Approximationsalgorithmen

7. Vorlesung: Metrisches k-Zentrum

– Basierend auf Folien von Joachim Spoerhase –

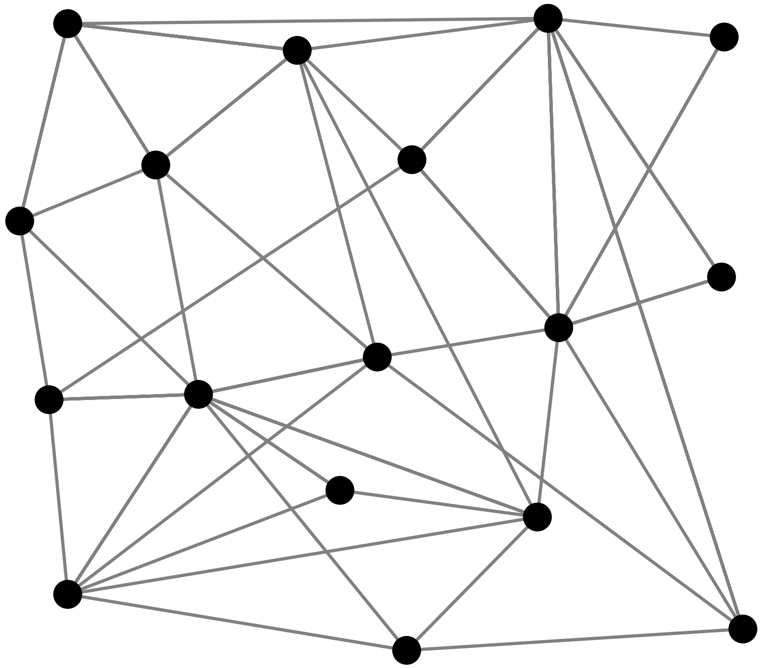
Philipp Kindermann Lehrstuhl für Informatik I WS 2018/19

Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$

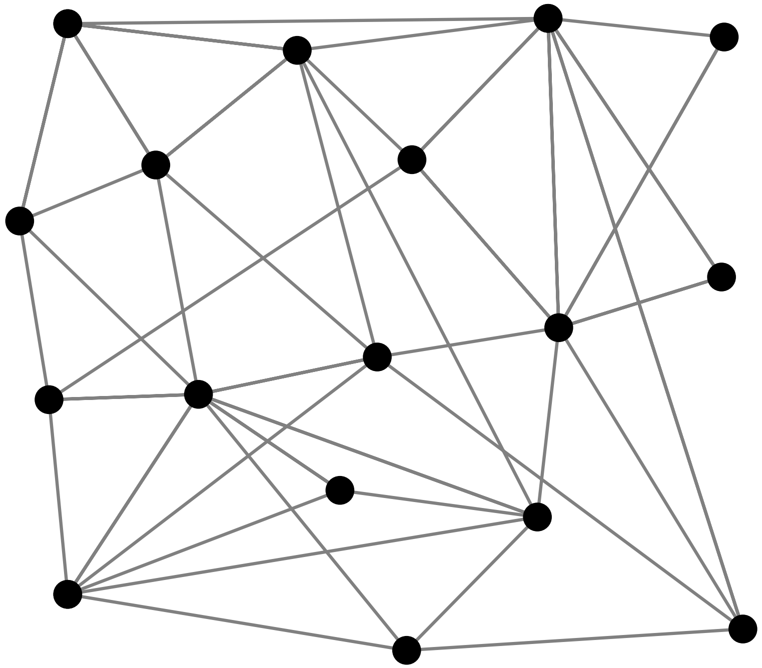
Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$



Das metrische k-ZENTRUM-Problem

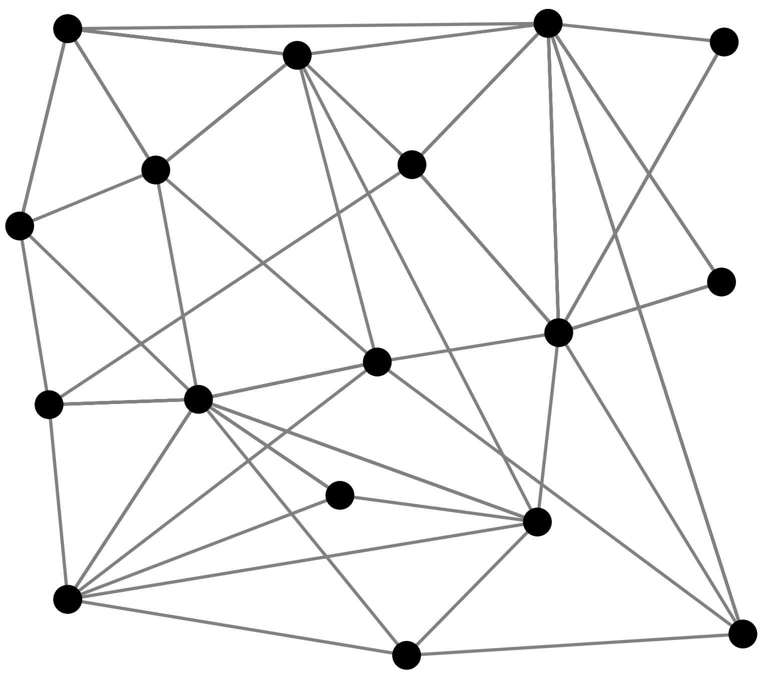
Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c \colon E \to \mathbb{Q}_{\geq 0}$



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c \colon E \to \mathbb{Q}_{\geq 0}$

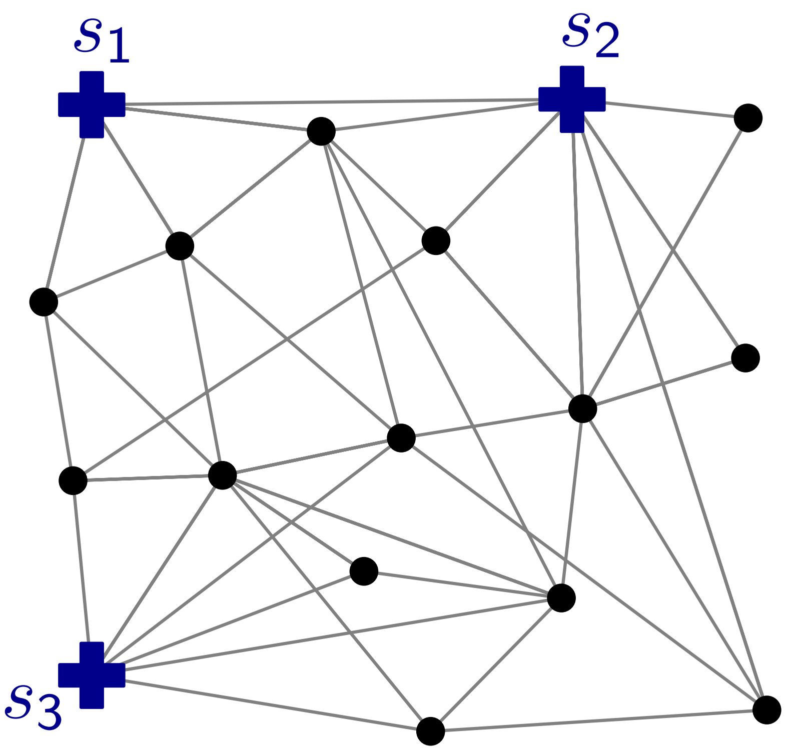
Knotenmenge $S \subseteq V$



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

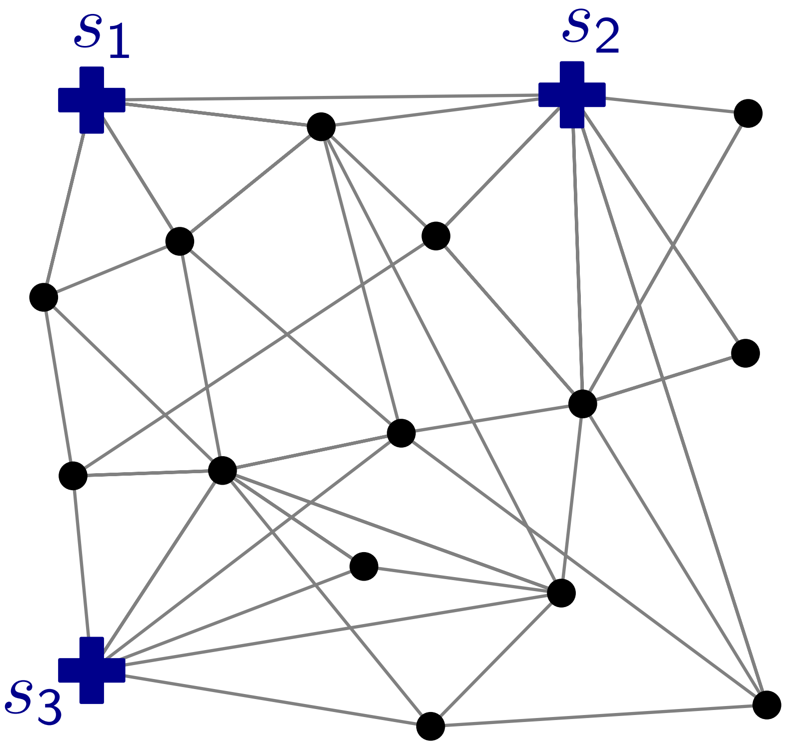
Knotenmenge $S \subseteq V$



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

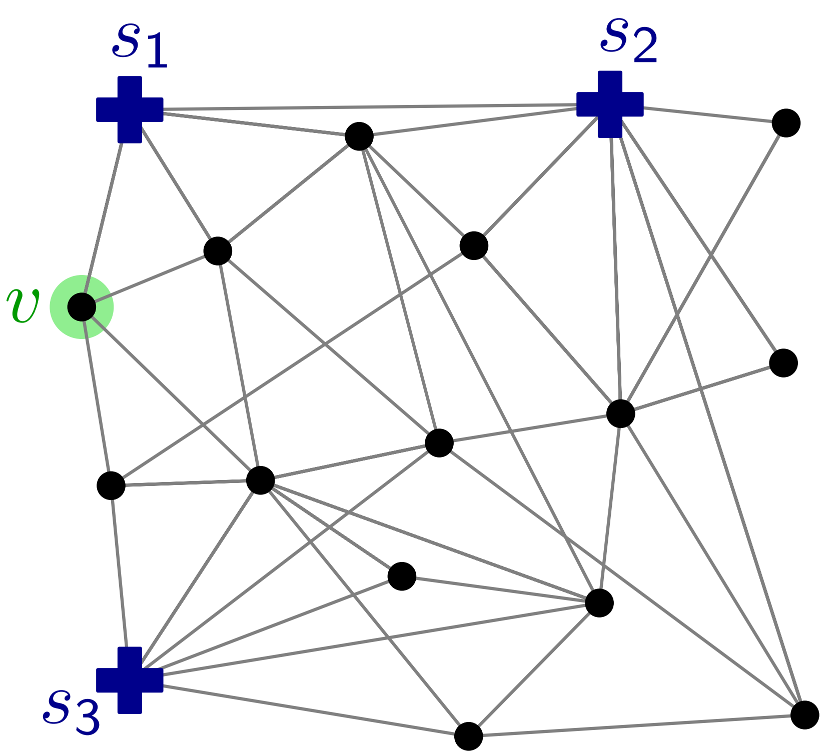
Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

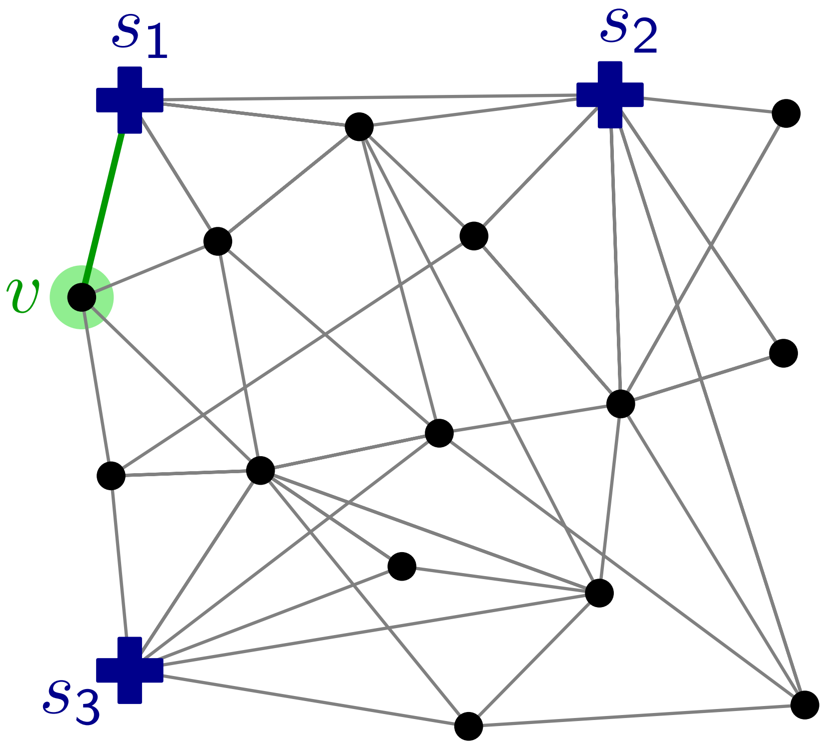
Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.



Das metrische k-ZENTRUM-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c \colon E \to \mathbb{Q}_{\geq 0}$

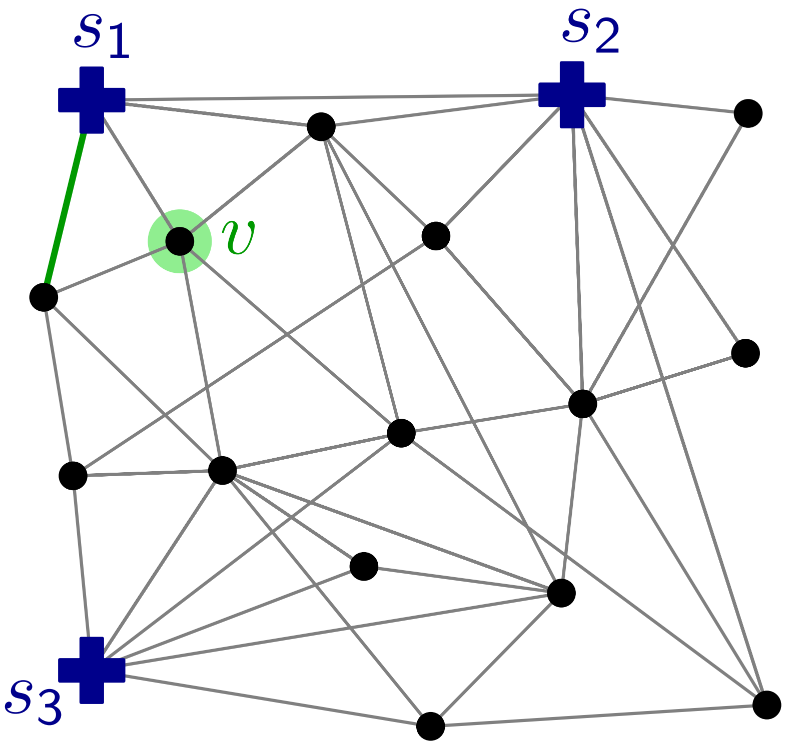
Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

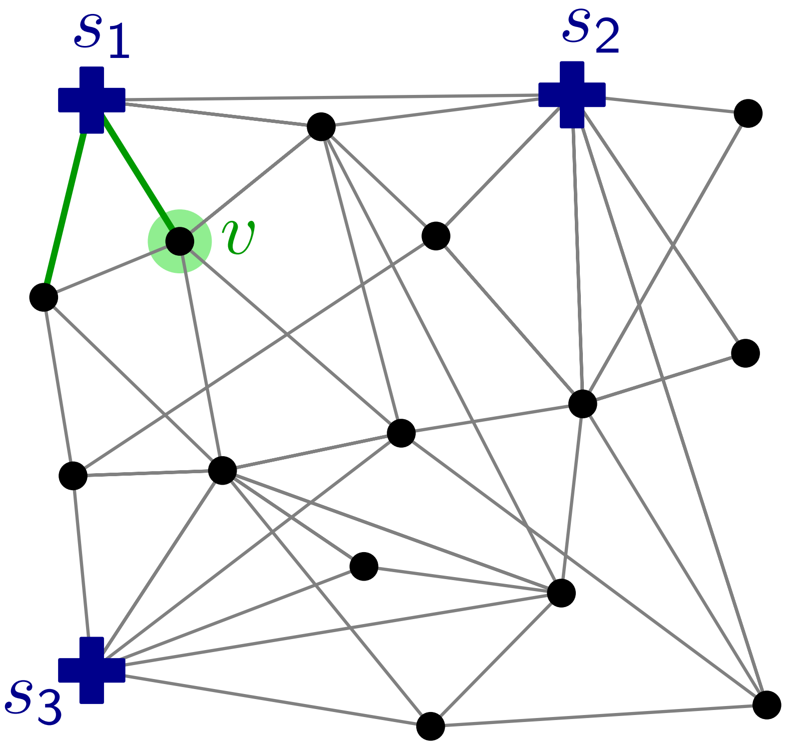
Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.



Das metrische k-ZENTRUM-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

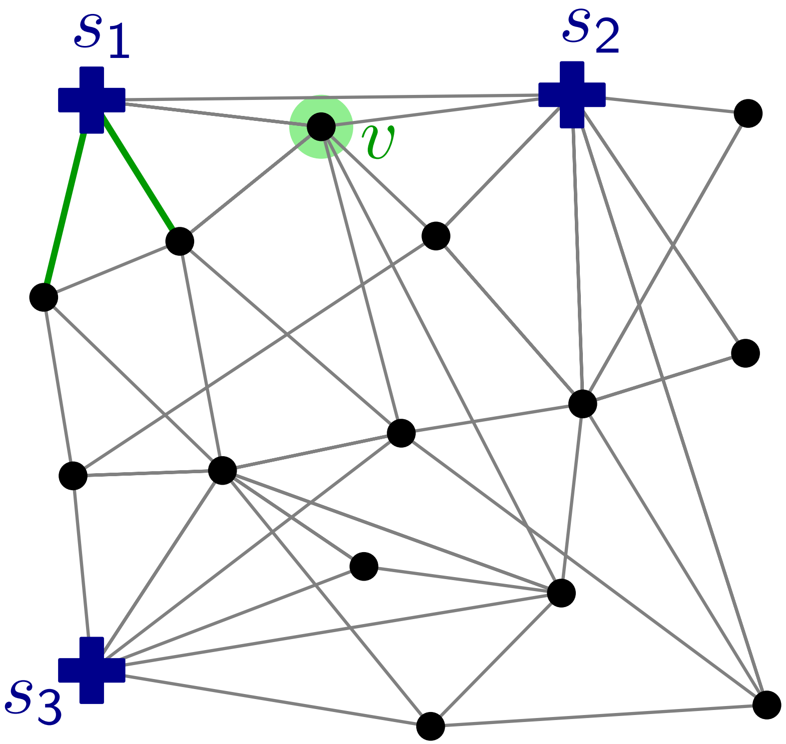
Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c \colon E \to \mathbb{Q}_{\geq 0}$

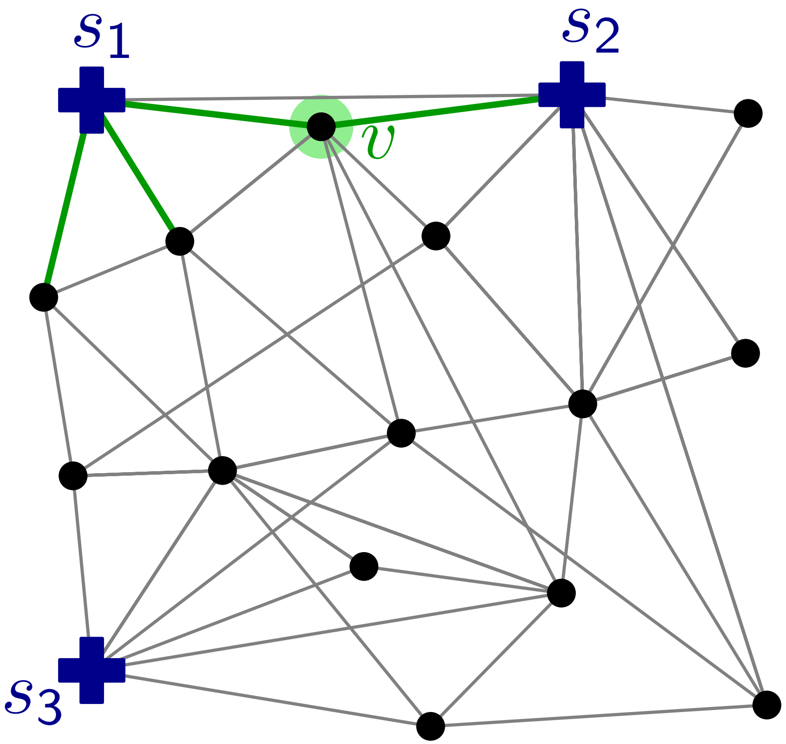
Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

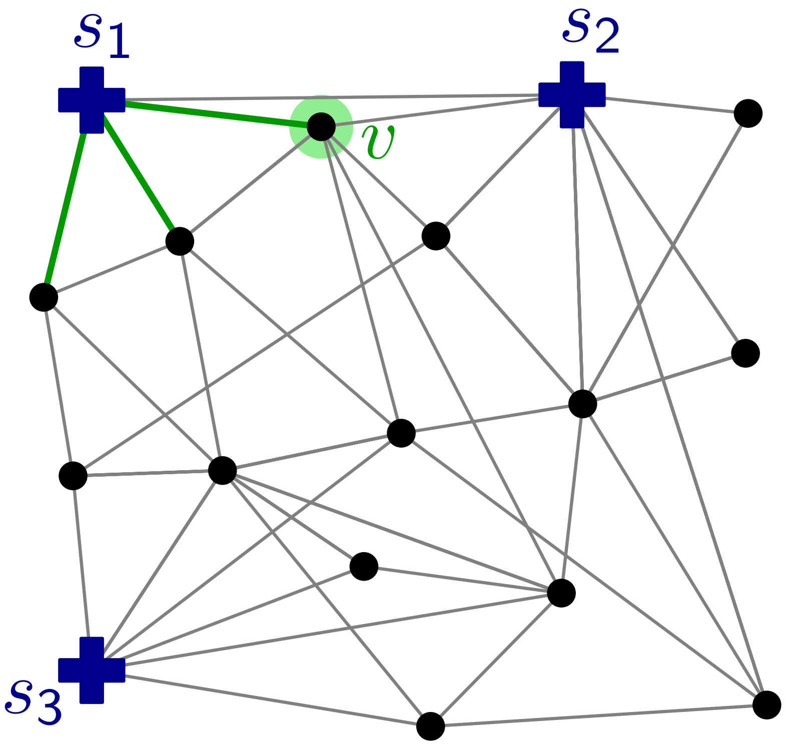
Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c \colon E \to \mathbb{Q}_{\geq 0}$

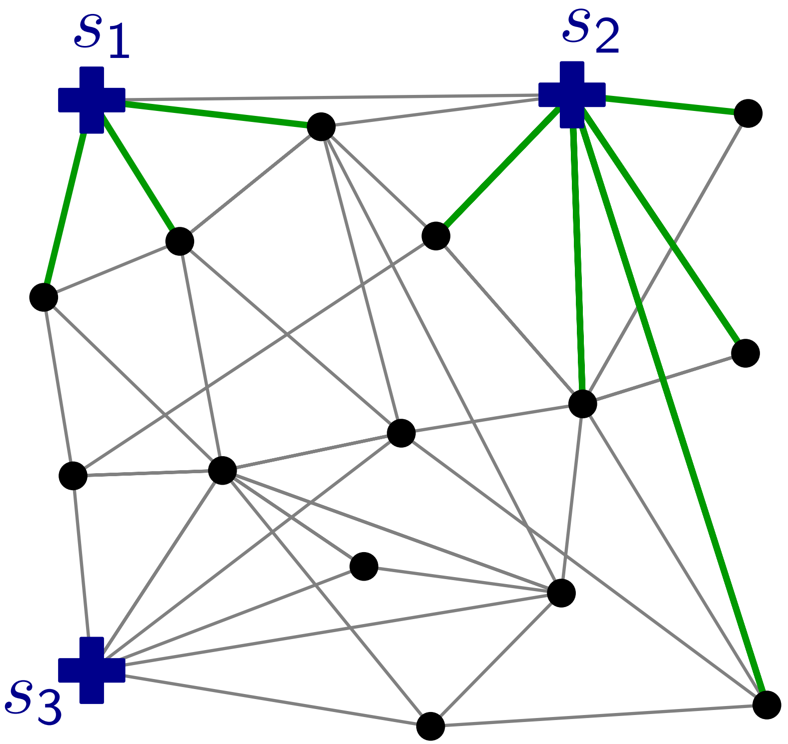
Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

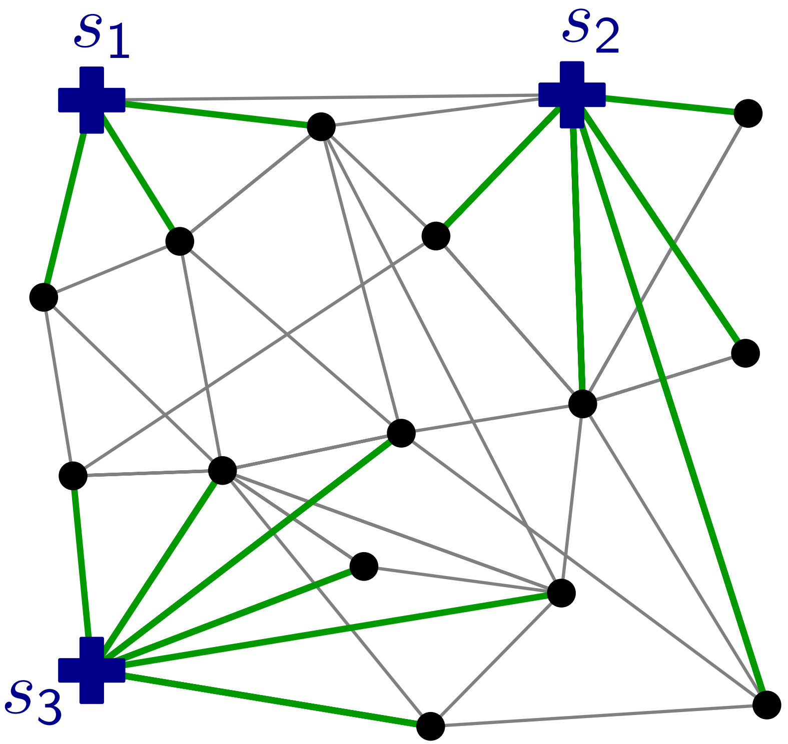
Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.



Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.

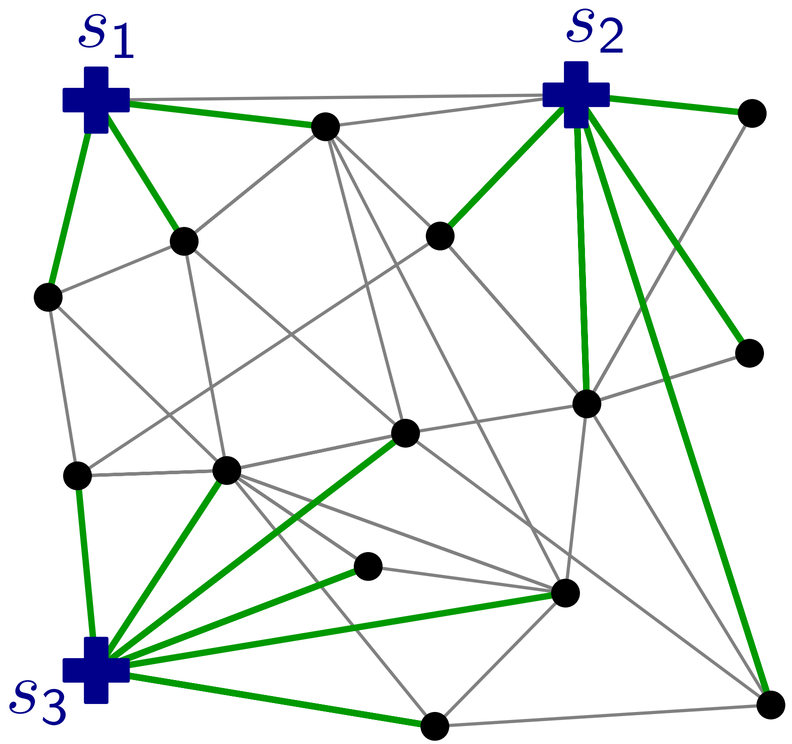


Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.

$\text{cost}(S) := \max_{v \in V} c(v, S)$

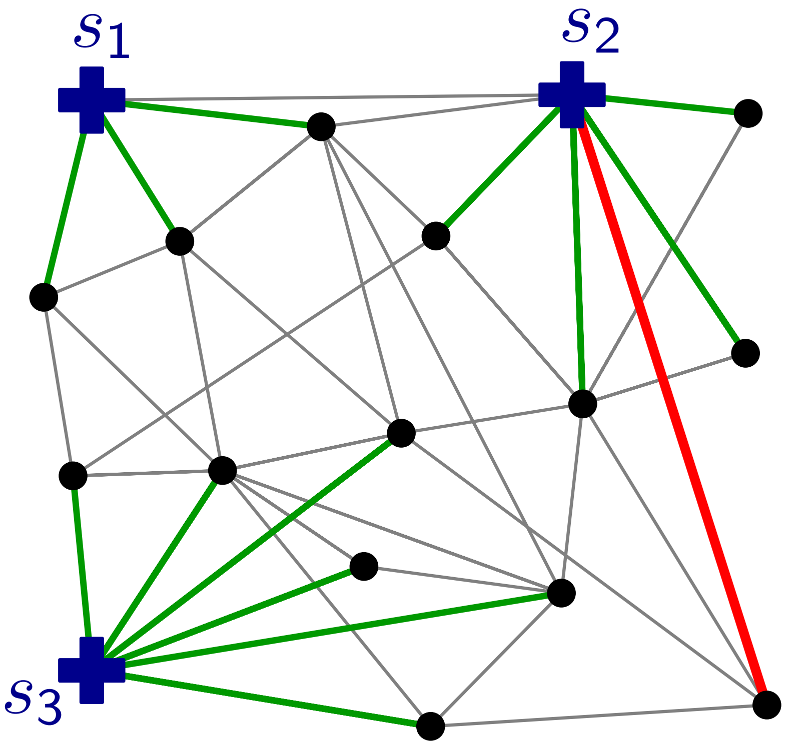


Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.

$\text{cost}(S) := \max_{v \in V} c(v, S)$

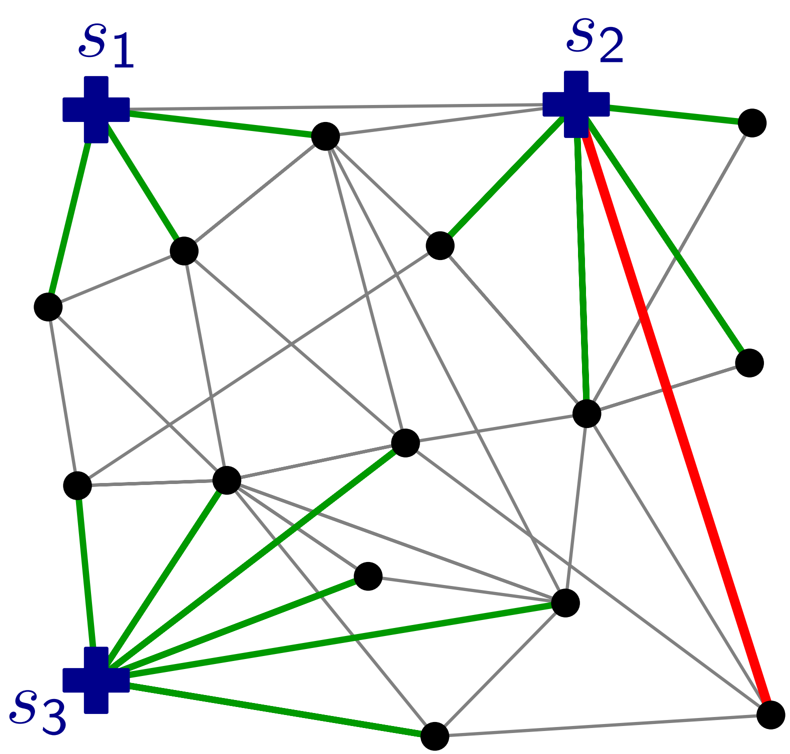


Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c \colon E \to \mathbb{Q}_{\geq 0}$

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.

Gesucht: Eine Knotenmenge S, so dass $\mathrm{cost}(S) := \max_{v \in V} c(v, S)$ minimal ist.

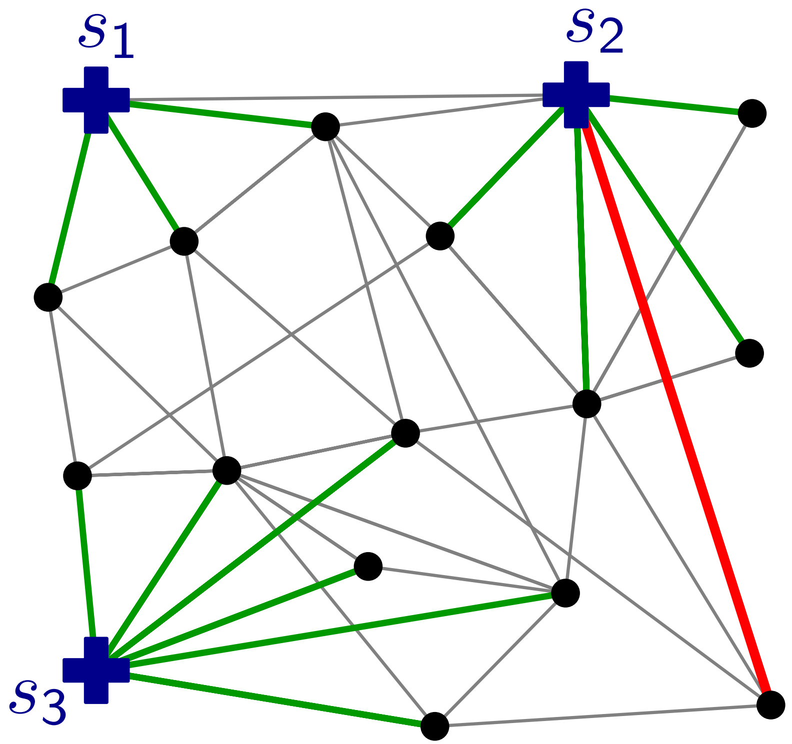


Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$ und eine natürliche Zahl $k \leq |V|$.

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.

Gesucht: Eine Knotenmenge S, so dass $\mathrm{cost}(S) := \max_{v \in V} c(v, S)$ minimal ist.

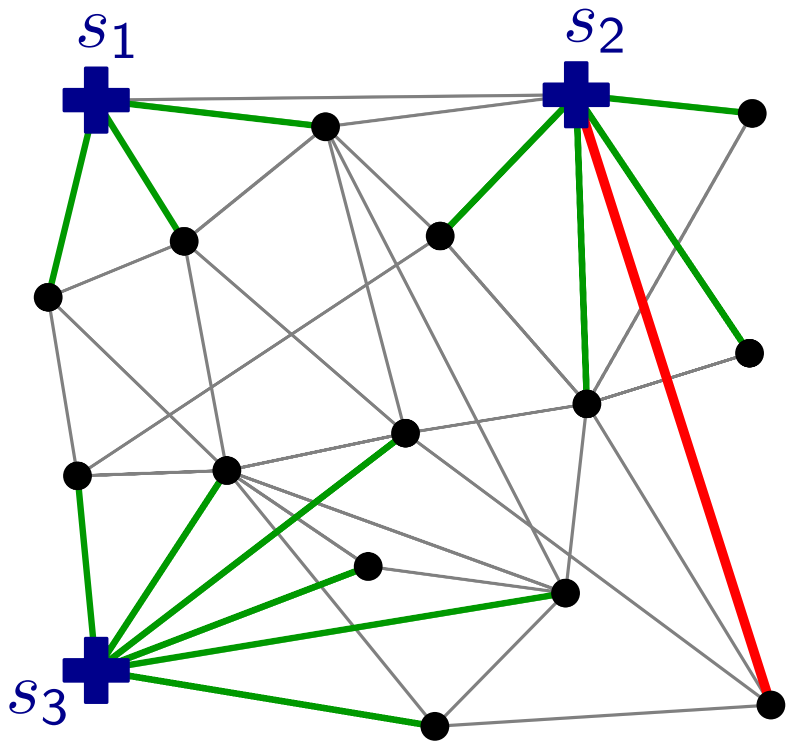


Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c \colon E \to \mathbb{Q}_{\geq 0}$ und eine natürliche Zahl $k \leq |V|$.

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.

Gesucht: Eine k-elementige Knotenmenge S, so dass $\text{cost}(S) := \max_{v \in V} c(v, S)$ minimal ist.

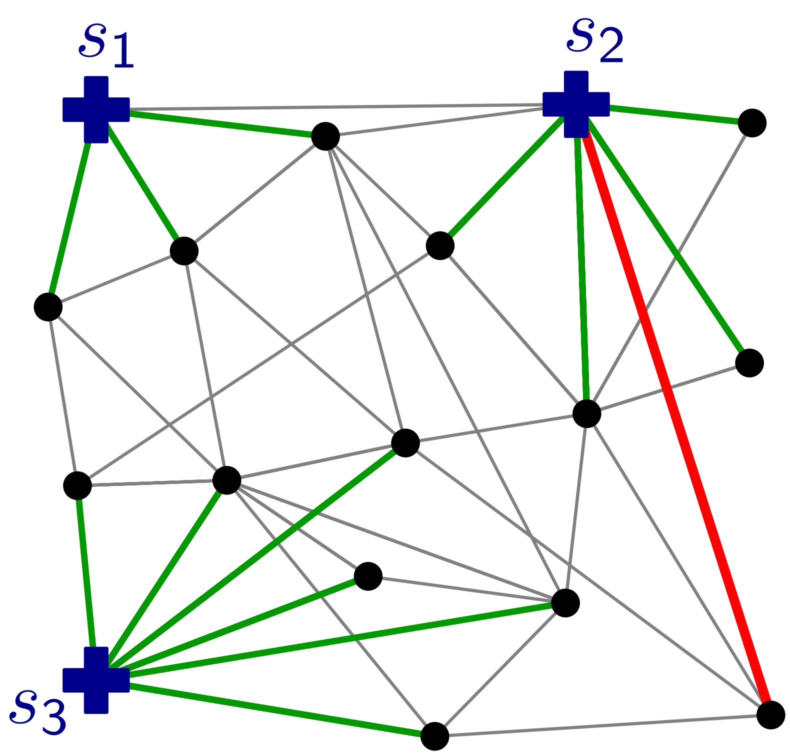


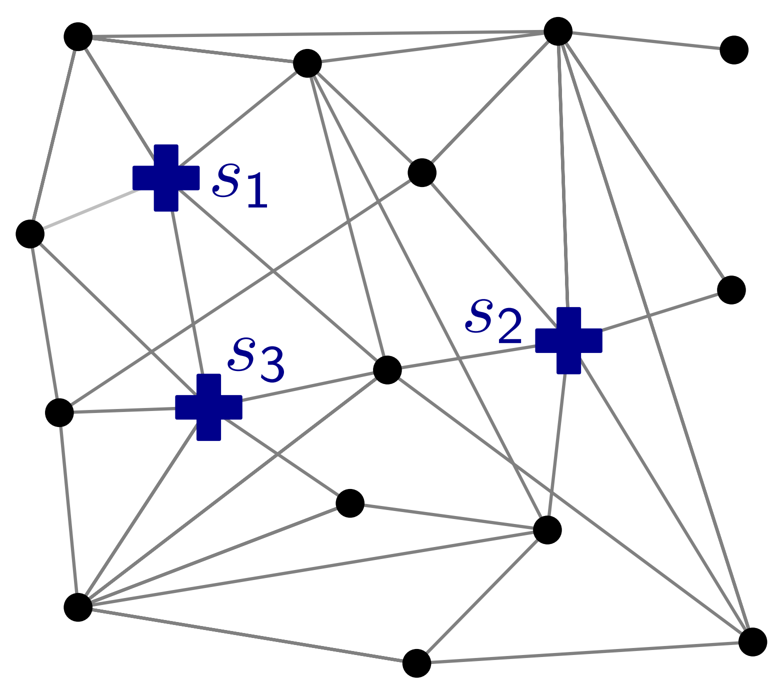
Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$ und eine natürliche Zahl $k \leq |V|$.

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.

Gesucht: Eine k-elementige Knotenmenge S, so dass $\text{cost}(S) := \max_{v \in V} c(v, S)$ minimal ist.



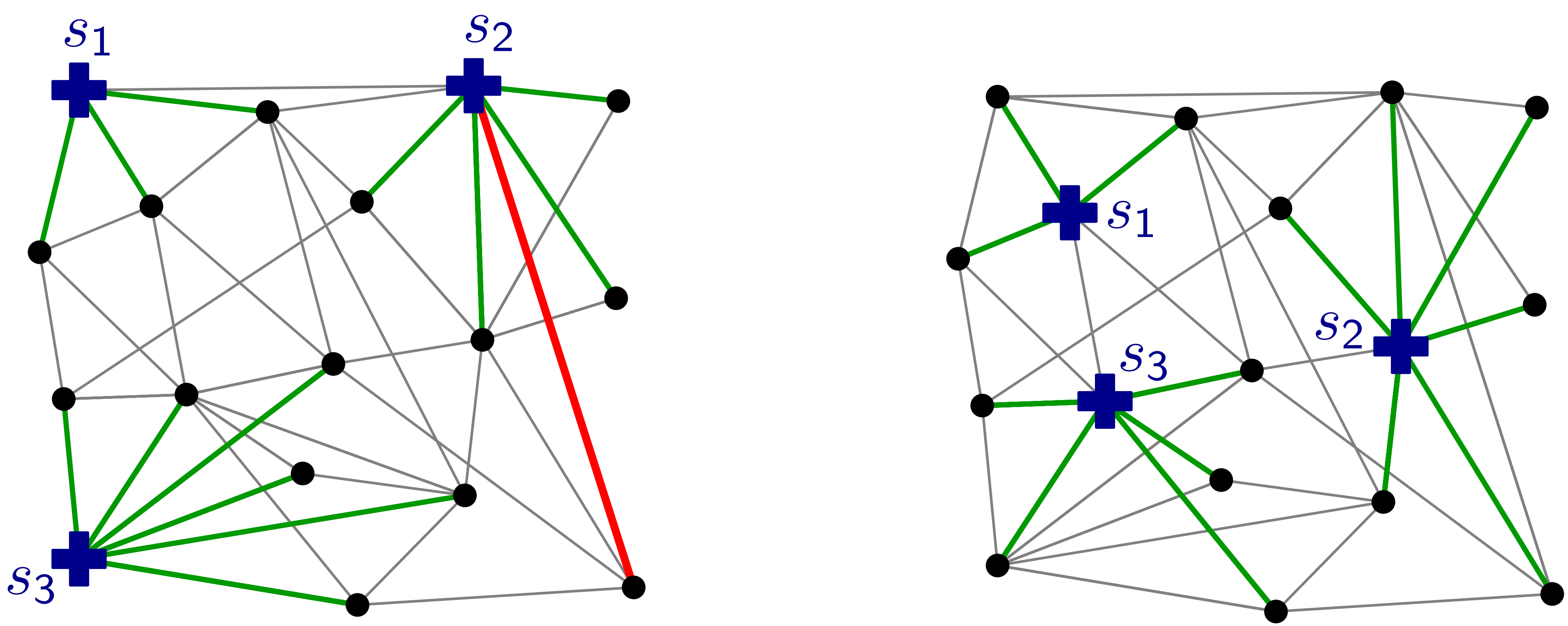


Das metrische k-Zentrum-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$ und eine natürliche Zahl $k \leq |V|$.

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.

Gesucht: Eine k-elementige Knotenmenge S, so dass $\mathrm{cost}(S) := \max_{v \in V} c(v, S)$ minimal ist.

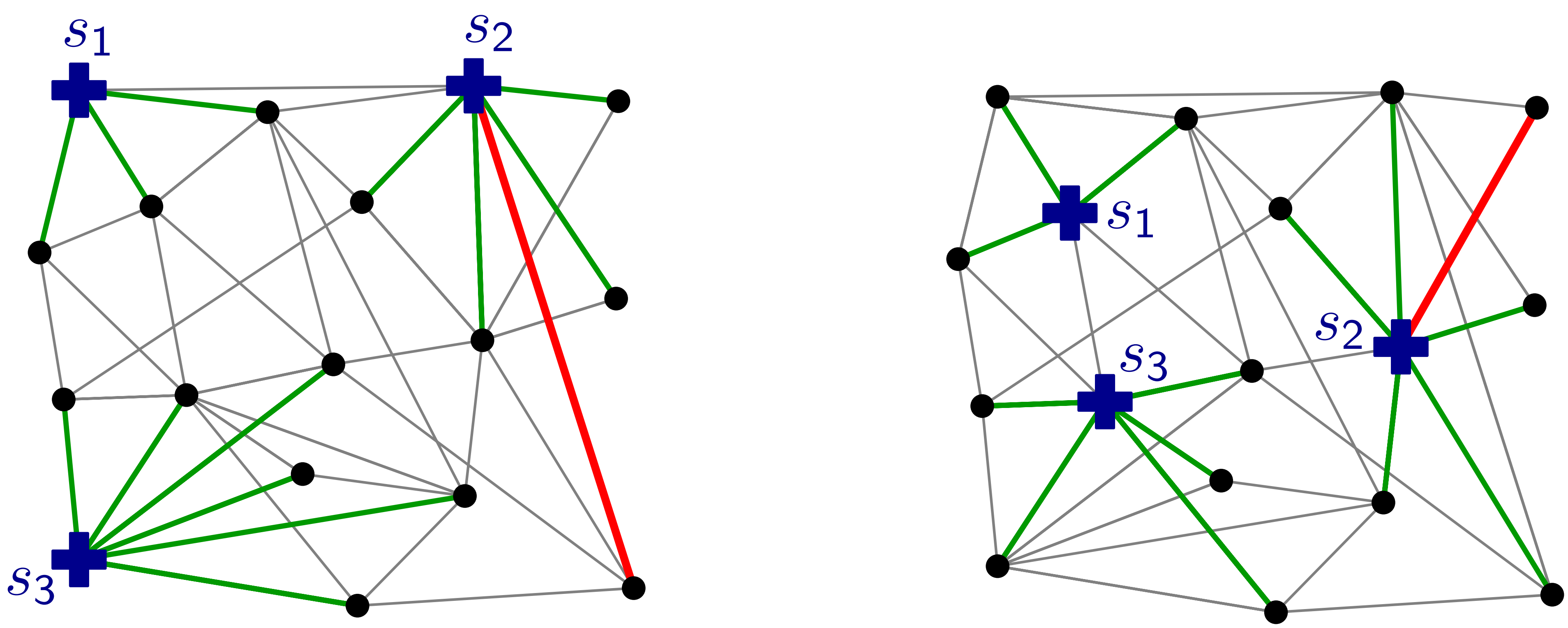


Das metrische k-Zentrum-Problem

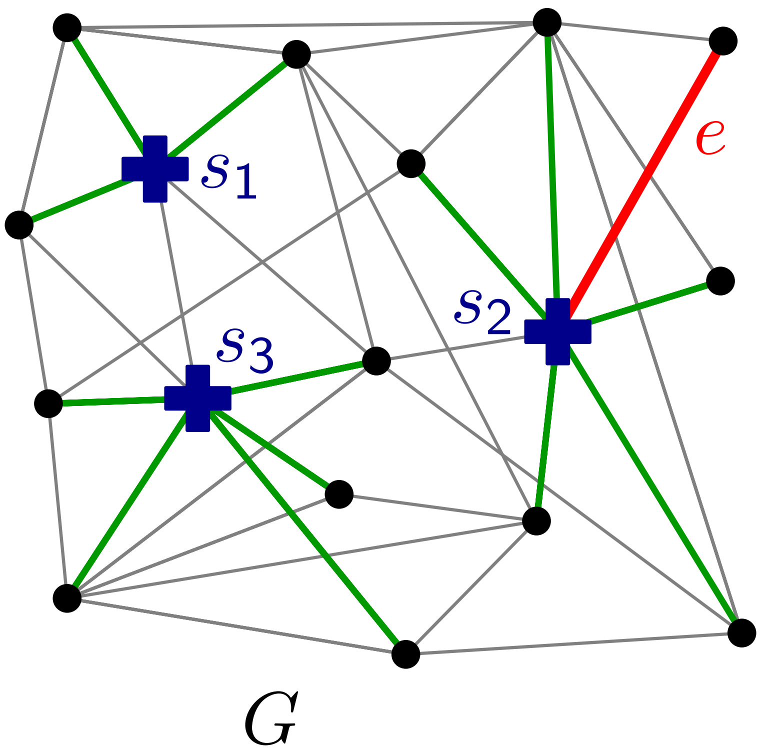
Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$ und eine natürliche Zahl $k \leq |V|$.

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten einer günstigsten Kante von v zu einem Knoten aus S.

Gesucht: Eine k-elementige Knotenmenge S, so dass $\text{cost}(S) := \max_{v \in V} c(v, S)$ minimal ist.

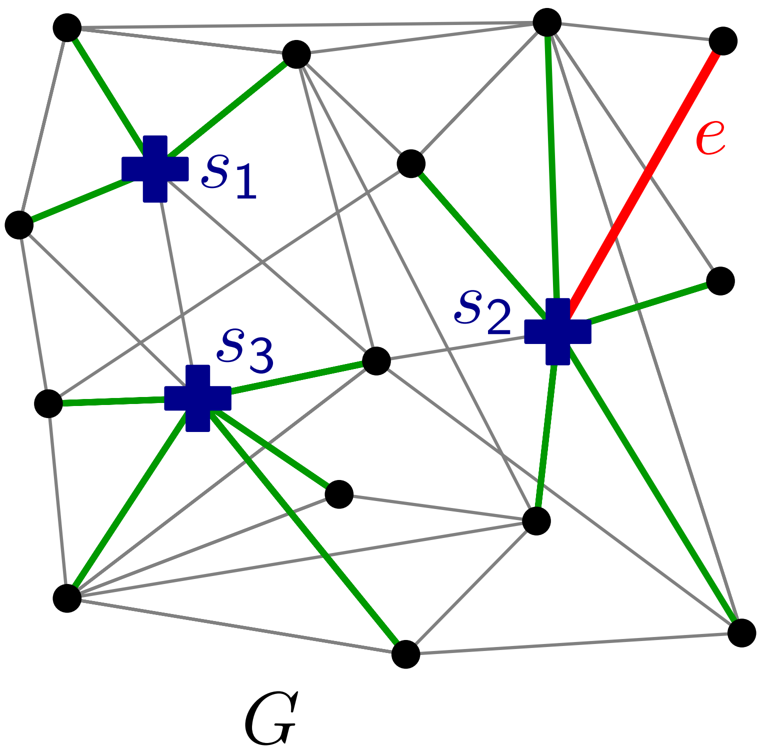


Parametrisches Pruning



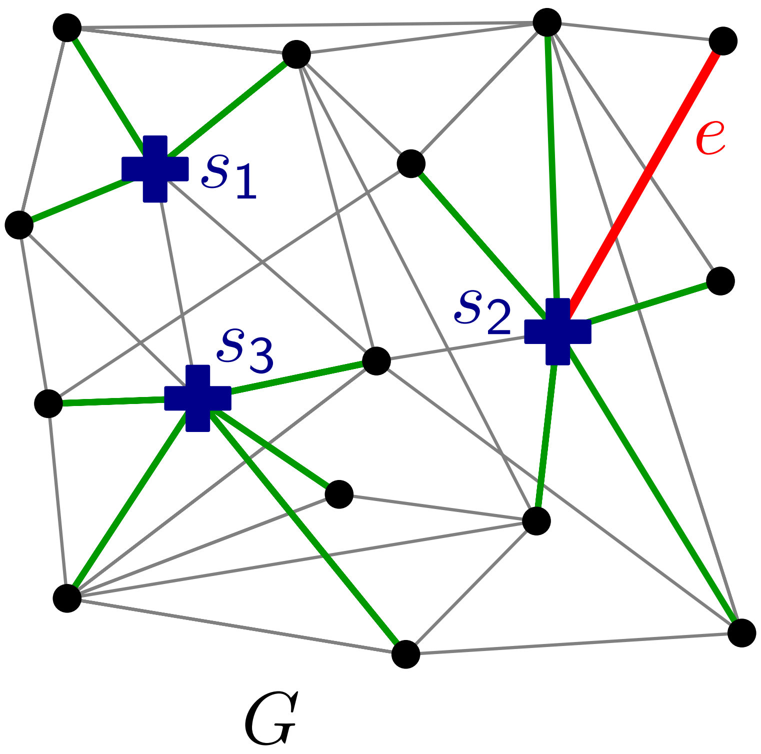
Parametrisches Pruning

Angenommen, wir wissen, dass OPT $= c(e_j)$.



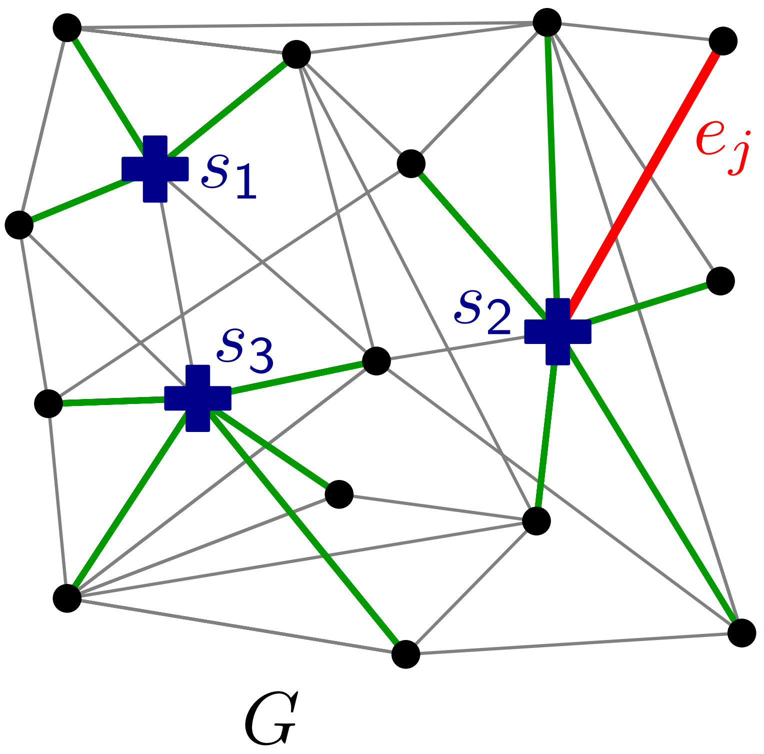
Parametrisches Pruning

Sei $E = \{e_1, \ldots, e_m\}$ mit $c(e_1) \leq \ldots \leq c(e_m)$.
Angenommen, wir wissen, dass $\mathrm{OPT} = c(e_j)$.



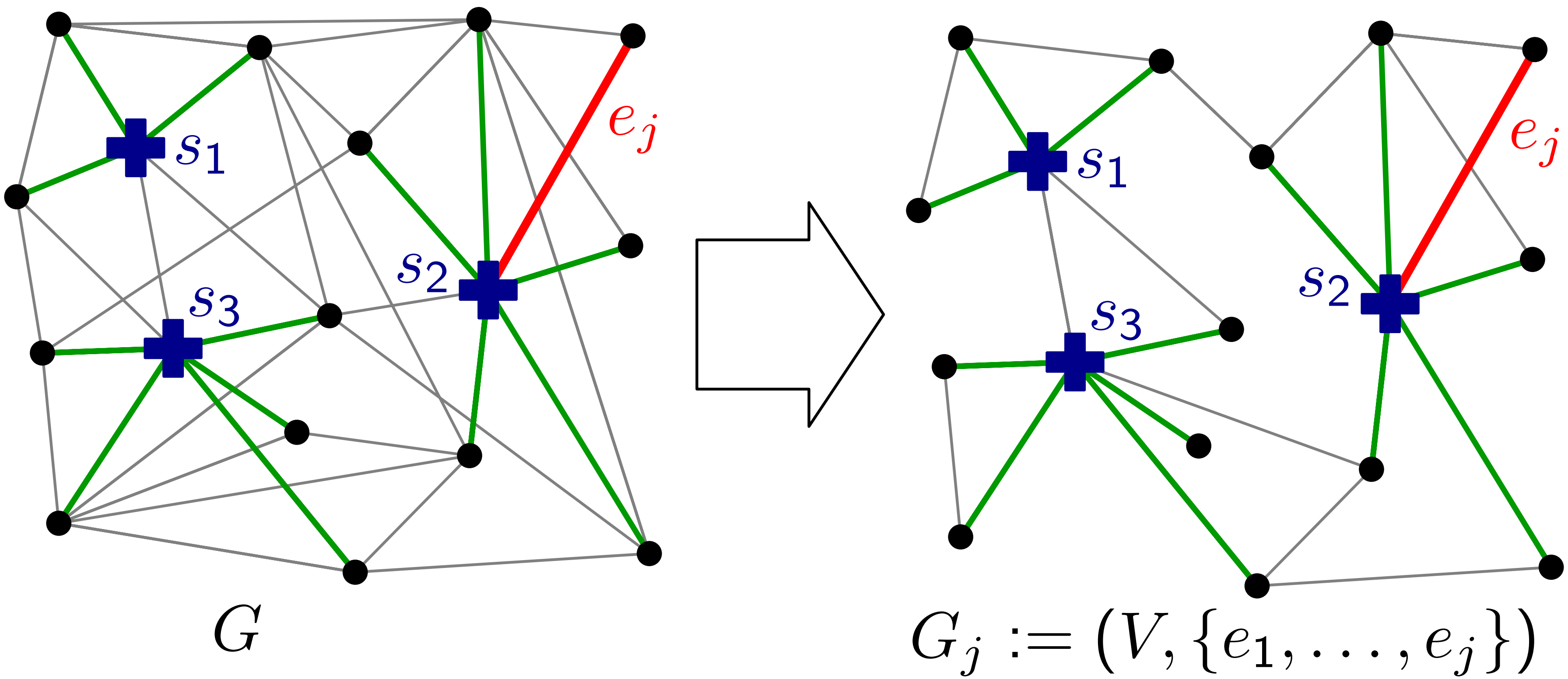
Parametrisches Pruning

Sei $E = \{e_1, \dots, e_m\}$ mit $c(e_1) \leq \ldots \leq c(e_m)$.
Angenommen, wir wissen, dass OPT $= c(e_j)$.



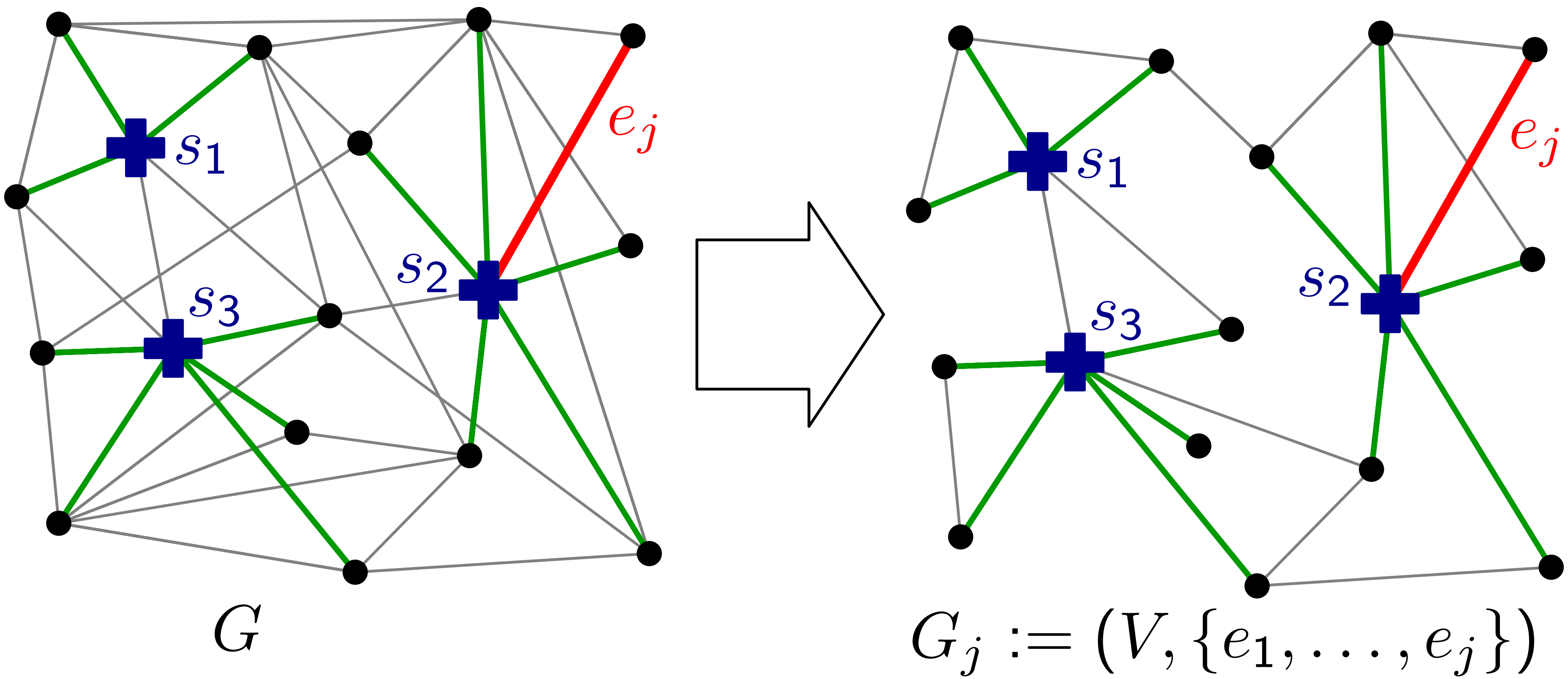
Parametrisches Pruning

Sei $E = \{e_1, \ldots, e_m\}$ mit $c(e_1) \leq \ldots \leq c(e_m)$.
Angenommen, wir wissen, dass $\mathrm{OPT} = c(e_j)$.



Parametrisches Pruning

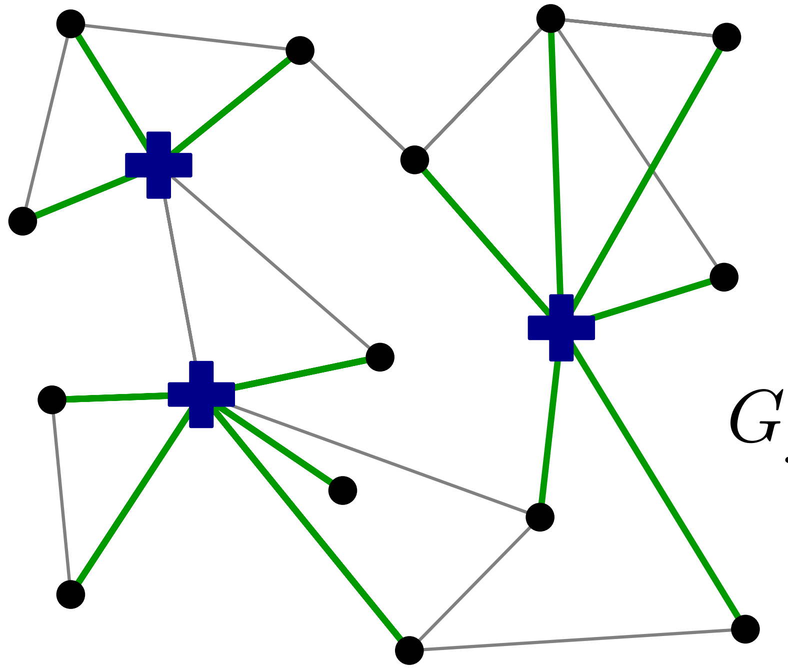
Sei $E = \{e_1, \ldots, e_m\}$ mit $c(e_1) \leq \ldots \leq c(e_m)$.
Angenommen, wir wissen, dass $\mathrm{OPT} = c(e_j)$.



…probieren alle G_i aus.

Dominierende Menge

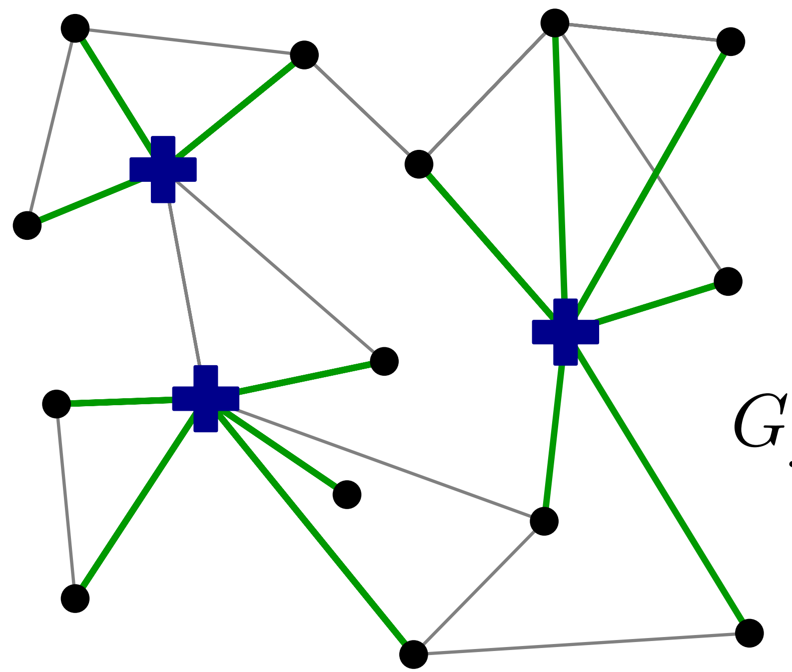
Def.



$$G_j := (V, \{e_1, \ldots, e_j\})$$

Dominierende Menge

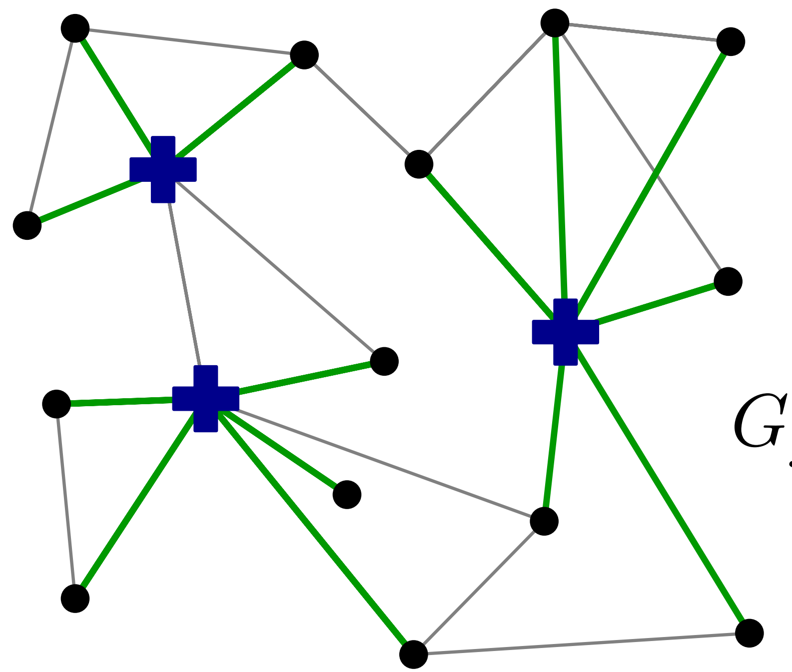
Def. Eine Knotenmenge D eines Graphen H heißt **dominierend**, wenn jeder Knoten in D liegt oder adjazent zu einem Knoten aus D ist.



$$G_j := (V, \{e_1, \ldots, e_j\})$$

Dominierende Menge

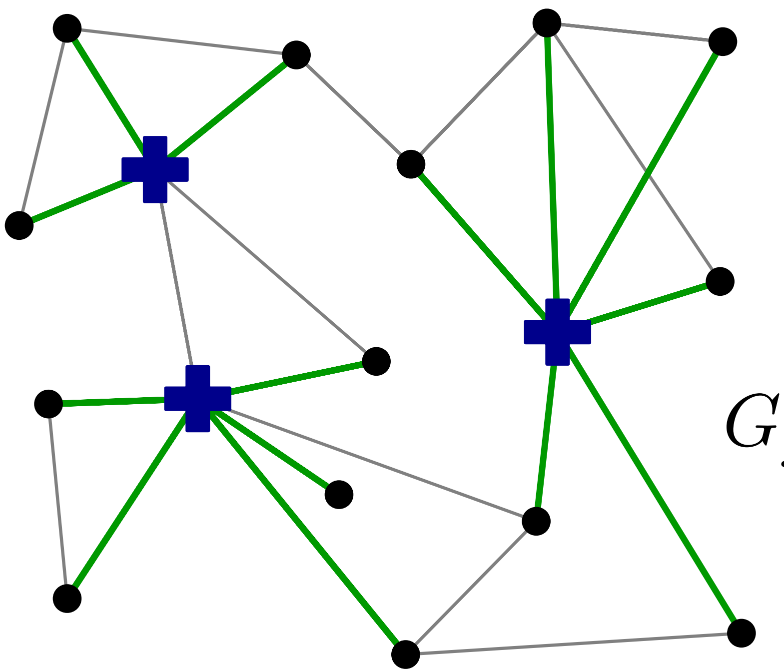
Def. Eine Knotenmenge D eines Graphen H heißt **dominierend**, wenn jeder Knoten in D liegt oder adjazent zu einem Knoten aus D ist. Die Kardinalität einer kleinsten dominierenden Menge wird mit $\mathrm{dom}(H)$ bezeichnet.



$$G_j := (V, \{e_1, \ldots, e_j\})$$

Dominierende Menge

Def. Eine Knotenmenge D eines Graphen H heißt **dominierend**, wenn jeder Knoten in D liegt oder adjazent zu einem Knoten aus D ist. Die Kardinalität einer kleinsten dominierenden Menge wird mit $\mathrm{dom}(H)$ bezeichnet.

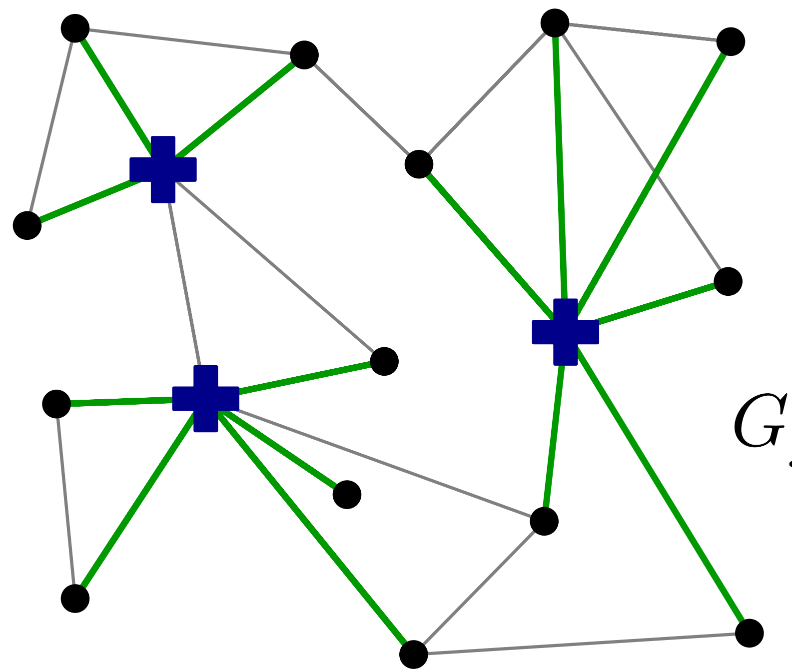


$$\mathrm{dom}(G_j) \leq k$$

$$G_j := (V, \{e_1, \ldots, e_j\})$$

Dominierende Menge

Def. Eine Knotenmenge D eines Graphen H heißt **dominierend**, wenn jeder Knoten in D liegt oder adjazent zu einem Knoten aus D ist. Die Kardinalität einer kleinsten dominierenden Menge wird mit $\mathrm{dom}(H)$ bezeichnet.



$\mathrm{dom}(G_j) \leq k$

$G_j := (V, \{e_1, \ldots, e_j\})$

... aber die Berechnung von $\mathrm{dom}(H)$ ist NP-schwer.

Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

Quadrat eines Graphen

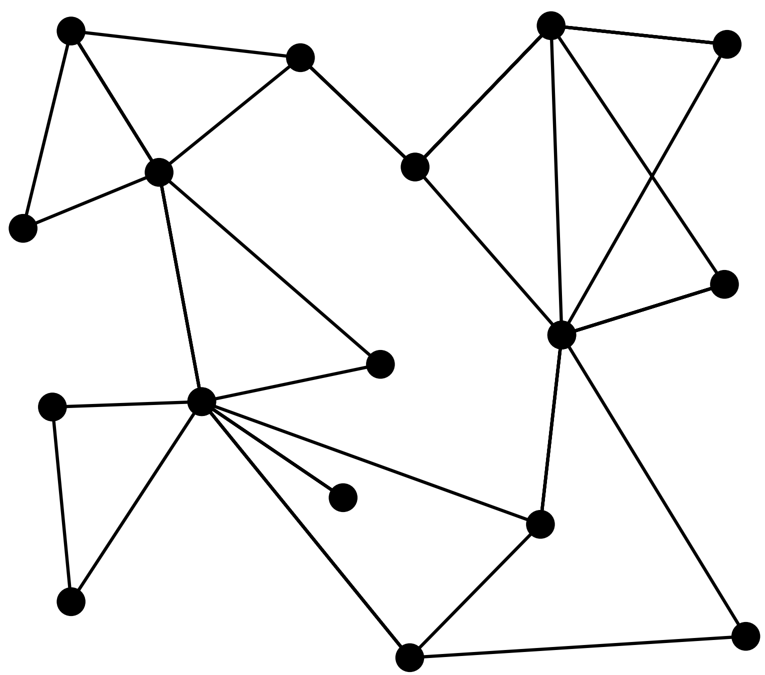
Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H.

Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H.

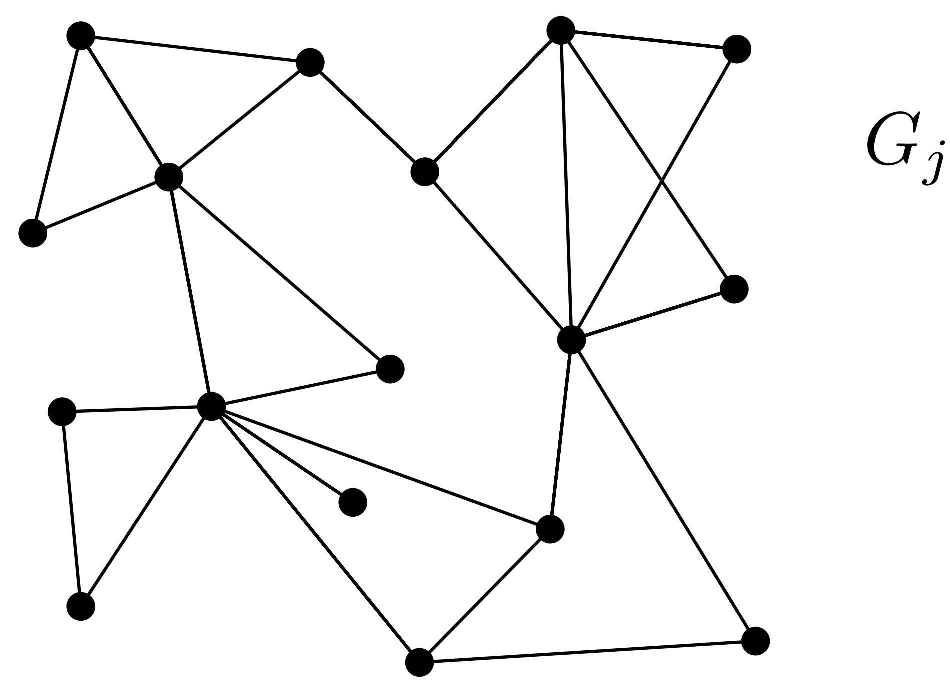


G_j

Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

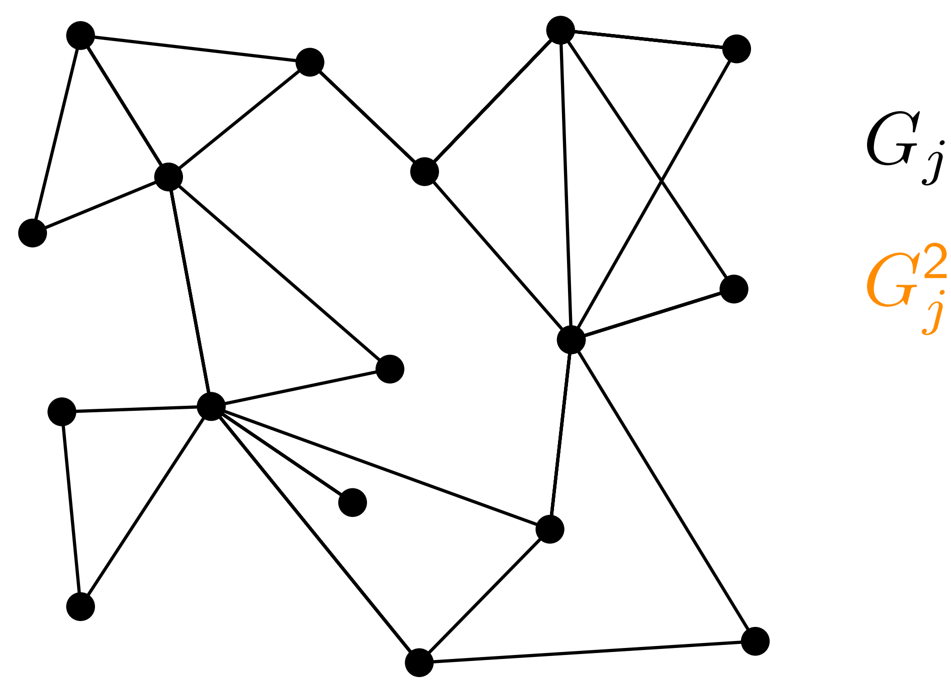
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

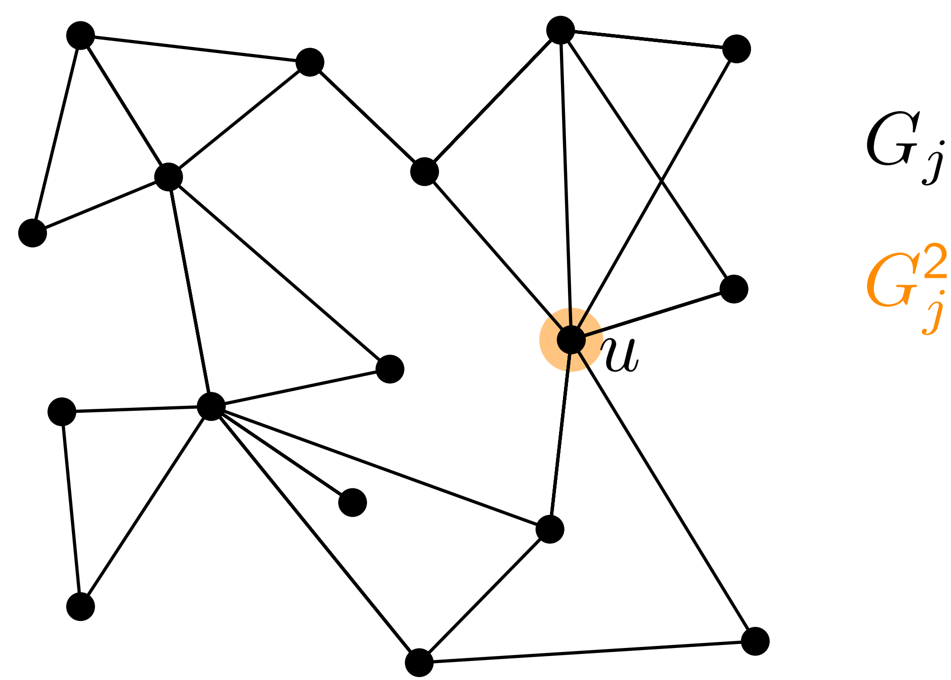
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter" Version von G_j

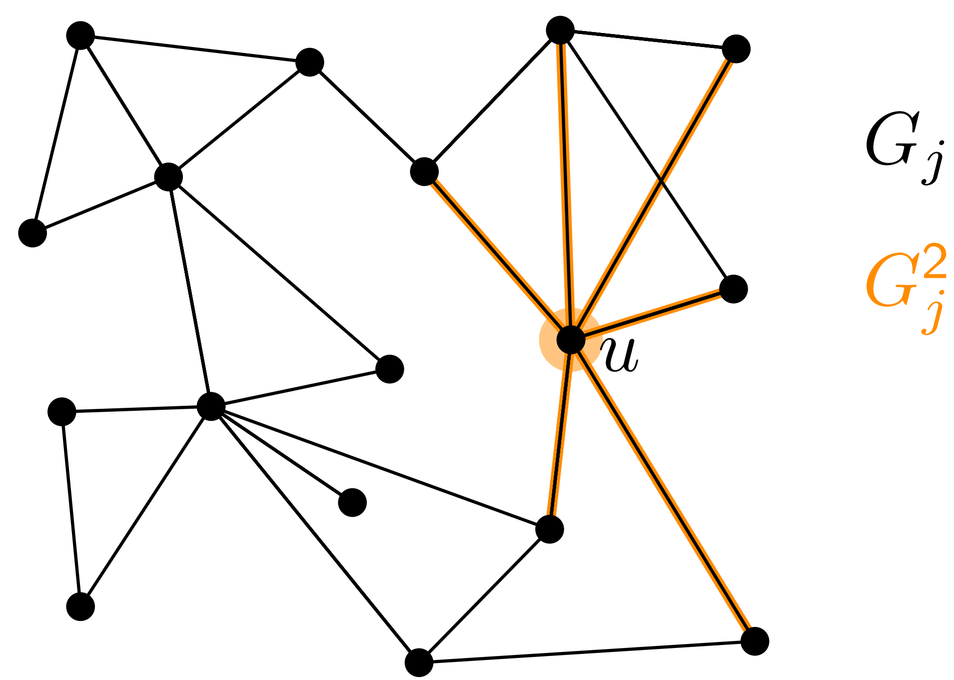
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

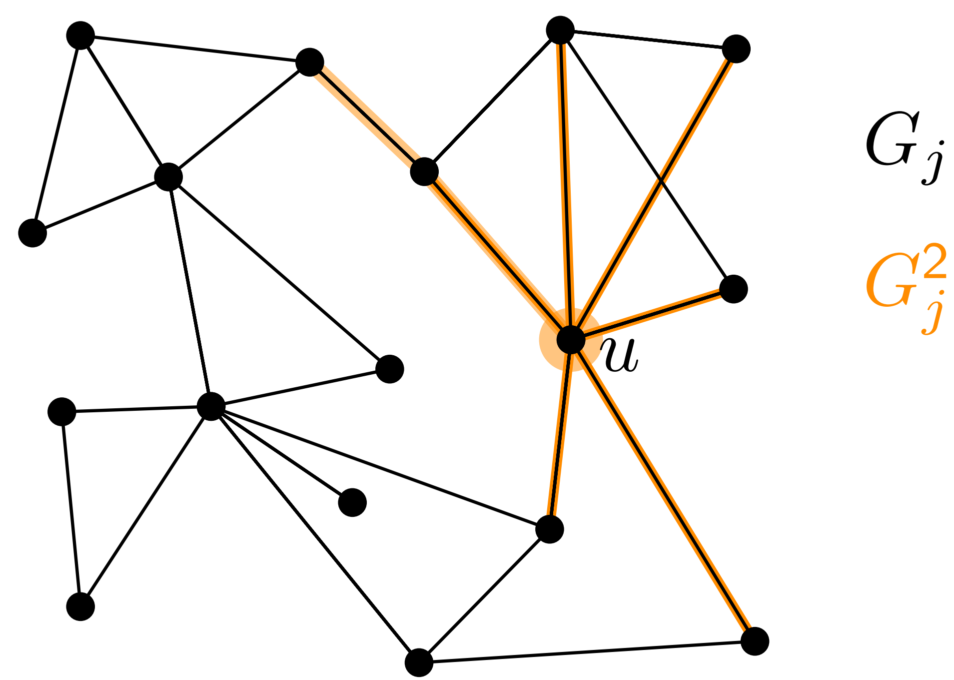
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

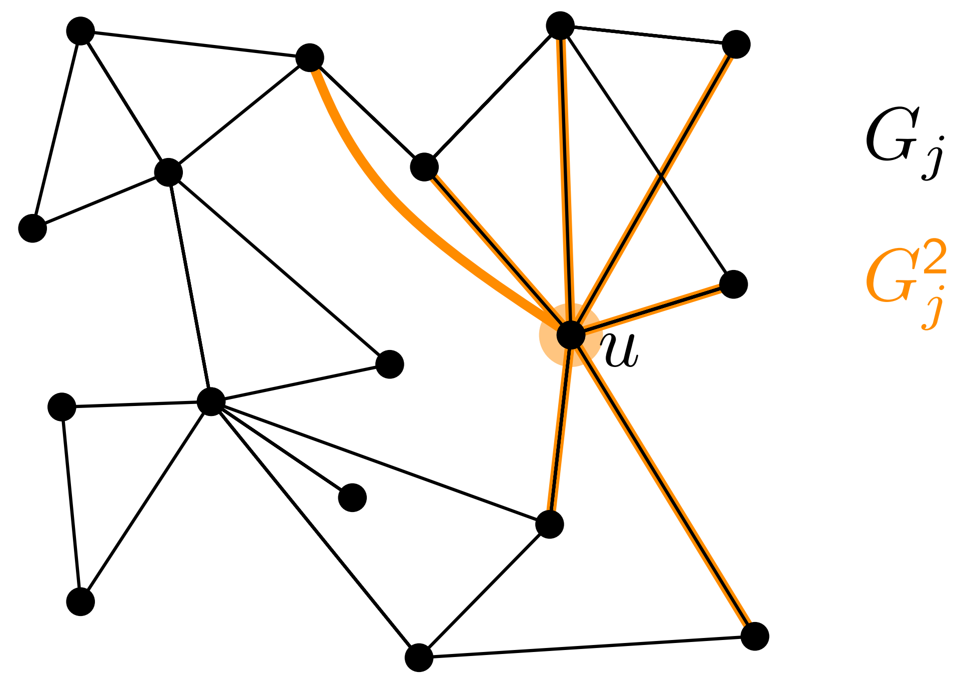
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

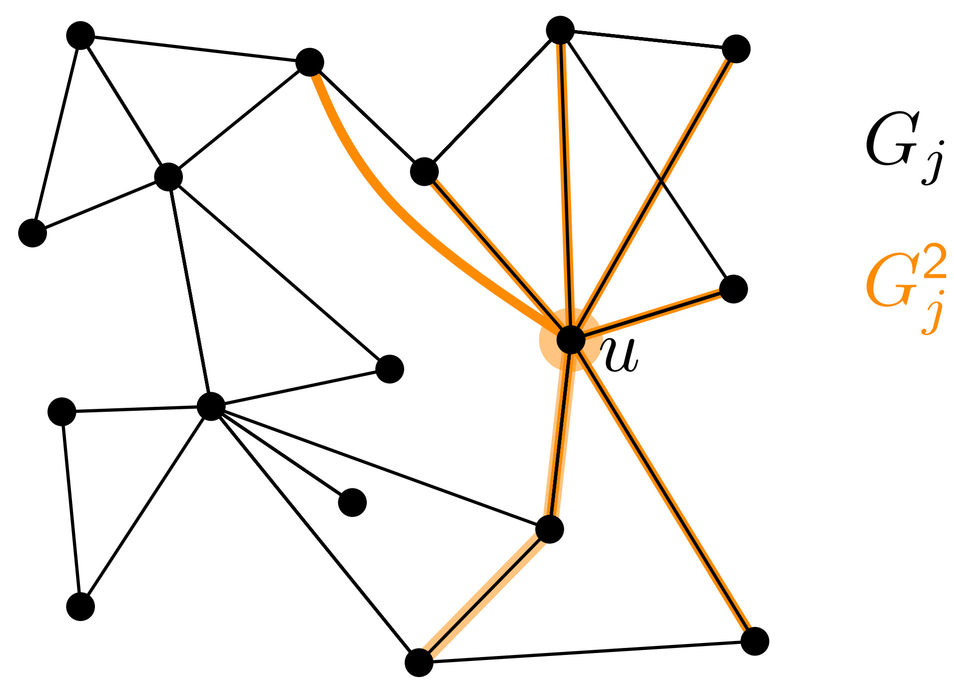
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

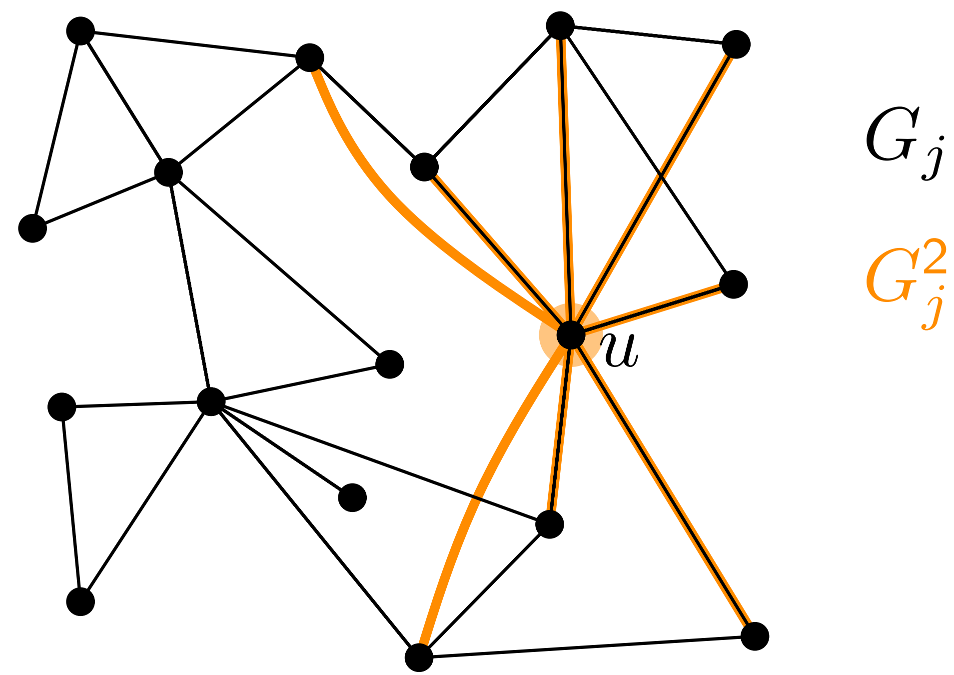
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

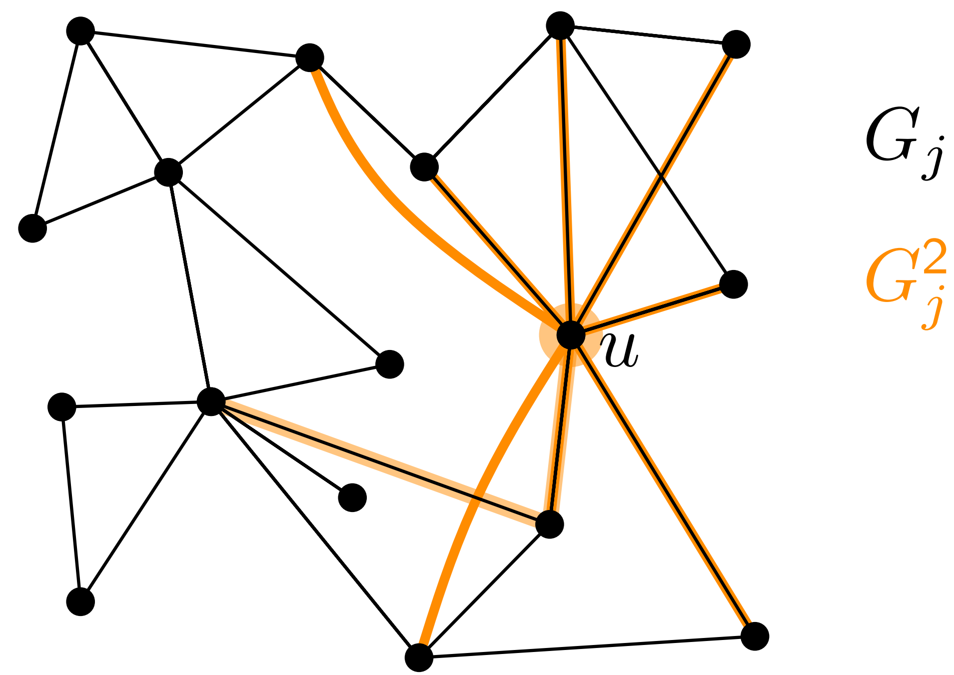
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

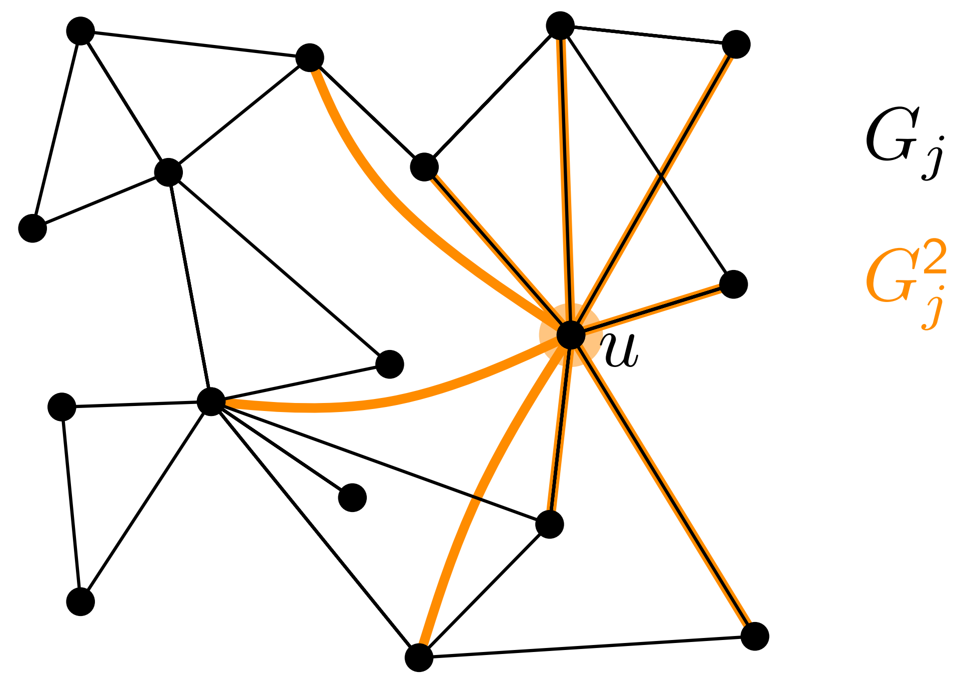
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

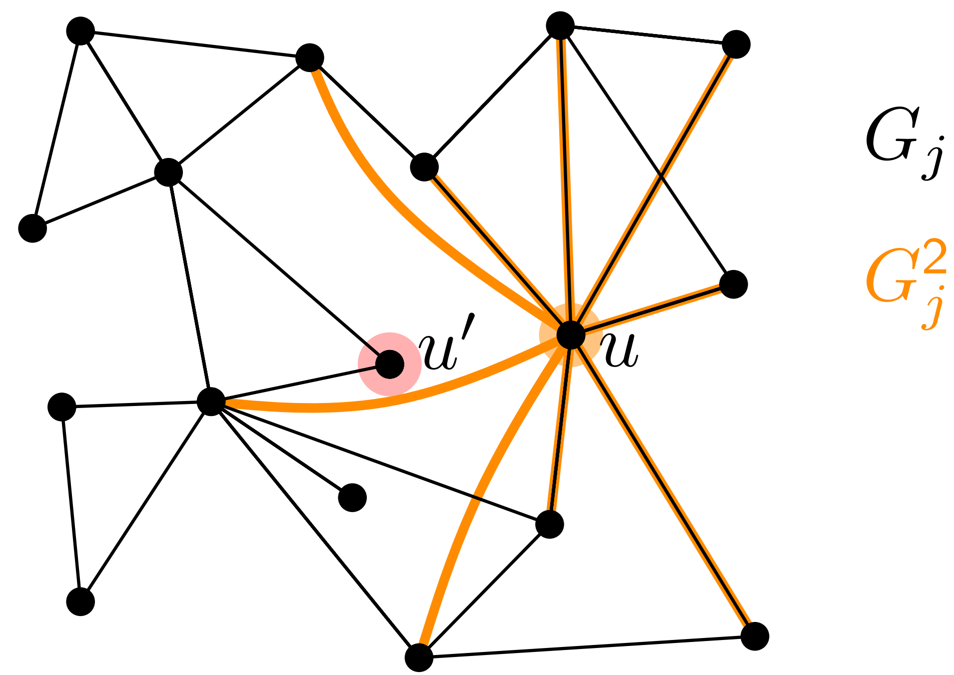
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

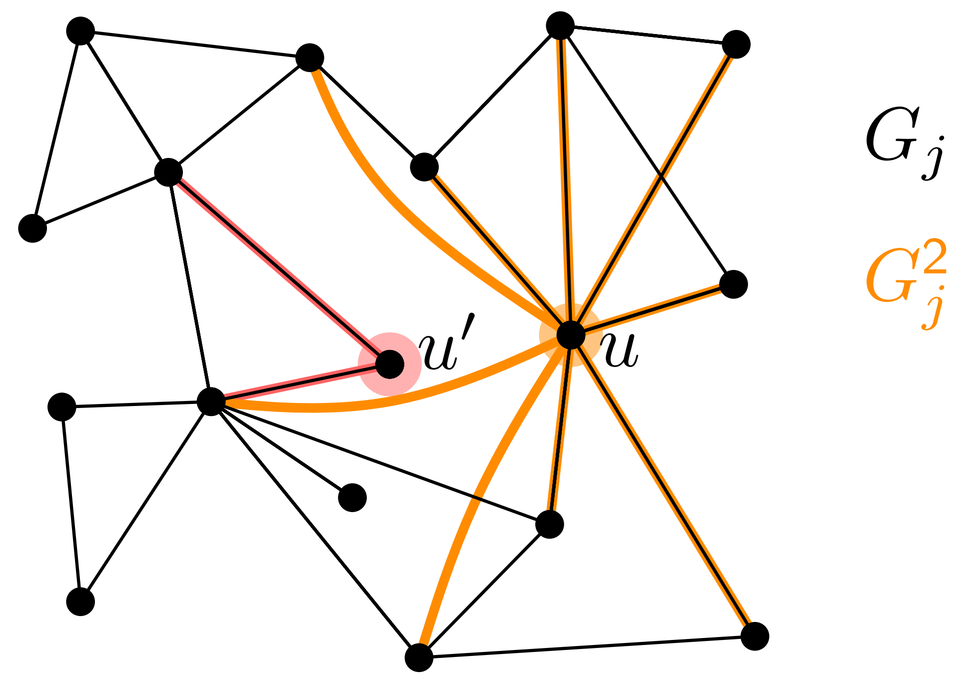
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

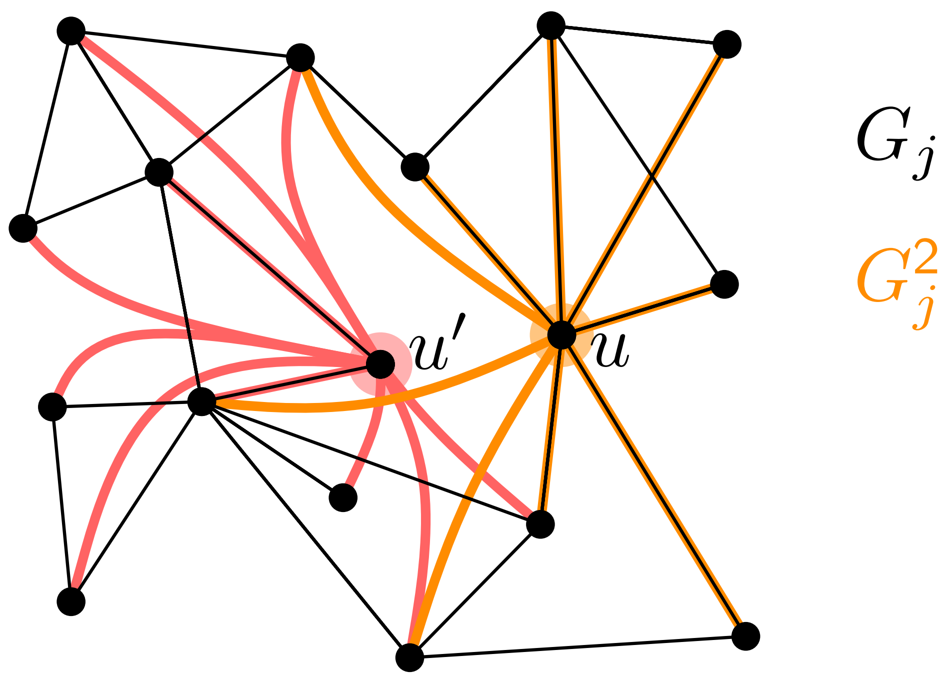
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.



Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.

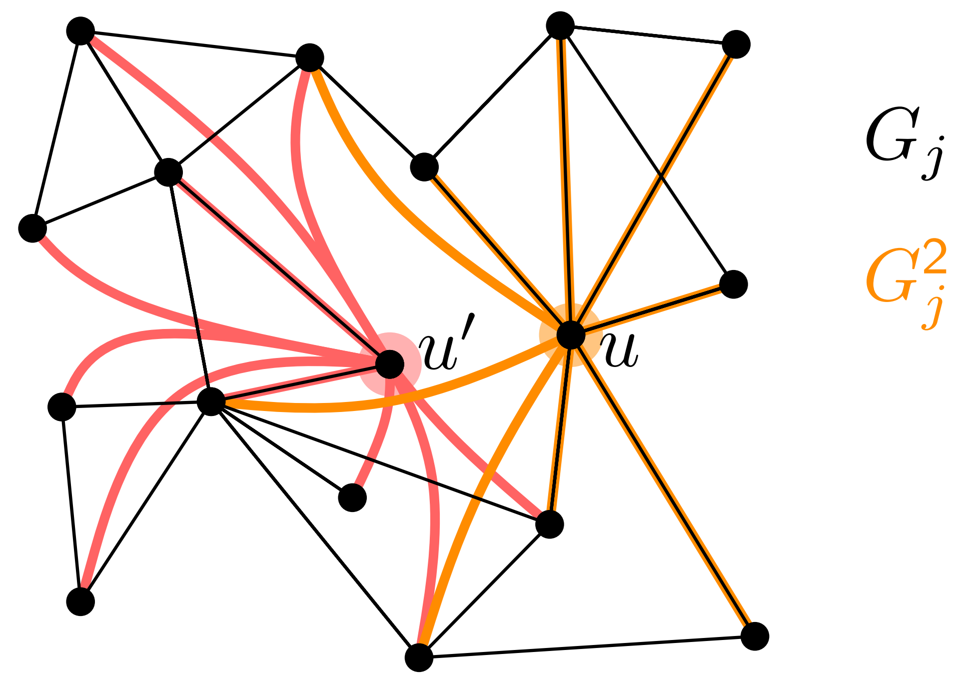


Quadrat eines Graphen

Ausweg: Finde kleine dominierende Menge in „vergröberter“ Version von G_j

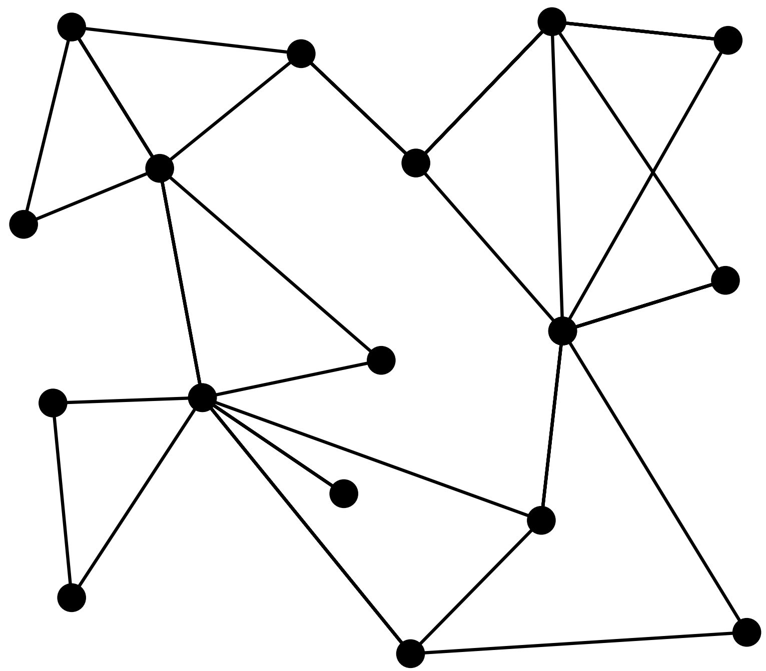
Def. Das **Quadrat** H^2 eines Graphen H hat die gleiche Knotenmenge wie H. Zwei Knoten $u \neq v$ sind genau dann adjazent in H^2, wenn es in H einen u–v-Weg mit höchstens zwei Kanten gibt.

Dominierende Menge in G_j^2 mit $\leq k$ Elementen ist 2-Approximation für metrisches k-Zentrum



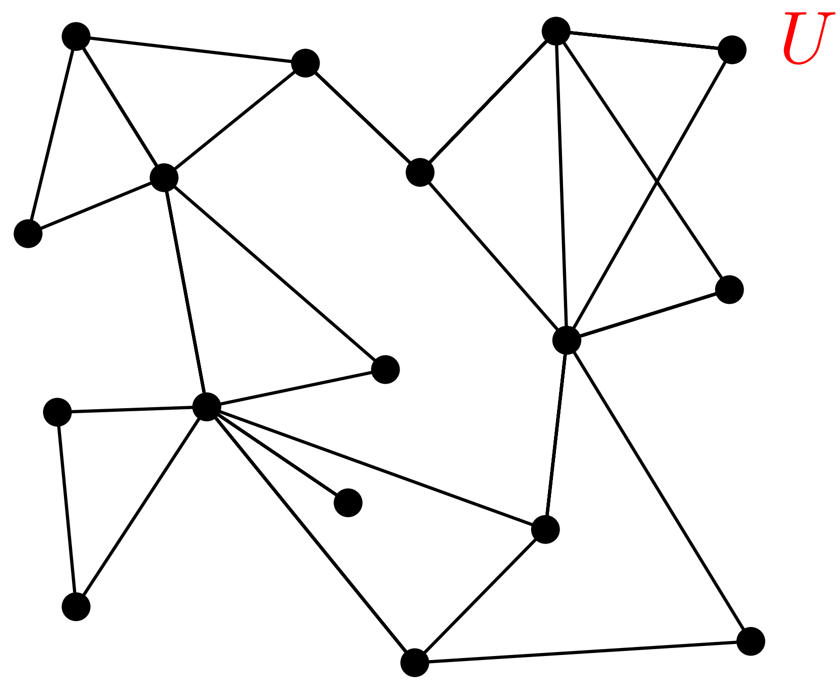
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind.



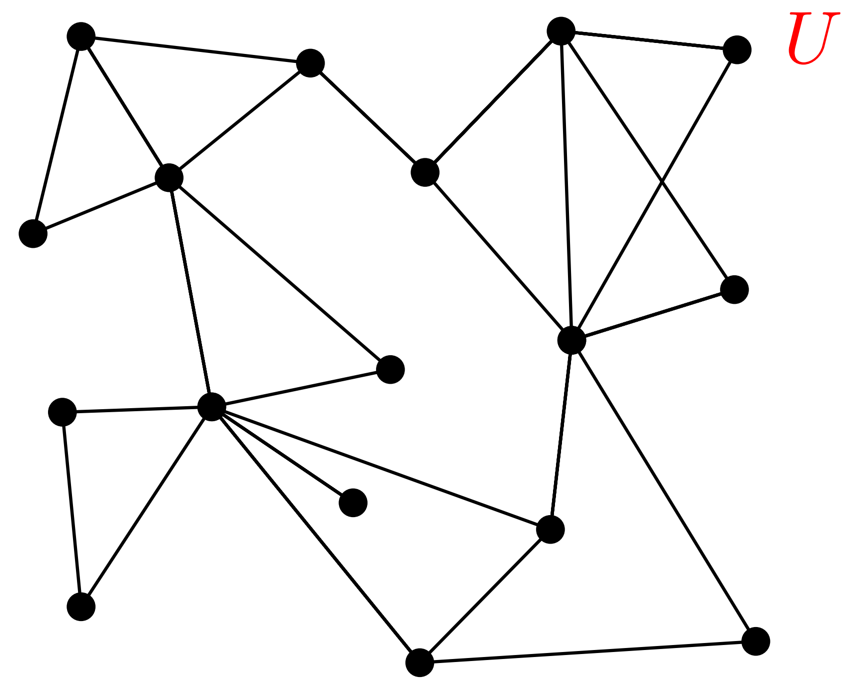
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind.



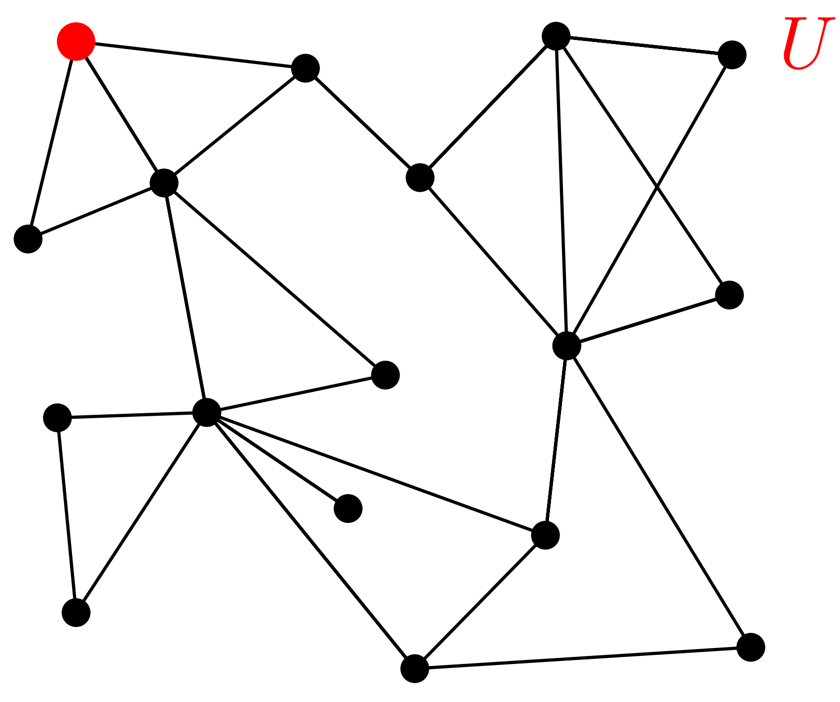
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



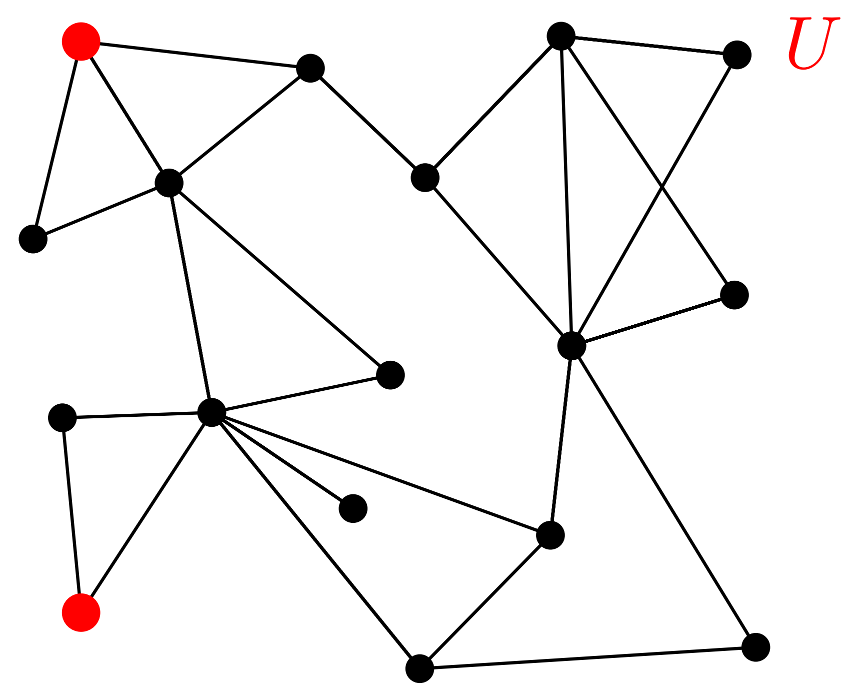
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



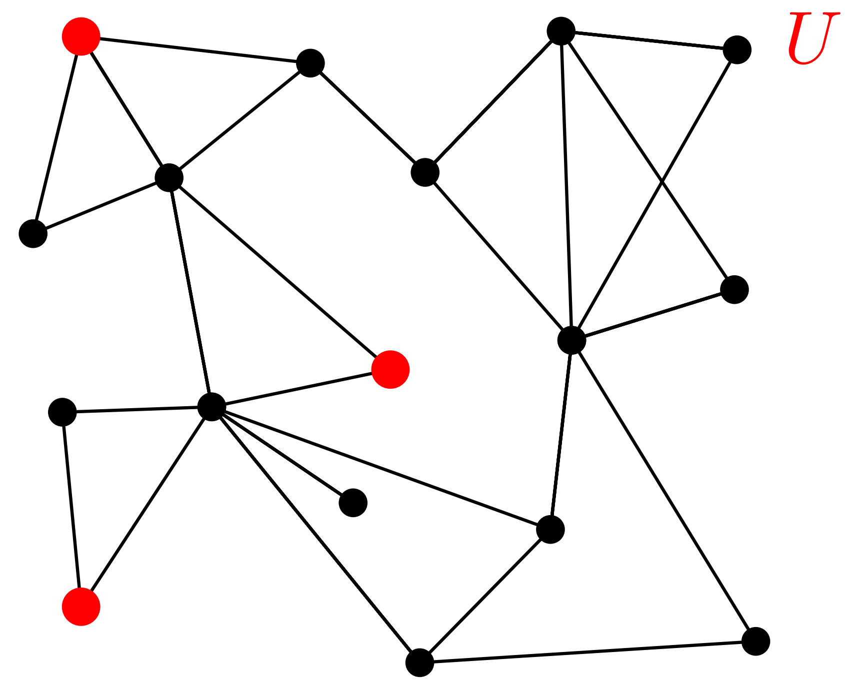
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



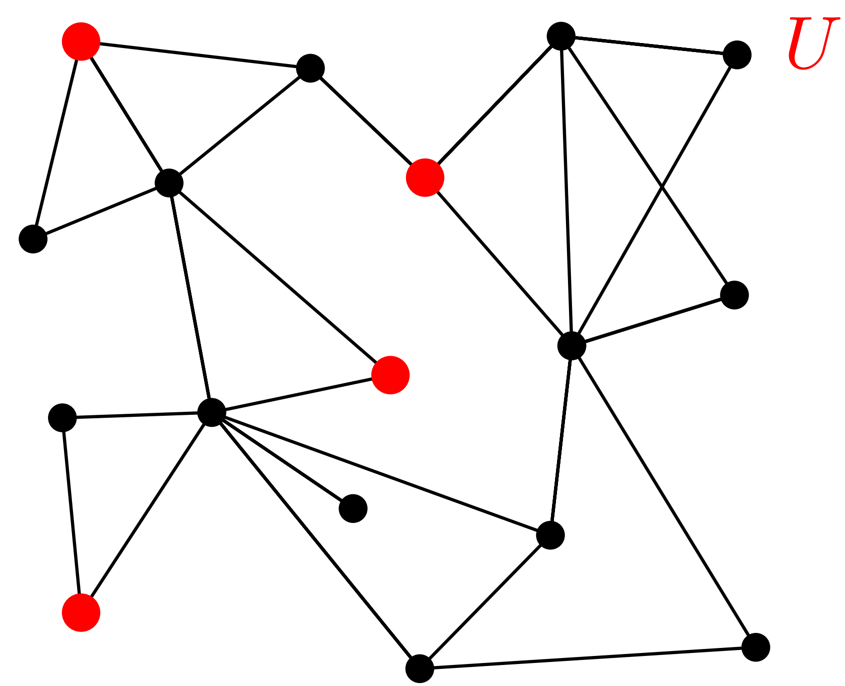
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



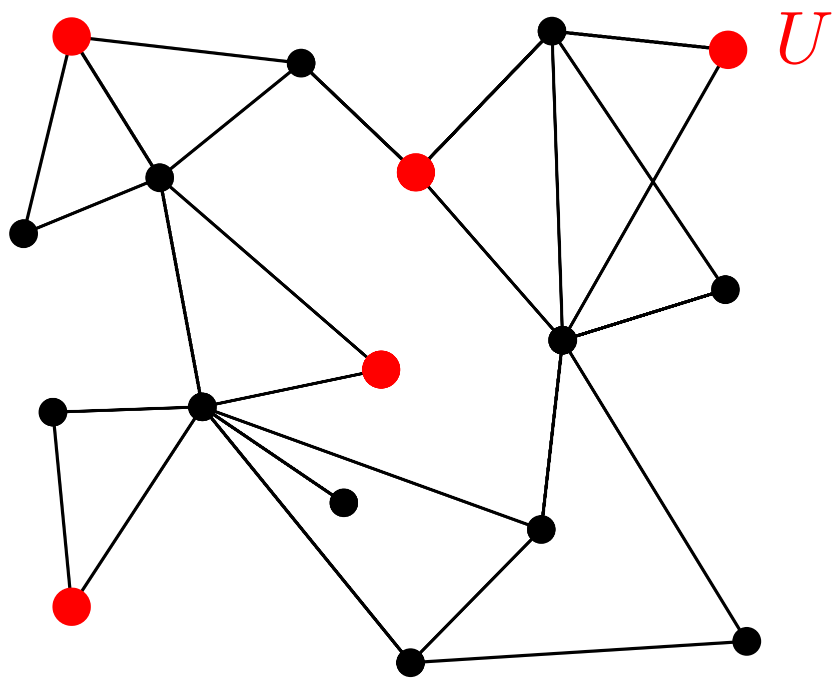
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



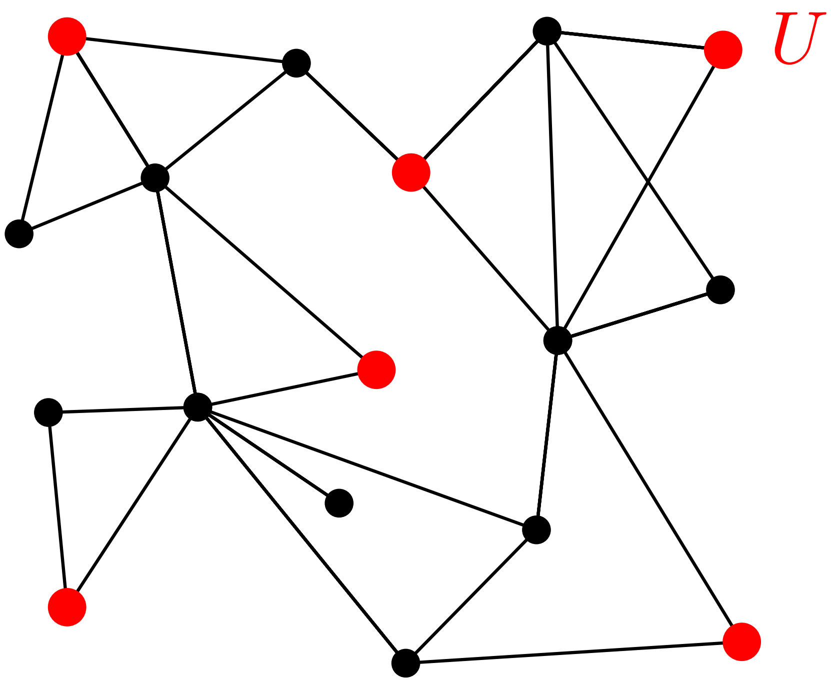
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



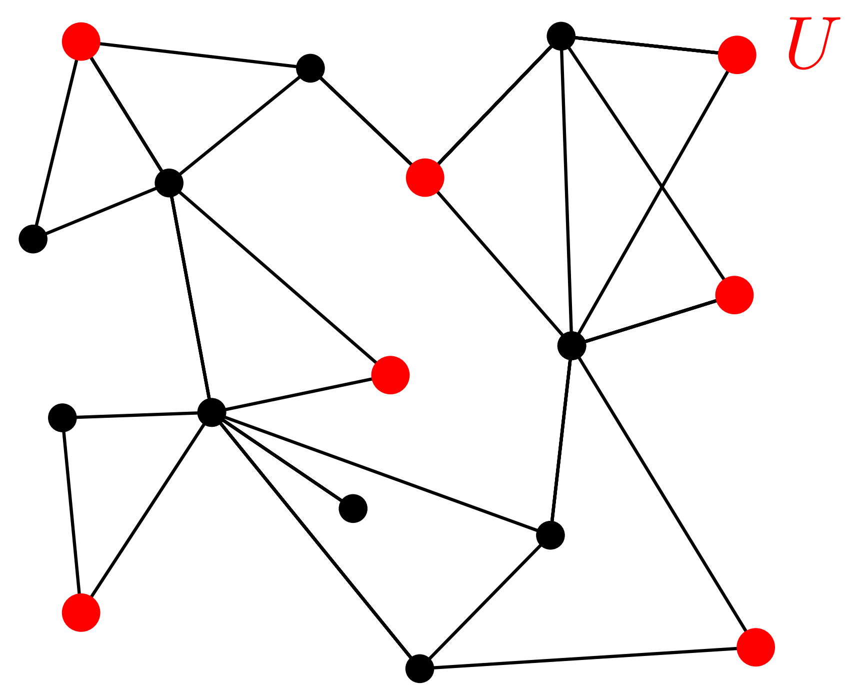
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



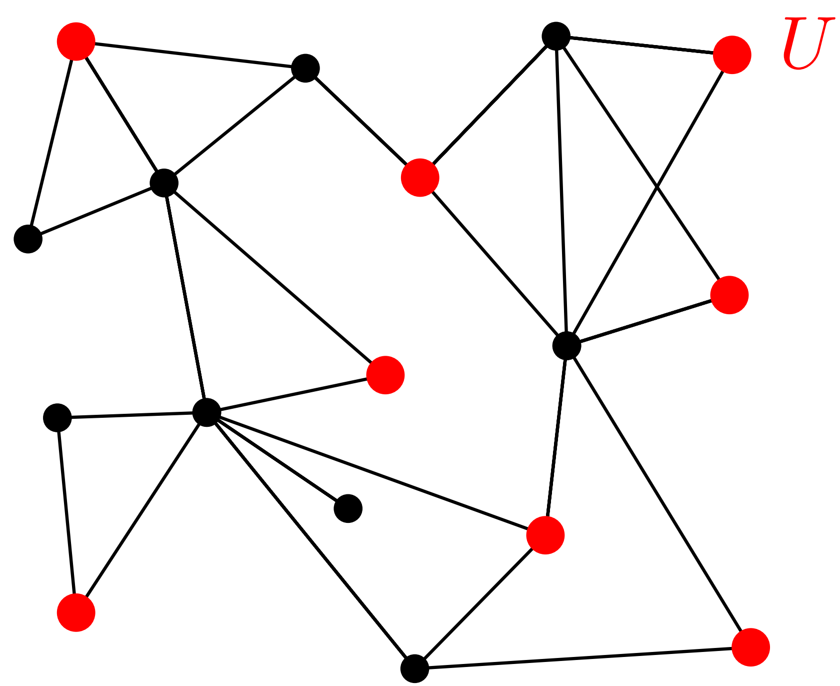
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



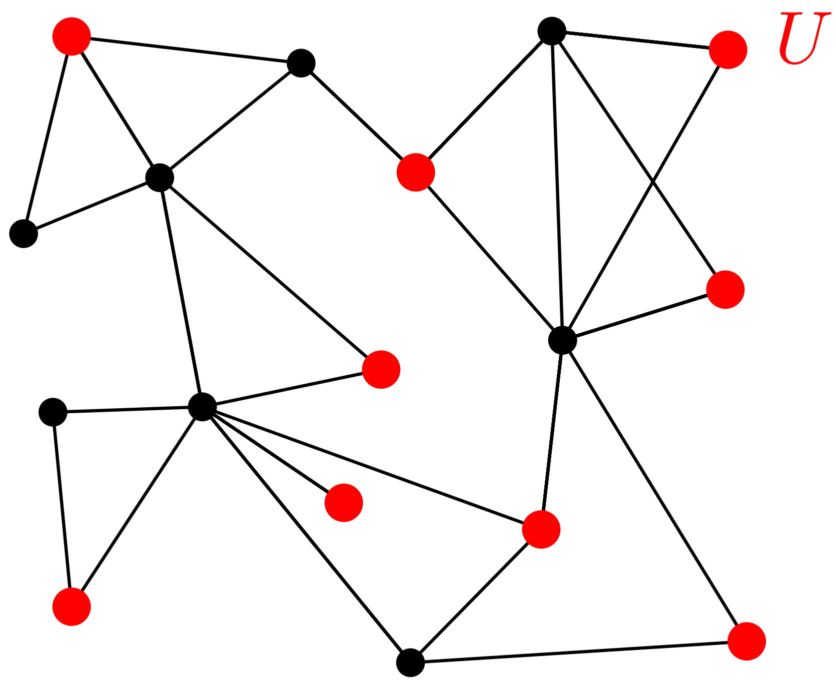
Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



Unabhängige Mengen

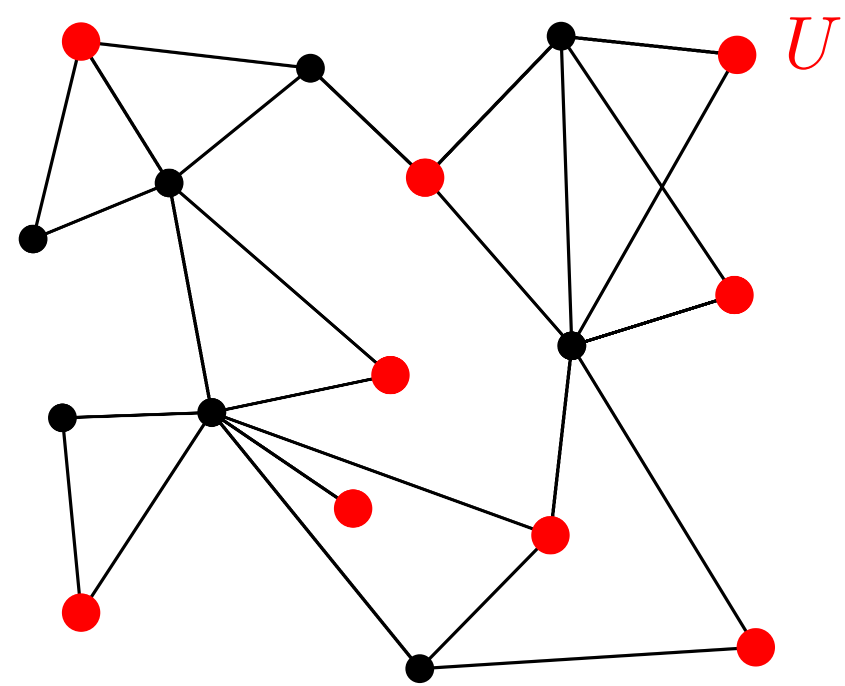
Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.



Unabhängige Mengen

Def. Eine Knotenmenge U in einem Graphen heißt **unabhängig**, wenn keine zwei Knoten aus U adjazent sind. Sie heißt darüberhinaus **nicht erweiterbare unabhängige Menge**, wenn es keine echte Obermenge von Knoten gibt, die auch unabhängig ist.

Nicht erweiterbare unabhängige Menge ist dominierend

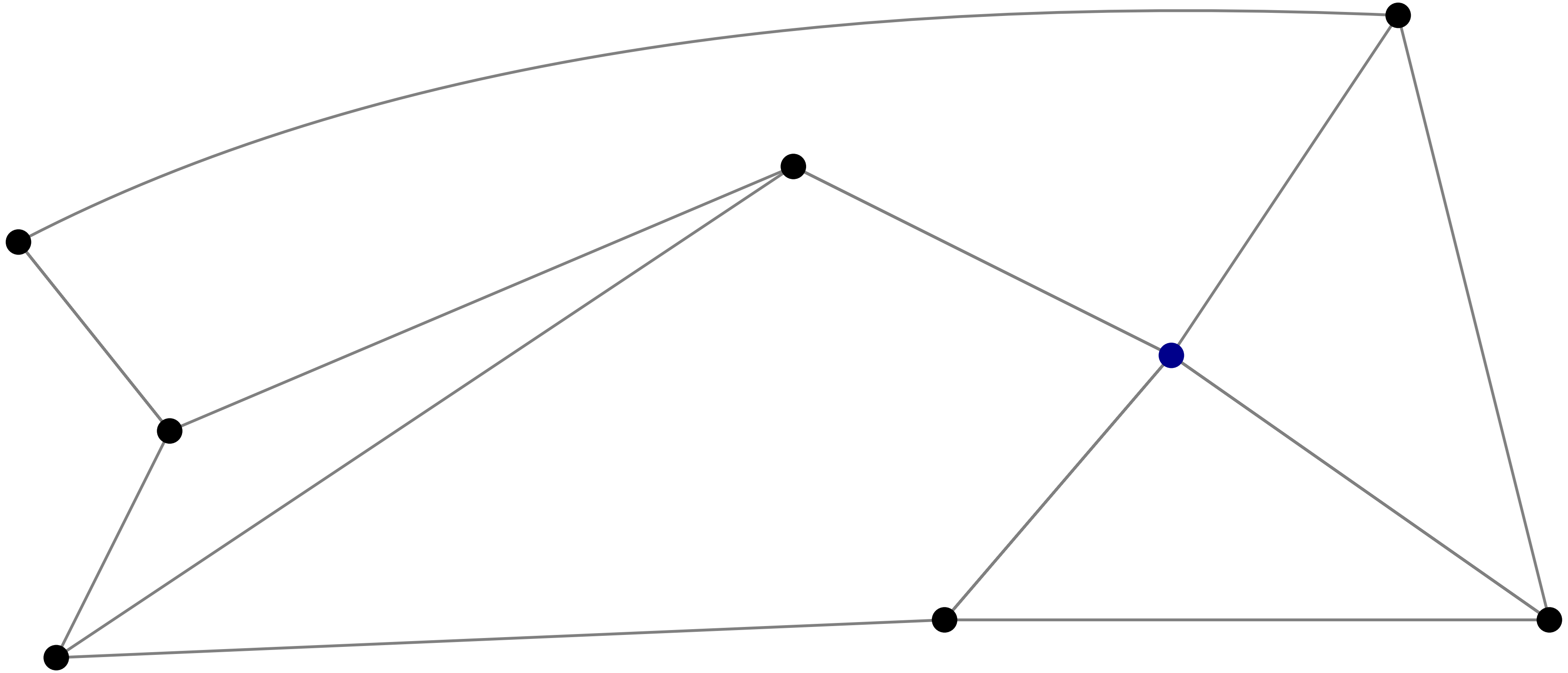


Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$

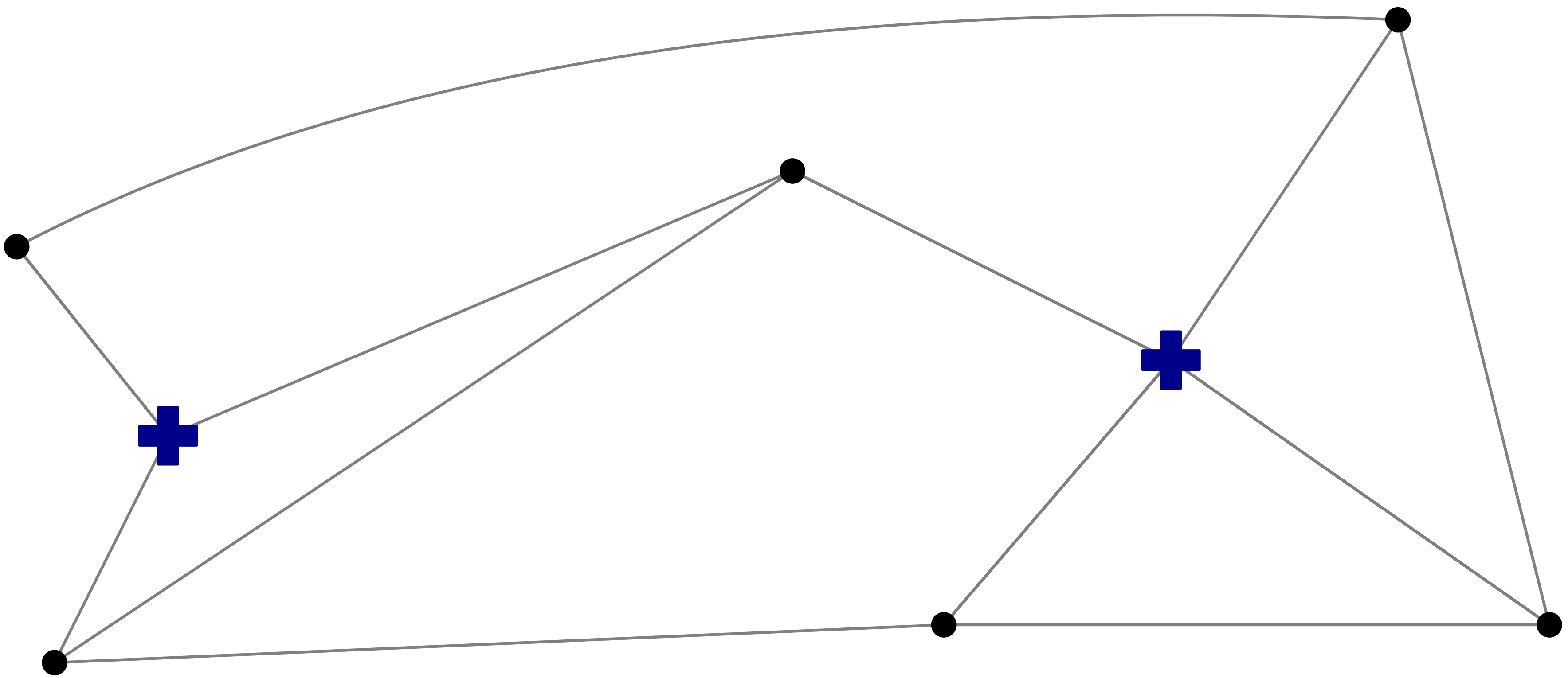
Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$



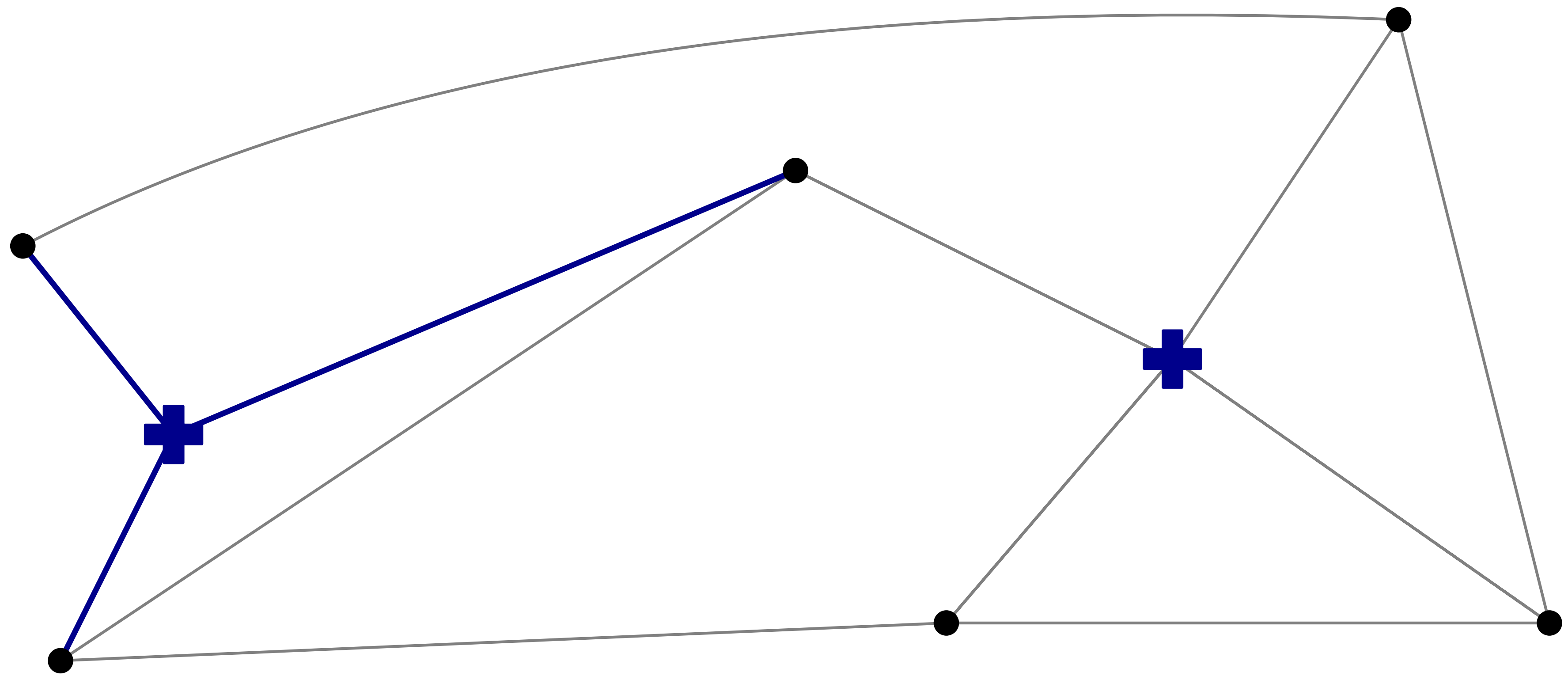
Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$



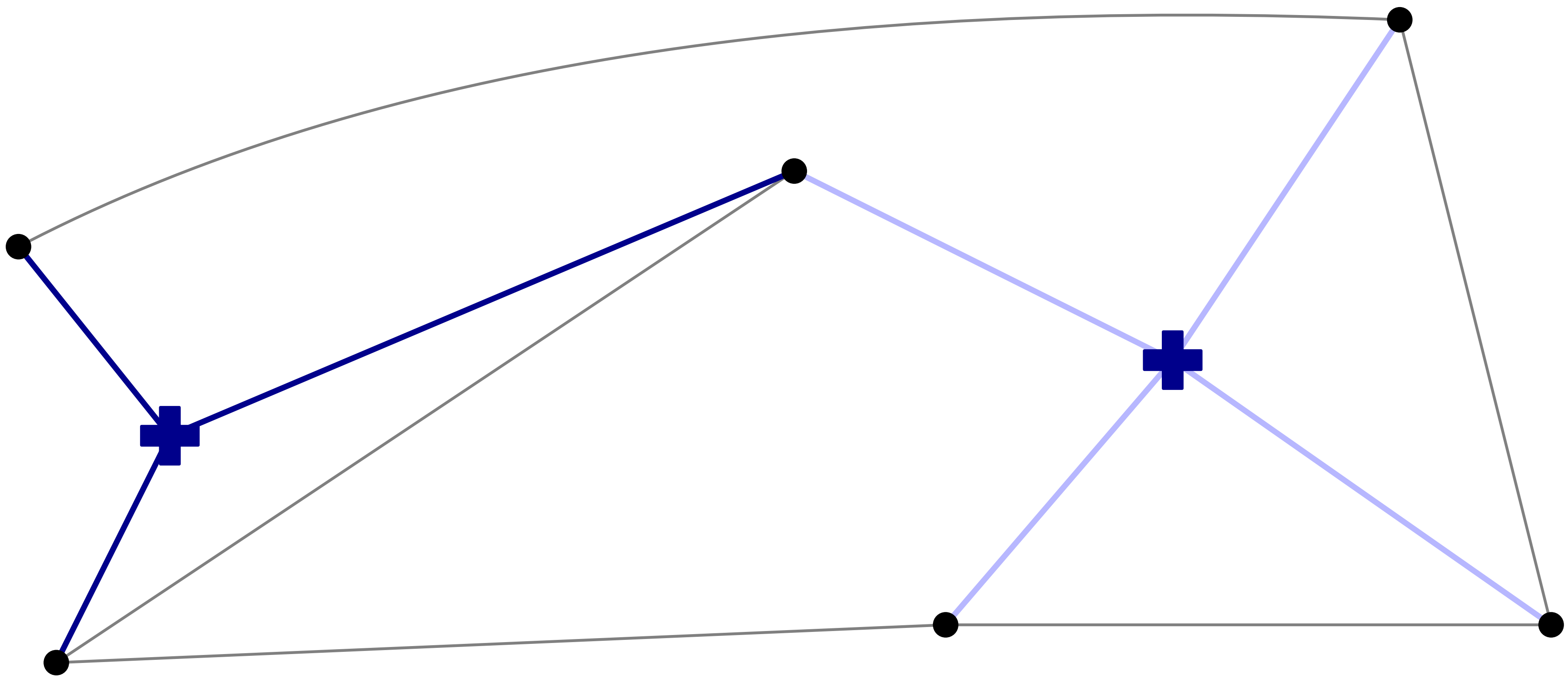
Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$



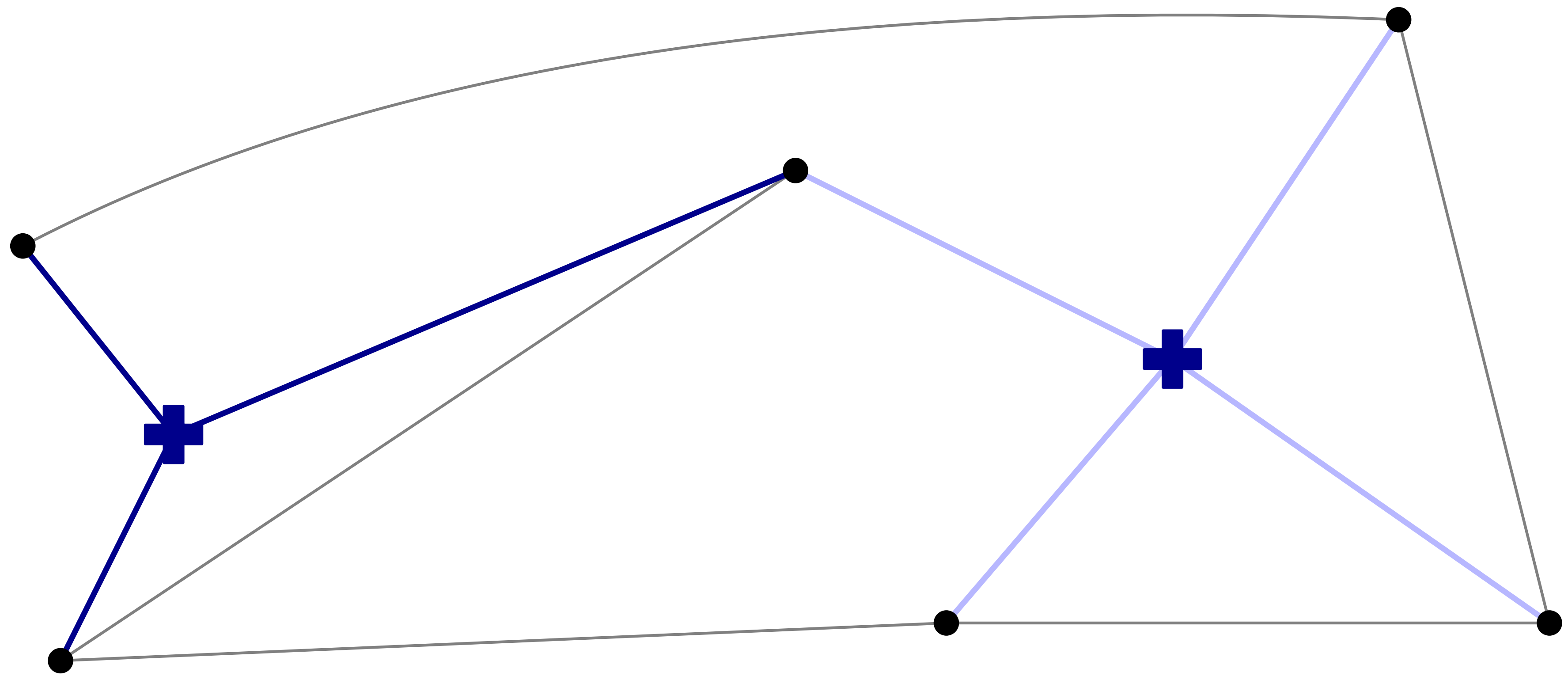
Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$



Unabhängige Mengen in H^2

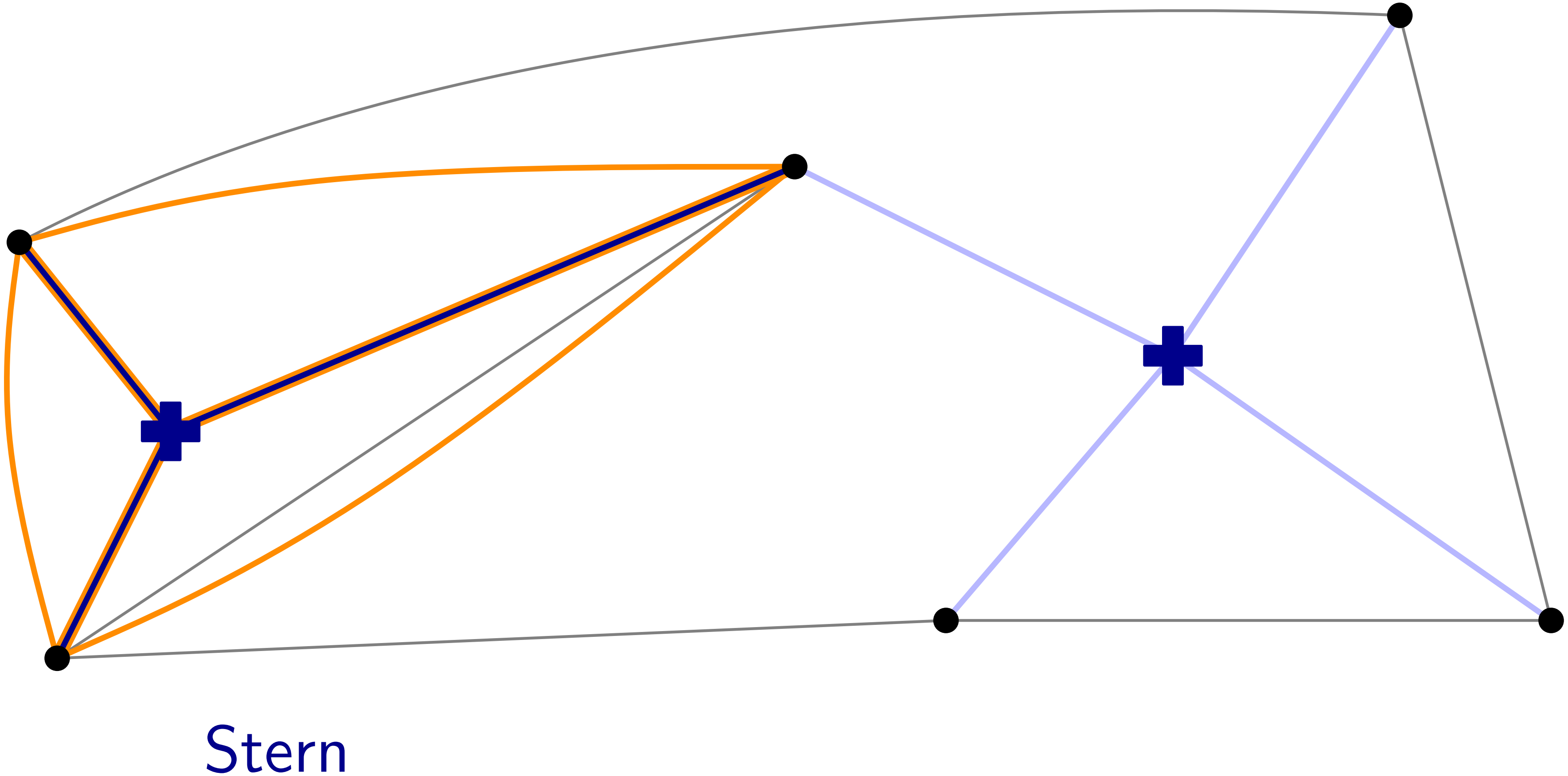
Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$



Stern

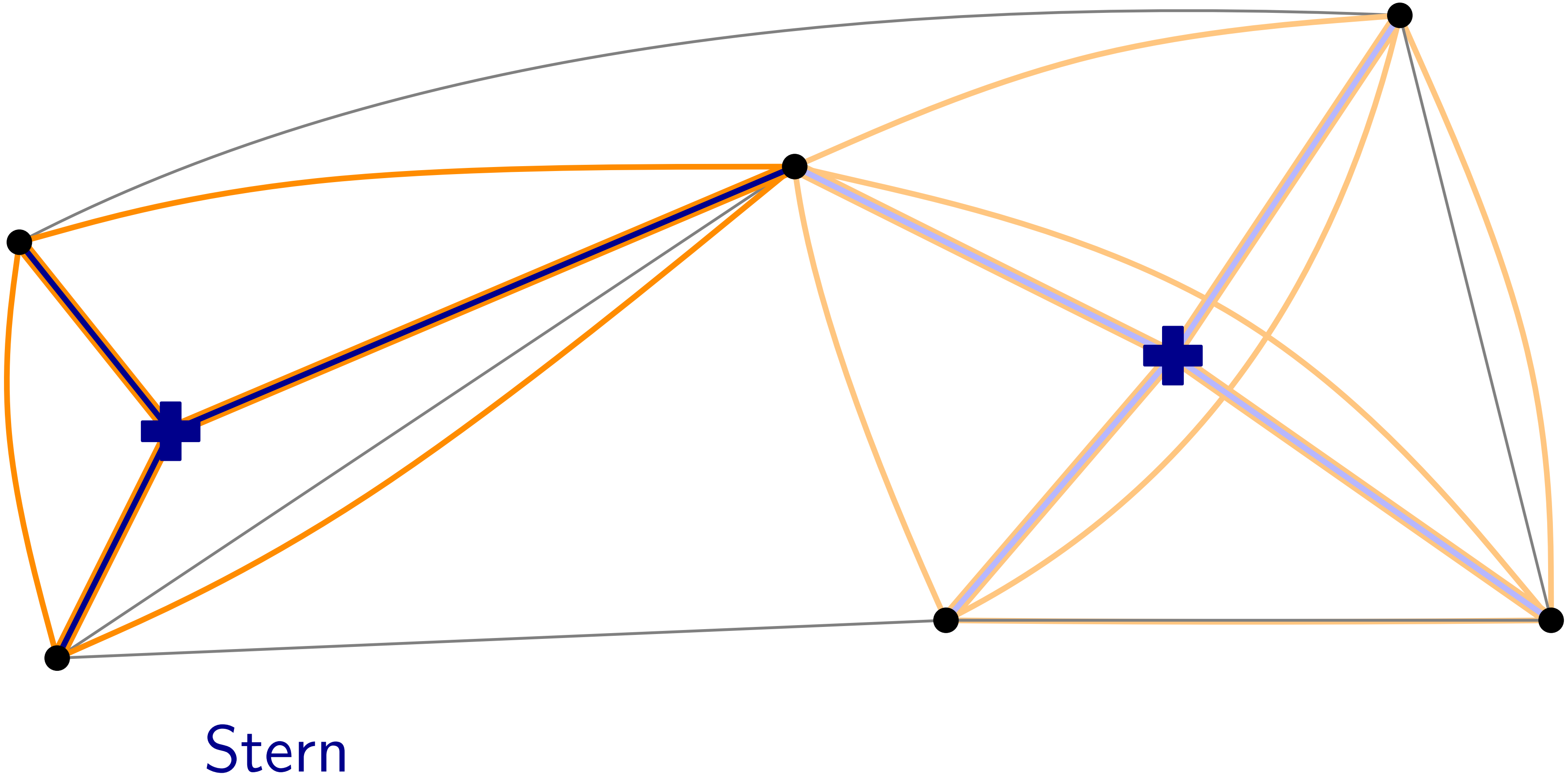
Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$



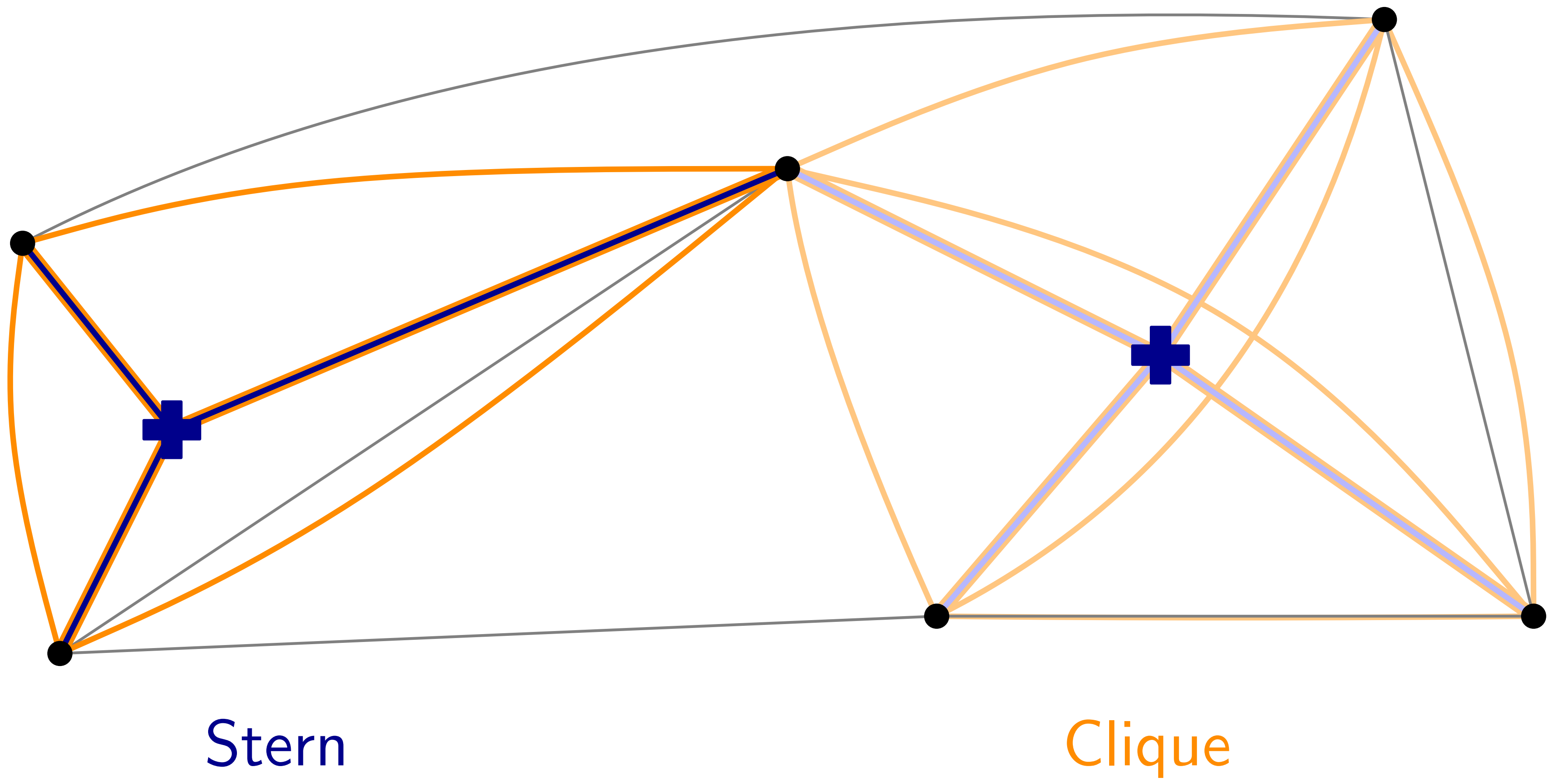
Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$



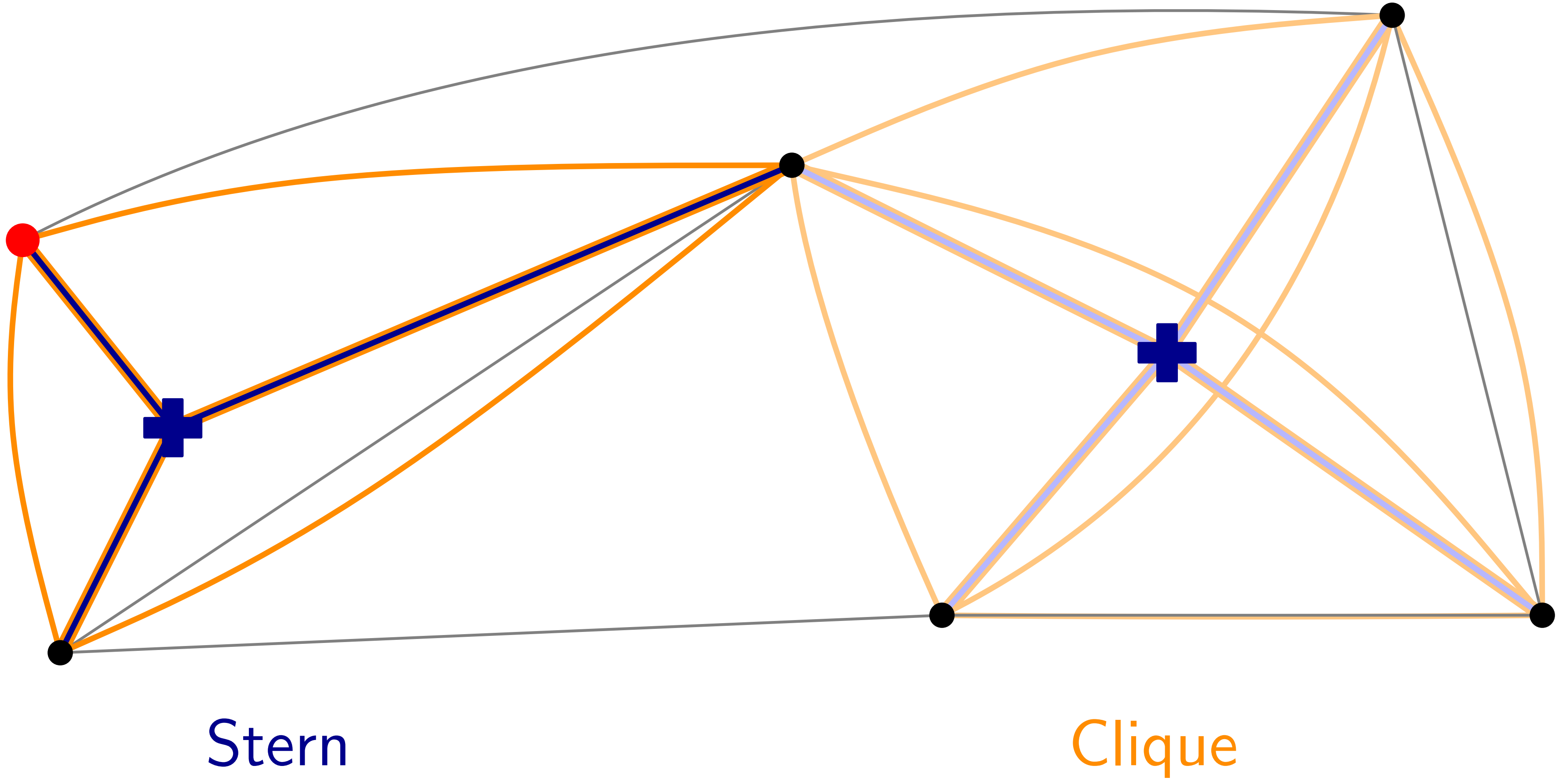
Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$



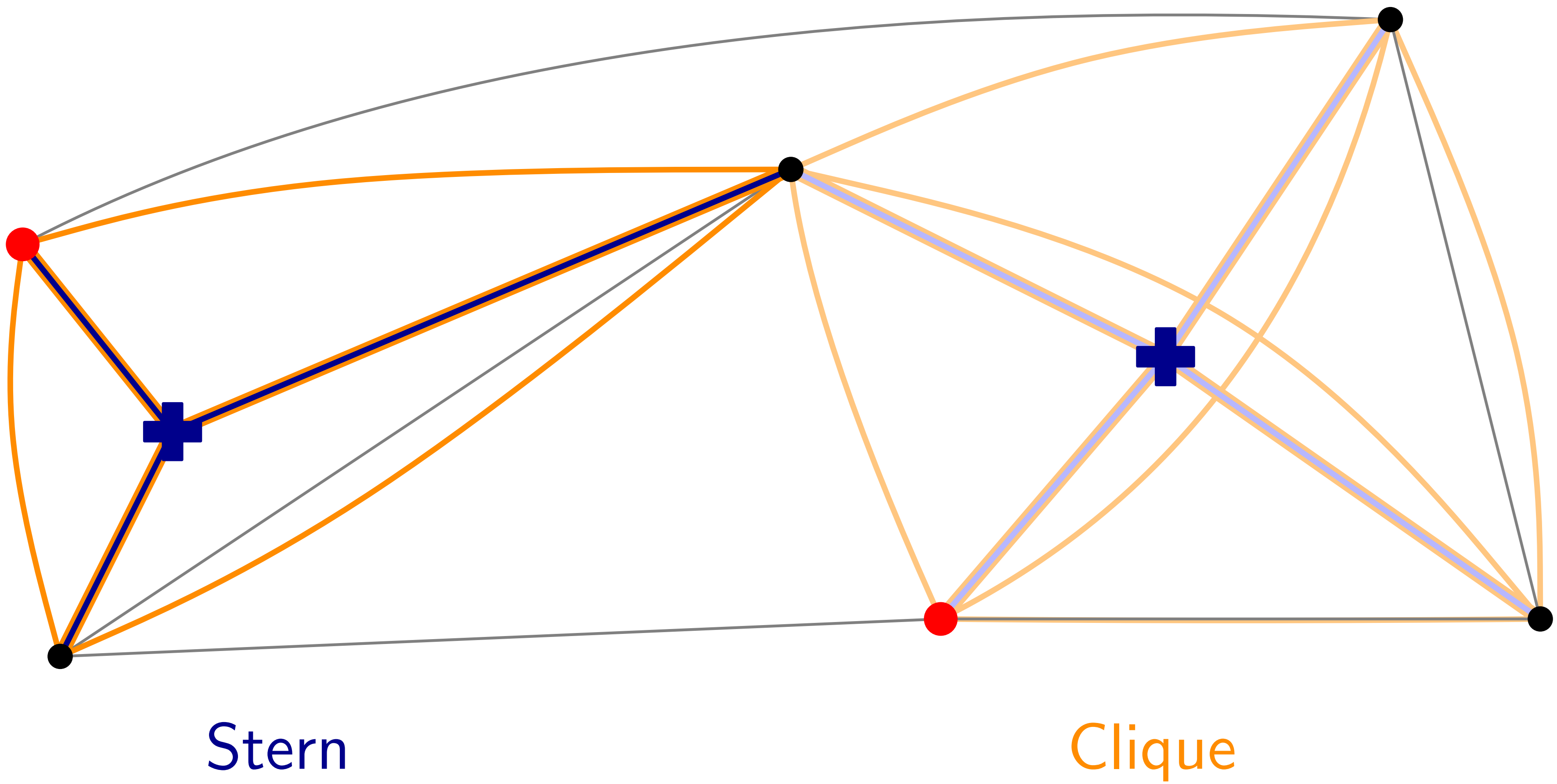
Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \le \mathrm{dom}(H)$



Unabhängige Mengen in H^2

Lemma. Sei H ein Graph und U eine unabhängige Menge in H^2. Dann gilt $|U| \leq \mathrm{dom}(H)$



Faktor 2 für metrisches k-Zentrum

Algorithmus Metrisches-k-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

Faktor 2 für metrisches k-Zentrum

Algorithmus Metrisches-k-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

Faktor 2 für metrisches k-Zentrum

Algorithmus Metrisches-k-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

Konstruiere G_j^2

Faktor 2 für metrisches k-Zentrum

Algorithmus Metrisches-k-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

Konstruiere G_j^2

Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

Faktor 2 für metrisches k-Zentrum

Algorithmus Metrisches-k-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

Konstruiere G_j^2

Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

if $|U_j| \leq k$ **then**

return U_j

Faktor 2 für metrisches k-Zentrum

Algorithmus Metrisches-k-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

Konstruiere G_j^2

Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

if $|U_j| \leq k$ **then**

return U_j

Lemma. Sei j wie am Ende des Algorithmus, dann gilt $c(e_j) \leq \text{OPT}$.

Faktor 2 für metrisches k-Zentrum

Algorithmus Metrisches-k-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

Konstruiere G_j^2

Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

if $|U_j| \leq k$ **then**

return U_j

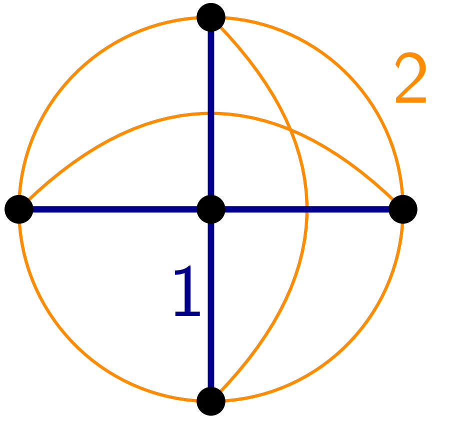
Lemma. Sei j wie am Ende des Algorithmus, dann gilt $c(e_j) \leq \mathrm{OPT}$.

Satz. Obiger Algorithmus ist ein Faktor-2-Approximationsalgorithmus für das metrische k-Zentrum-Problem.

Geht's besser?

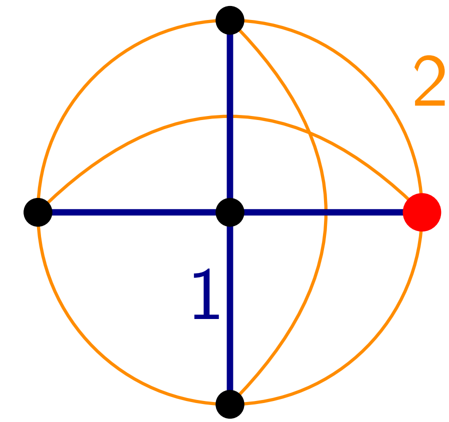
Analyse scharf?

Geht's besser?



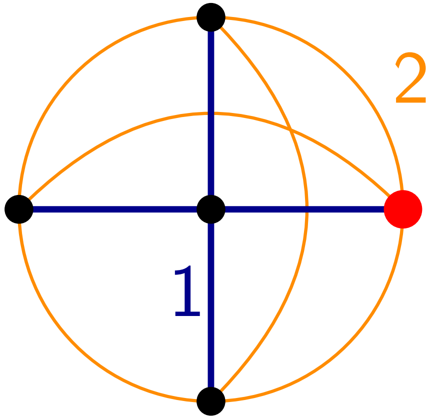
Analyse scharf?

Geht's besser?



Analyse scharf?

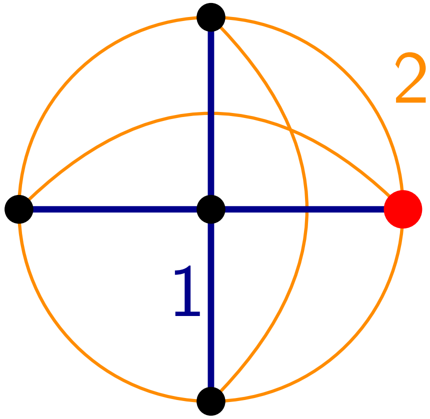
Geht's besser?



Analyse scharf?

Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-$\textsc{Zentrum}$-Problem, wobei $\epsilon > 0$.

Geht's besser?

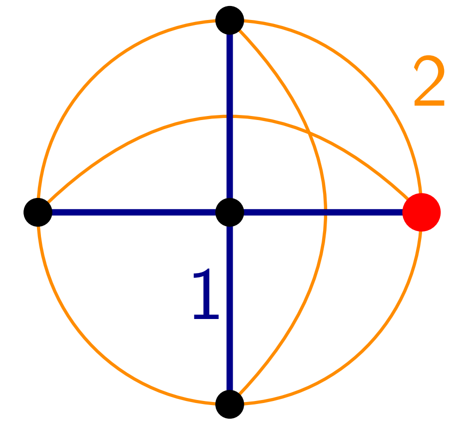


Analyse scharf?

Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-$\textsc{Zentrum}$-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.

Geht's besser?

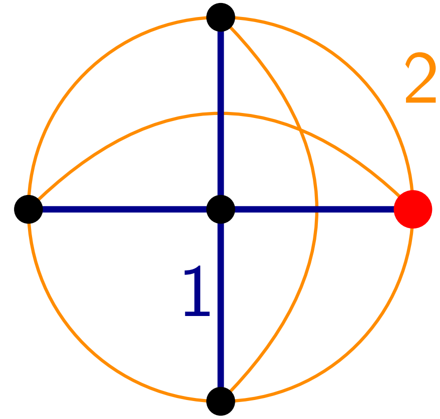


Analyse scharf?

Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-$\textsc{Zentrum}$-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k

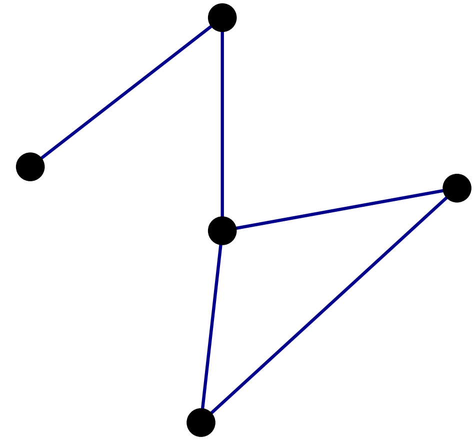
Geht's besser?



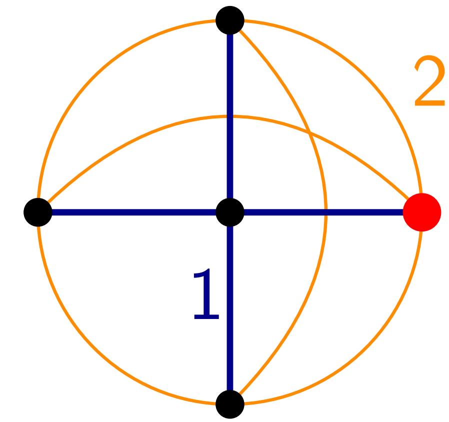
Analyse scharf?

Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-$\textsc{Zentrum}$-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k



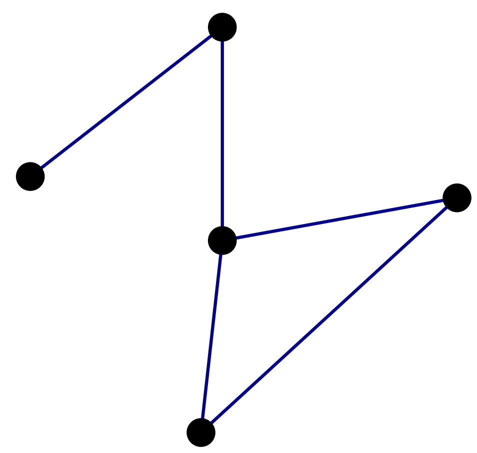
Geht's besser?



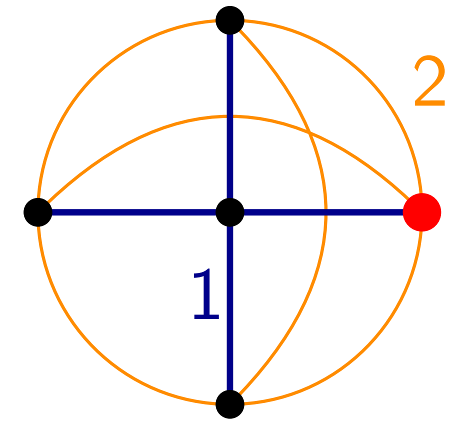
Analyse scharf?

Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-$\textsc{Zentrum}$-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k
Konstr. vollst. Graph $G' = (V, E \cup E')$



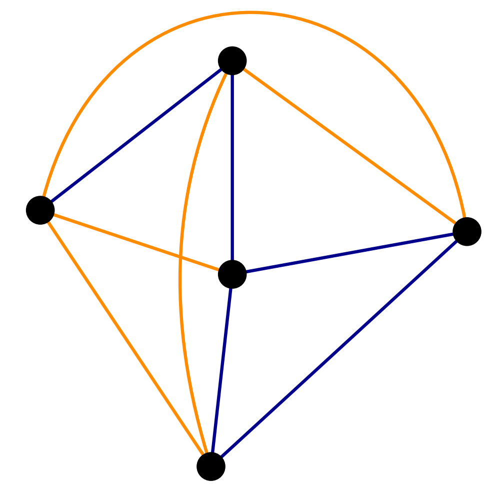
Geht's besser?



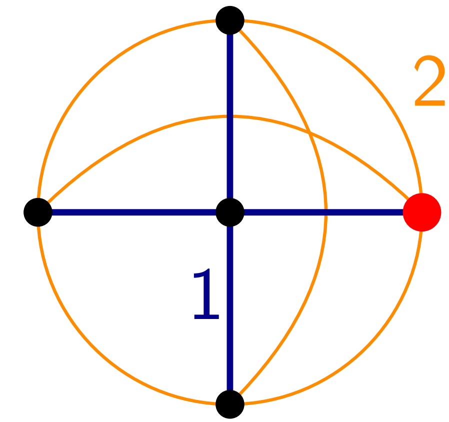
Analyse scharf?

Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-$\textsc{Zentrum}$-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k
Konstr. vollst. Graph $G' = (V, E \cup E')$



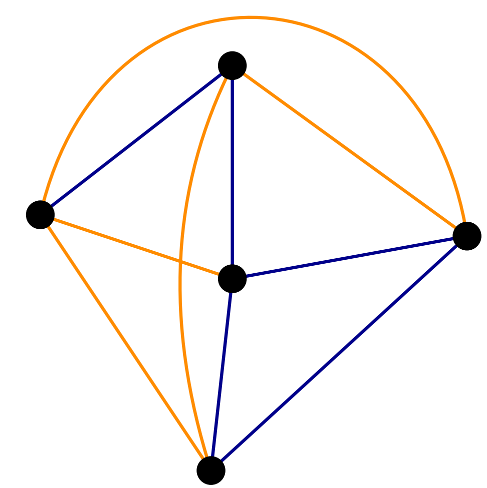
Geht's besser?



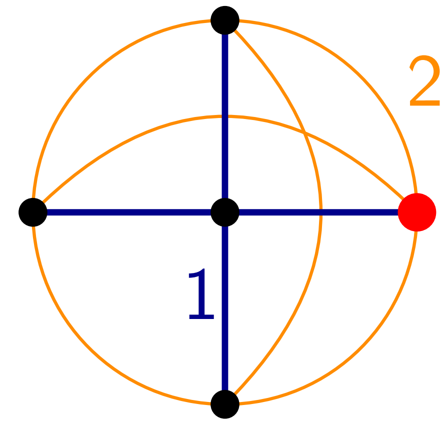
Analyse scharf?

Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-Zentrum-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k
Konstr. vollst. Graph $G' = (V, E \cup E')$
mit $c(e) = \begin{cases} 1, & \text{falls } e \in E \\ 2, & \text{falls } e \in E' \end{cases}$



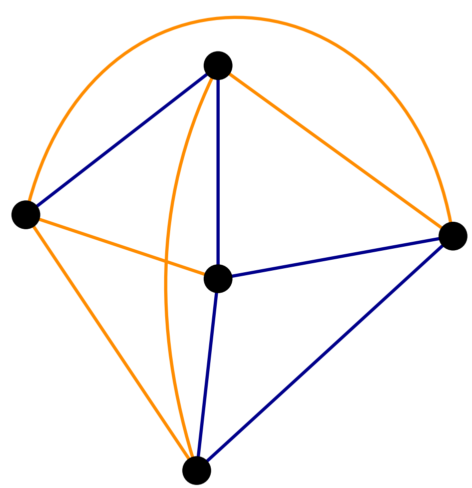
Geht's besser?



Analyse scharf?

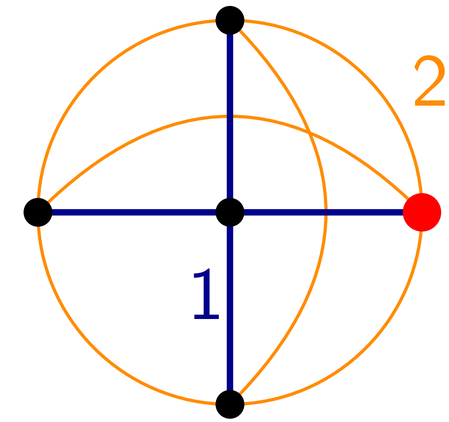
Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-ZENTRUM-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k
Konstr. vollst. Graph $G' = (V, E \cup E')$
mit $c(e) = \begin{cases} 1, & \text{falls } e \in E \\ 2, & \text{falls } e \in E' \end{cases}$



S: metrisches k-Zentrum
Falls dom$(G) \leq k$, dann cost$(S) = 1$

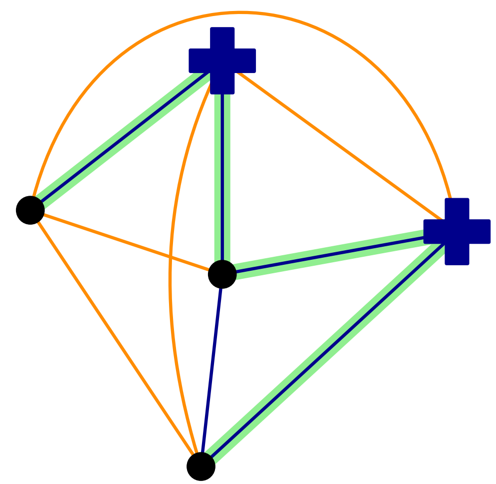
Geht's besser?



Analyse scharf?

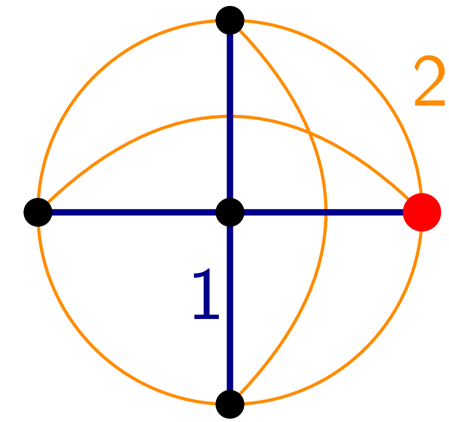
Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-$\textsc{Zentrum}$-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k
Konstr. vollst. Graph $G' = (V, E \cup E')$
mit $c(e) = \begin{cases} 1, & \text{falls } e \in E \\ 2, & \text{falls } e \in E' \end{cases}$



S: metrisches k-Zentrum
Falls $\mathrm{dom}(G) \leq k$, dann $\mathrm{cost}(S) = 1$

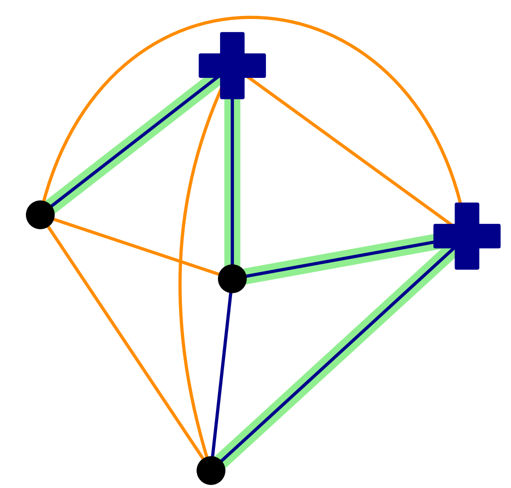
Geht's besser?



Analyse scharf?

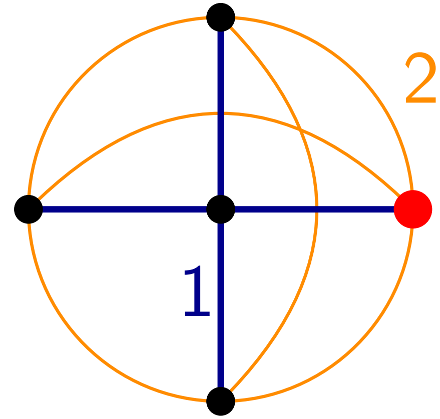
Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-Zentrum-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k
Konstr. vollst. Graph $G' = (V, E \cup E')$
mit $c(e) = \begin{cases} 1, & \text{falls } e \in E \\ 2, & \text{falls } e \in E' \end{cases}$



S: metrisches k-Zentrum
Falls dom$(G) \leq k$, dann cost$(S) = 1$
Falls dom$(G) > k$, dann cost$(S) = 2$

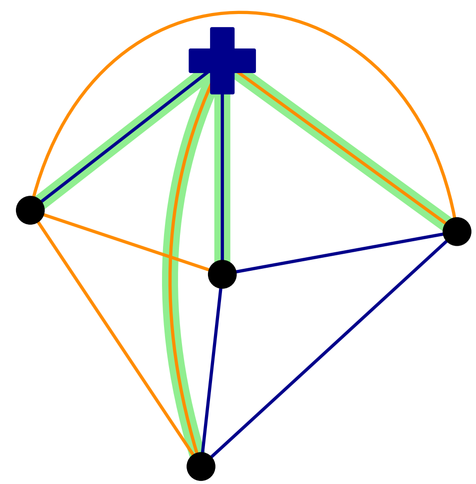
Geht's besser?



Analyse scharf?

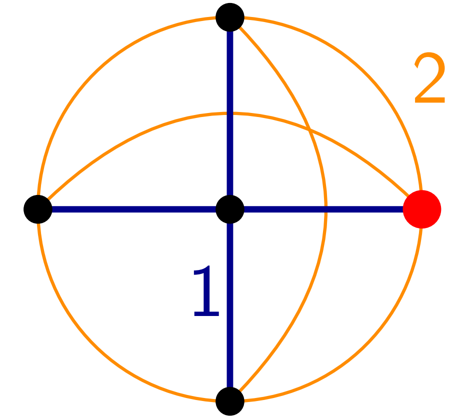
Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-$\textsc{Zentrum}$-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k
Konstr. vollst. Graph $G' = (V, E \cup E')$
mit $c(e) = \begin{cases} 1, & \text{falls } e \in E \\ 2, & \text{falls } e \in E' \end{cases}$



S: metrisches k-Zentrum
Falls $\mathrm{dom}(G) \leq k$, dann $\mathrm{cost}(S) = 1$
Falls $\mathrm{dom}(G) > k$, dann $\mathrm{cost}(S) = 2$

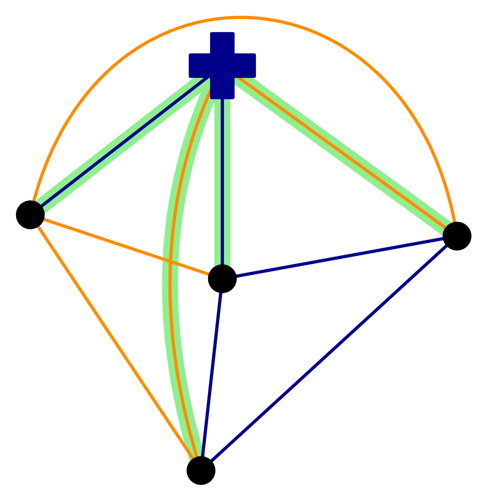
Geht's besser?



Analyse scharf?

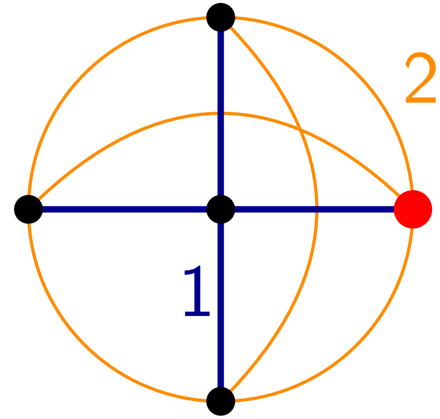
Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-ZENTRUM-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k
Konstr. vollst. Graph $G' = (V, E \cup E')$
mit $c(e) = \begin{cases} 1, & \text{falls } e \in E \\ 2, & \text{falls } e \in E' \end{cases}$



S: metrisches k-Zentrum
Falls dom$(G) \leq k$, dann cost$(S) = 1$
Falls dom$(G) > k$, dann cost$(S) = 2$

Geht's besser?

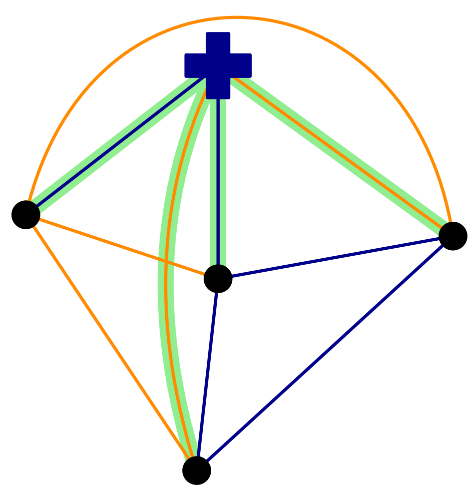


Analyse scharf?

Satz. Unter der Annahme P $\neq$ NP gibt es keinen Faktor-$(2-\epsilon)$-Approximationsalgorithmus für das metrische k-$\textsc{Zentrum}$-Problem, wobei $\epsilon > 0$.

Beweis. Reduktion von dominierende Menge.
Geg.: $G = (V, E)$, k
Konstr. vollst. Graph $G' = (V, E \cup E')$
mit $c(e) = \begin{cases} 1, & \text{falls } e \in E \\ 2, & \text{falls } e \in E' \end{cases}$

$\triangle$-Ungleichung erfüllt



S: metrisches k-Zentrum
Falls $\mathrm{dom}(G) \leq k$, dann $\mathrm{cost}(S) = 1$
Falls $\mathrm{dom}(G) > k$, dann $\mathrm{cost}(S) = 2$

Metrisches k-ZENTRUM-Problem

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$ und eine natürliche Zahl $k \leq |V|$.

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten der günstigsten Kante von v zu einem Knoten aus S.

Gesucht: Eine k-elementige Knotenmenge S, so dass $\mathrm{cost}(S) := \max_{v \in V} c(v, S)$ minimal ist.

Metrisches $\cancel{k}$-Zentrum-Problem

gewichtetes

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$ und eine natürliche Zahl $k \leq |V|$.

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten der günstigsten Kante von v zu einem Knoten aus S.

Gesucht: Eine k-elementige Knotenmenge S, so dass cost$(S) := \max_{v \in V} c(v, S)$ minimal ist.

Metrisches ~~k~~-Zentrum-Problem

gewichtetes

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$ ~~und eine natürliche Zahl $k \leq |V|$~~., Knotengewichte $w\colon V \to \mathbb{Q}_{\geq 0}$ und eine Gewichtsschranke $W \in \mathbb{Q}_+$

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten der günstigsten Kante von v zu einem Knoten aus S.

Gesucht: Eine k-elementige Knotenmenge S, so dass $\text{cost}(S) := \max_{v \in V} c(v, S)$ minimal ist.

Metrisches ~~k~~-Zentrum-Problem

gewichtetes

Gegeben: Ein vollständiger Graph $G = (V, E)$ mit Metrik $c\colon E \to \mathbb{Q}_{\geq 0}$ ~~und eine natürliche Zahl $k \leq |V|$.~~, Knotengewichte $w\colon V \to \mathbb{Q}_{\geq 0}$ und eine Gewichtsschranke $W \in \mathbb{Q}_+$

Für jede Knotenmenge $S \subseteq V$ bezeichne $c(v, S)$ die Kosten der günstigsten Kante von v zu einem Knoten aus S.

Knotenmenge S mit Gewicht höchstens W

Gesucht: Eine ~~k elementige Knotenmenge S~~, so dass $\text{cost}(S) := \max_{v \in V} c(v, S)$ minimal ist.

Die gewichtete Version

Algorithmus Metrisches-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

Konstruiere G_j^2

Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

if $|U_j| \leq k$ **then**

return U_j

Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-$\textsc{Zentrum}$

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

  Konstruiere G_j^2

  Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

  if $|U_j| \leq k$ **then**

    return U_j

Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-$\textsc{Zentrum}$

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

Konstruiere G_j^2

Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

if $|U_j| \leq k$ **then**

return U_j

könnte sehr großes Gewicht haben

Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

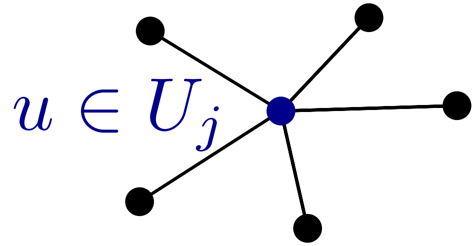
  Konstruiere G_j^2

  Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

  if $|U_j| \leq k$ **then**

    return U_j

könnte sehr großes Gewicht haben



Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

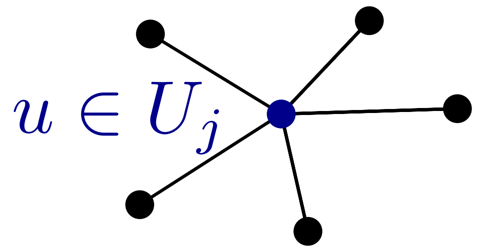
  Konstruiere G_j^2

  Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

  if $|U_j| \leq k$ **then**

    return U_j

könnte sehr großes Gewicht haben



$s_j(u) :=$ leichtester Knoten aus $N_{G_j}(u) \cup \{u\}$

Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

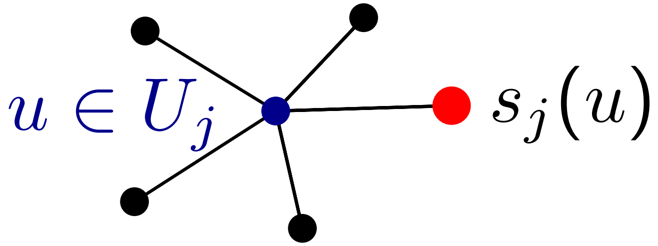
Konstruiere G_j^2

Finde nicht erweiterb. unabhängige Menge U_j in G_j^2

könnte sehr großes Gewicht haben

if $|U_j| \leq k$ **then**

return U_j



$s_j(u) :=$ leichtester Knoten aus $N_{G_j}(u) \cup \{u\}$

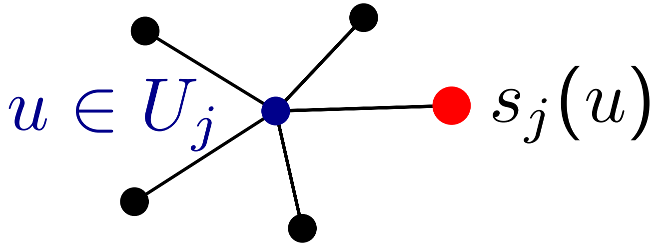
Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-$\textsc{Zentrum}$

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

- Konstruiere G_j^2
- Finde nicht erweiterb. unabhängige Menge U_j in G_j^2
- Berechne $S_j := \{ s_j(u) \mid u \in U_j \}$
- **if** $|U_j| \leq k$ **then**
  - **return** U_j



$s_j(u) :=$ leichtester Knoten aus $N_{G_j}(u) \cup \{u\}$

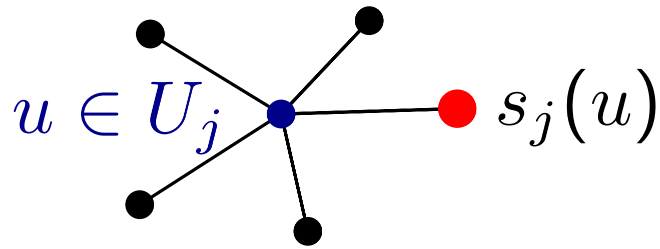
Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

- Konstruiere G_j^2
- Finde nicht erweiterb. unabhängige Menge U_j in G_j^2
- Berechne $S_j := \{\, s_j(u) \mid u \in U_j \,\}$
- **if** $\cancel{|U_j| \leq k}$ $w(S_j) \leq W$ **then**
  - **return** $\cancel{U_j}$ S_j



$s_j(u) :=$ leichtester Knoten aus $N_{G_j}(u) \cup \{u\}$

Satz. Obiger Algorithmus liefert eine 3-Approximation für das gewichtete metrische Zentrum-Problem.

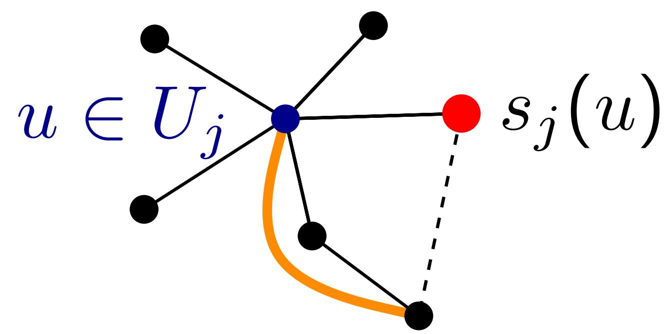
Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \le \ldots \le c(e_m)$

for $j = 1, \ldots, m$ **do**

- Konstruiere G_j^2
- Finde nicht erweiterb. unabhängige Menge U_j in G_j^2
- Berechne $S_j := \{ s_j(u) \mid u \in U_j \}$
- **if** $\cancel{|U_j| \le k}$ **then** $w(S_j) \le W$
  - **return** $\cancel{U_j}$ S_j



$s_j(u) :=$ leichtester Knoten aus $N_{G_j}(u) \cup \{u\}$

Satz. Obiger Algorithmus liefert eine 3-Approximation für das gewichtete metrische Zentrum-Problem.

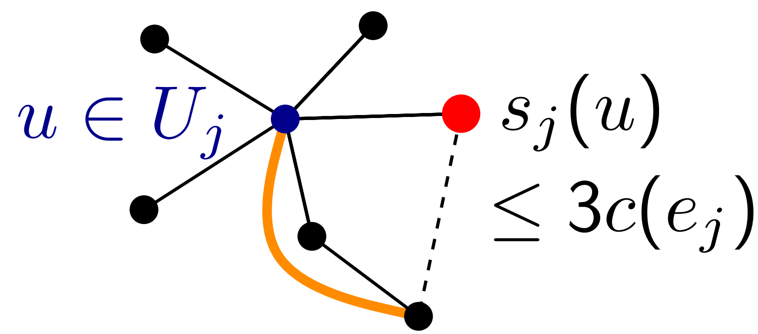
Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \leq \ldots \leq c(e_m)$

for $j = 1, \ldots, m$ **do**

- Konstruiere G_j^2
- Finde nicht erweiterb. unabhängige Menge U_j in G_j^2
- Berechne $S_j := \{\, s_j(u) \mid u \in U_j \,\}$
- **if** $\cancel{|U_j| \leq k}$ $w(S_j) \leq W$ **then**
  - **return** $\cancel{U_j}$ S_j



$s_j(u) :=$ leichtester Knoten aus $N_{G_j}(u) \cup \{u\}$

Satz. Obiger Algorithmus liefert eine 3-Approximation für das gewichtete metrische Zentrum-Problem.

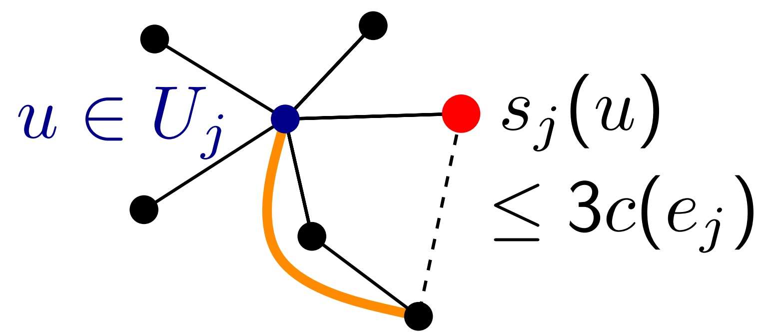
Die gewichtete Version

Algorithmus Metrisches-Gewichtetes-Zentrum

Sortiere die Kanten von G gemäß $c(e_1) \le \ldots \le c(e_m)$

for $j = 1, \ldots, m$ **do**

- Konstruiere G_j^2
- Finde nicht erweiterb. unabhängige Menge U_j in G_j^2
- Berechne $S_j := \{\, s_j(u) \mid u \in U_j \,\}$
- **if** $\cancel{|U_j| \le k}$ $w(S_j) \le W$ **then**
  - **return** $\cancel{U_j}$ S_j



$s_j(u) :=$ leichtester Knoten aus $N_{G_j}(u) \cup \{u\}$

Satz. Obiger Algorithmus liefert eine 3-Approximation für das gewichtete metrische Zentrum-Problem.