Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Approximationsalgorithmen

3. Vorlesung: Teil 1 – ShortestSuperString

– Folien von Joachim Spoerhase –

Alexander Wolff Lehrstuhl für Informatik I WS 2018/19

ShortestSuperString

Gegeben sei eine Menge $\{s_1, \ldots, s_n\} \subseteq \Sigma^+$ von Strings über einem endlichen Alphabet Σ. Gesucht ist ein **kürzester String** s, der jeden String s_i, $i = 1, \ldots, n$ als Teilstring enthält.

ShortestSuperString

Gegeben sei eine Menge $\{s_1, \ldots, s_n\} \subseteq \Sigma^+$ von Strings über einem endlichen Alphabet Σ. Gesucht ist ein **kürzester String** s, der jeden String s_i, $i = 1, \ldots, n$ als Teilstring enthält.

Bsp.: $U := \{cbaa, abc, bcb\}$

ShortestSuperString

Gegeben sei eine Menge $\{s_1, \ldots, s_n\} \subseteq \Sigma^+$ von Strings über einem endlichen Alphabet Σ. Gesucht ist ein **kürzester String** s, der jeden String s_i, $i = 1, \ldots, n$ als Teilstring enthält.

Bsp.: $U := \{cbaa, abc, bcb\}$

$$abcbaa$$

$$\begin{array}{l} abc \\ bcb \\ cbaa \end{array}$$

ShortestSuperString

Gegeben sei eine Menge $\{s_1, \ldots, s_n\} \subseteq \Sigma^+$ von Strings über einem endlichen Alphabet Σ. Gesucht ist ein **kürzester String** s, der jeden String s_i, $i = 1, \ldots, n$ als Teilstring enthält.

Bsp.: $U := \{cbaa, abc, bcb\}$

„überdeckt“ alle Strings aus U

$abcbaa$

$$\begin{array}{l} abc \\ bcb \\ cbaa \end{array}$$

ShortestSuperString

Gegeben sei eine Menge $\{s_1, \ldots, s_n\} \subseteq \Sigma^+$ von Strings über einem endlichen Alphabet Σ. Gesucht ist ein **kürzester String** s, der jeden String s_i, $i = 1, \ldots, n$ als Teilstring enthält.

Bsp.: $U := \{cbaa, abc, bcb\}$

„überdeckt“ alle Strings aus U

Ohne Einschränkung:
Kein String s_i ist Teilstring eines anderen Strings s_j

$abcbaa$

$$\begin{array}{l} abc \\ bcb \\ cbaa \end{array}$$

Beziehung SSS und SETCOVER

Lemma. $\mathrm{OPT}_{\mathrm{SC}} \leq 2 \cdot \mathrm{OPT}$

Beziehung SSS und SetCover

Lemma. $\mathsf{OPT}_{\mathsf{SC}} \leq 2 \cdot \mathsf{OPT}$

Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathsf{OPT}$.

Beziehung SSS und SetCover

Lemma. $\mathrm{OPT}_{\mathrm{SC}} \leq 2 \cdot \mathrm{OPT}$

Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathrm{OPT}$.

linkestes Auftreten eines Strings $s_{b_1} \in U$

Beziehung SSS und SetCover

Lemma. $\mathrm{OPT}_{\mathrm{SC}} \leq 2 \cdot \mathrm{OPT}$

Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathrm{OPT}$.

Beziehung SSS und SetCover

Lemma. $\mathsf{OPT}_{\mathsf{SC}} \leq 2 \cdot \mathsf{OPT}$

Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathsf{OPT}$.

Beziehung SSS und SetCover

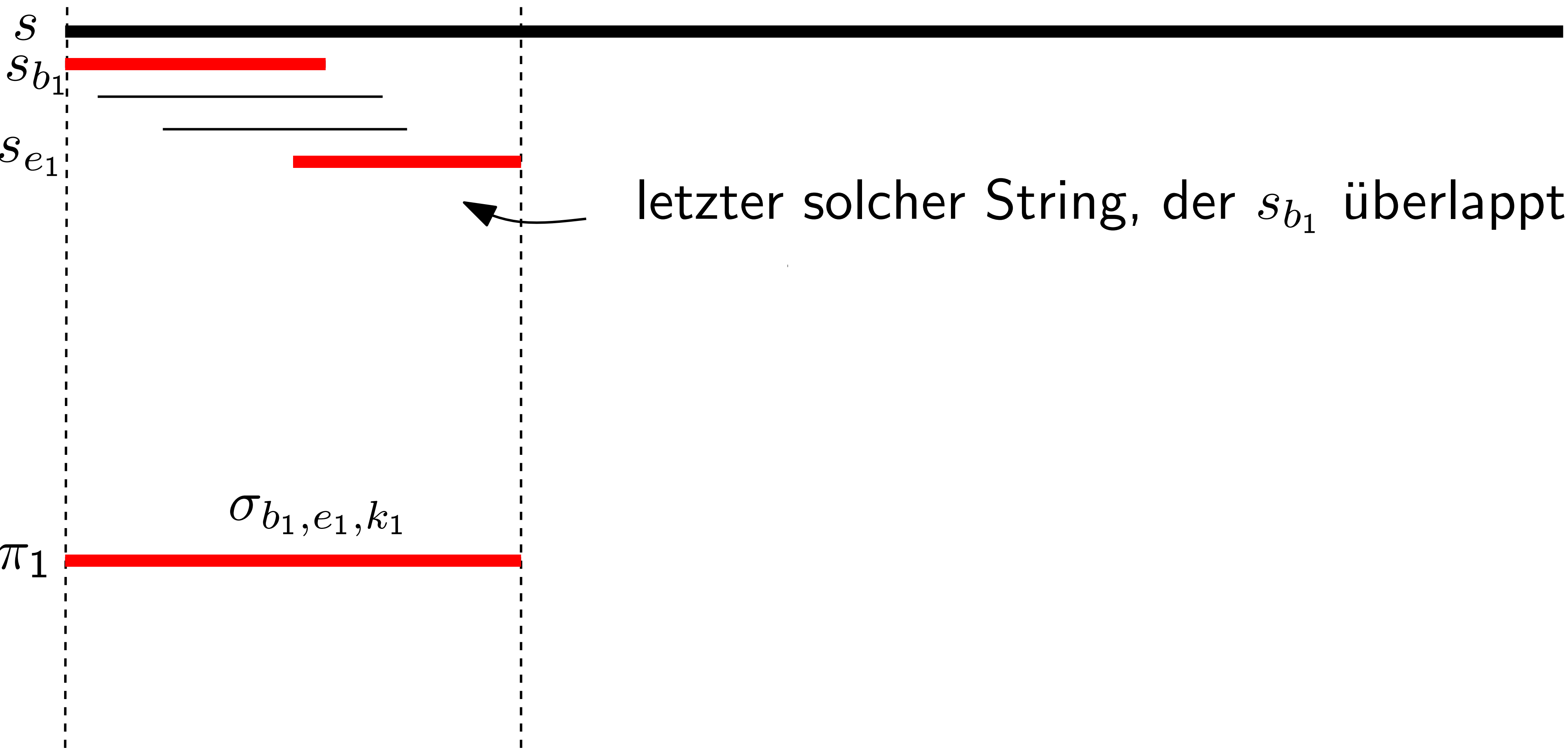
Lemma. $\mathrm{OPT}_{\mathrm{SC}} \leq 2 \cdot \mathrm{OPT}$

Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathrm{OPT}$.

Beziehung SSS und SetCover

Lemma. $\mathsf{OPT}_{\mathsf{SC}} \leq 2 \cdot \mathsf{OPT}$

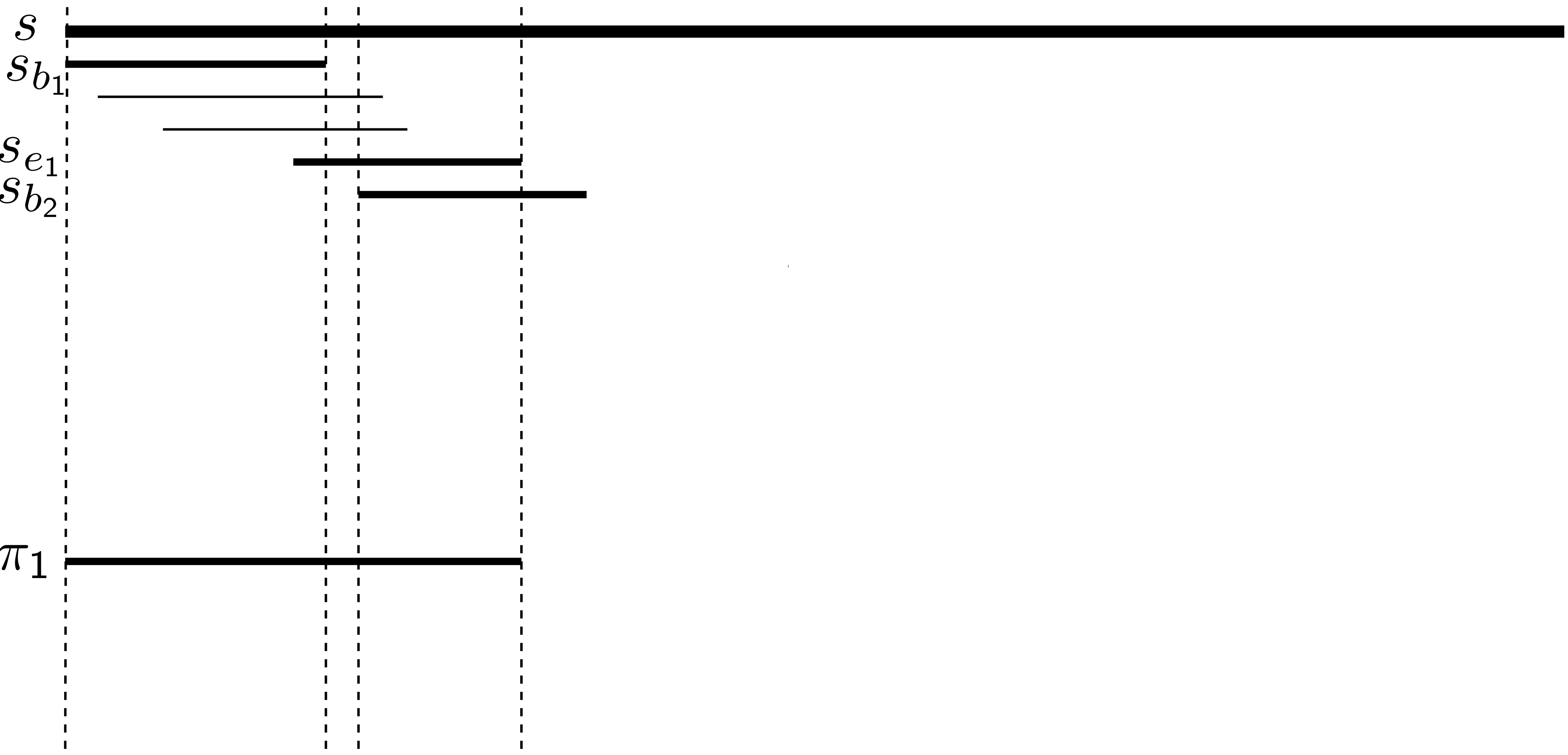
Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathsf{OPT}$.



Beziehung SSS und SetCover

Lemma. $\mathsf{OPT}_{\mathsf{SC}} \leq 2 \cdot \mathsf{OPT}$

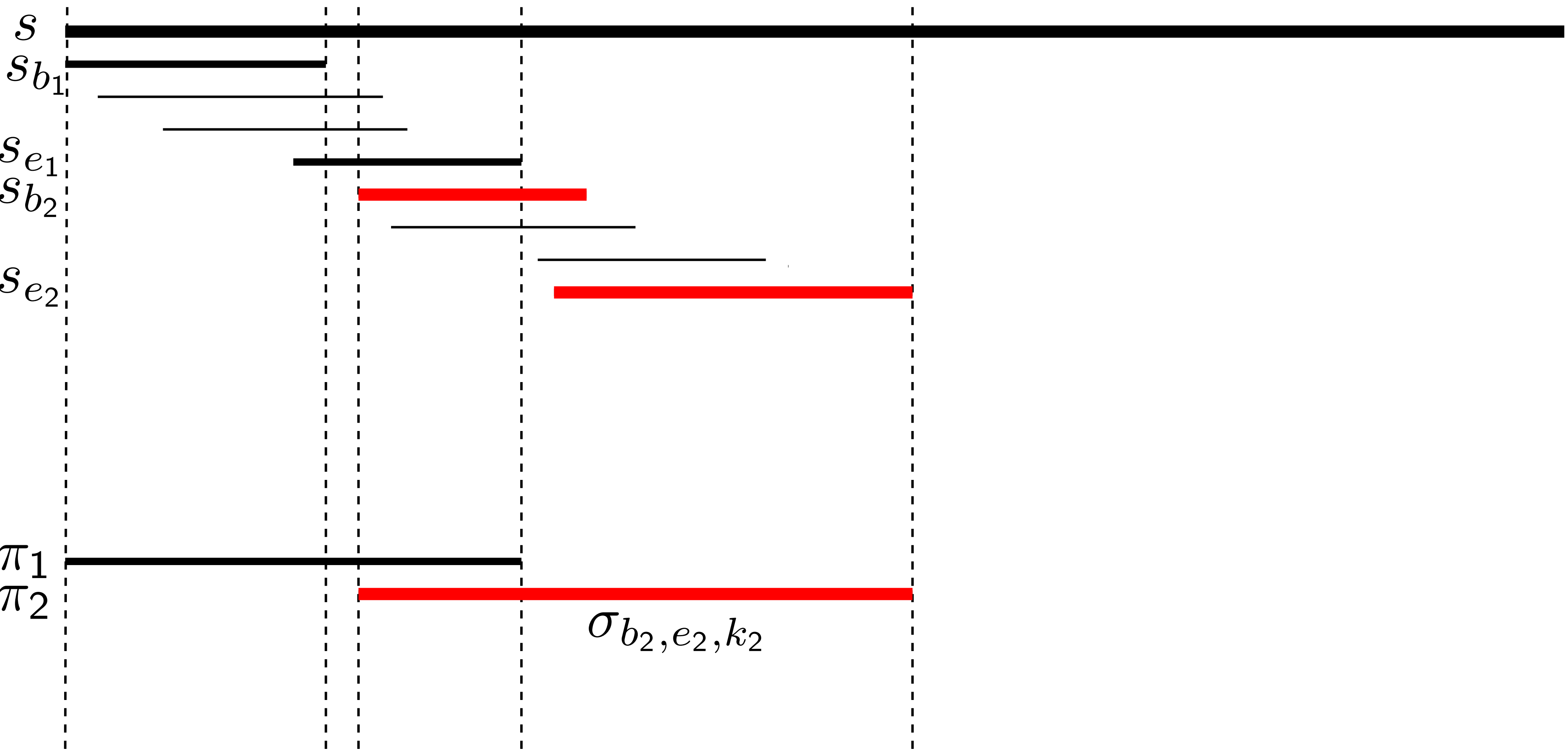
Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathsf{OPT}$.



Beziehung SSS und SetCover

Lemma. $\mathrm{OPT}_{\mathrm{SC}} \leq 2 \cdot \mathrm{OPT}$

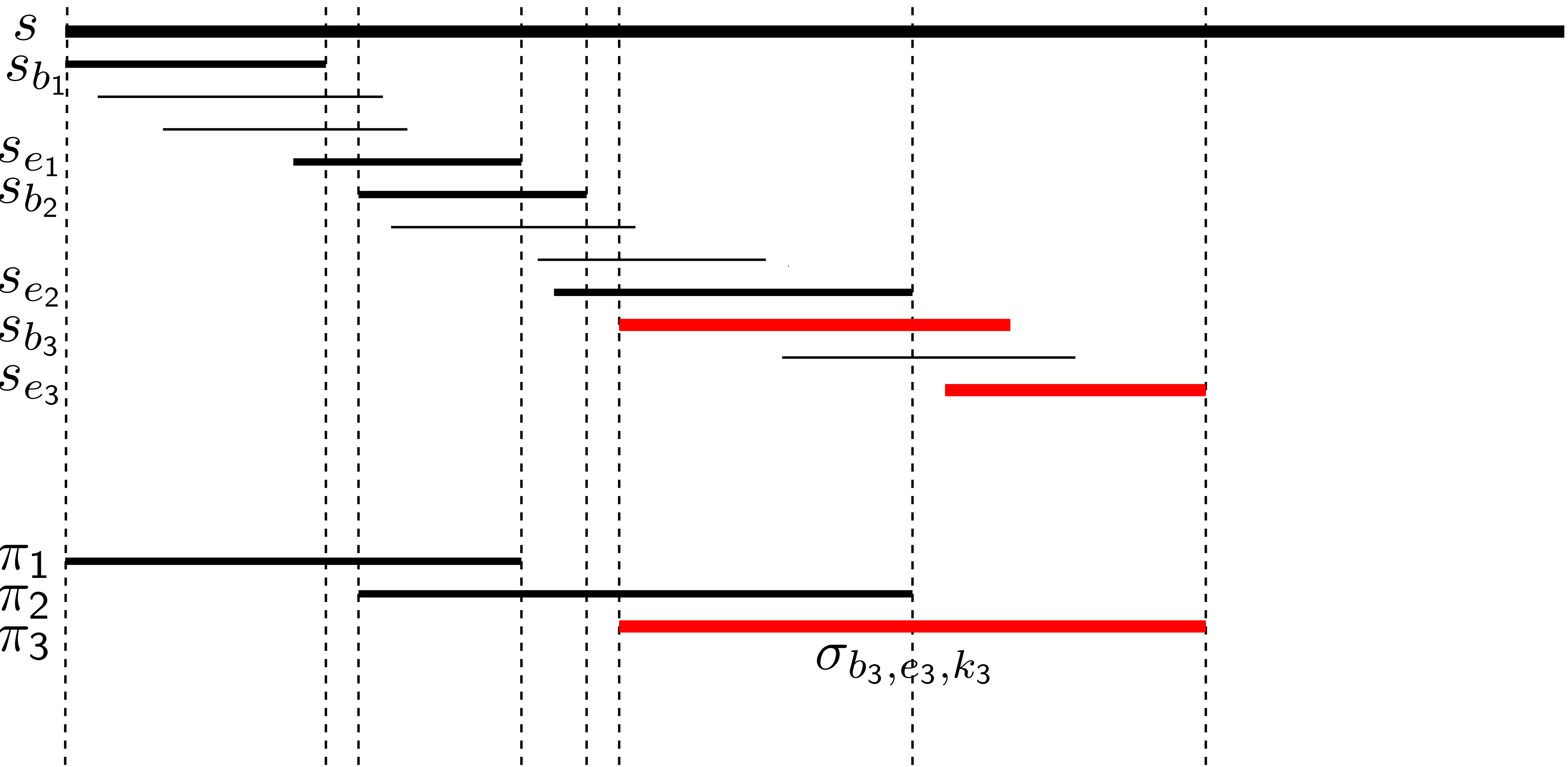
Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathrm{OPT}$.



Beziehung SSS und SetCover

Lemma. $\mathrm{OPT}_{\mathrm{SC}} \leq 2 \cdot \mathrm{OPT}$

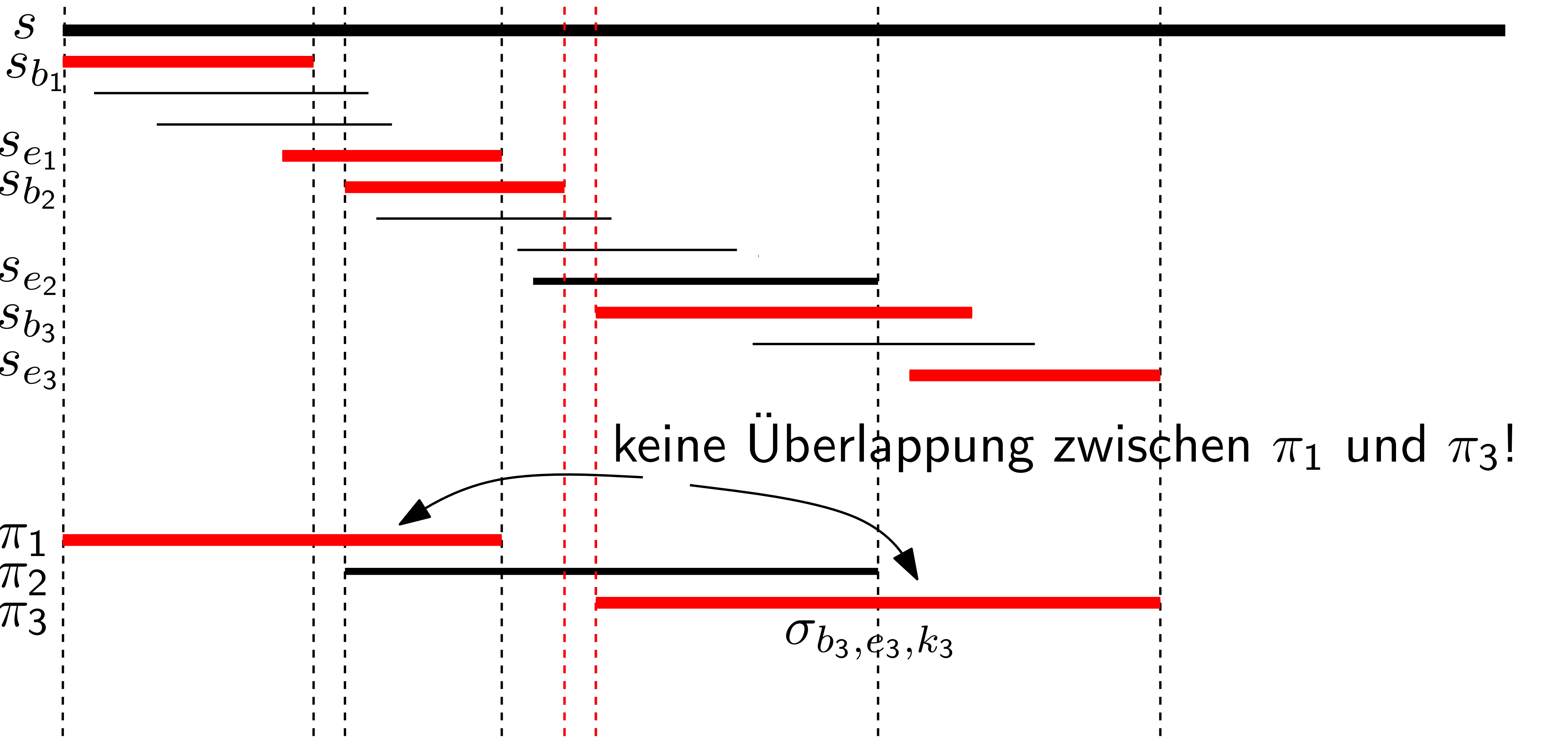
Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathrm{OPT}$.



Beziehung SSS und SetCover

Lemma. $\mathrm{OPT}_{\mathrm{SC}} \leq 2 \cdot \mathrm{OPT}$

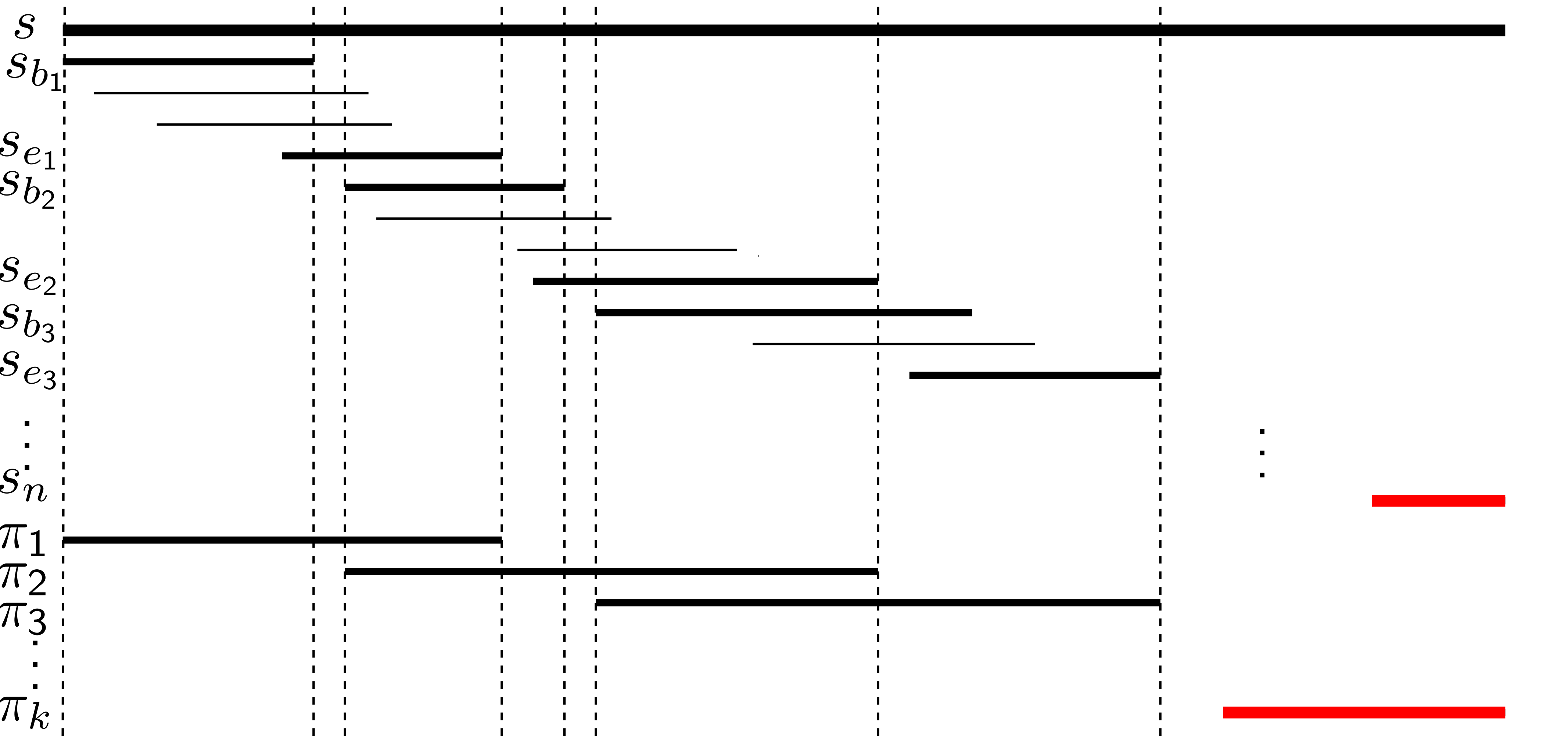
Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathrm{OPT}$.



Beziehung SSS und SetCover

Lemma. $\mathrm{OPT}_{\mathrm{SC}} \leq 2 \cdot \mathrm{OPT}$

Beweis.
Betrachte optimalen String s.
Konstruiere Überdeckung mit Kosten $\leq 2|s| = 2 \cdot \mathrm{OPT}$.



Beziehung SSS und SetCover

Lemma. $\mathrm{OPT}_{\mathrm{SC}} \leq 2 \cdot \mathrm{OPT}$

Beweis.

- Jeder String $s_i \in U$ ist in einem Teilstring π_j enthalten

Beziehung SSS und $\textsc{SetCover}$

Lemma. $\mathsf{OPT}_{\mathsf{SC}} \leq 2 \cdot \mathsf{OPT}$

Beweis.

- Jeder String $s_i \in U$ ist in einem Teilstring π_j enthalten
- $\mathsf{set}(\pi_1), \ldots, \mathsf{set}(\pi_k)$ ist Lösung für $\textsc{SetCover}$-Instanz mit Kosten $\sum_i |\pi_i|$

Beziehung SSS und SetCover

Lemma. $\mathsf{OPT}_{\mathsf{SC}} \leq 2 \cdot \mathsf{OPT}$

Beweis.

- Jeder String $s_i \in U$ ist in einem Teilstring π_j enthalten
- $\mathrm{set}(\pi_1), \ldots, \mathrm{set}(\pi_k)$ ist Lösung für SetCover-Instanz mit Kosten $\sum_i |\pi_i|$
- Teilstrings π_j, π_{j+2} sind überschneidungsfrei

Beziehung SSS und SetCover

Lemma. $\mathsf{OPT}_{\mathsf{SC}} \leq 2 \cdot \mathsf{OPT}$

Beweis.

- Jeder String $s_i \in U$ ist in einem Teilstring π_j enthalten
- $\mathsf{set}(\pi_1), \ldots, \mathsf{set}(\pi_k)$ ist Lösung für SetCover-Instanz mit Kosten $\sum_i |\pi_i|$
- Teilstrings π_j, π_{j+2} sind überschneidungsfrei
- Jedes Auftreten eines Zeichens liegt in höchstens **zwei** (aufeinanderfolgenden) Teilstrings π_j und π_{j+1}

Beziehung SSS und SetCover

Lemma. $\mathsf{OPT}_{\mathsf{SC}} \leq 2 \cdot \mathsf{OPT}$

Beweis.

- Jeder String $s_i \in U$ ist in einem Teilstring π_j enthalten
- $\mathsf{set}(\pi_1), \ldots, \mathsf{set}(\pi_k)$ ist Lösung für SetCover-Instanz mit Kosten $\sum_i |\pi_i|$
- Teilstrings π_j, π_{j+2} sind überschneidungsfrei
- Jedes Auftreten eines Zeichens liegt in höchstens **zwei** (aufeinanderfolgenden) Teilstrings π_j und π_{j+1}
- $\sum_i |\pi_i| \leq 2|s| = 2 \cdot \mathsf{OPT}$

Algorithmus für SSS

- Konstruiere Instanz U, $\mathcal{S}$, c für $\textsc{SetCover}$ wie beschrieben

- Sei $\mathsf{set}(\pi_1), \ldots, \mathsf{set}(\pi_k)$ Lösung, die vom Greedy-Algorithmus für $\textsc{SetCover}$ zurückgegeben wird

- Gib $\pi_1 \circ \cdots \circ \pi_k$ als Lösung zurück

Approximations für SSS

Satz. Obiger Approximationsalgorithmus für ShortestSuperstring hat Approximationsgüte $2\mathcal{H}_n$.

Approximations für SSS

Satz. Obiger Approximationsalgorithmus für SHORTESTSUPERSTRING hat Approximationsgüte $2\mathcal{H}_n$.

Beweis.

Die Kosten der Lösung sind beschränkt durch

$$\mathcal{H}_n \cdot \mathsf{OPT}_{\mathsf{SC}} \leq 2 \cdot \mathcal{H}_n \cdot \mathsf{OPT}$$

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Approximationsalgorithmen

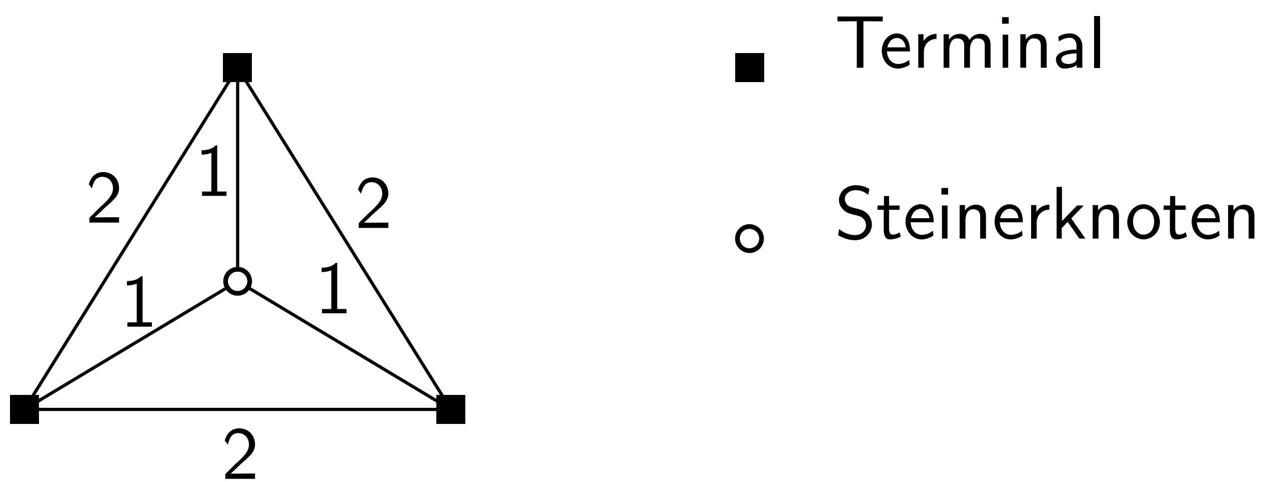
3. Vorlesung: Teil 2 –
SteinerTree & approximationserhaltende Reduktion

– Folien von Joachim Spoerhase –

Alexander Wolff Lehrstuhl für Informatik I WS 2018/19

SteinerTree

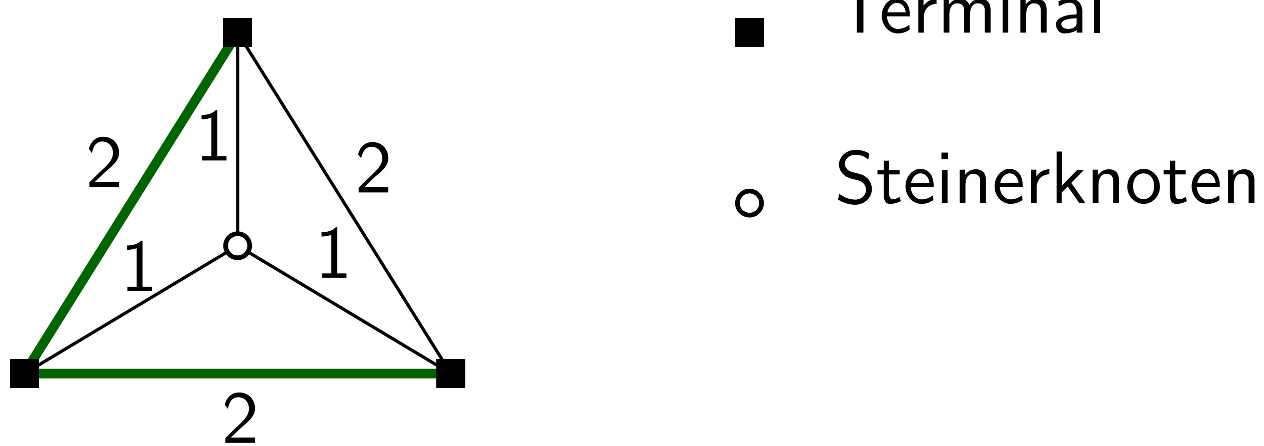
Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{Q}^+$ sowie eine Partitionierung von V in eine Menge T von **Terminalen** und eine Menge S von **Steinerknoten**.
Gesucht ist ein Teilbaum $B = (V', E')$ von G, der alle Terminale enthält, d.h. $T \subseteq V'$, und der minimale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



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.

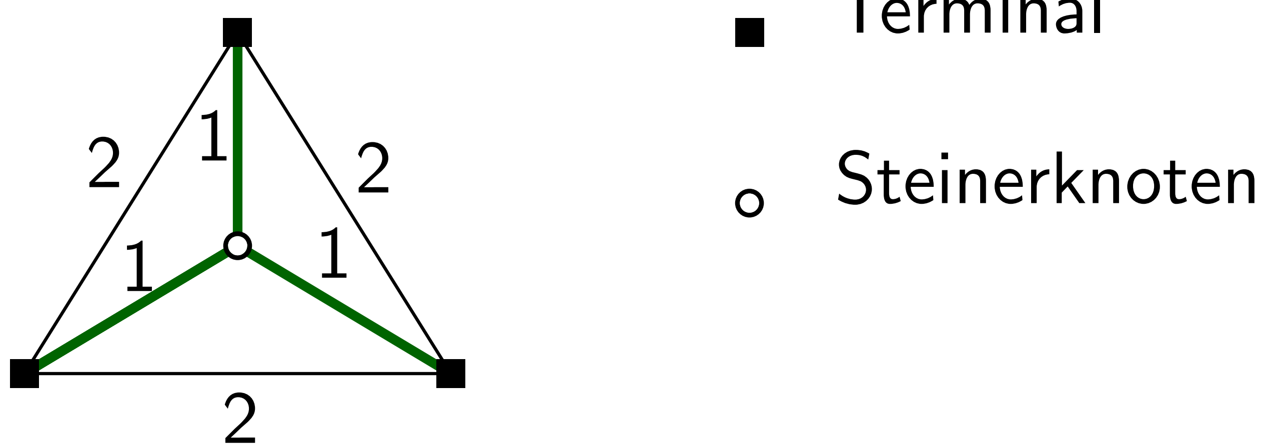
zul. Lösung mit Kosten 4



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.

optimale Lösung mit Kosten 3

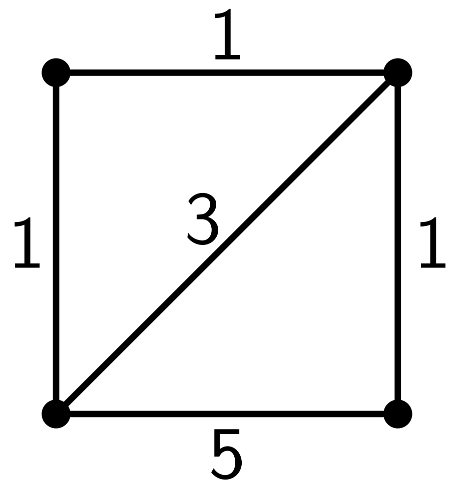


Metrisches SteinerTree

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

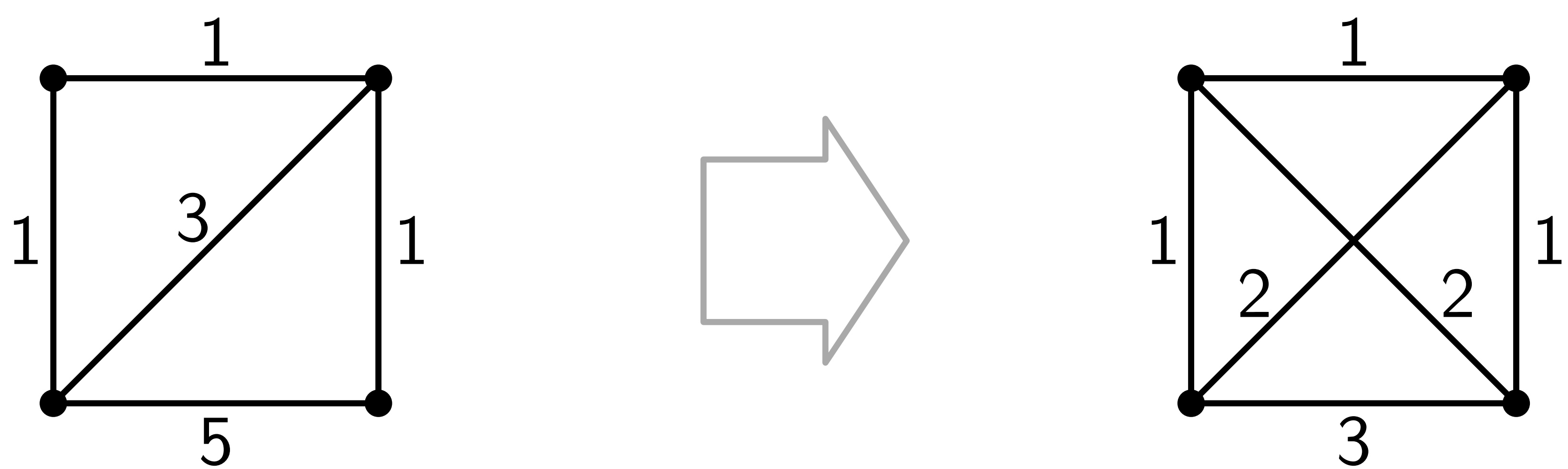
Einschränkung von SteinerTree, bei der der Graph G vollständig und metrisch sein muss, d.h. für jedes Tripel u, v, w von Knoten gilt $c(u, w) \leq c(u, v) + c(v, w)$.



nicht 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 $c(u, w) \leq c(u, v) + c(v, w)$.

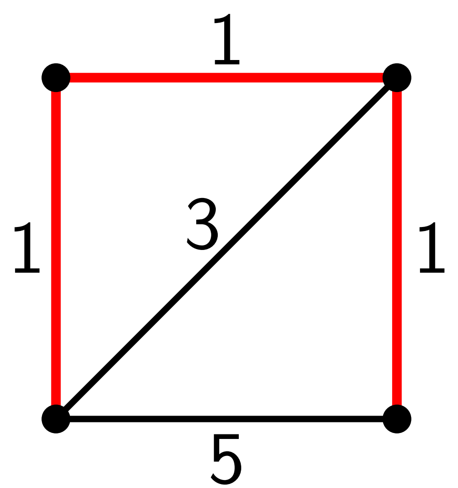


nicht metrisch

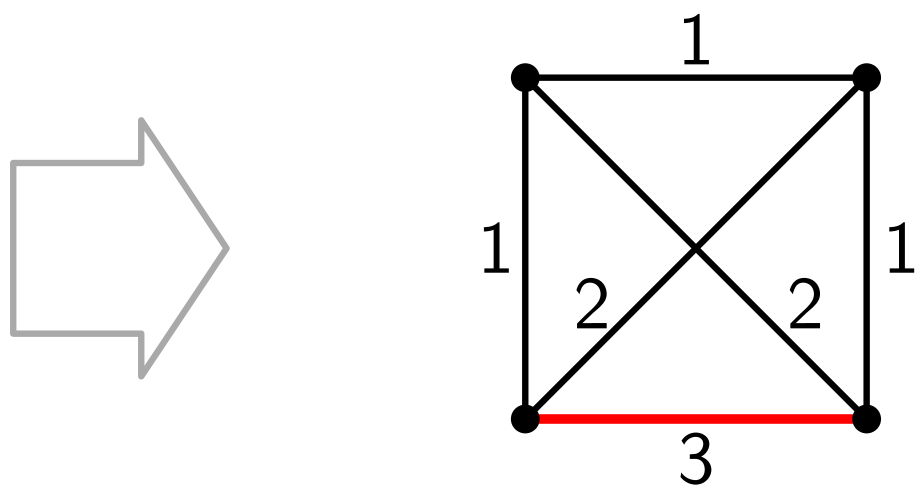
metrisch

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 metrisch



metrisch

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 $\mathsf{OPT}_{\Pi_2}(I_2) \leq \mathsf{OPT}_{\Pi_1}(I_1)$.

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

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

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion 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.

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

Approximationserhaltende Reduktion

Satz. Seien Π_1, Π_2 Minimierungsprobleme, für die es eine approximationserhaltende Reduktion 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 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 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 $\textsc{SteinerTree}$

Satz. Es gibt eine approximationserhaltende Reduktion von $\textsc{SteinerTree}$ auf metrisches $\textsc{SteinerTree}$.

Beweis. „Abbildung f“

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

Metrisches SteinerTree

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

Beweis. „Abbildung f"

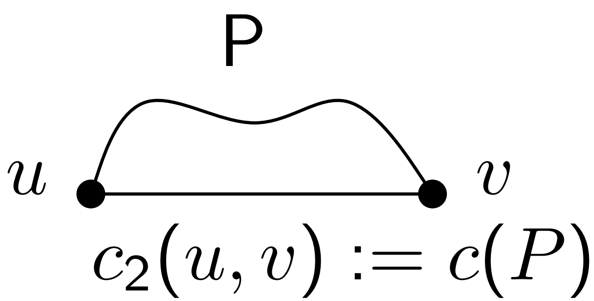
- Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = S \mathbin{\dot\cup} T$
- Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch S, T wie bei I_1

Metrisches SteinerTree

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

Beweis. „Abbildung f“

- Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = S \mathbin{\dot\cup} T$
- Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch S, T wie bei I_1
- $c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1, für alle $u, v \in V$

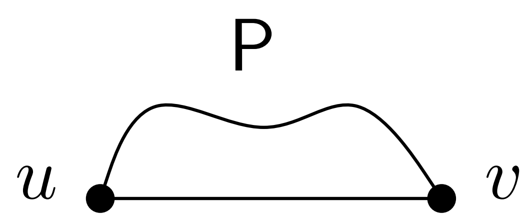


Metrisches SteinerTree

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

Beweis. „Abbildung f“

- Instanz I_1 von SteinerTree: Graph $G_1 = (V, E_1)$, Kantengewichte c_1, Partitionierung $V = S \,\dot{\cup}\, T$
- Metrische Instanz $I_2 := f(I_1)$: vollständiger Graph $G_2 = (V, E_2)$, jedoch S, T wie bei I_1
- $c_2(u, v) :=$ Länge eines kürzesten u–v-Weges in G_1, für alle $u, v \in V$
- $c_2(u, v) \leq c_1(u, v)$ für alle $(u, v) \in E$



$c_2(u, v) := c(P)$

Metrisches SteinerTree

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

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

- Sei B^* optimaler SB für I_1

Metrisches SteinerTree

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

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

- Sei B^* optimaler SB für I_1
- B^* ist auch zulässige Lösung für I_2, da $E_1 \subseteq E_2$ und Knotenmengen V, S, T gleich

Metrisches SteinerTree

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

Beweis. „$\text{OPT}(I_2) \leq \text{OPT}(I_1)$ "

- Sei B^* optimaler SB für I_1
- B^* ist auch zulässige Lösung für I_2, da $E_1 \subseteq E_2$ und Knotenmengen V, S, T gleich
- $\text{OPT}(I_2) \leq c_2(B^*) \leq c_1(B^*) = \text{OPT}(I_1)$

Metrisches SteinerTree

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

Beweis. „Abbildung g“

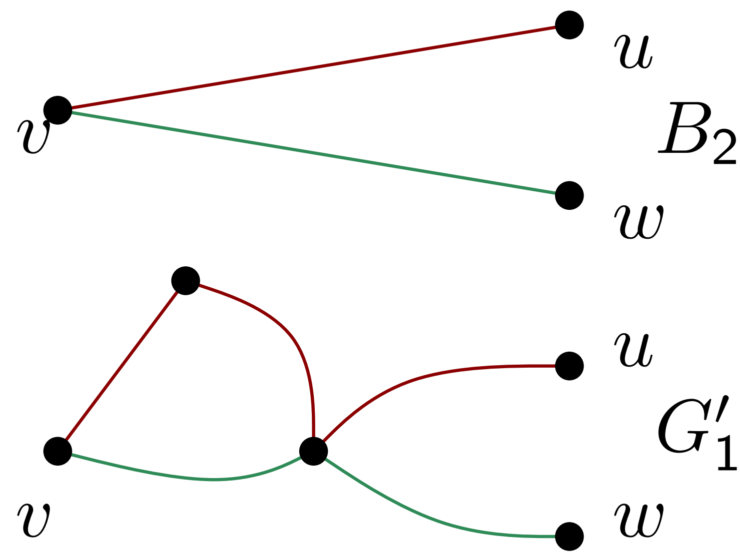
- Sei B_2 Steinerbaum für G_2

Metrisches SteinerTree

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

Beweis. „Abbildung g“

- 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

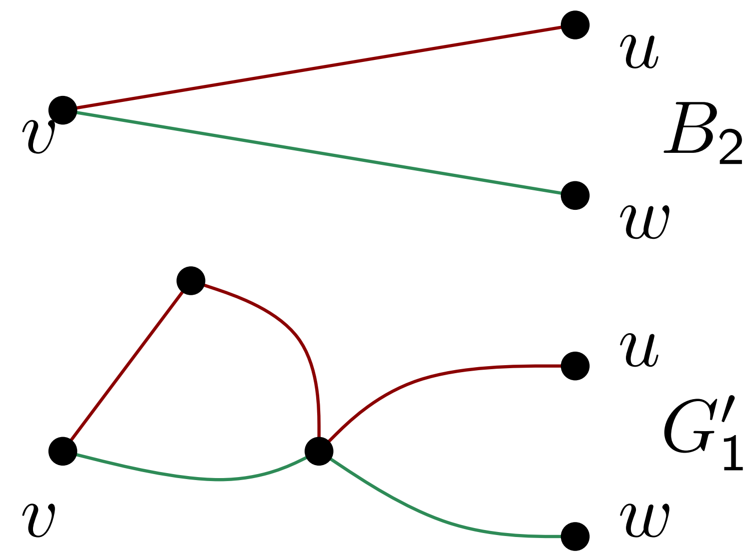


Metrisches SteinerTree

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

Beweis. „Abbildung g“

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

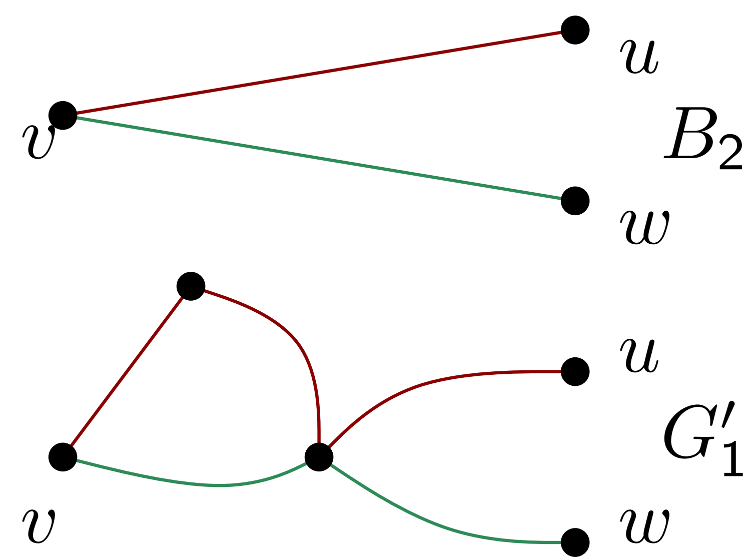


Metrisches SteinerTree

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

Beweis. „Abbildung g“

- 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

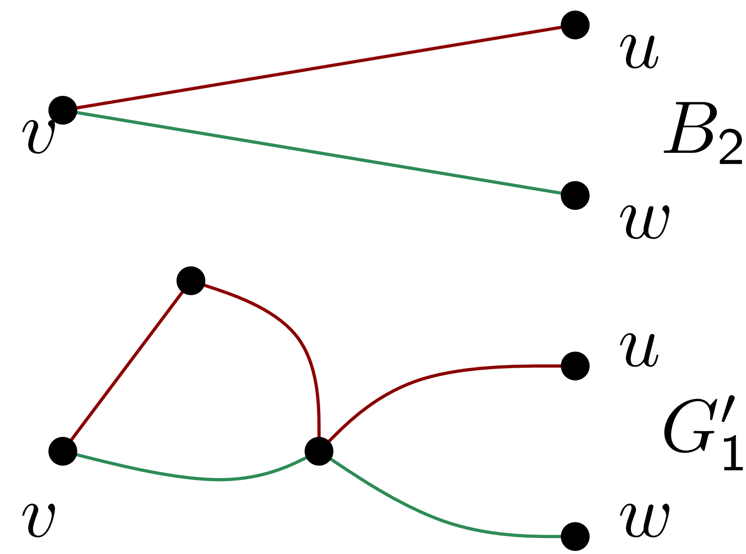


Metrisches SteinerTree

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

Beweis. „Abbildung g"

- 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
- G_1' nicht immer Baum

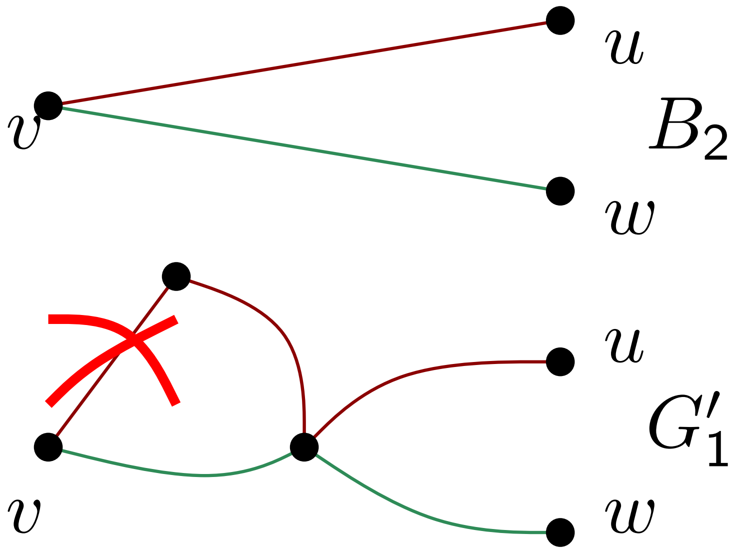


Metrisches $\textsc{SteinerTree}$

Satz. Es gibt eine approximationserhaltende Reduktion von $\textsc{SteinerTree}$ auf metrisches $\textsc{SteinerTree}$.

Beweis. „Abbildung g“

- 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
- G_1' nicht immer Baum
- Betrachte Spannbaum B_1 von G_1'

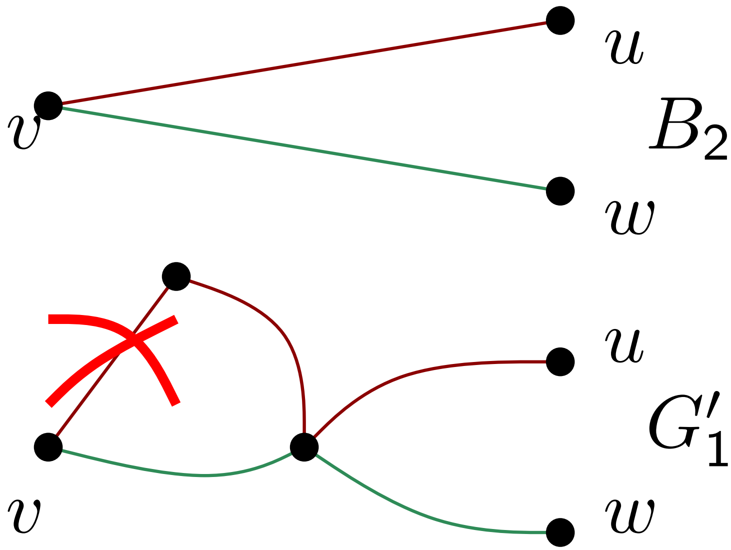


Metrisches SteinerTree

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

Beweis. „Abbildung g“

- 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
- G_1' nicht immer Baum
- Betrachte Spannbaum B_1 von G_1'
- $\rightsquigarrow$ Steinerbaum B_1 für G_1

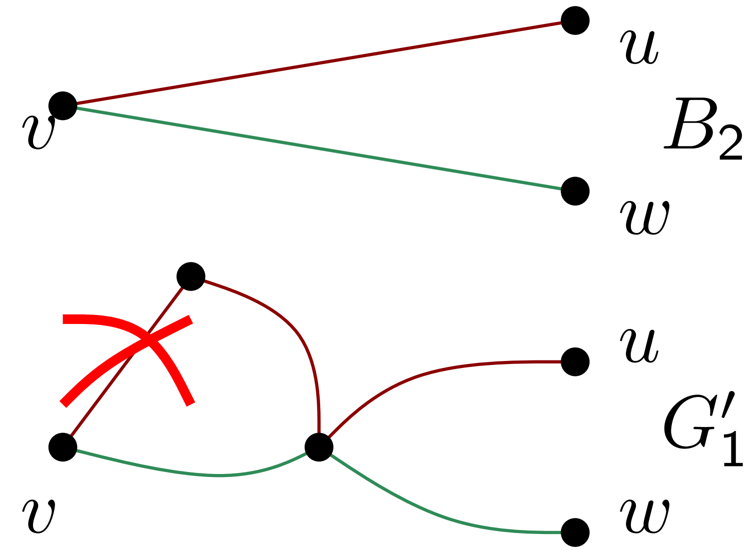


Metrisches SteinerTree

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

Beweis. „Abbildung g“

- 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
- G_1' nicht immer Baum
- Betrachte Spannbaum B_1 von G_1'
- $\rightsquigarrow$ Steinerbaum B_1 für G_1
- $c_1(B_1) \leq c_1(G_1') \leq c_2(B_2)$

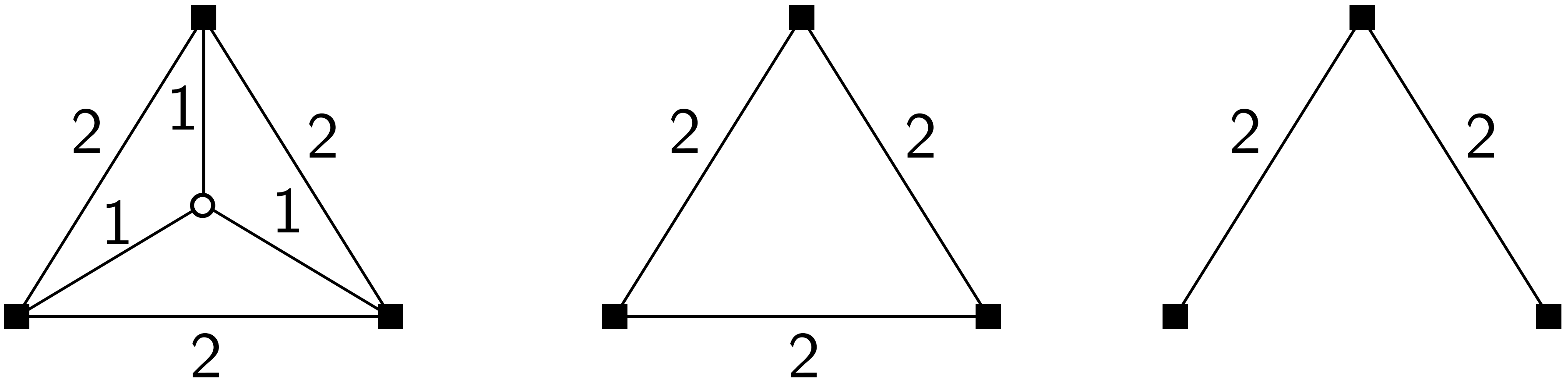


2-Approximation für SteinerTree

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

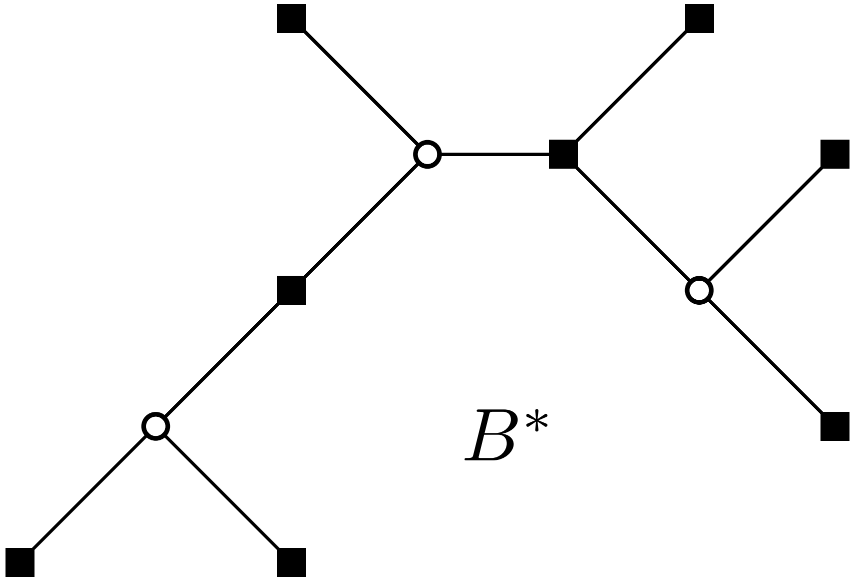
Satz. Gegeben sei eine Instanz für metrisches SteinerTree. Sei B ein minimaler Spannbaum (MSB) für den durch die Terminalmenge T induzierten Subgraphen $G[T]$.
Dann gilt $c(B) \leq 2 \cdot \mathrm{OPT}$.



G $\qquad$ $G[T]$ $\qquad$ B

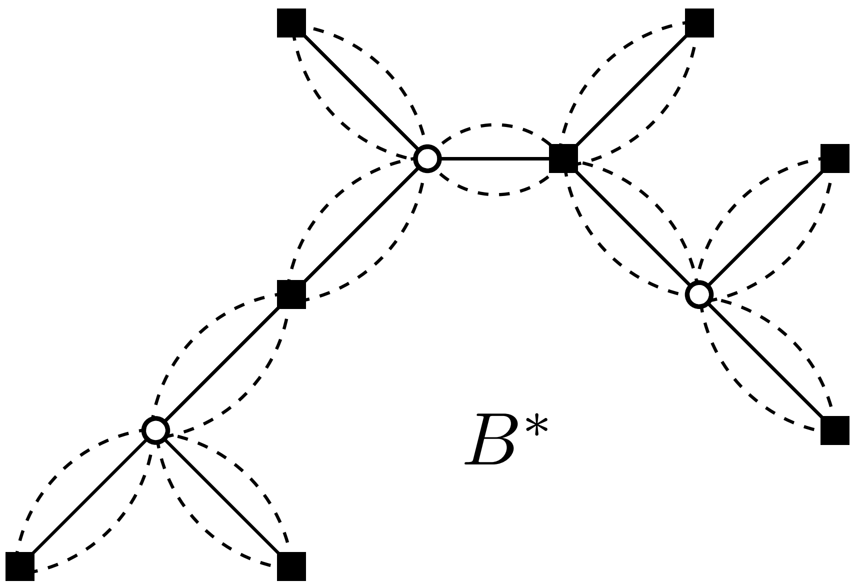
Beweis

- Betrachte optimalen Steinerbaum B^*



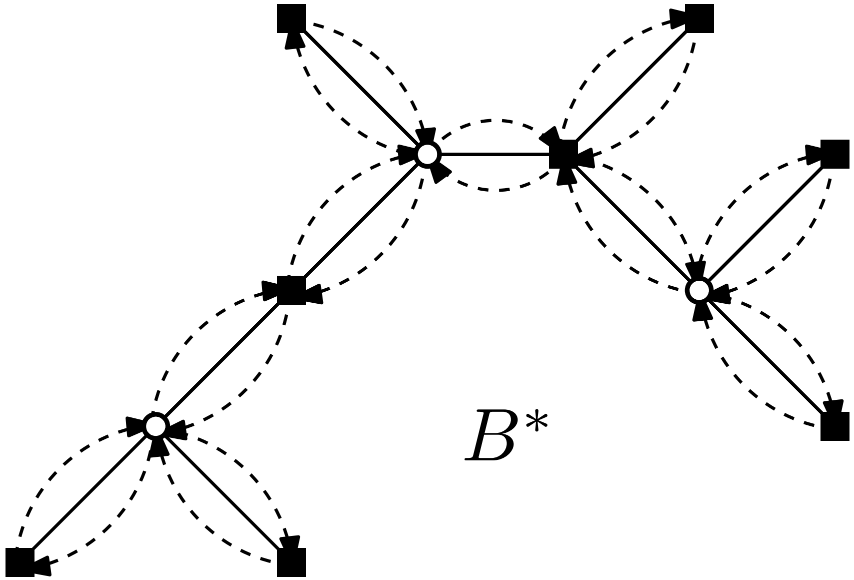
Beweis

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



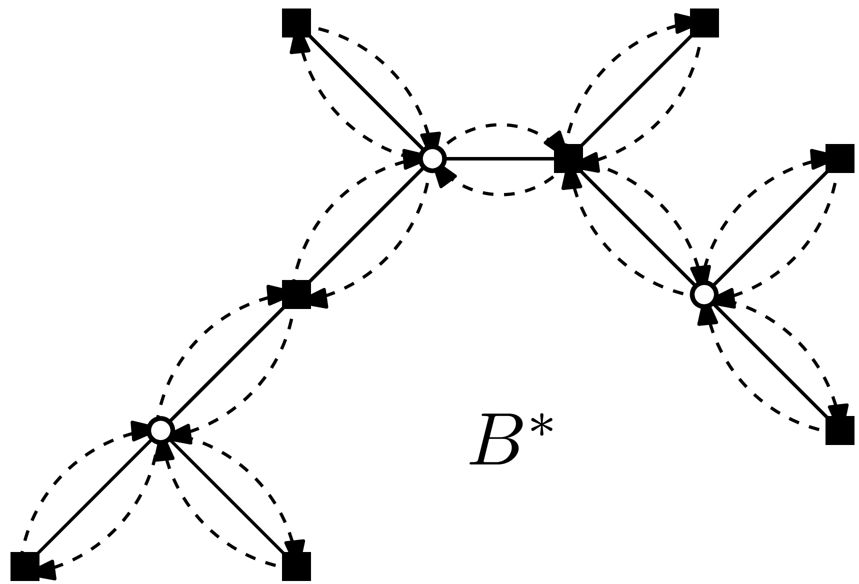
Beweis

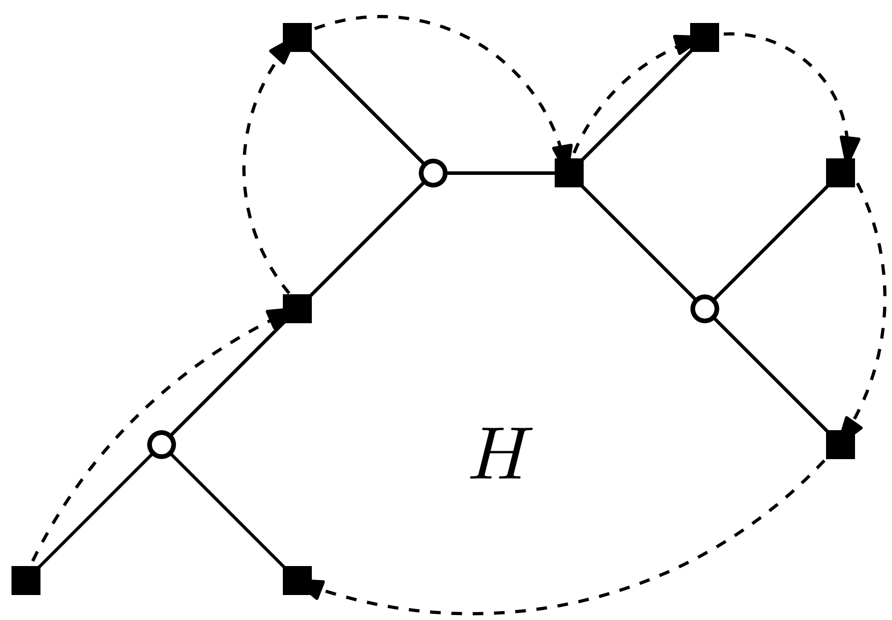
- Betrachte optimalen Steinerbaum B^*
- Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$
- Finde eine Eulertour T' in B' $\rightsquigarrow$ $c(T') = c(B') = 2 \cdot \mathrm{OPT}$



Beweis

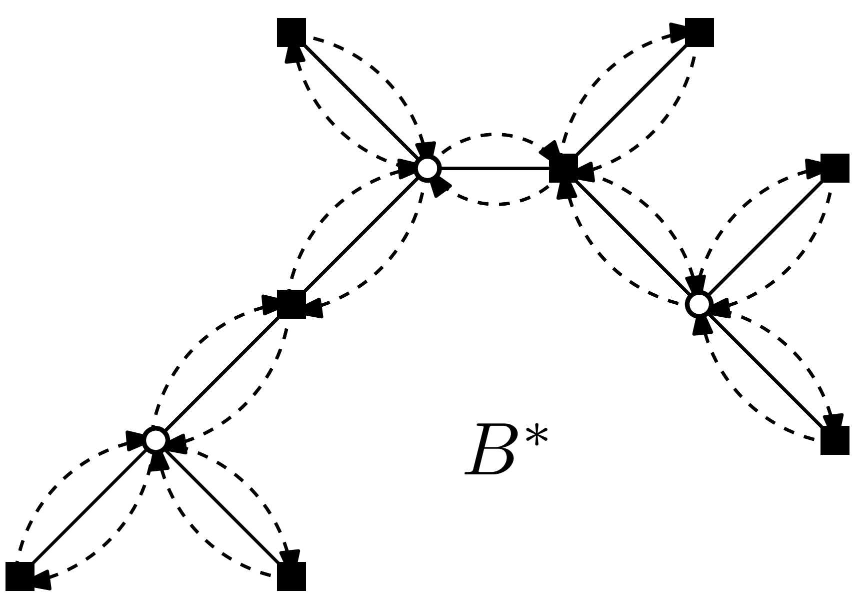
- Betrachte optimalen Steinerbaum B^*
- Verdopple alle Kanten aus B^* $\rightsquigarrow$ eulerscher (Multi-)Graph B' mit Kosten $c(B') = 2 \cdot \mathrm{OPT}$
- Finde eine Eulertour T' in B' $\rightsquigarrow$ $c(T') = c(B') = 2 \cdot \mathrm{OPT}$
- Hamiltonpfad H für $G[T]$ durch „Abkürzen“ von Steinerknoten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

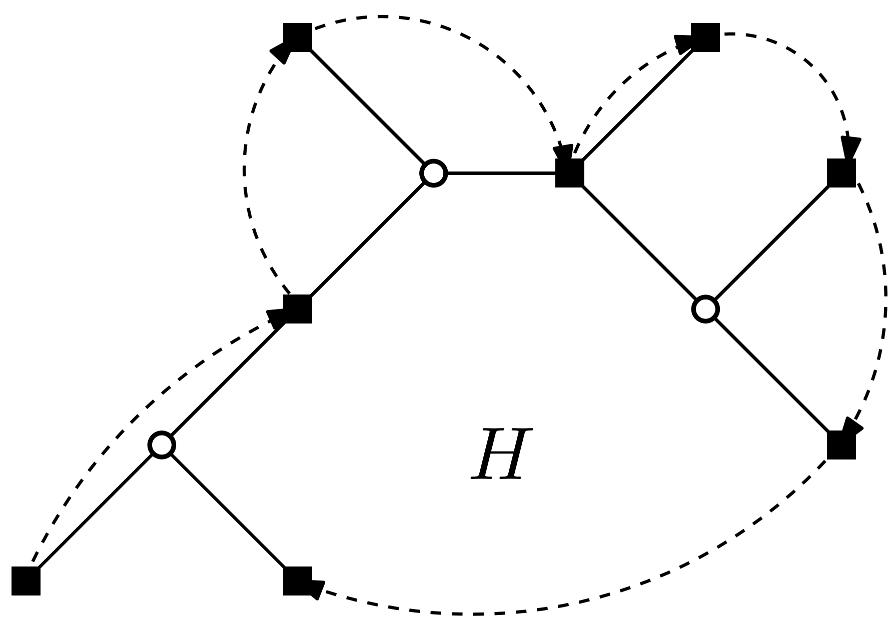




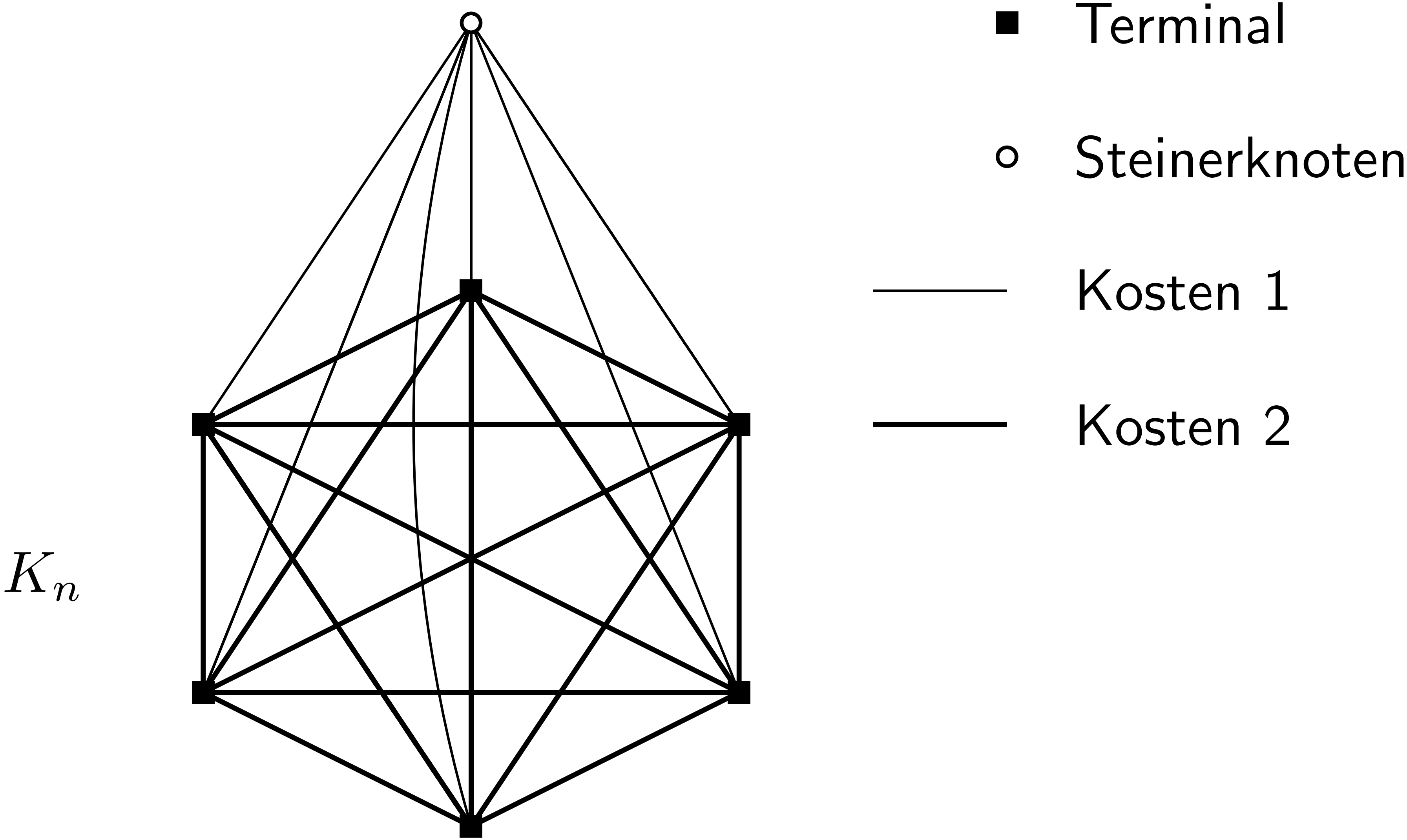
Beweis

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



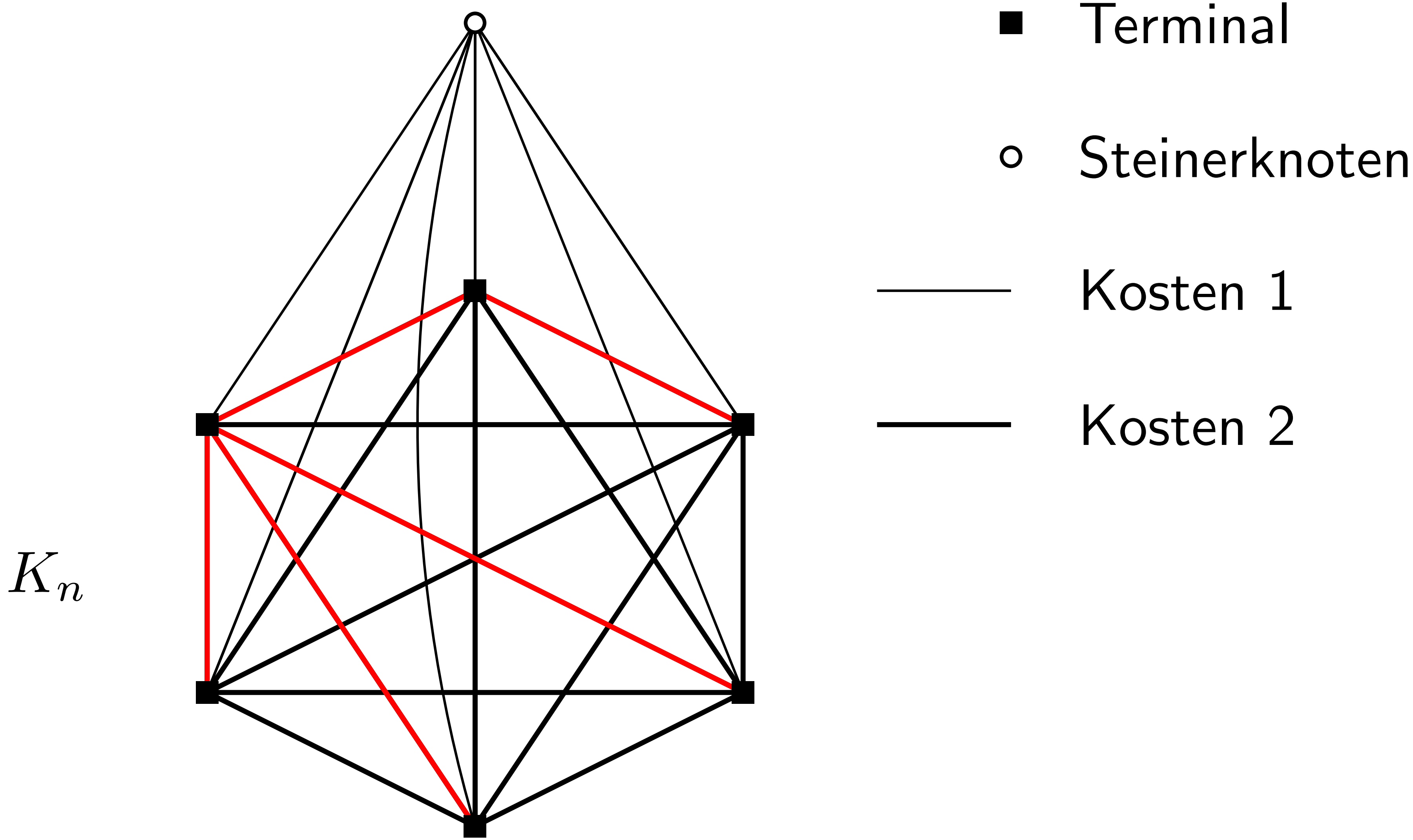


Scharfes Beispiel



Scharfes Beispiel

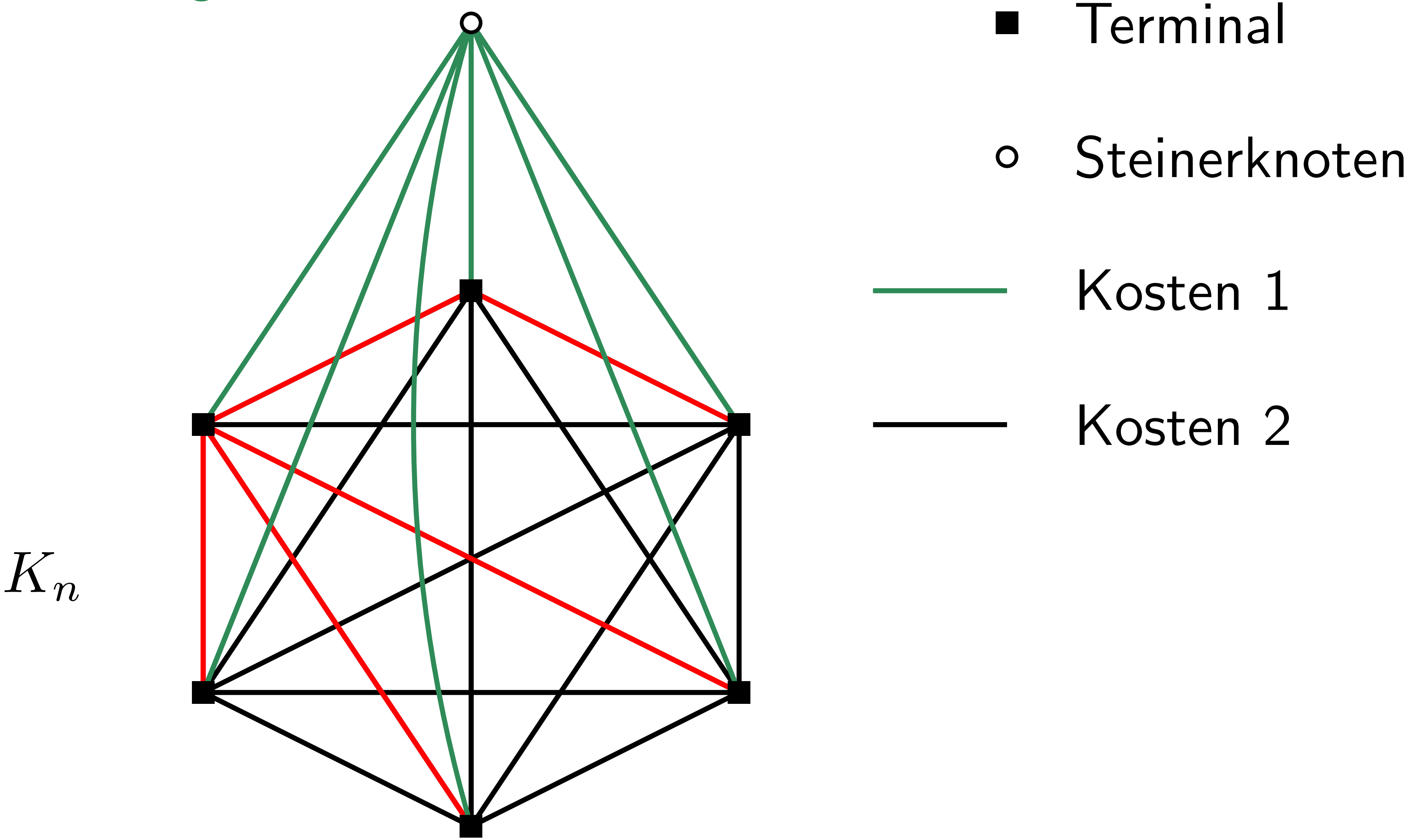
MST von $G[T]$ mit Kosten $2(n-1)$



Scharfes Beispiel

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

Optimale Lösung mit Kosten n



Scharfes Beispiel

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

Optimale Lösung mit Kosten n

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

K_n