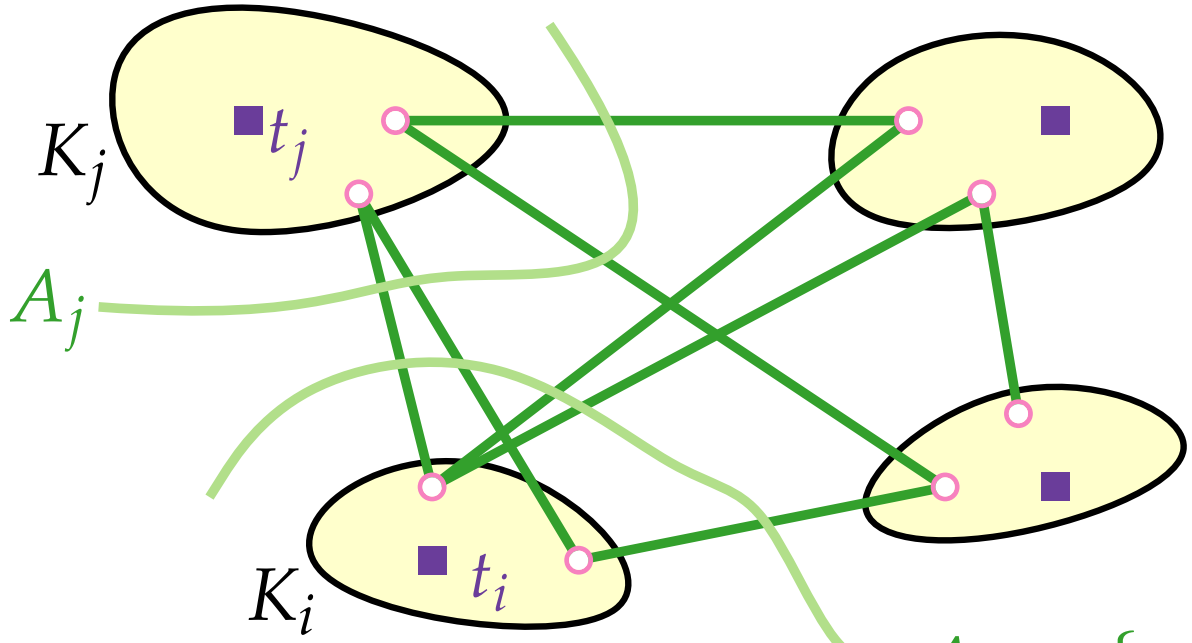
$A_i = \{uv \in A : u \in K_i, v \notin K_i\}$

Beobachtung. $A = \bigcup_{i=1}^{k} A_i$ und $c(A) =$

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:



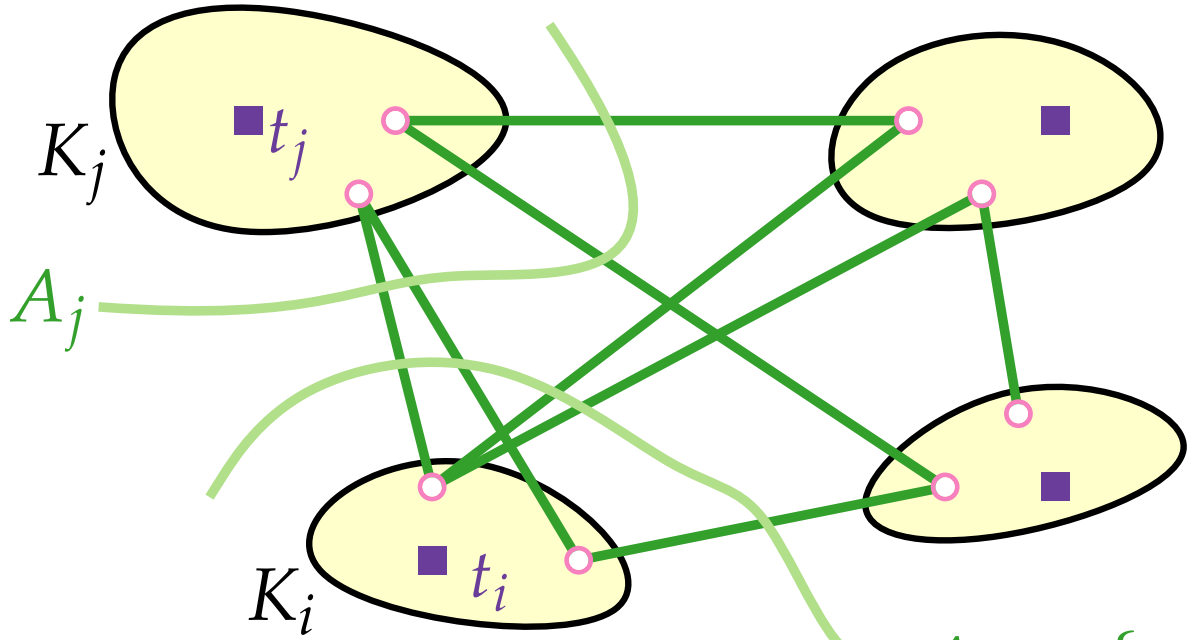
$A_i = \{uv \in A : u \in K_i, v \notin K_i\}$

Beobachtung. $A = \bigcup_{i=1}^{k} A_i$ und $c(A) = \frac{1}{2} \sum_{i=1}^{k} c(A_i)$

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:



$A_i = \{uv \in A : u \in K_i, v \notin K_i\}$

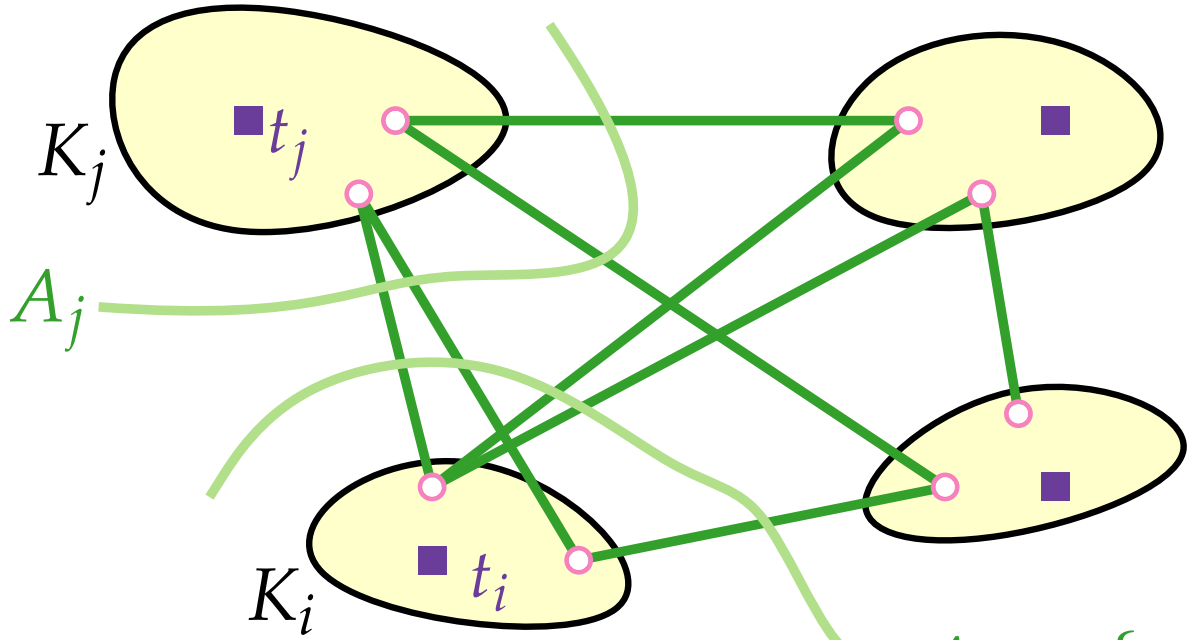
Beobachtung. $A = \bigcup_{i=1}^{k} A_i$ und $c(A) = \frac{1}{2} \sum_{i=1}^{k} c(A_i)$

$\Rightarrow \mathrm{OPT} = c(A) \geq$

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:



$A_i = \{uv \in A : u \in K_i, v \notin K_i\}$

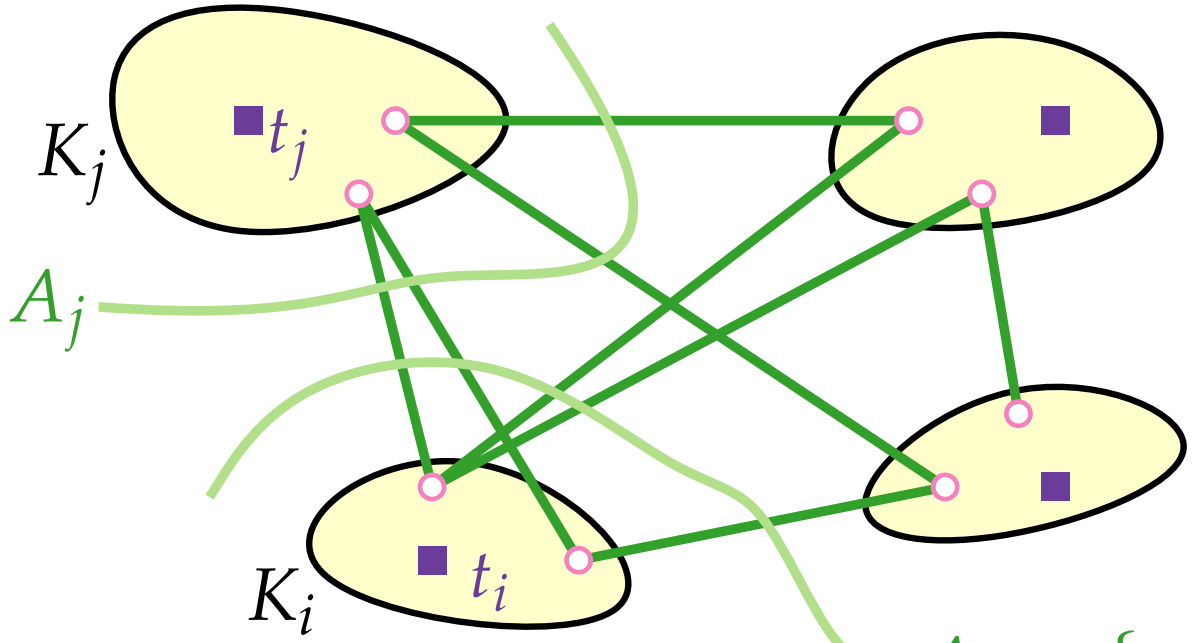
Beobachtung. $A = \bigcup_{i=1}^{k} A_i$ und $c(A) = \frac{1}{2} \sum_{i=1}^{k} c(A_i)$

$\Rightarrow \mathrm{OPT} = c(A) \geq \frac{1}{2} \sum_{i=1}^{k} c(C_i) \geq$

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MEHRWEGESCHNITT.

Beweis. Betrachte optimalen Mehrwegeschnitt A:



$A_i = \{uv \in A : u \in K_i, v \notin K_i\}$

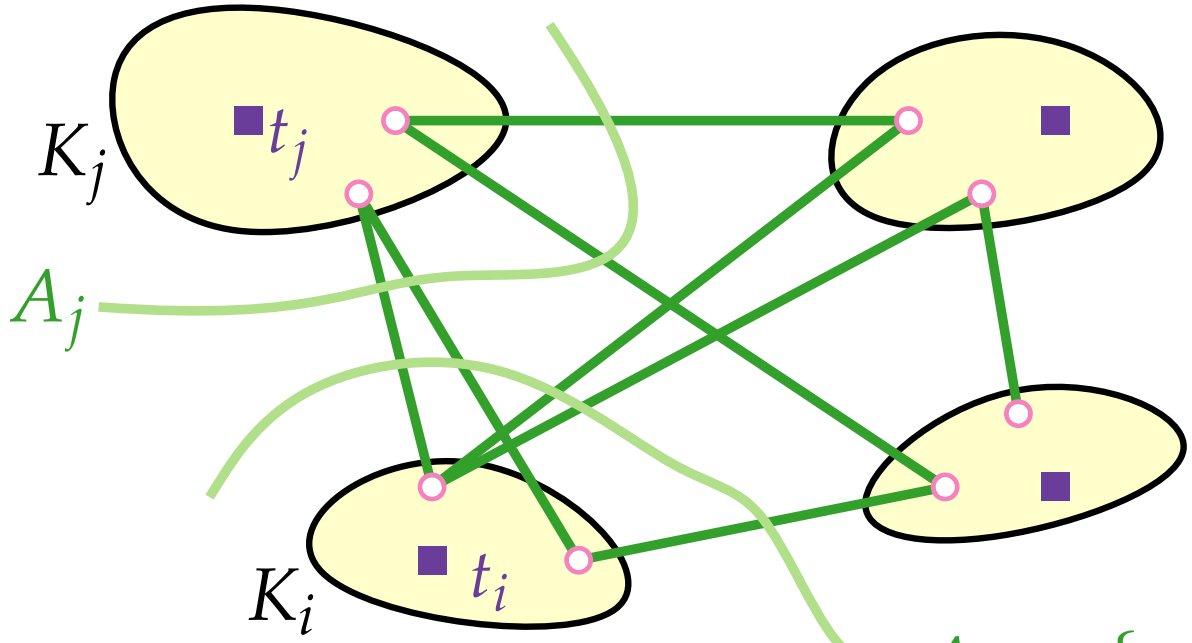
Beobachtung. $A = \bigcup_{i=1}^{k} A_i$ und $c(A) = \frac{1}{2} \sum_{i=1}^{k} c(A_i)$

$\Rightarrow \mathrm{OPT} = c(A) \geq \frac{1}{2} \sum_{i=1}^{k} c(C_i) \geq \frac{1}{2} \cdot \frac{k}{k-1} c(C)$.

Approximationsgüte

Satz. Obiger Algorithmus ist ein Faktor-$(2 - 2/k)$-Approximationsalgorithmus für MehrwegeSchnitt.

Beweis. Betrachte optimalen Mehrwegeschnitt A:



$A_i = \{uv \in A : u \in K_i, v \notin K_i\}$

Beobachtung. $A = \bigcup_{i=1}^{k} A_i$ und $c(A) = \frac{1}{2}\sum_{i=1}^{k} c(A_i)$

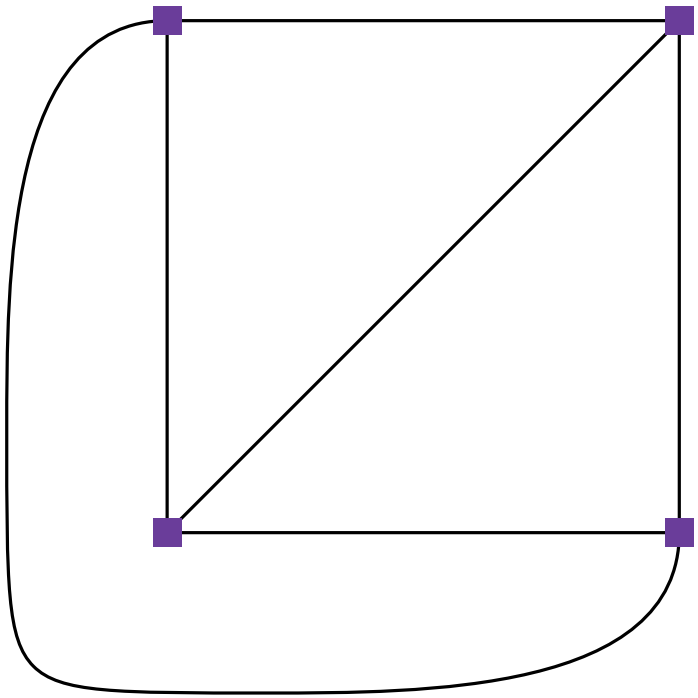
$$\Rightarrow \mathrm{OPT} = c(A) \geq \frac{1}{2}\sum_{i=1}^{k} c(C_i) \geq \frac{1}{2} \cdot \frac{k}{k-1} \overbrace{c(C)}^{\mathrm{ALG}}.$$

Analyse scharf?

K_k

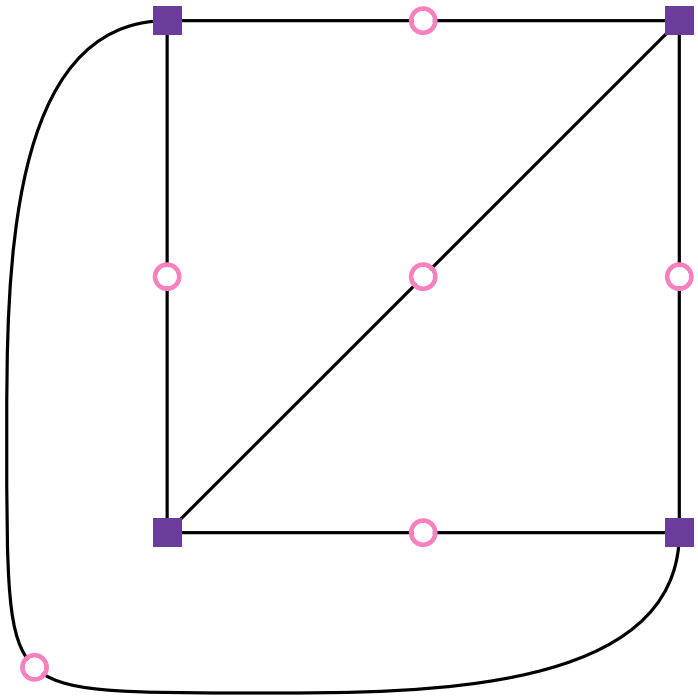
Analyse scharf?

K_k



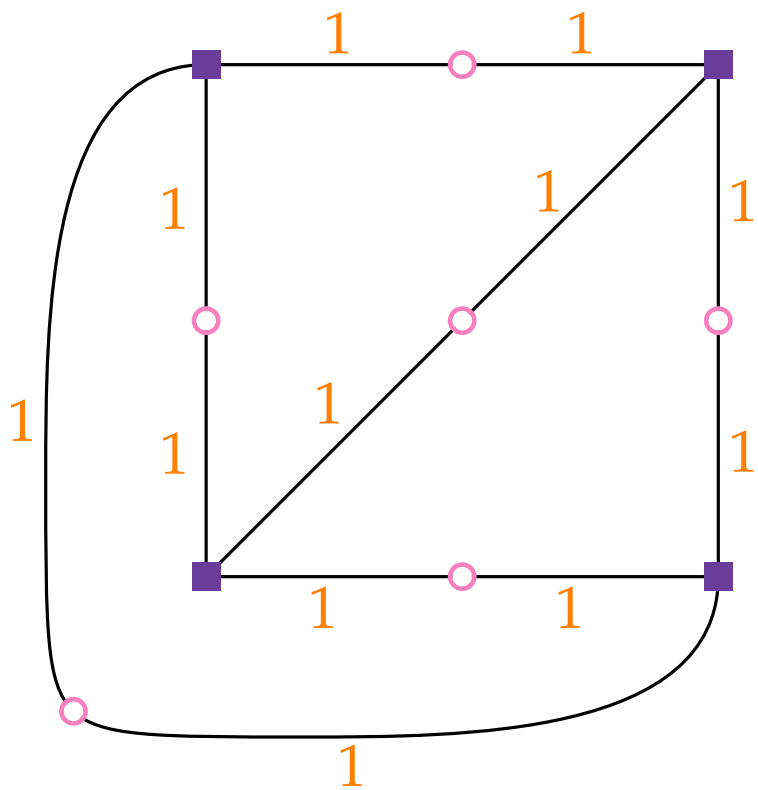
Analyse scharf?

K_k



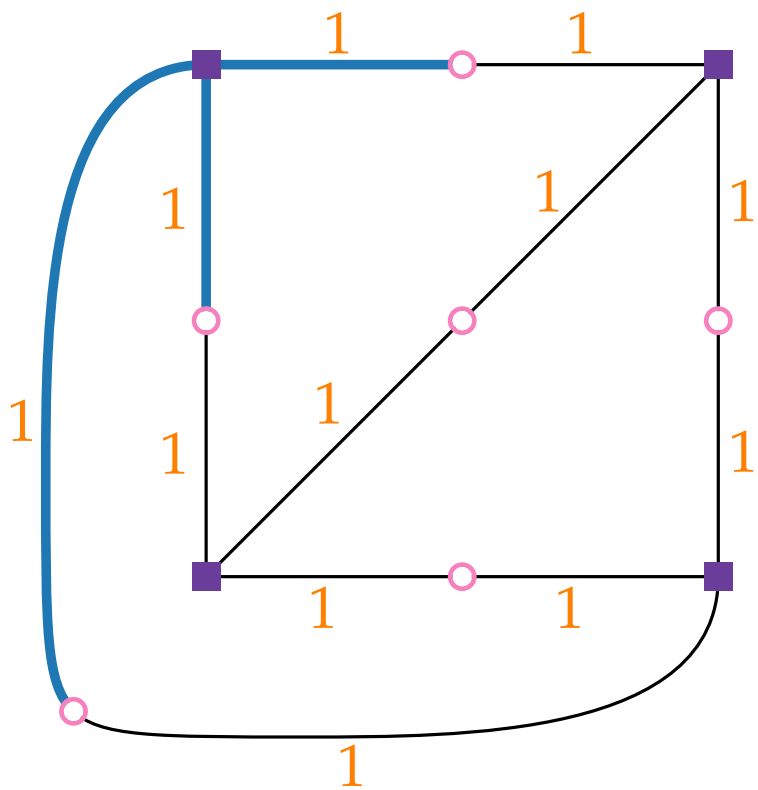
Analyse scharf?

K_k



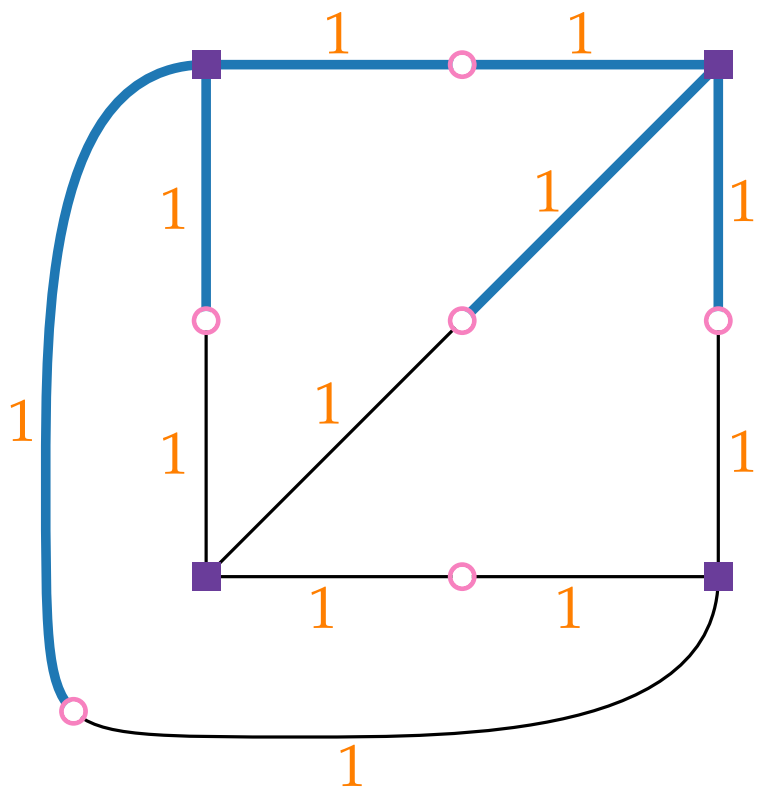
Analyse scharf?

K_k



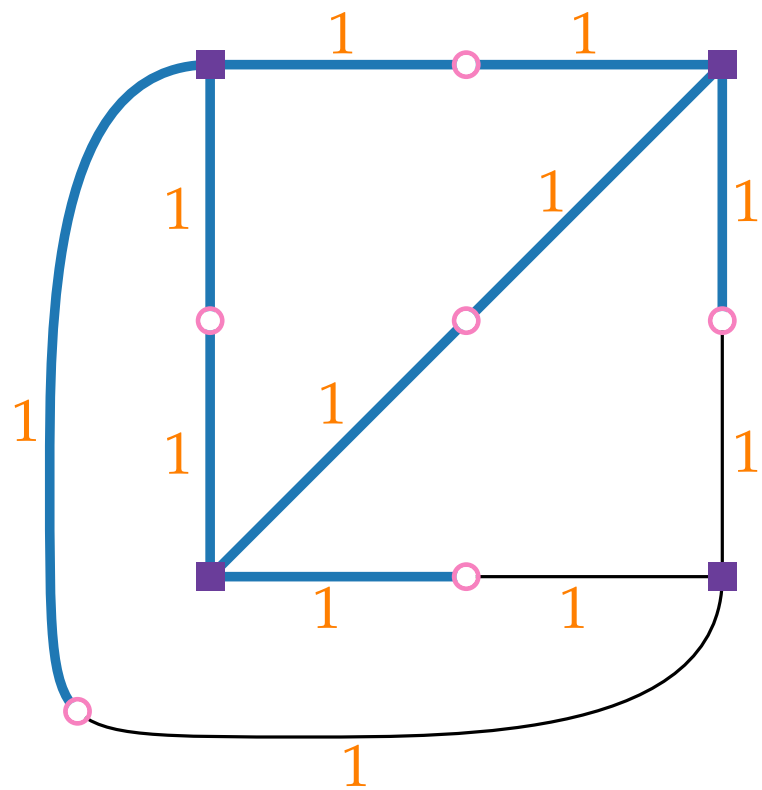
Analyse scharf?

K_k



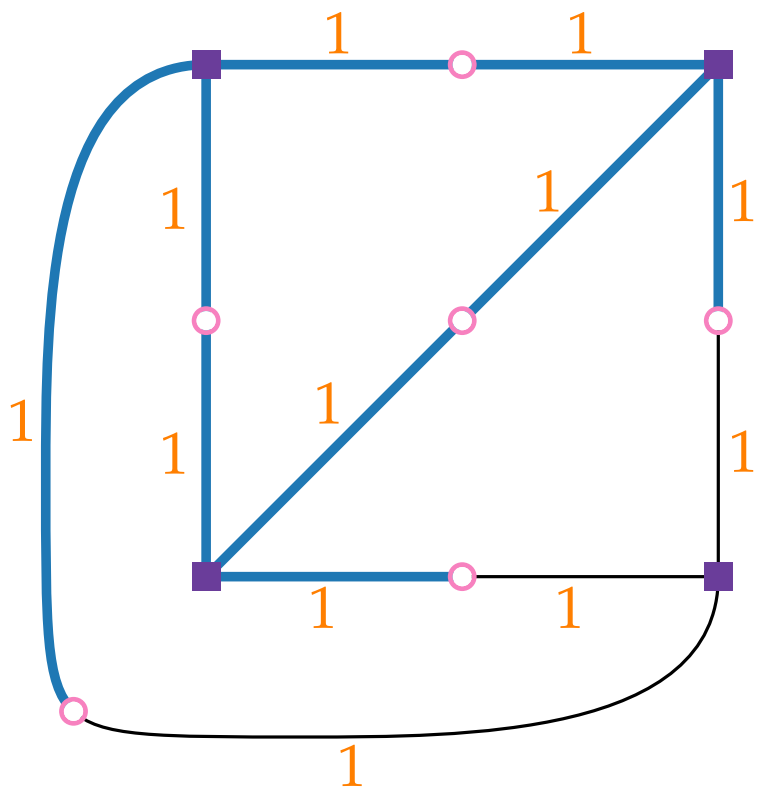
Analyse scharf?

K_k



Analyse scharf?

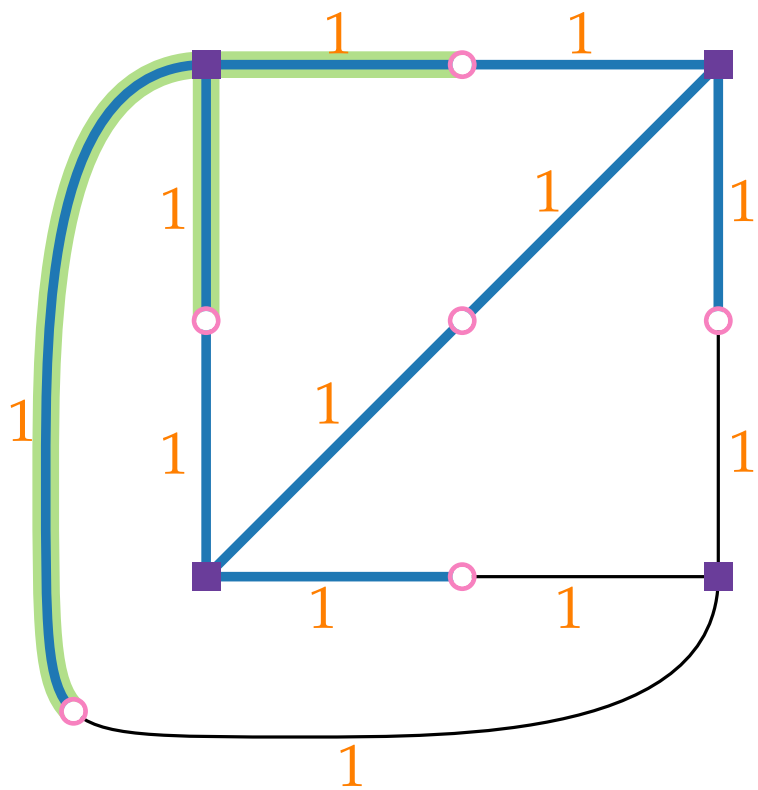
K_k



$$\text{ALG} = (k-1)(k-1)$$

Analyse scharf?

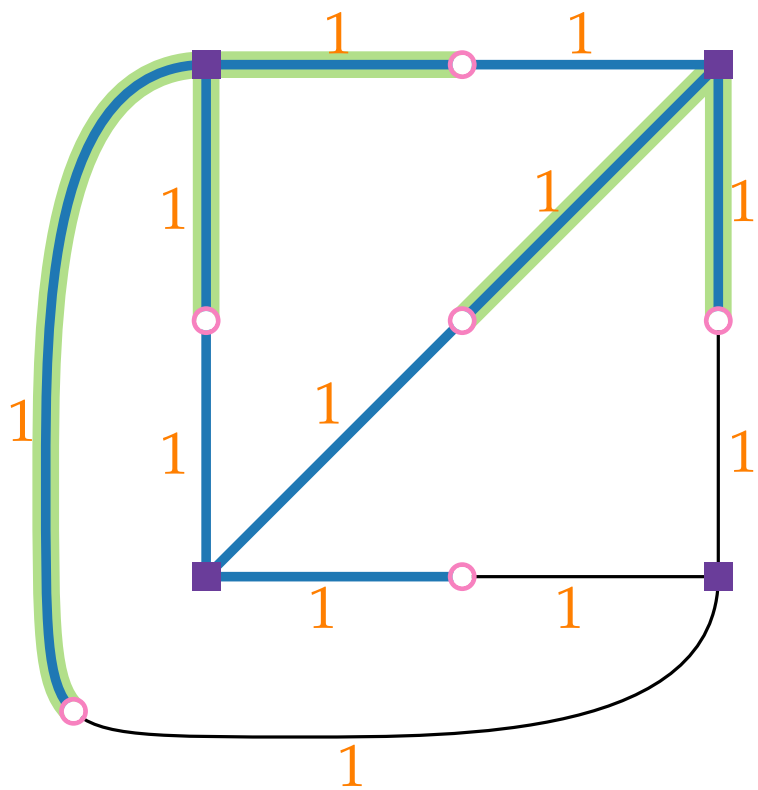
K_k



$$\text{ALG} = (k-1)(k-1)$$

Analyse scharf?

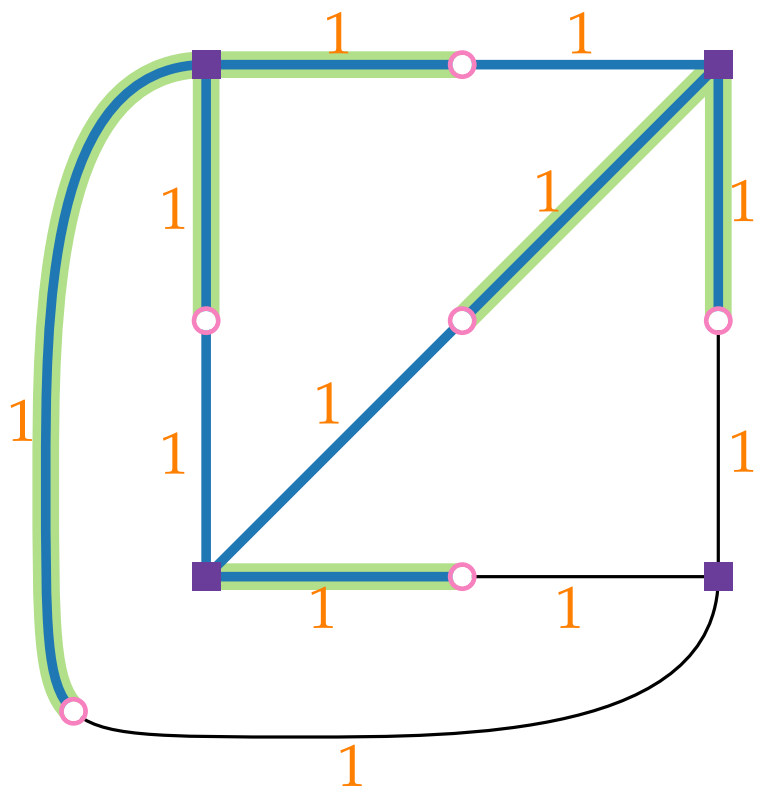
K_k



$$\text{ALG} = (k-1)(k-1)$$

Analyse scharf?

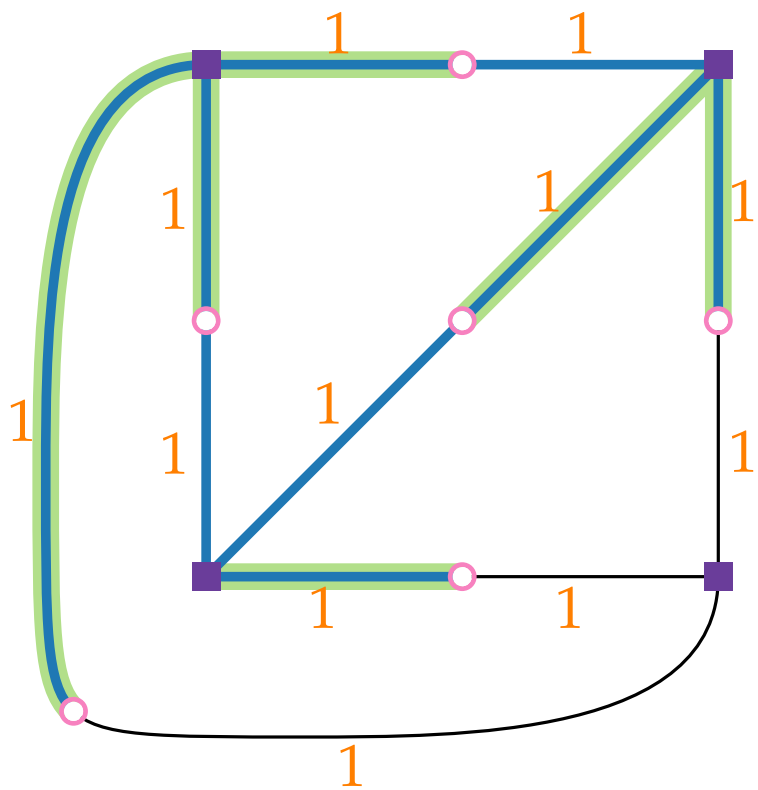
K_k



$$\text{ALG} = (k-1)(k-1)$$

Analyse scharf?

K_k

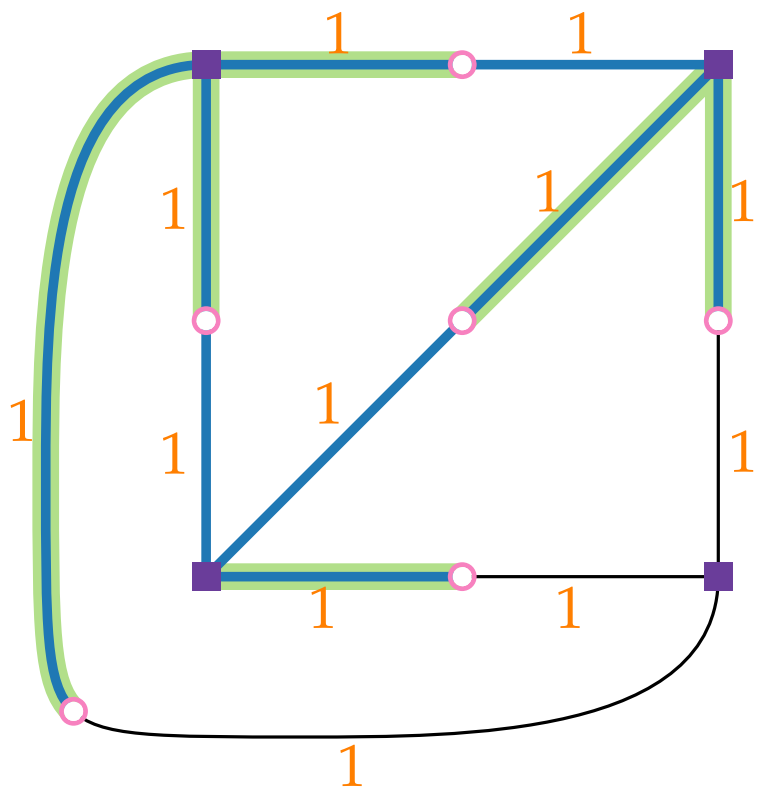


$\text{ALG} = (k-1)(k-1)$

$\text{OPT} = \sum_{i=1}^{k-1} i =$

Analyse scharf?

K_k

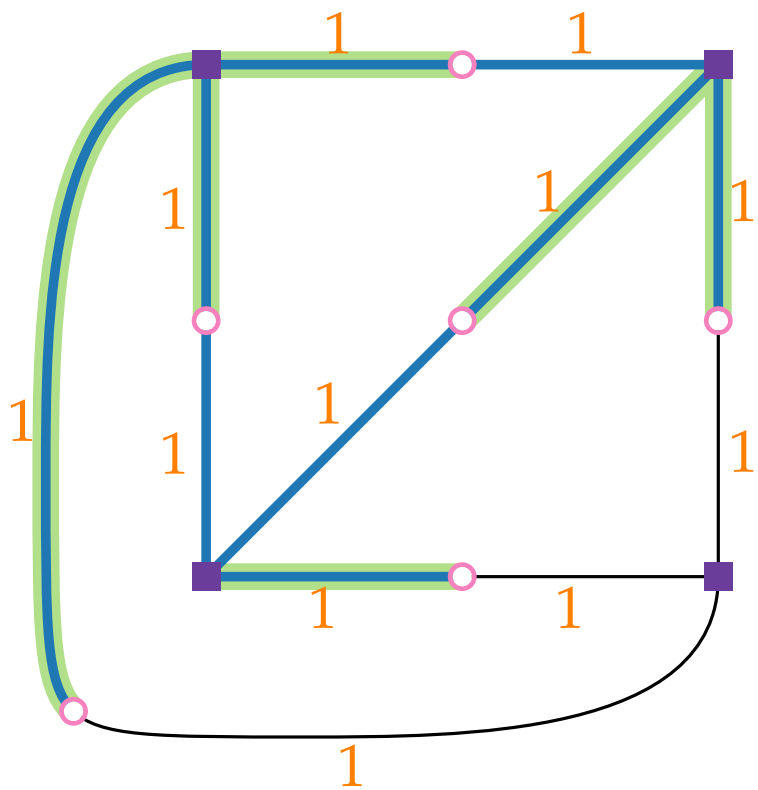


$$\text{ALG} = (k-1)(k-1)$$

$$\text{OPT} = \sum_{i=1}^{k-1} i = \frac{k\cdot(k-1)}{2}$$

Analyse scharf?

K_k



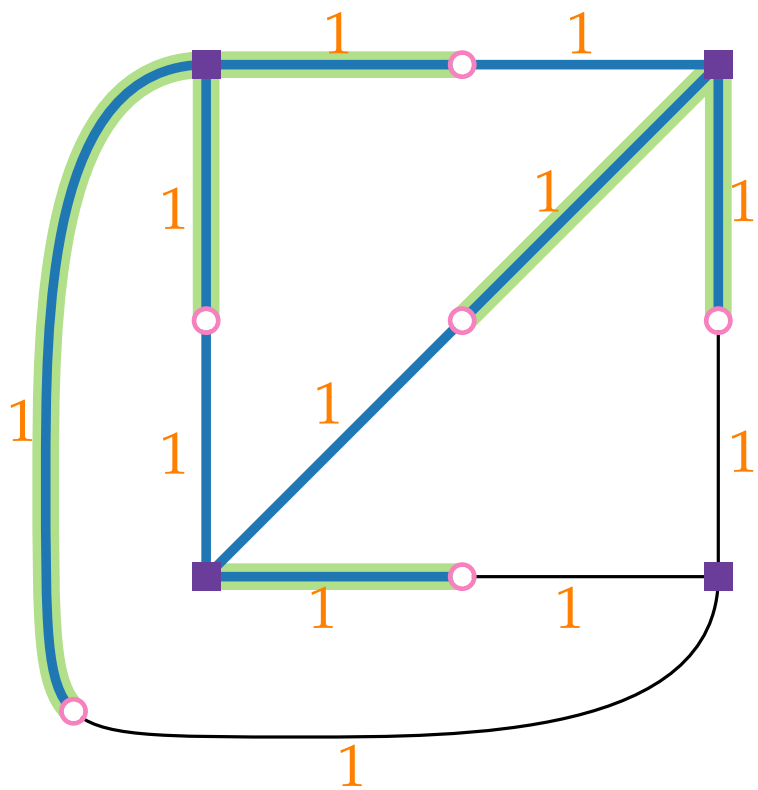
$$\text{ALG} = (k-1)(k-1)$$

$$\text{OPT} = \sum_{i=1}^{k-1} i = \frac{k\cdot(k-1)}{2}$$

$$\text{ALG}/\text{OPT} =$$

Analyse scharf?

K_k



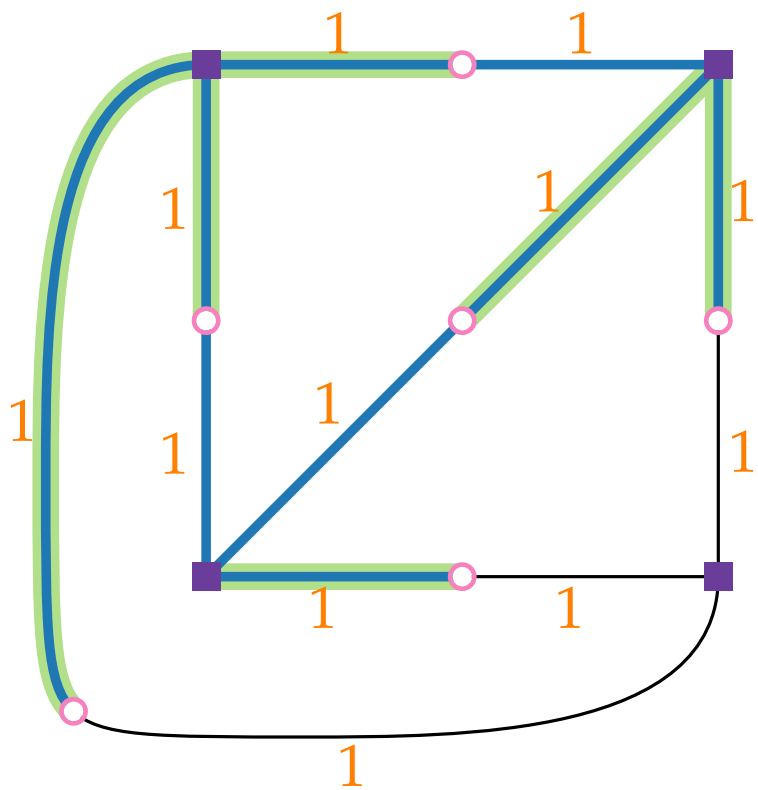
$$\text{ALG} = (k-1)(k-1)$$

$$\text{OPT} = \sum_{i=1}^{k-1} i = \frac{k\cdot(k-1)}{2}$$

$$\text{ALG}/\text{OPT} = \frac{2k-2}{k} =$$

Analyse scharf?

K_k



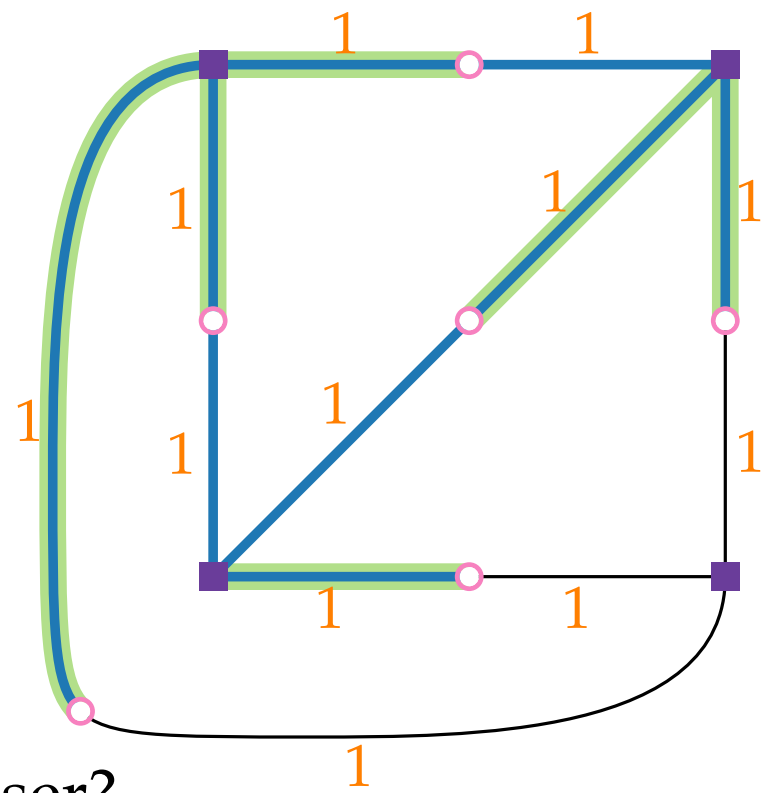
$$\text{ALG} = (k-1)(k-1)$$

$$\text{OPT} = \sum_{i=1}^{k-1} i = \frac{k\cdot(k-1)}{2}$$

$$\text{ALG}/\text{OPT} = \frac{2k-2}{k} = 2 - \frac{2}{k}$$

Analyse scharf?

K_k



besser?

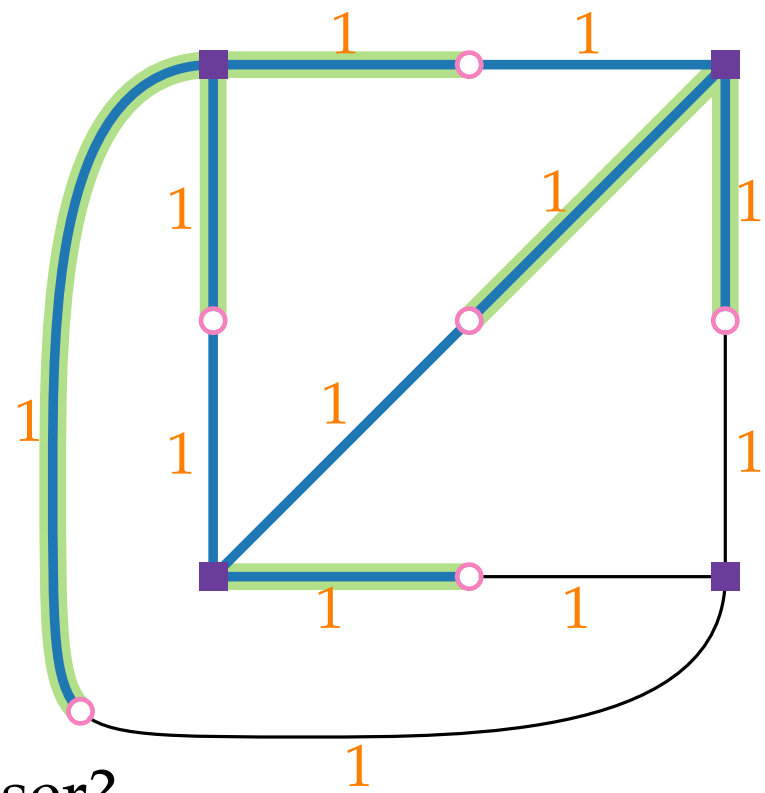
$$\text{ALG} = (k-1)(k-1)$$

$$\text{OPT} = \sum_{i=1}^{k-1} i = \frac{k\cdot(k-1)}{2}$$

$$\text{ALG}/\text{OPT} = \frac{2k-2}{k} = 2 - \frac{2}{k}$$

Analyse scharf?

K_k



$$\text{ALG} = (k-1)(k-1)$$

$$\text{OPT} = \sum_{i=1}^{k-1} i = \frac{k\cdot(k-1)}{2}$$

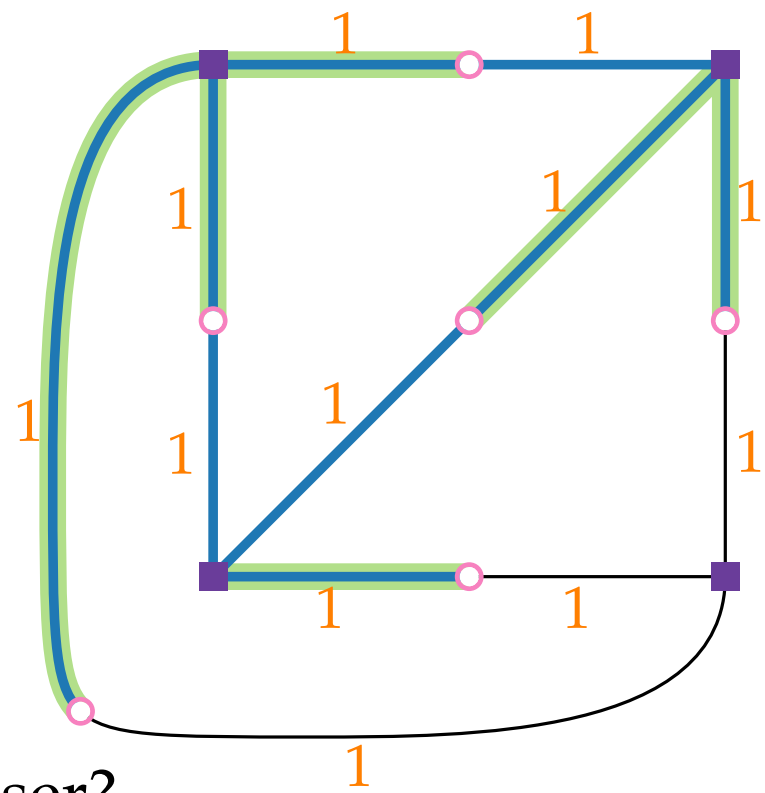
$$\text{ALG}/\text{OPT} = \frac{2k-2}{k} = 2 - \frac{2}{k}$$

besser?

Der beste bekannte Approximationsfaktor für MehrwegeSchnitt ist $1{,}2965 - \frac{1}{k}$ [Sharma & Vondrák '14]

Analyse scharf?

K_k



$$\text{ALG} = (k-1)(k-1)$$

$$\text{OPT} = \sum_{i=1}^{k-1} i = \frac{k\cdot(k-1)}{2}$$

$$\text{ALG}/\text{OPT} = \frac{2k-2}{k} = 2 - \frac{2}{k}$$

besser?

Der beste bekannte Approximationsfaktor für MehrwegeSchnitt ist $1{,}2965 - \frac{1}{k}$ [Sharma & Vondrák '14]

MehrwegeSchnitt lässt sich nicht mit Faktor $1{,}20016 - O(1/n)$ approximieren (außer P=NP)

[Bérczi, Chandrasekaran, Király & Madan '18]