Approximation Algorithms

Lecture 6:
k-Center via Parametric Pruning

Part I:
Metric k-Center

Alexander Wolff

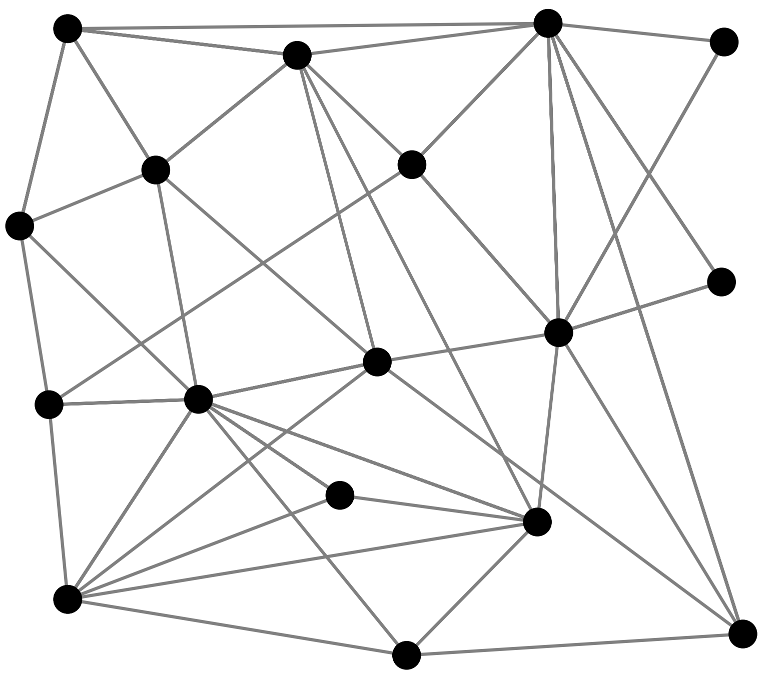
Winter term 2025

Metric k-Center

Given: A graph G

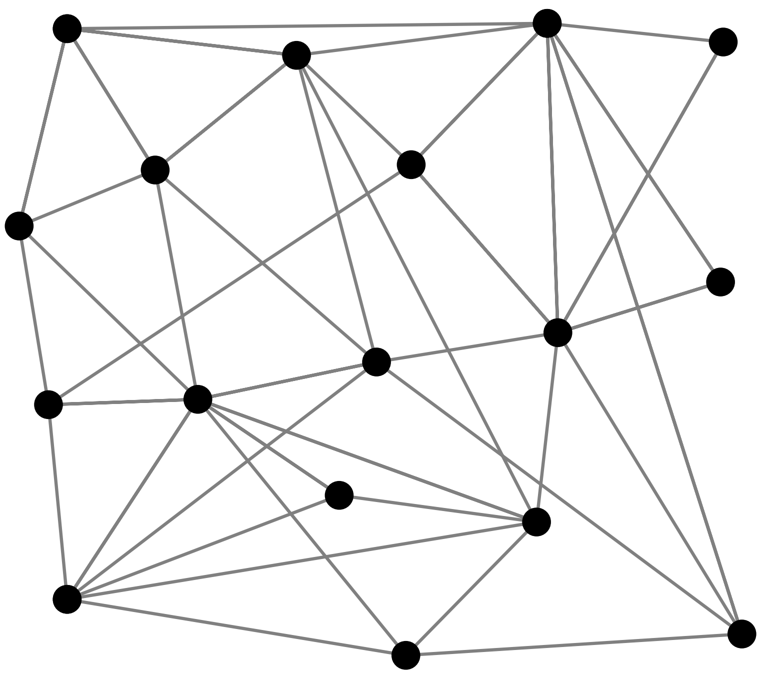
Metric k-Center

Given: A graph G



Metric k-Center

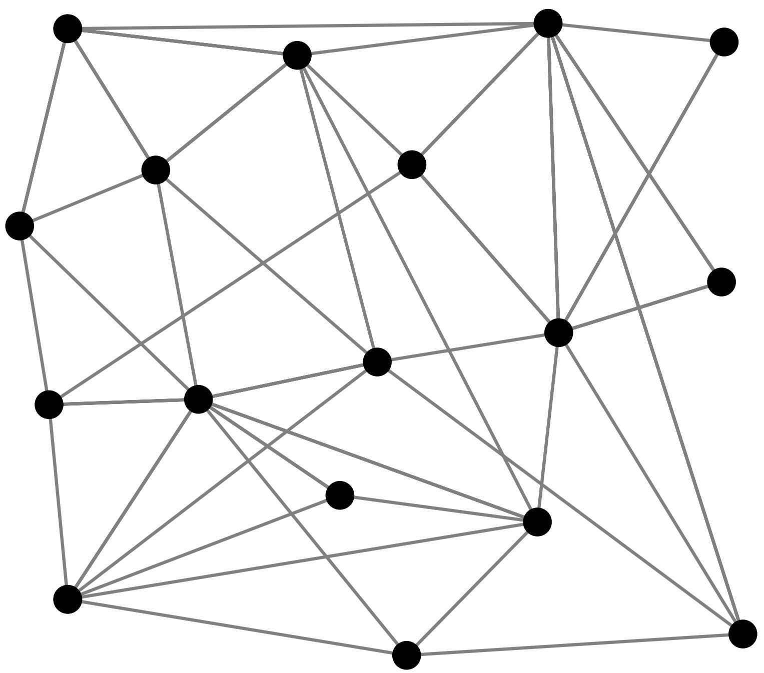
Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality



Metric k-CENTER

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

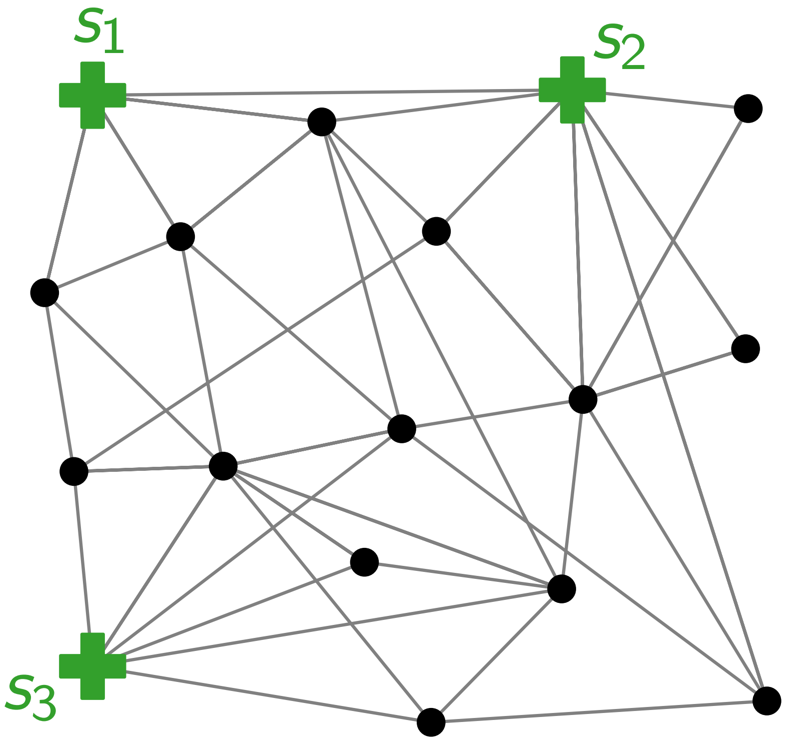
a set $S \subseteq V(G)$



Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

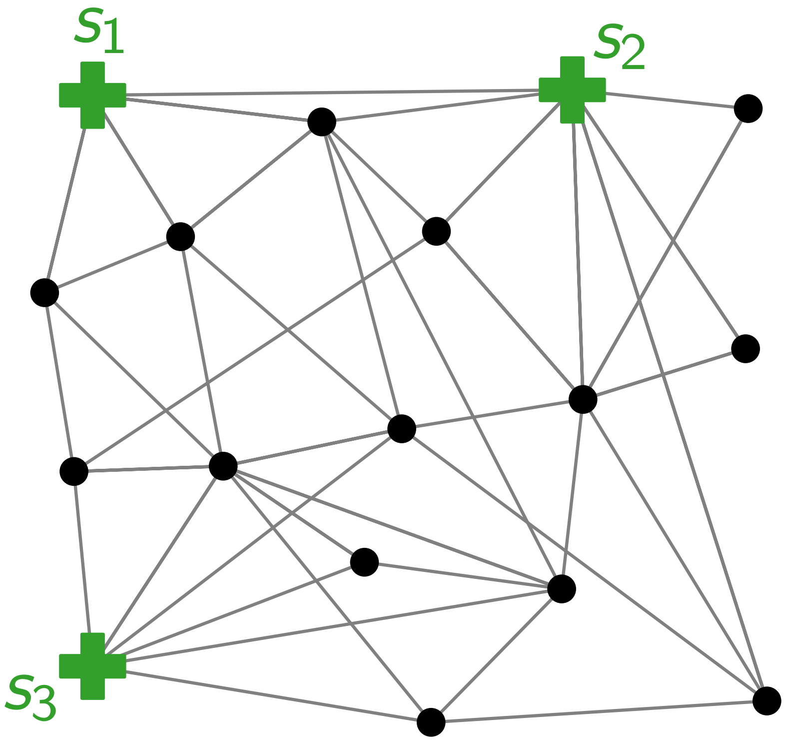
a set $S \subseteq V(G)$



Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

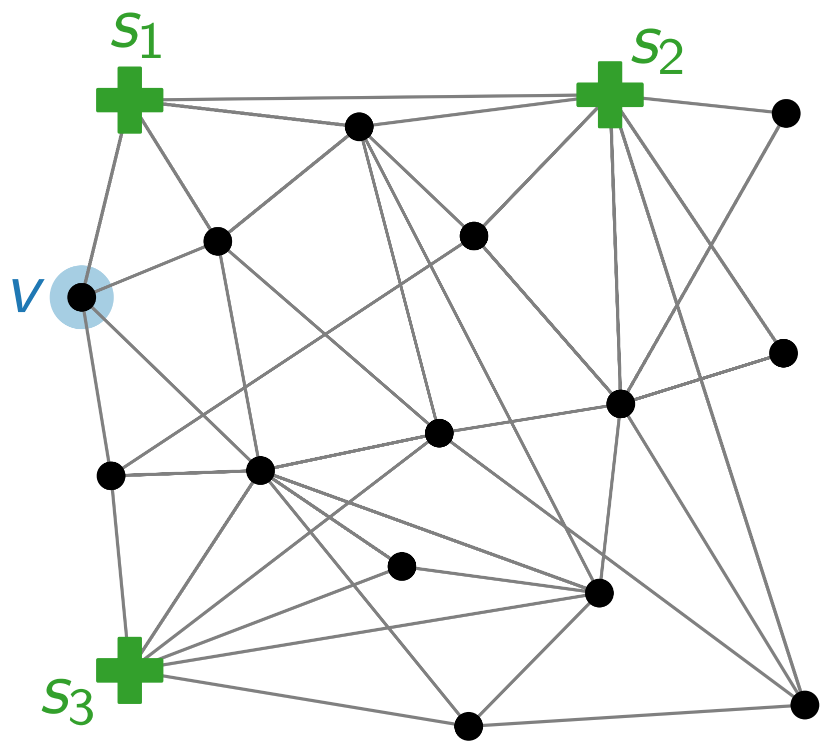
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c \colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

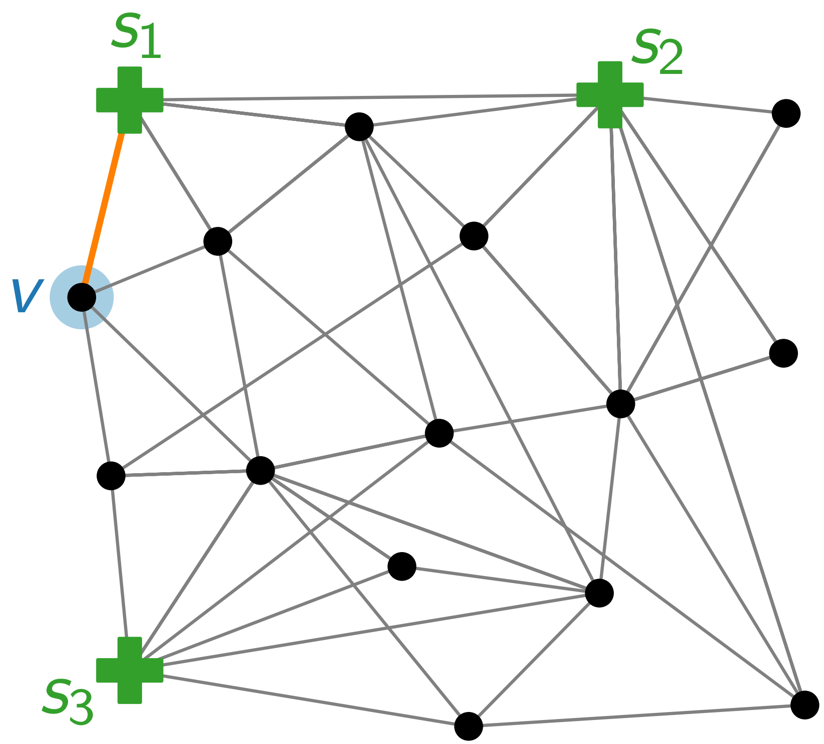
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

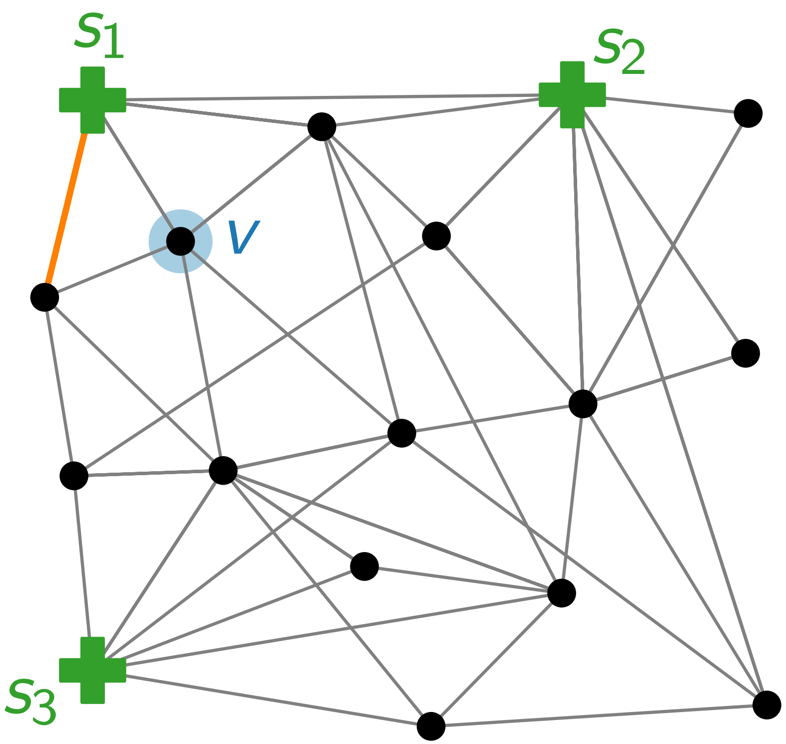
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c \colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

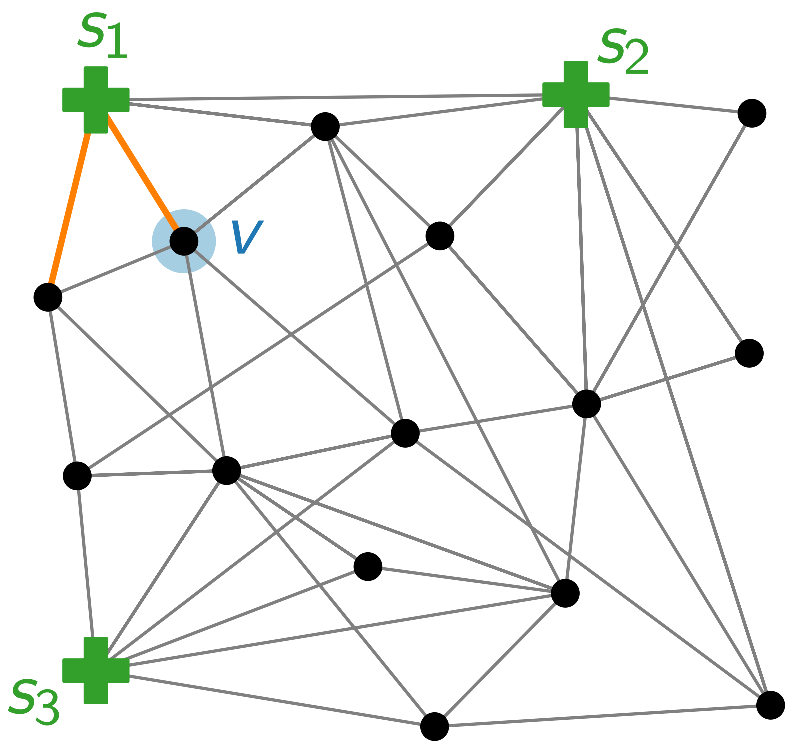
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

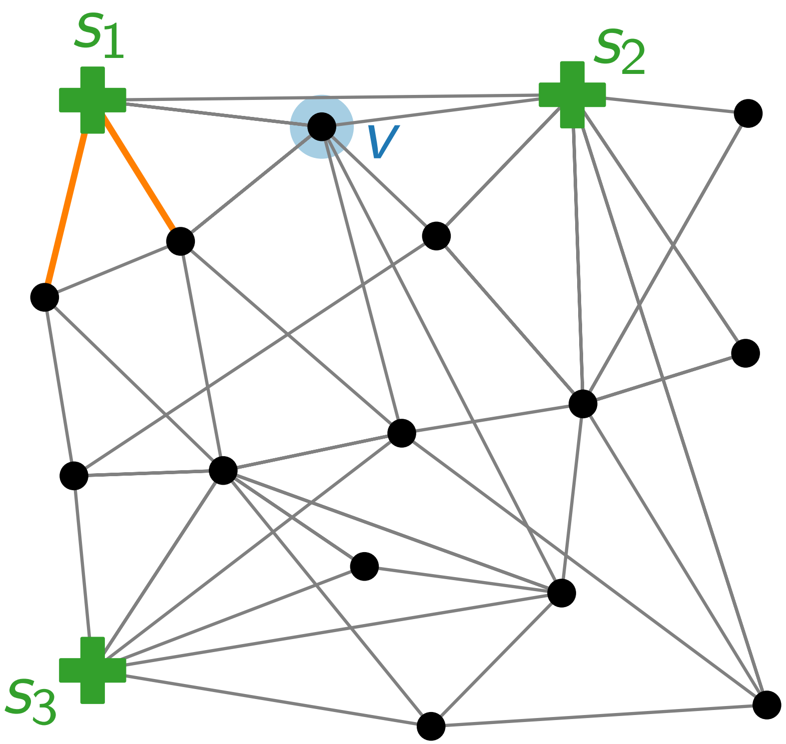
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c \colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

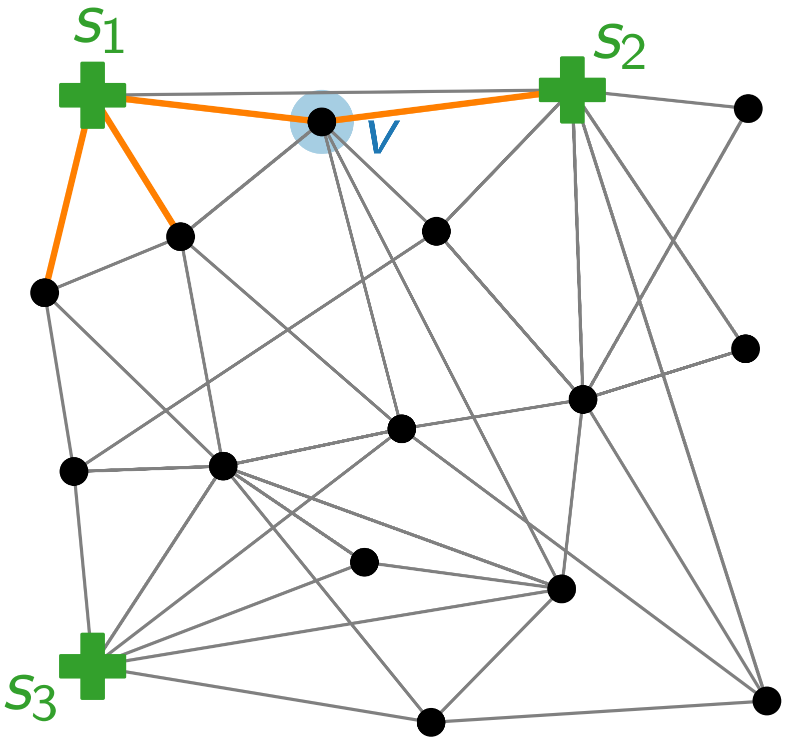
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

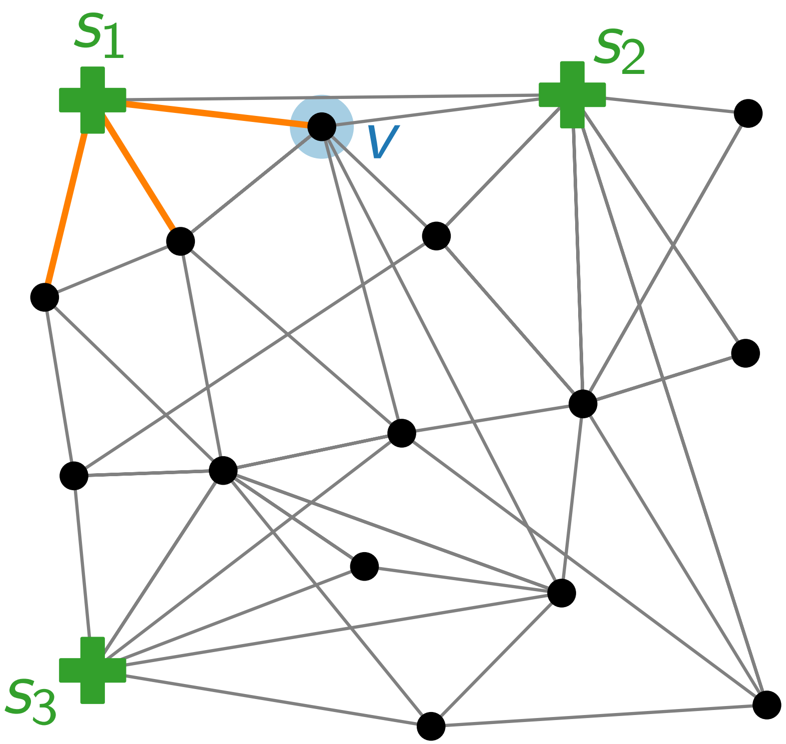
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

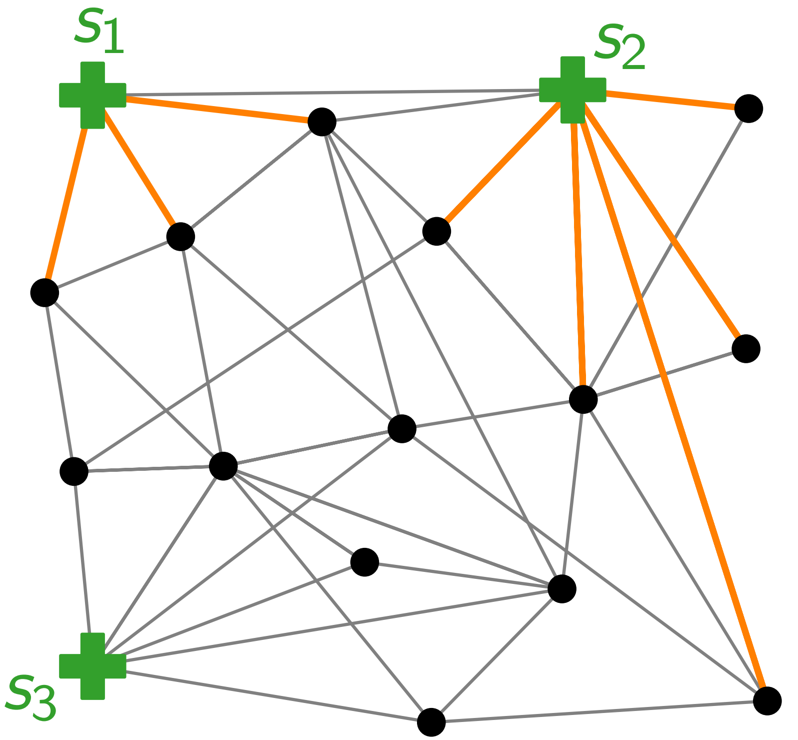
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c \colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

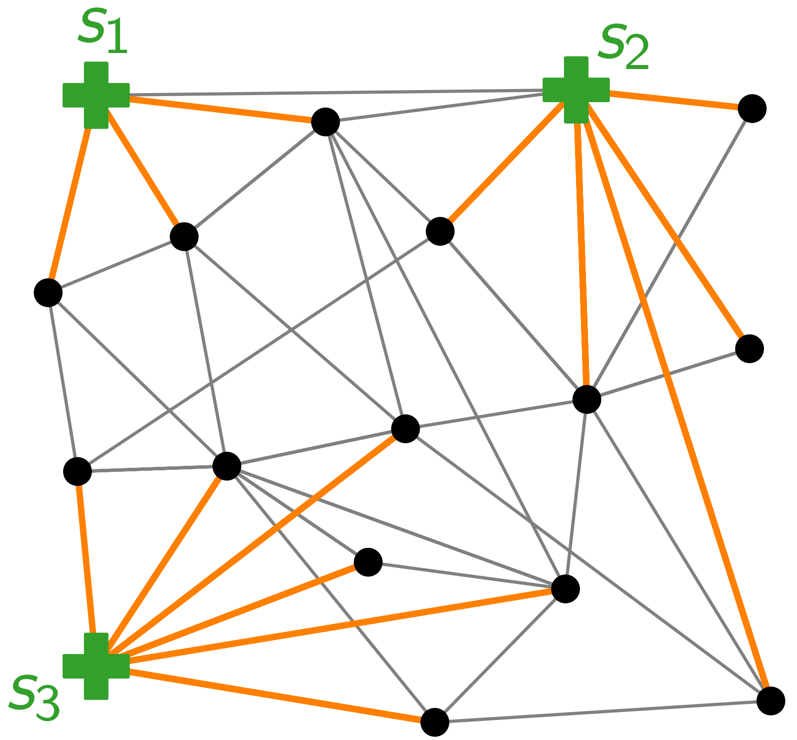
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

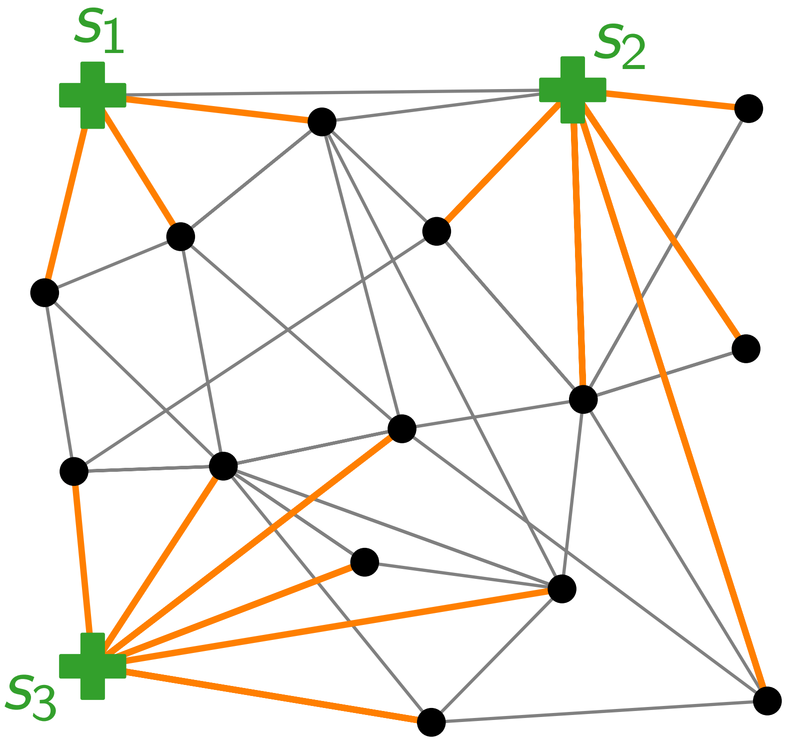
Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

Given: A complete graph G with edge costs $c \colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

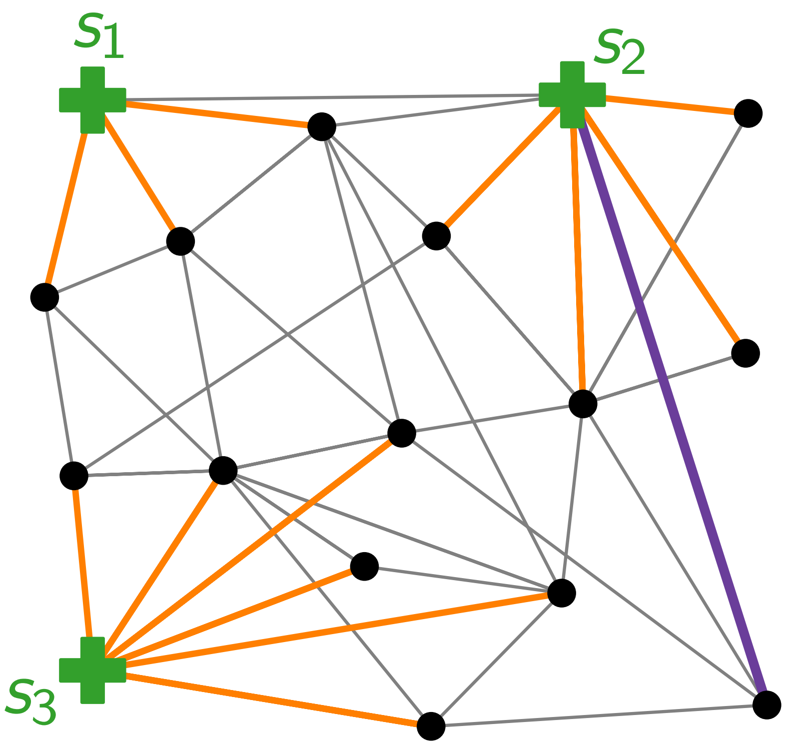


Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

$\max_{v \in V} c(v, S)$.

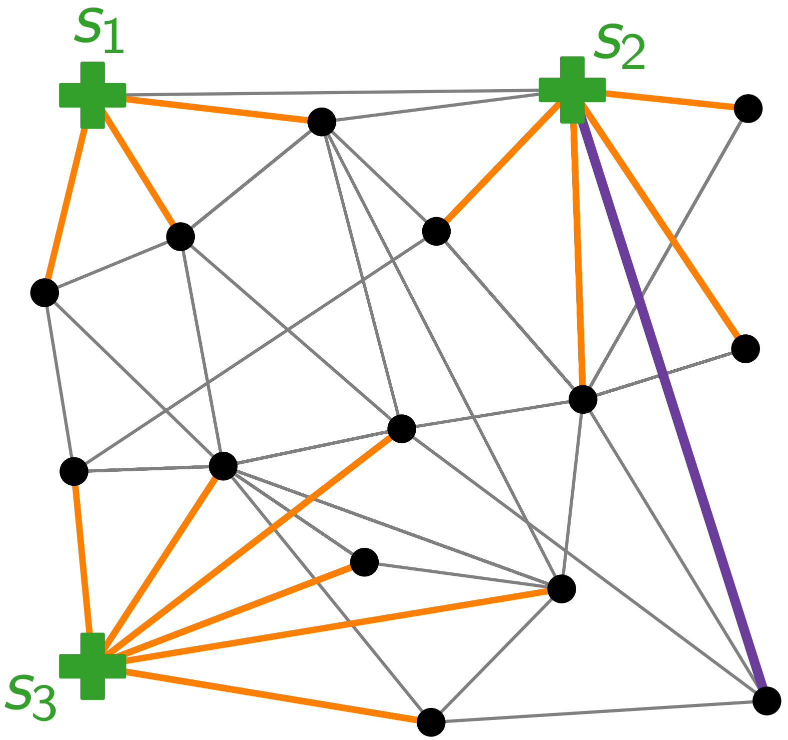


Metric k-Center

Given: A complete graph G with edge costs $c \colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality

Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

Find: A vertex set S that minimizes $\text{cost}(S) := \max_{v \in V} c(v, S)$.

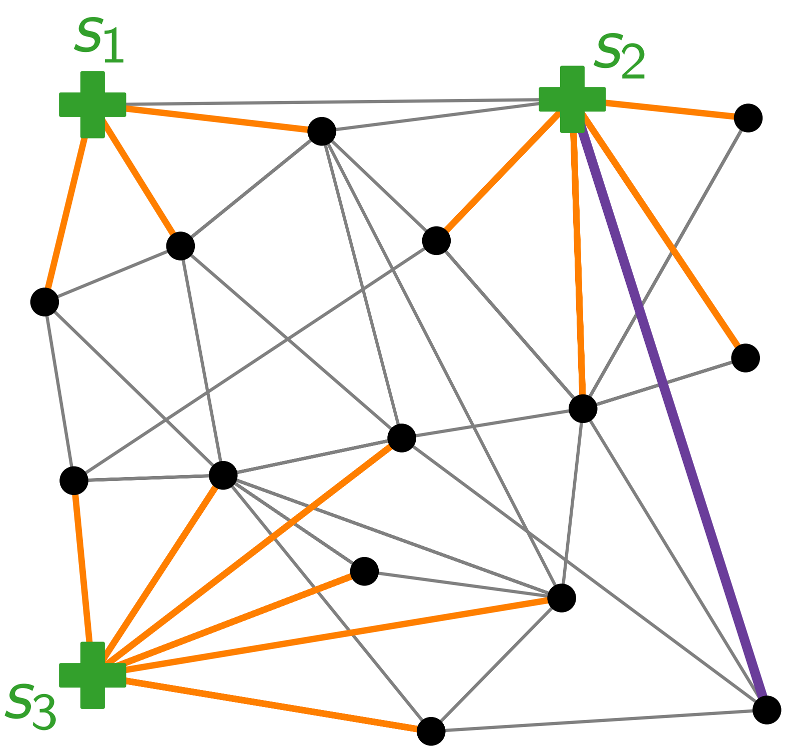


Metric k-Center

Given: A complete graph G with edge costs $c \colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality and a natural number $k \leq |V(G)|$.

Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

Find: A vertex set S that minimizes $\text{cost}(S) := \max_{v \in V} c(v, S)$.

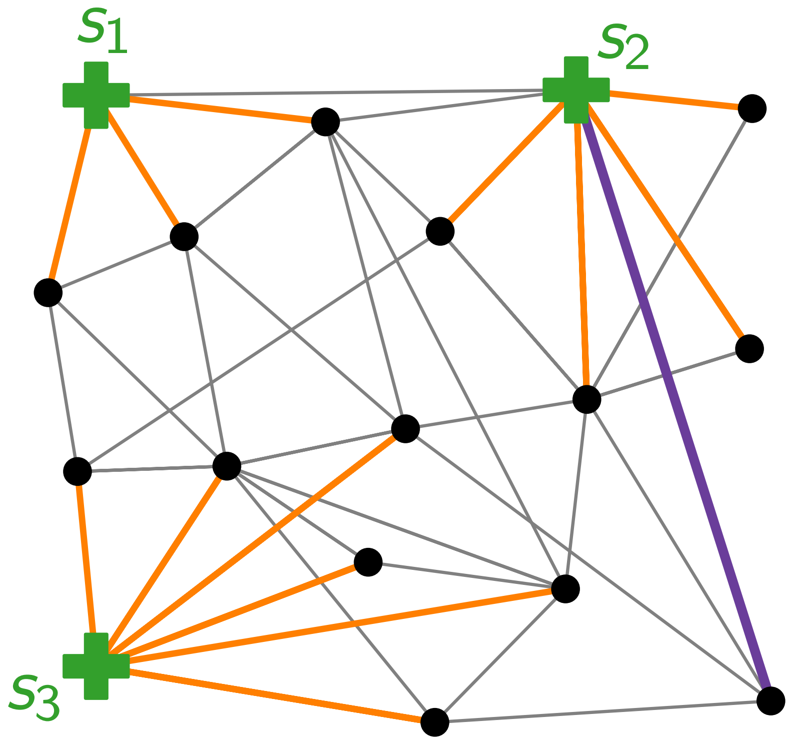


Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality and a natural number $k \leq |V(G)|$.

Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

Find: A k-element vertex set S that minimizes $\operatorname{cost}(S) := \max_{v \in V} c(v, S)$.

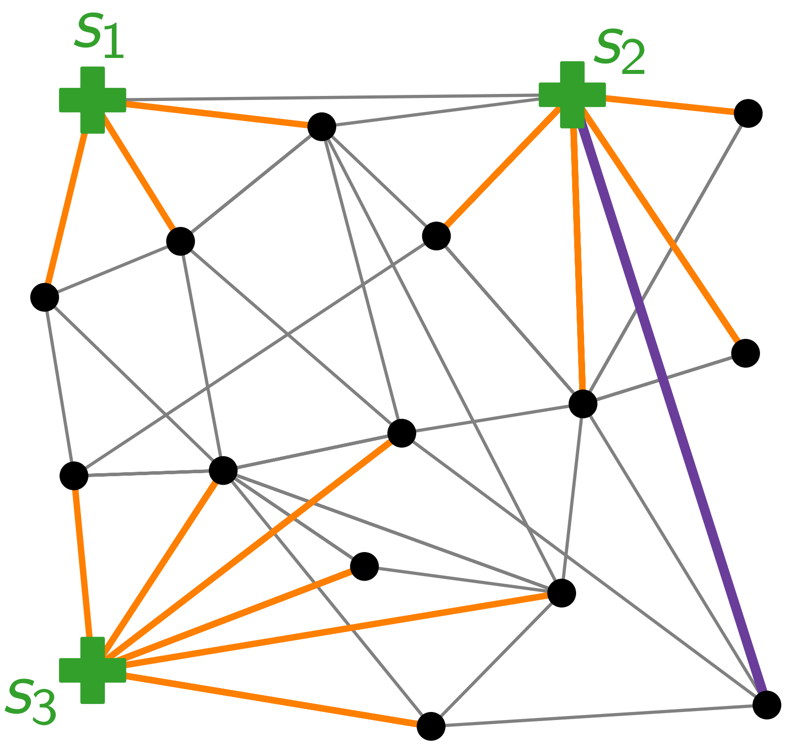


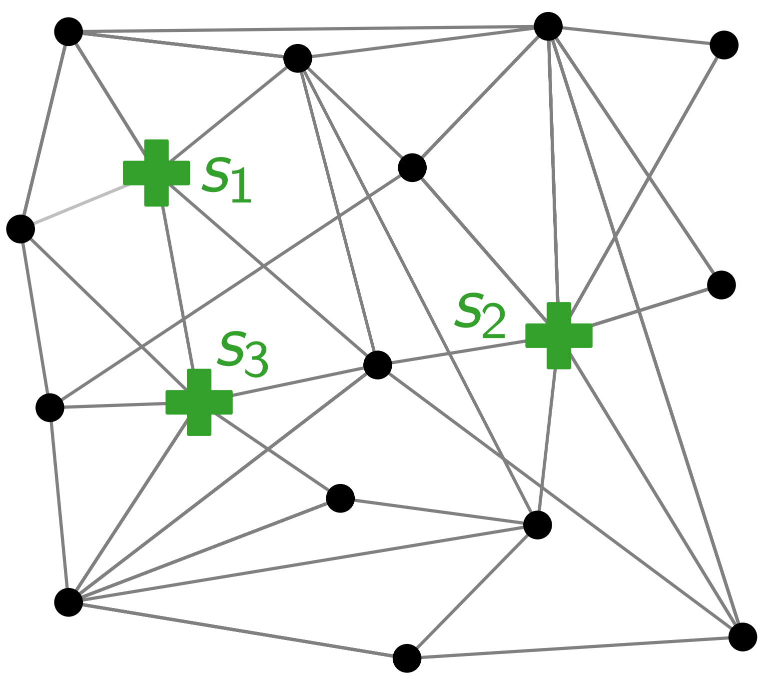
Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality and a natural number $k \leq |V(G)|$.

Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

Find: A k-element vertex set S that minimizes $\text{cost}(S) := \max_{v \in V} c(v, S)$.



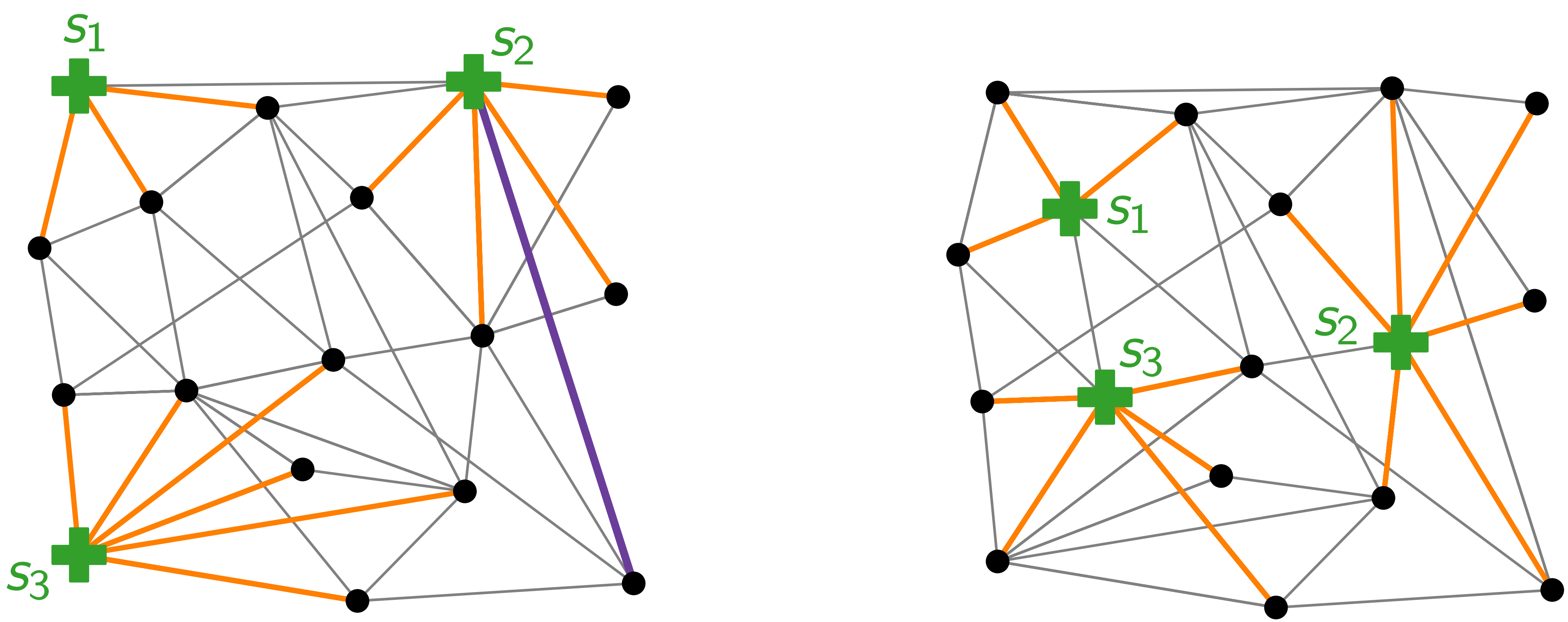


Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality and a natural number $k \leq |V(G)|$.

Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

Find: A k-element vertex set S that minimizes $\mathrm{cost}(S) := \max_{v \in V} c(v, S)$.

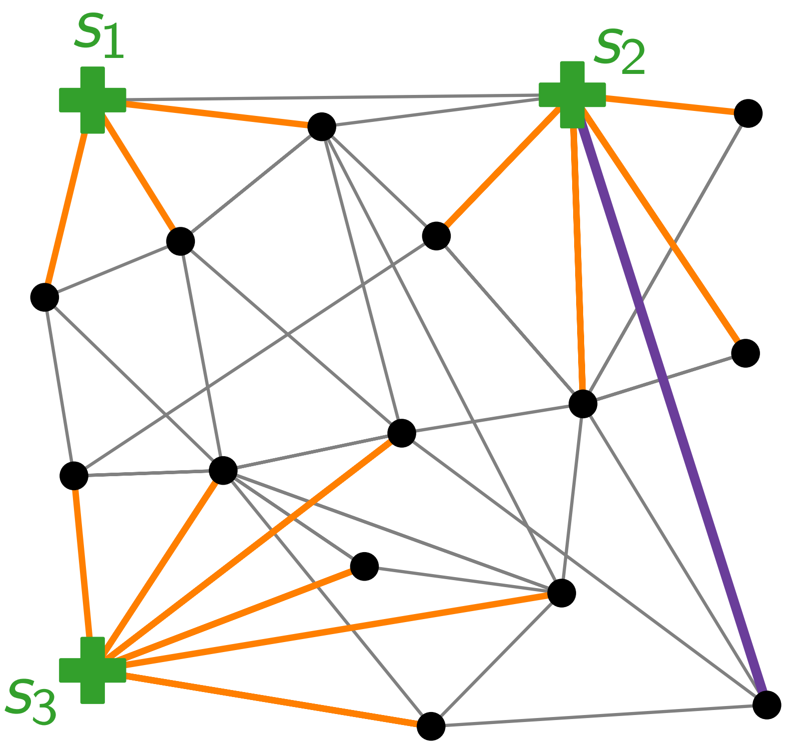


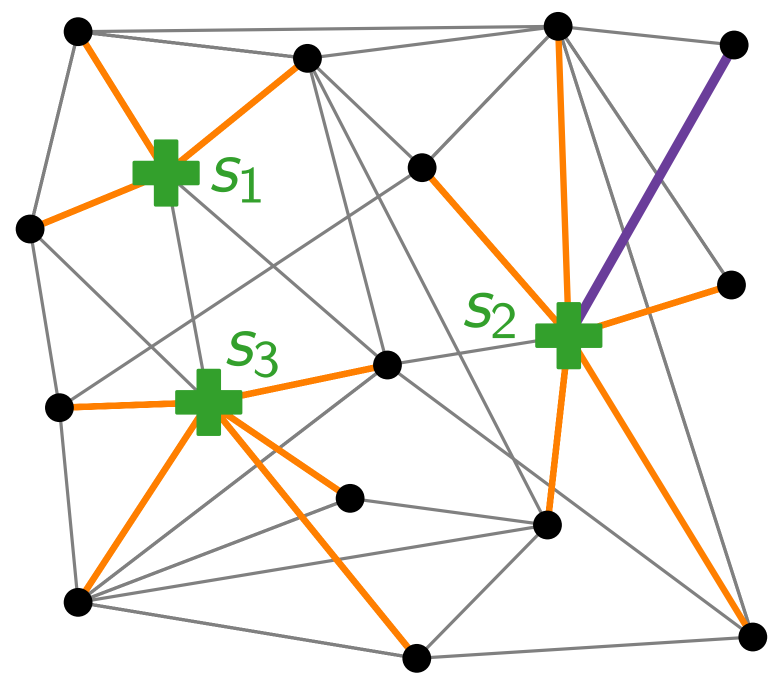
Metric k-Center

Given: A complete graph G with edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality and a natural number $k \leq |V(G)|$.

Given a set $S \subseteq V(G)$, $c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

Find: A k-element vertex set S that minimizes $\text{cost}(S) := \max_{v \in V} c(v, S)$.



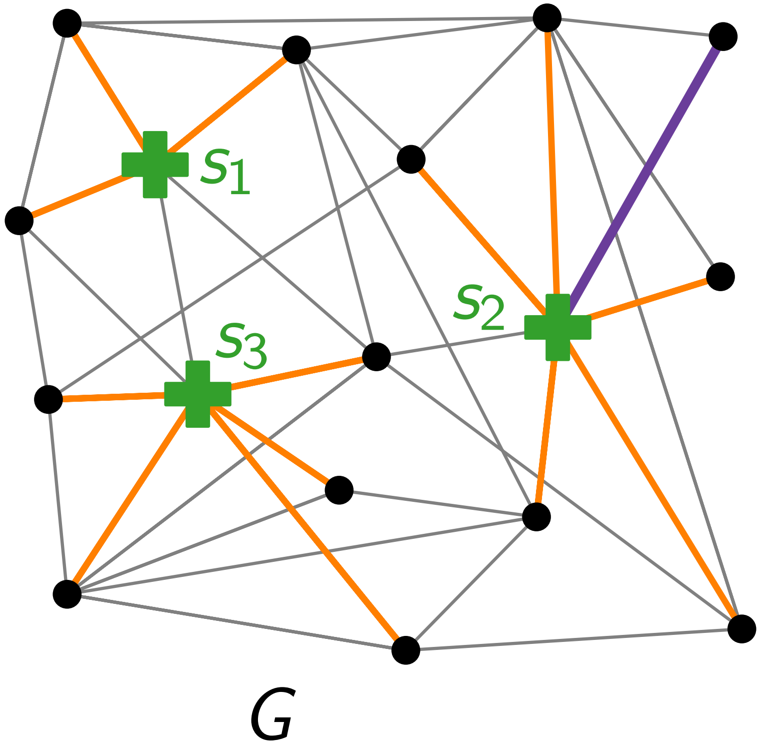


Approximation Algorithms

Lecture 6:
k-CENTER via Parametric Pruning

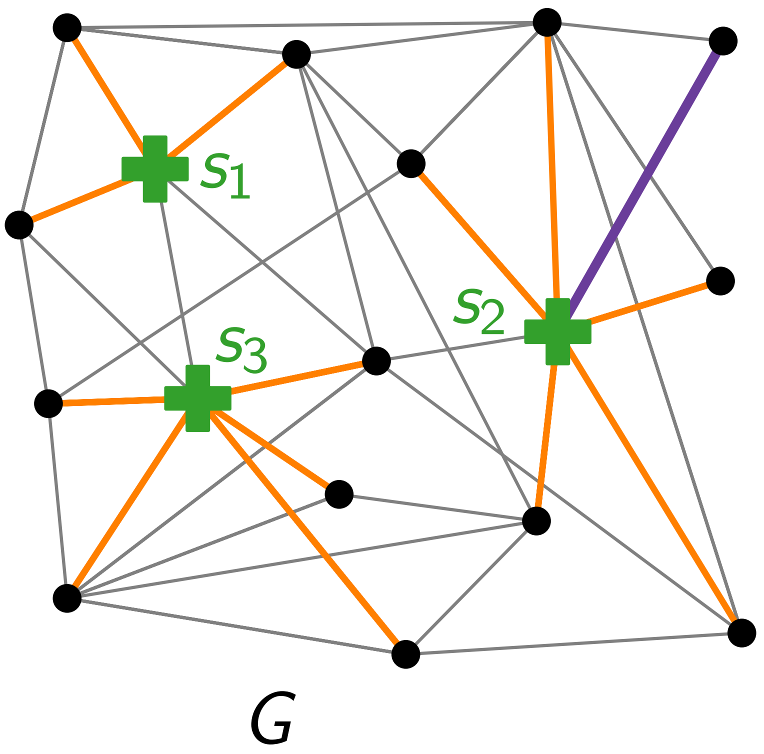
Part II:
Parametric Pruning

Parametric Pruning



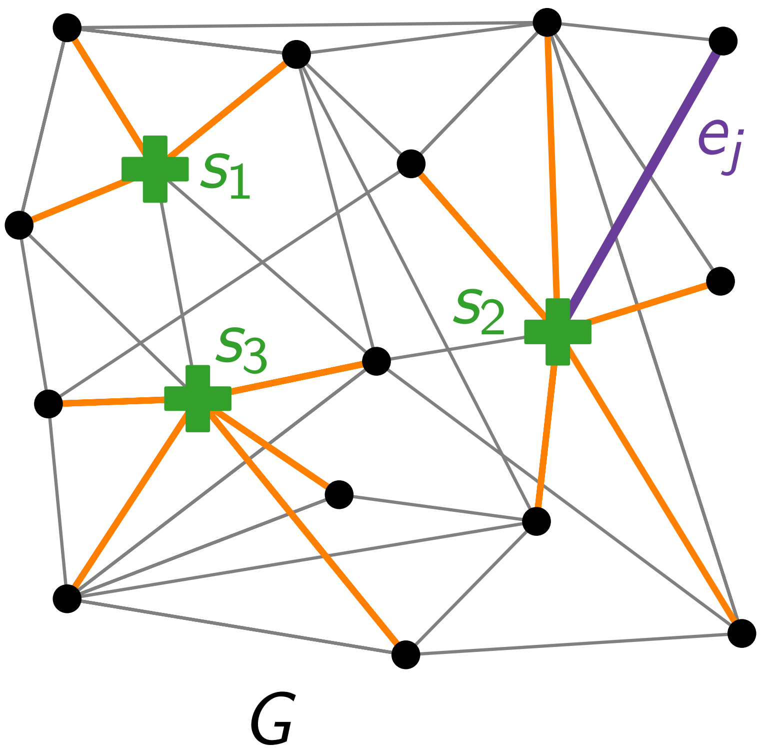
Parametric Pruning

Let $E(G) = \{e_1, \ldots, e_m\}$ with $c(e_1) \leq \cdots \leq c(e_m)$.



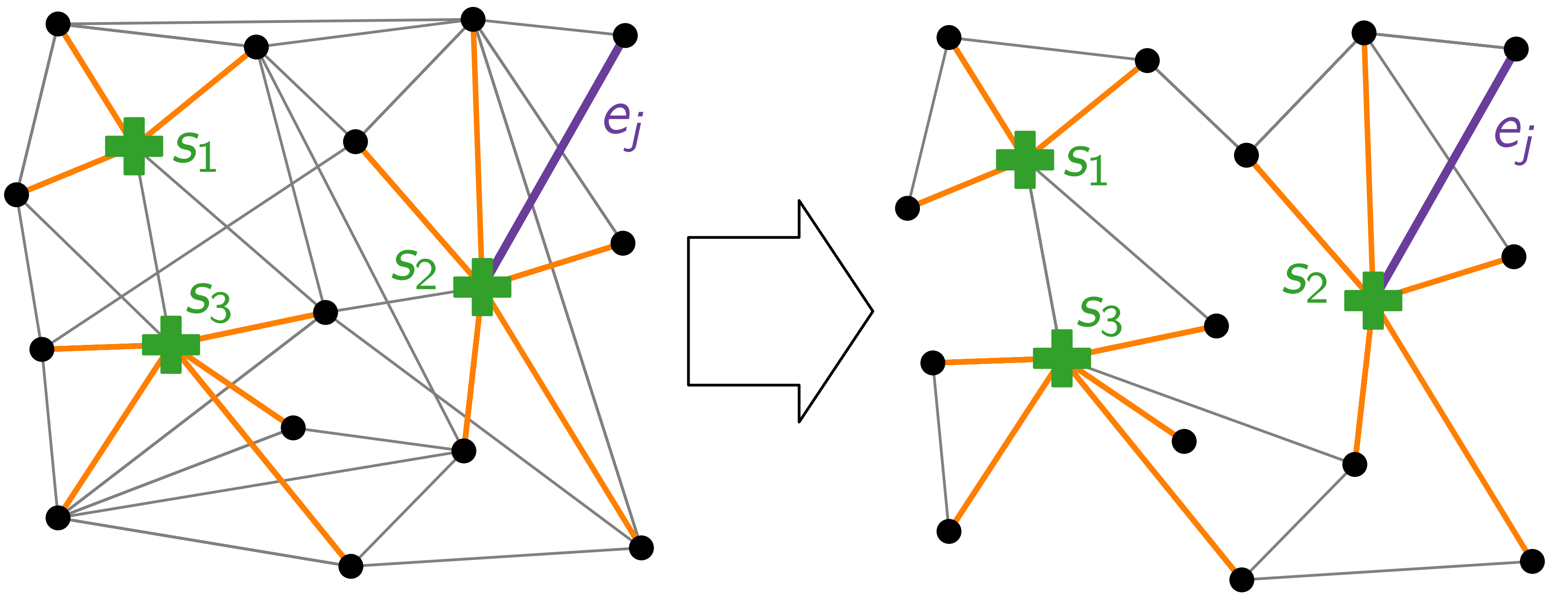
Parametric Pruning

Let $E(G) = \{e_1, \ldots, e_m\}$ with $c(e_1) \leq \cdots \leq c(e_m)$.
Suppose that we know that $\mathrm{OPT} = c(e_j)$.



Parametric Pruning

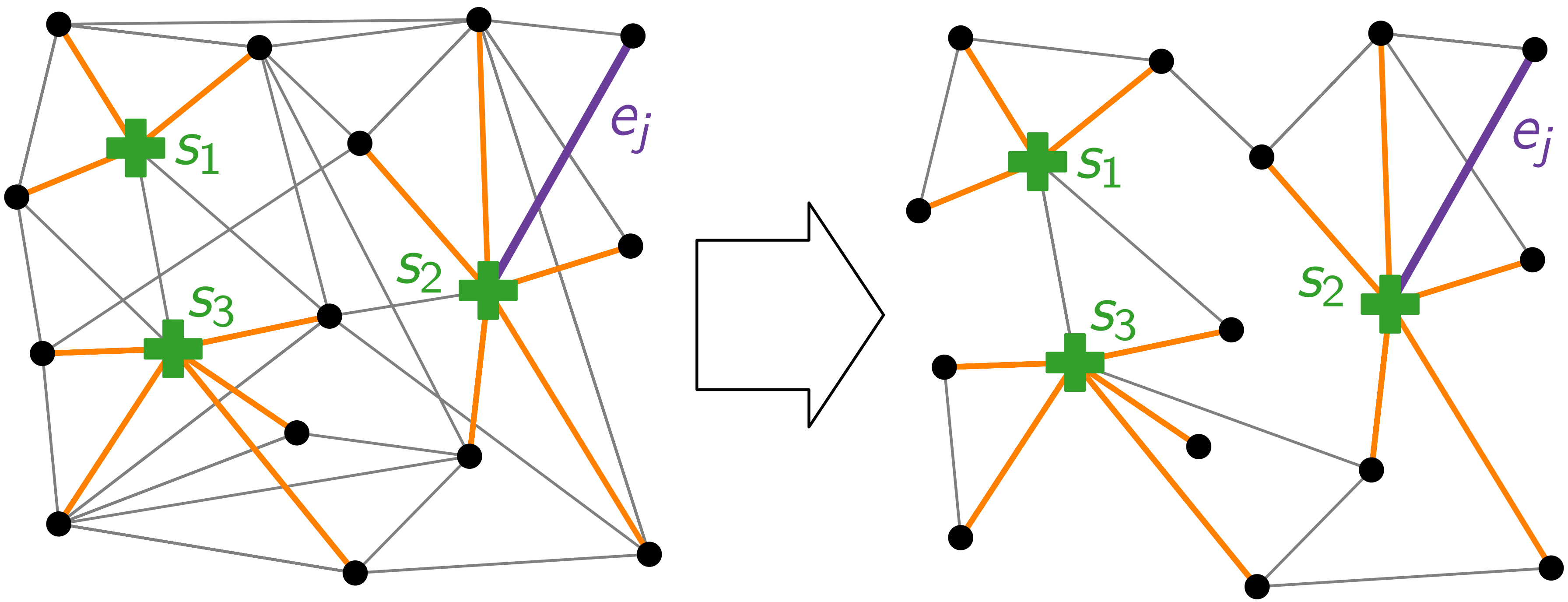
Let $E(G) = \{e_1, \ldots, e_m\}$ with $c(e_1) \leq \cdots \leq c(e_m)$.
Suppose that we know that $\text{OPT} = c(e_j)$.



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

Parametric Pruning

Let $E(G) = \{e_1, \ldots, e_m\}$ with $c(e_1) \leq \cdots \leq c(e_m)$.
Suppose that we know that $\mathrm{OPT} = c(e_j)$.

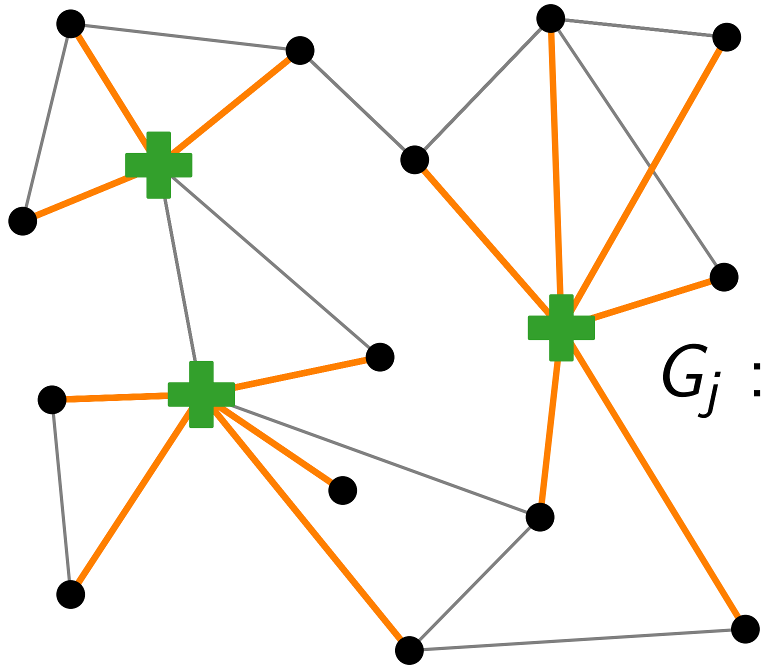


G $\qquad$ $G_j := (V(G), \{e_1, \ldots, e_j\})$

. . . try each G_j.

. . . try each G_j.

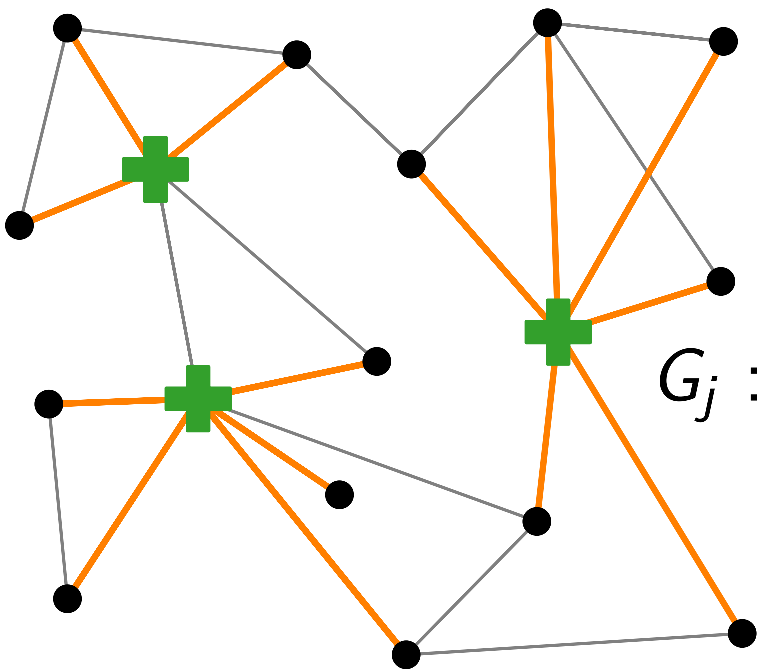
Def.



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

. . . try each G_j.

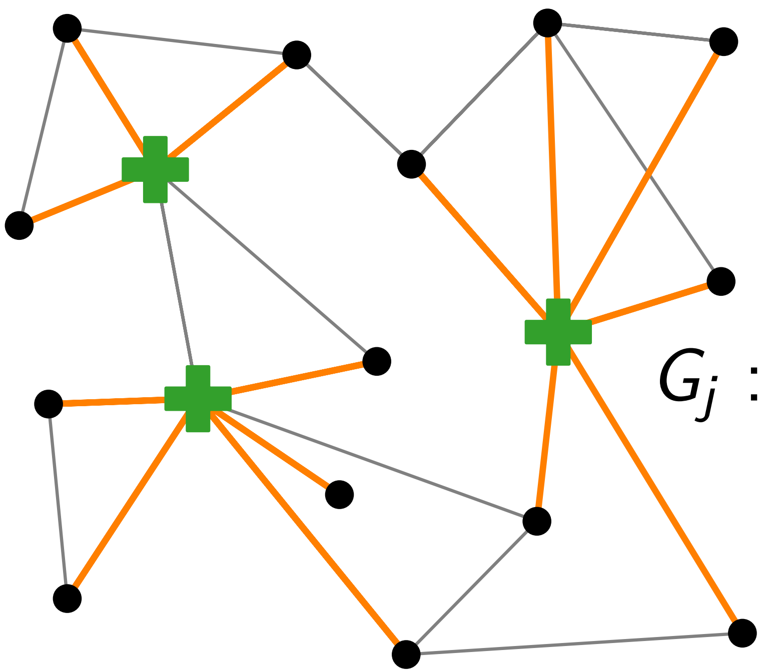
Def. A vertex set D of a graph H is **dominating** if each vertex is either in D or adjacent to a vertex in D.



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

. . . try each G_j.

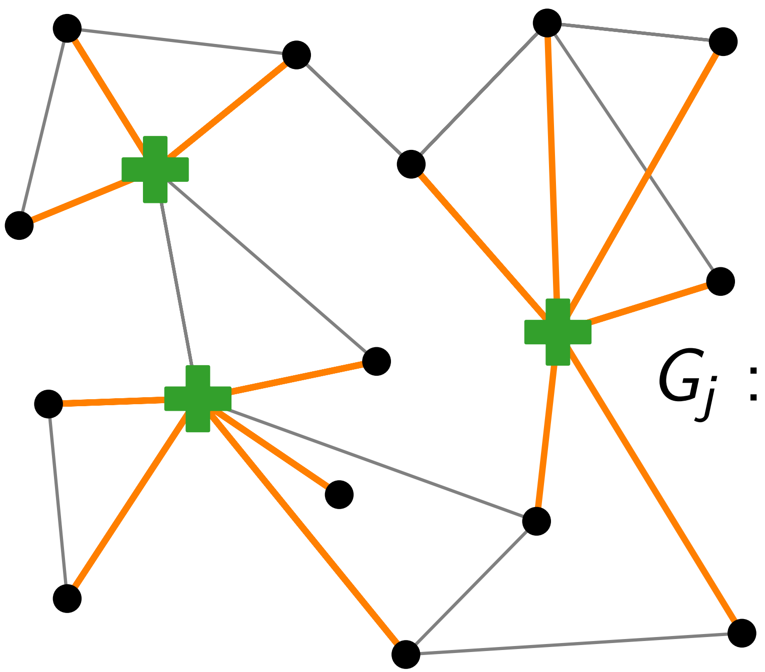
Def. A vertex set D of a graph H is **dominating** if each vertex is either in D or adjacent to a vertex in D. The cardinality of a smallest dominating set in H is denoted by $\mathrm{dom}(H)$.



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

...try each G_j.

Def. A vertex set D of a graph H is **dominating** if each vertex is either in D or adjacent to a vertex in D. The cardinality of a smallest dominating set in H is denoted by $\mathrm{dom}(H)$.

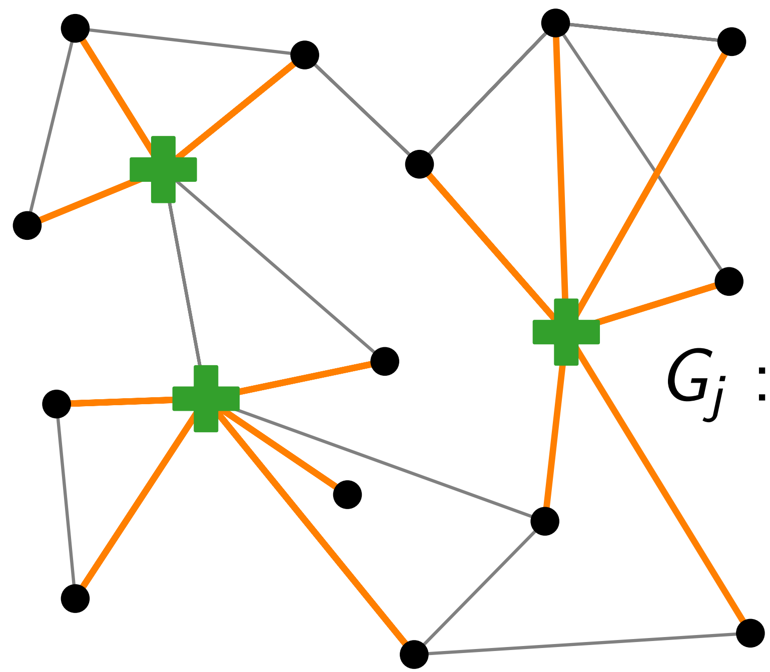


$$\mathrm{dom}(G_j) = 3$$

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

. . . try each G_j.

Def. A vertex set D of a graph H is **dominating** if each vertex is either in D or adjacent to a vertex in D.
The cardinality of a smallest dominating set in H is denoted by $\mathrm{dom}(H)$.



$\mathrm{dom}(G_j) = 3$

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

. . . but computing $\mathrm{dom}(H)$ is NP-hard.

Approximation Algorithms

Lecture 6:
k-Center via Parametric Pruning

Part III:
Square of a Graph

Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

Square of a Graph

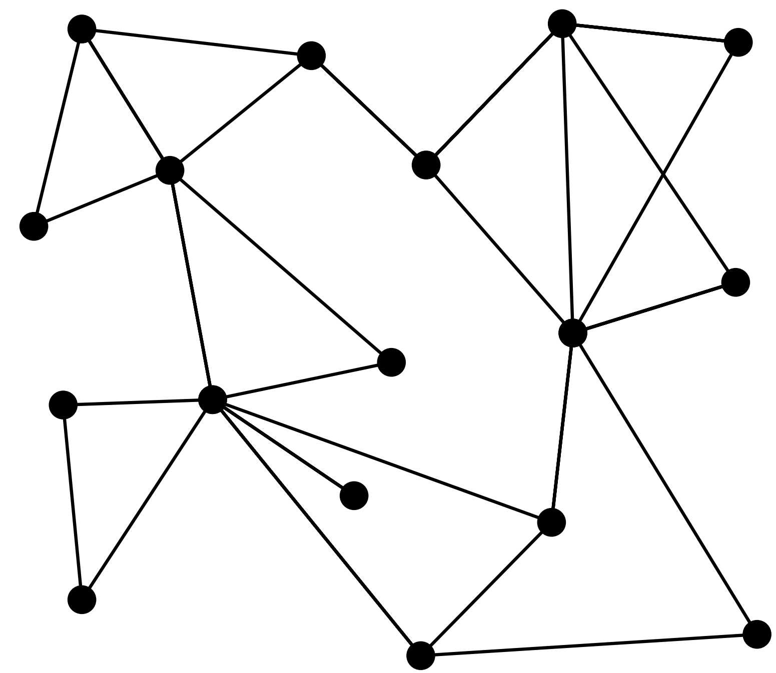
Idea: Find a small dominating set in a "coarsened" G_j.

Def. The **square** H^2 of a graph H has the same vertex set as H.

Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

Def. The **square** H^2 of a graph H has the same vertex set as H.

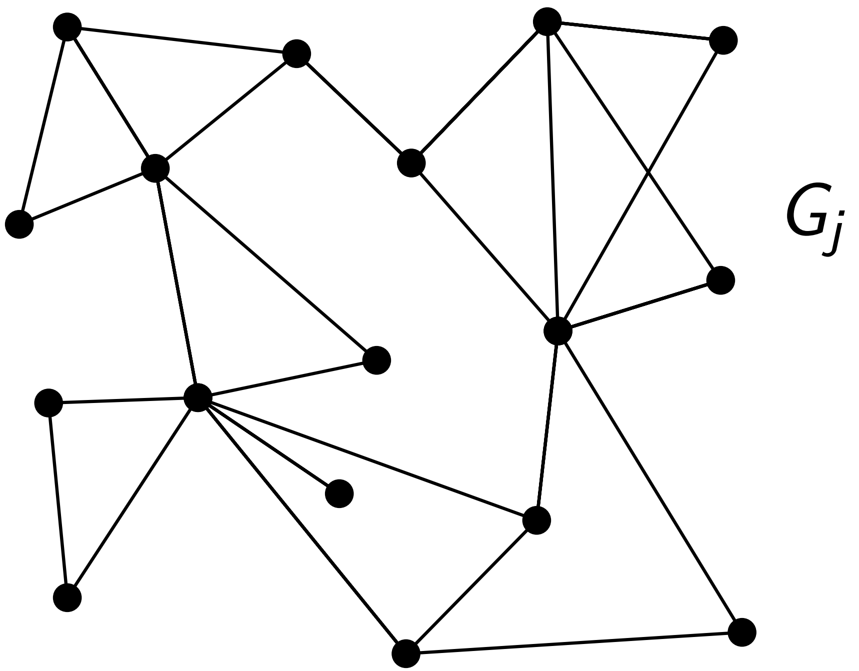


G_j

Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

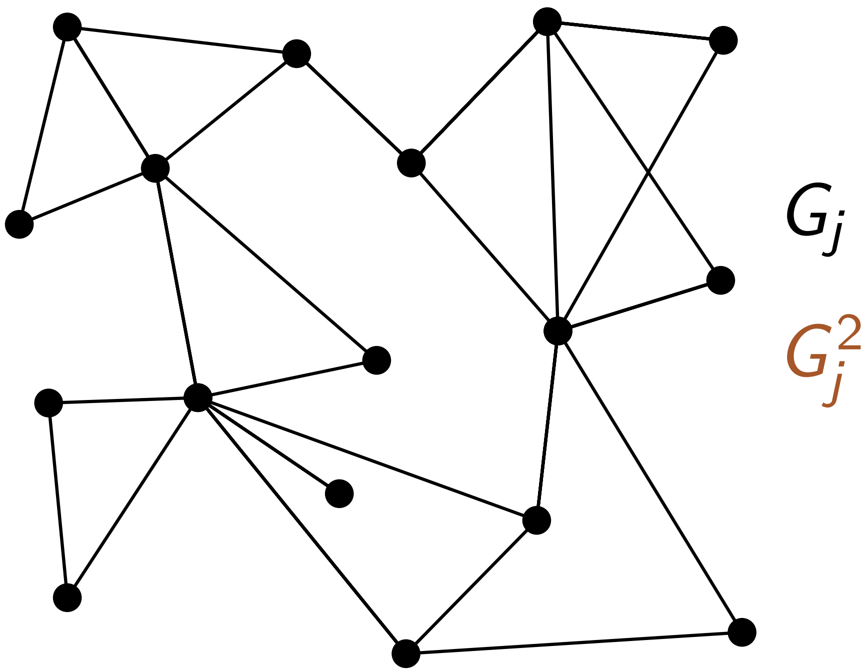
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

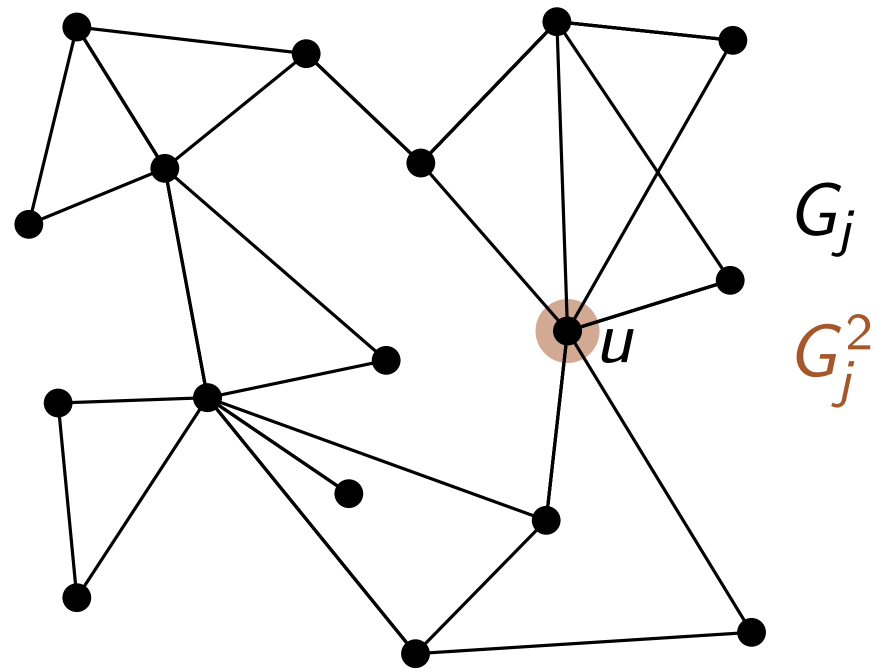
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

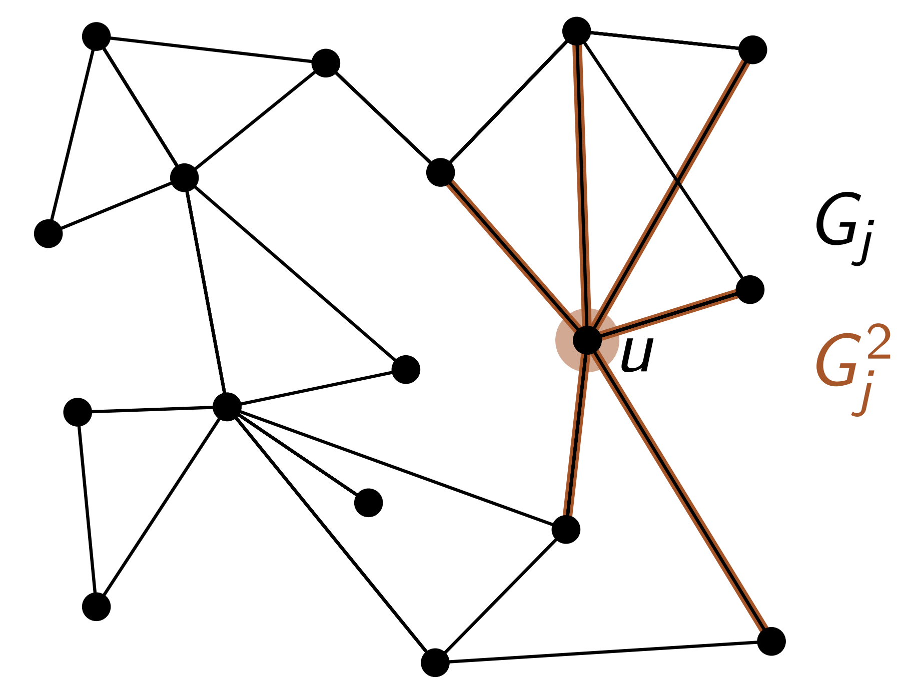
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

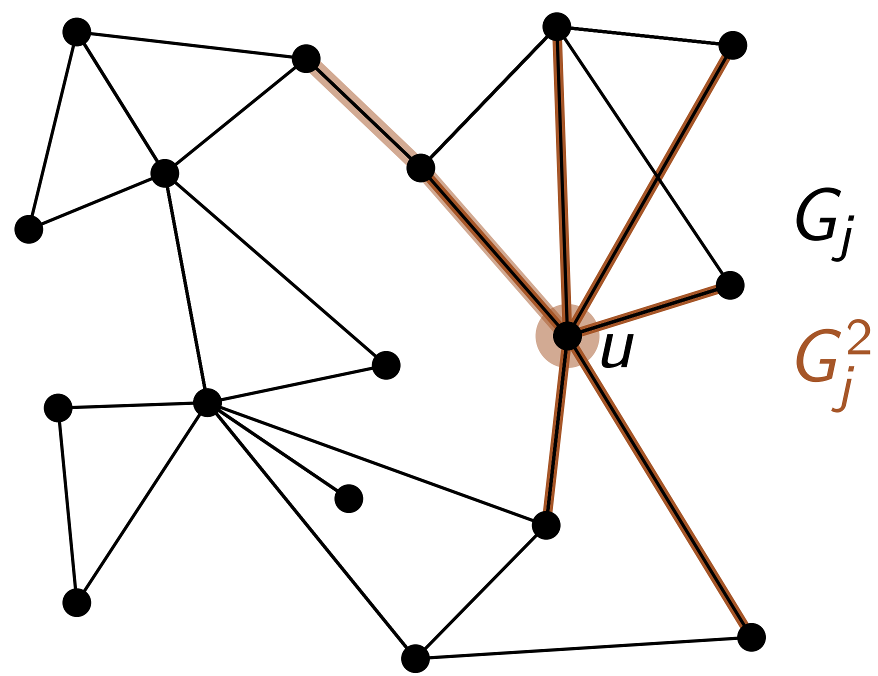
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

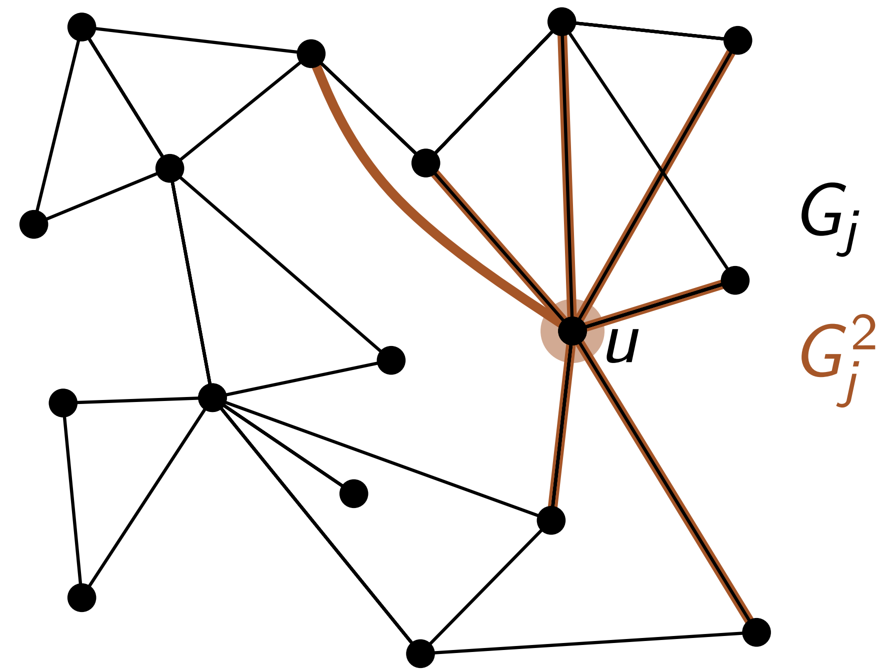
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

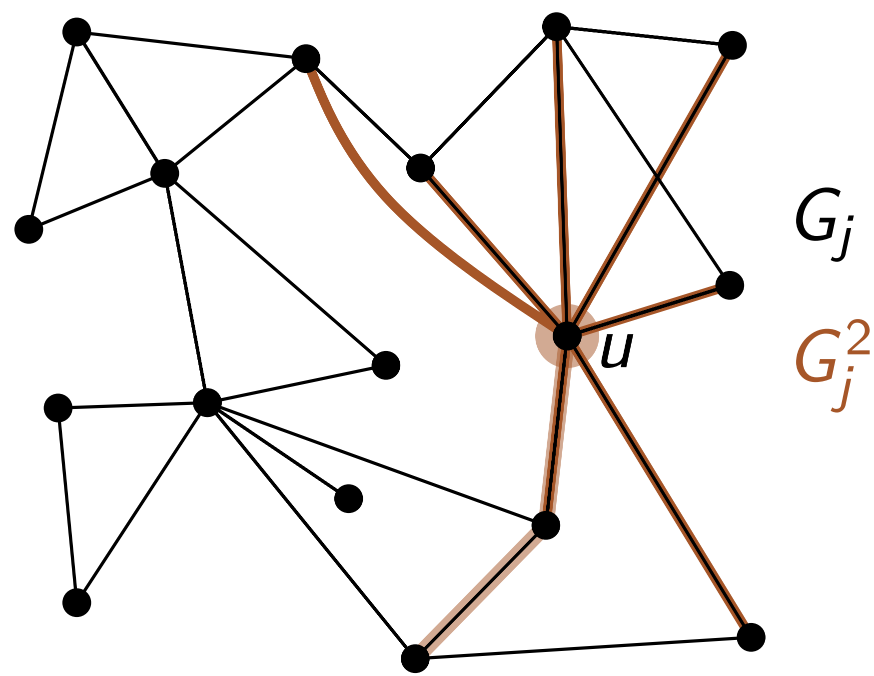
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

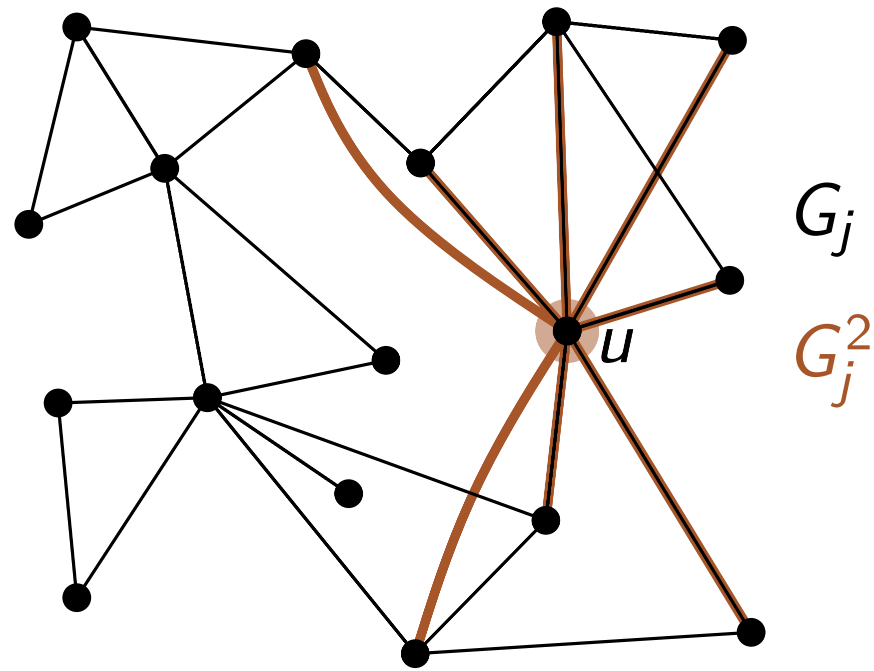
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

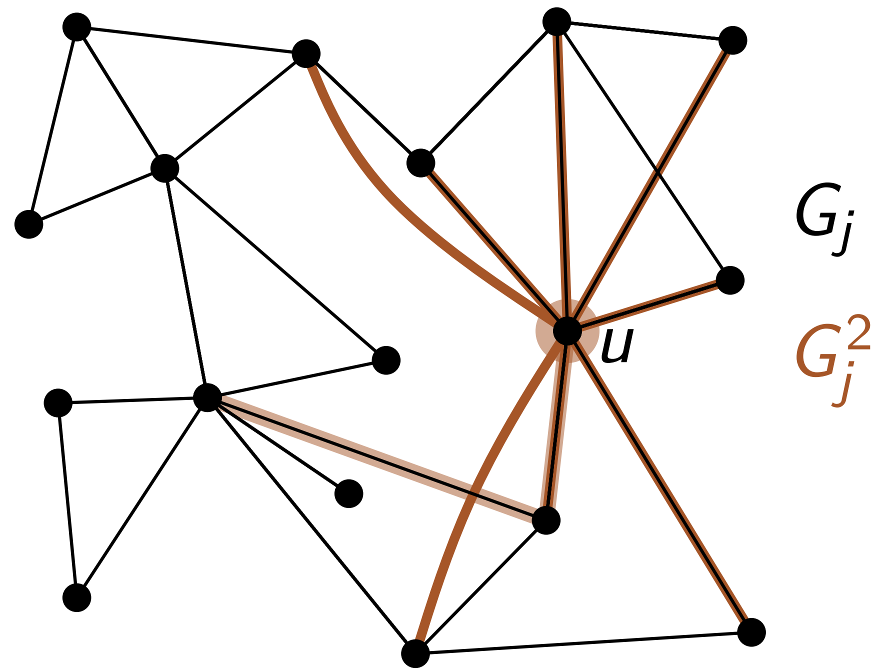
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

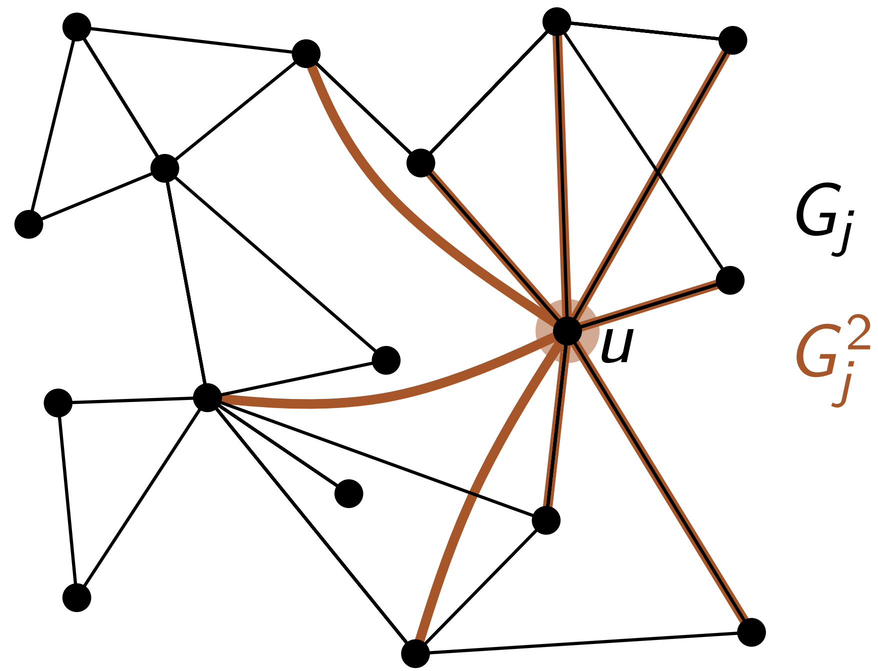
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

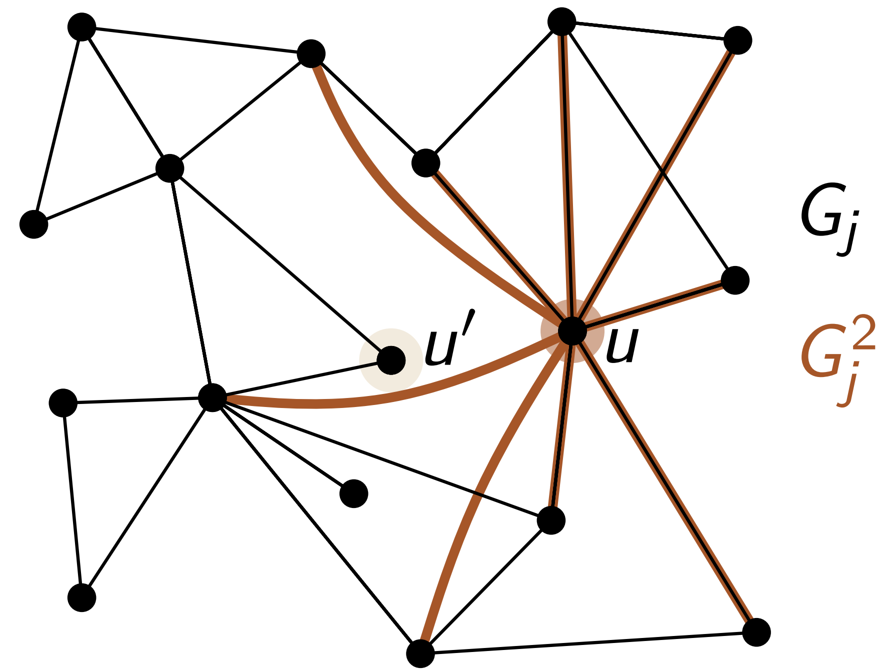
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a “coarsened” G_j.

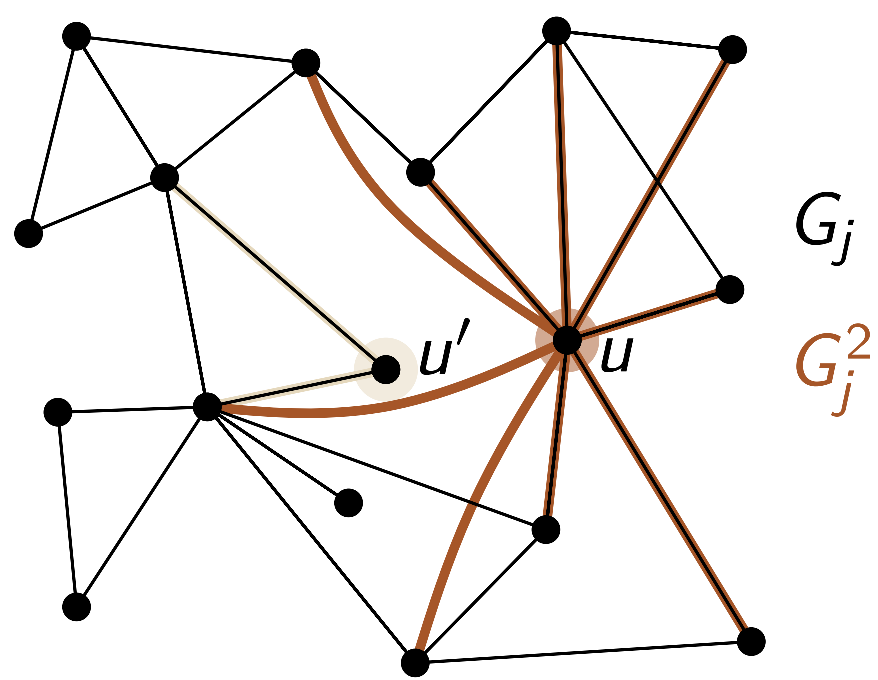
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a “coarsened” G_j.

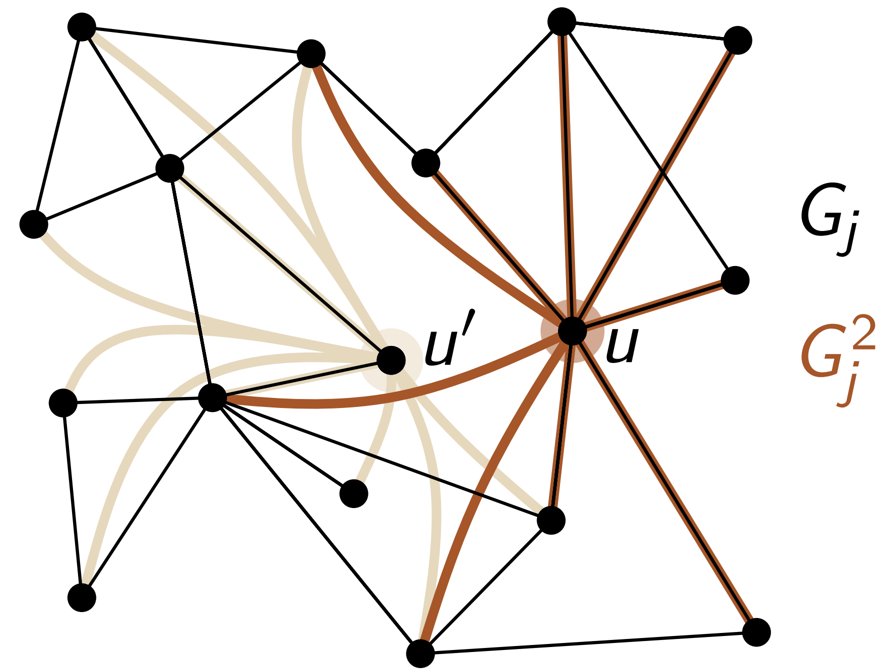
Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.



Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.

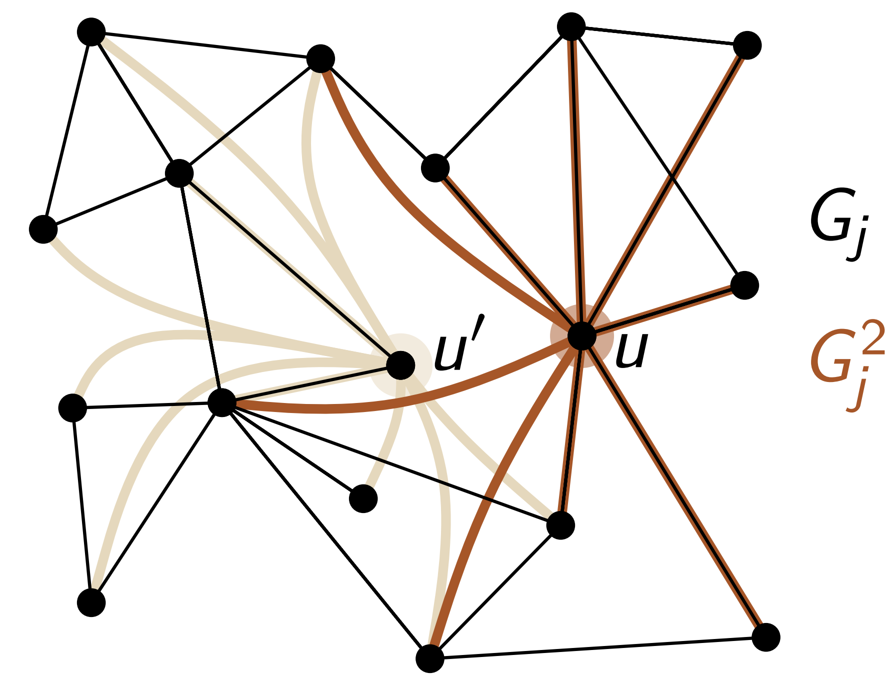


Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.

Obs. A dominating set of size at most k in G_j^2 is a -approximation for the metric k-Center of G.

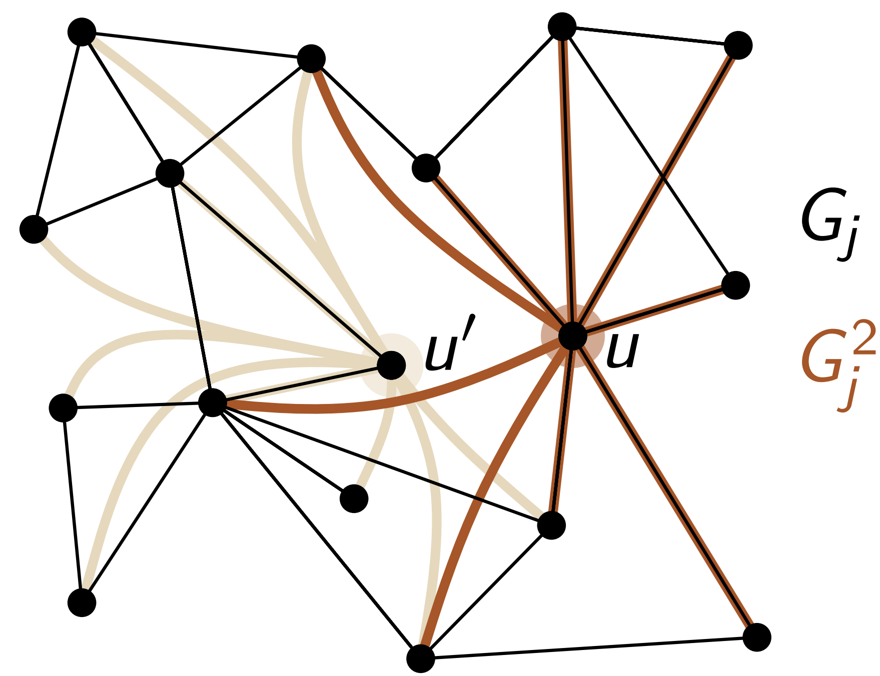


Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.

Obs. A dominating set of size at most k in G_j^2 is a 2-approximation for the metric k-CENTER of G.

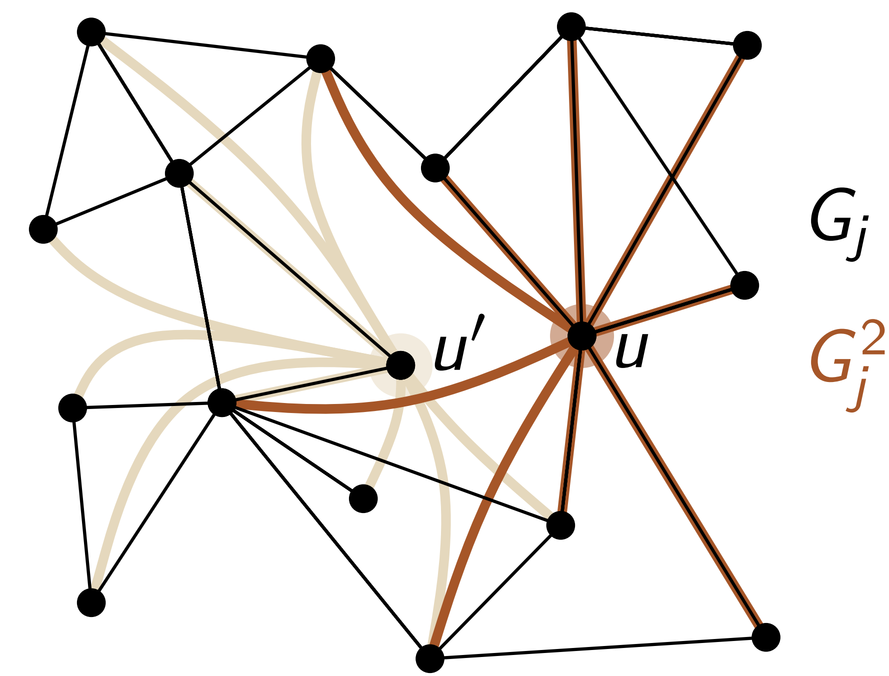


Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.

Obs. A dominating set of size at most k in G_j^2 is a 2-approximation for the metric k-Center of G.



Why?

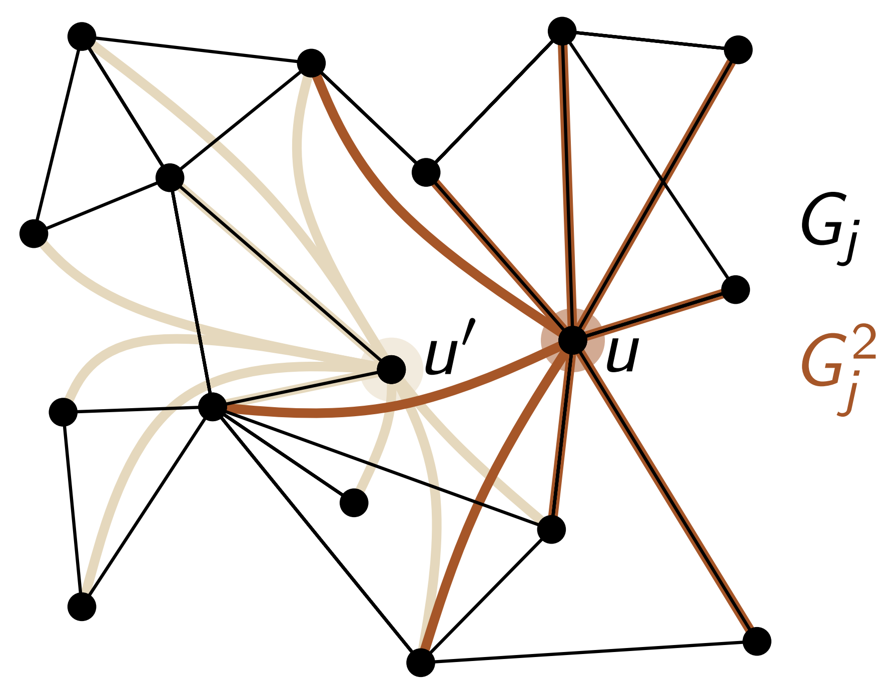
Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.

Obs. A dominating set of size at most k in G_j^2 is a 2-approximation for the metric k-CENTER of G.

Why? $\max_{e \in E(G_j)} c(e) = \mathrm{OPT}$



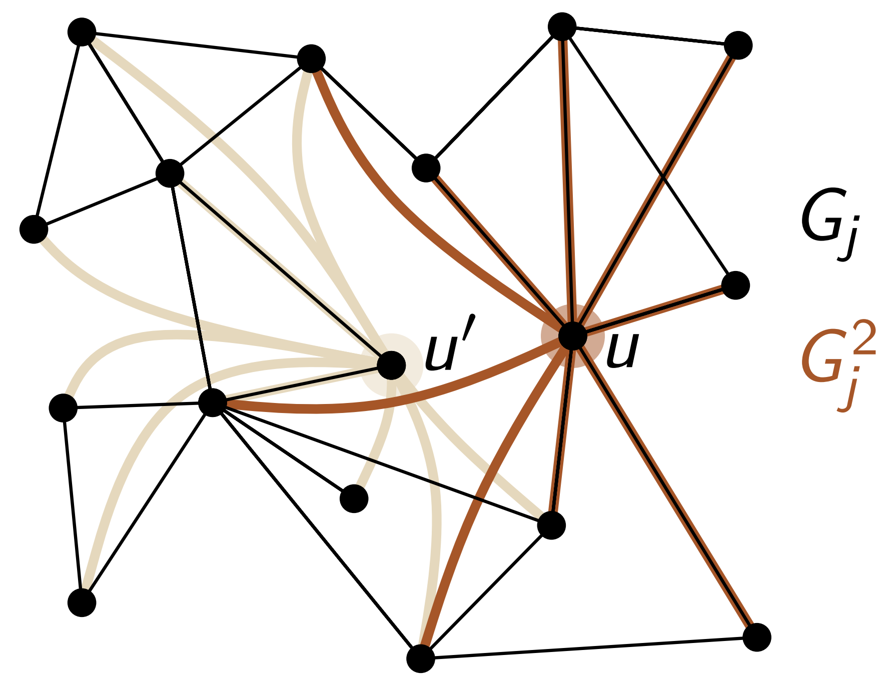
Square of a Graph

Idea: Find a small dominating set in a "coarsened" G_j.

Def. The **square** H^2 of a graph H has the same vertex set as H. Two vertices $u \neq v$ are adjacent in H^2 iff they are within distance at most **two** in H.

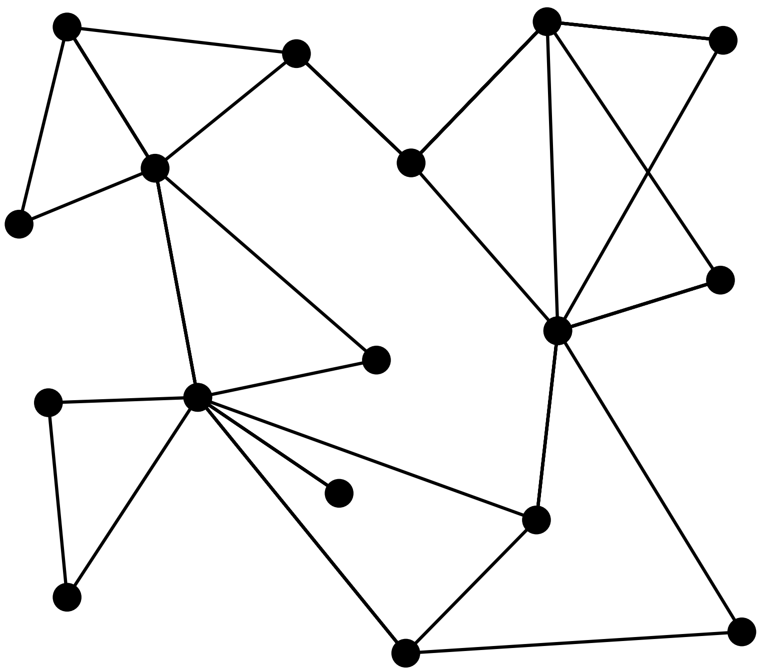
Obs. A dominating set of size at most k in G_j^2 is a 2-approximation for the metric k-CENTER of G.

Why? $\max_{e \in E(G_j)} c(e) = \mathrm{OPT}$ and edge costs are metric!



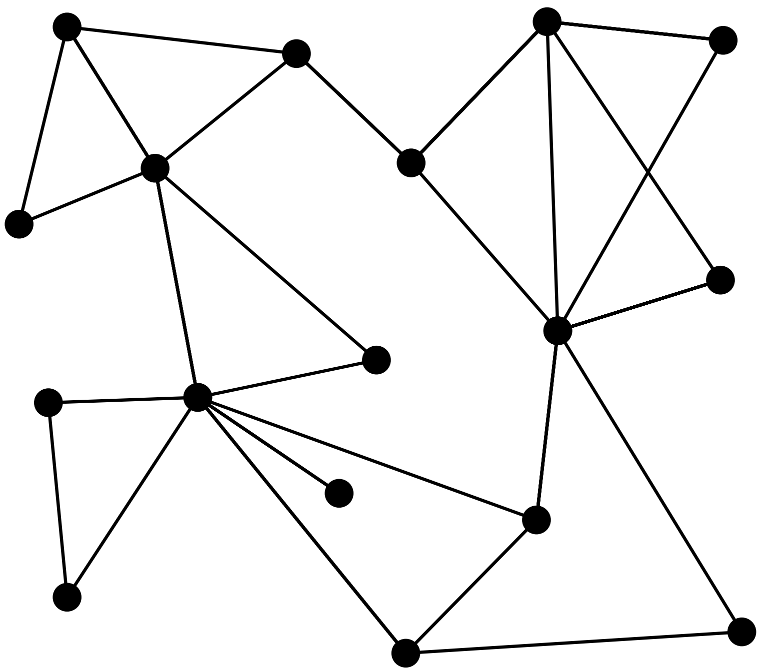
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.



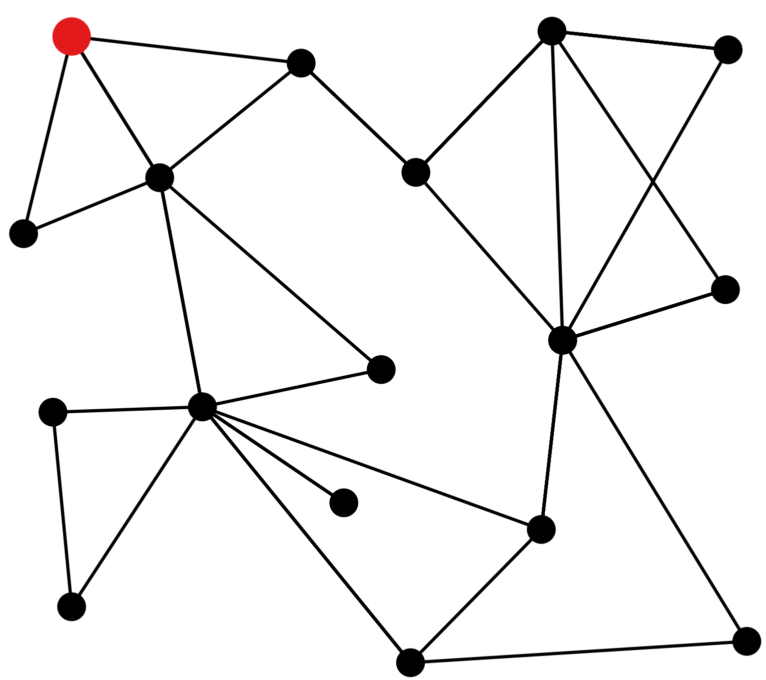
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



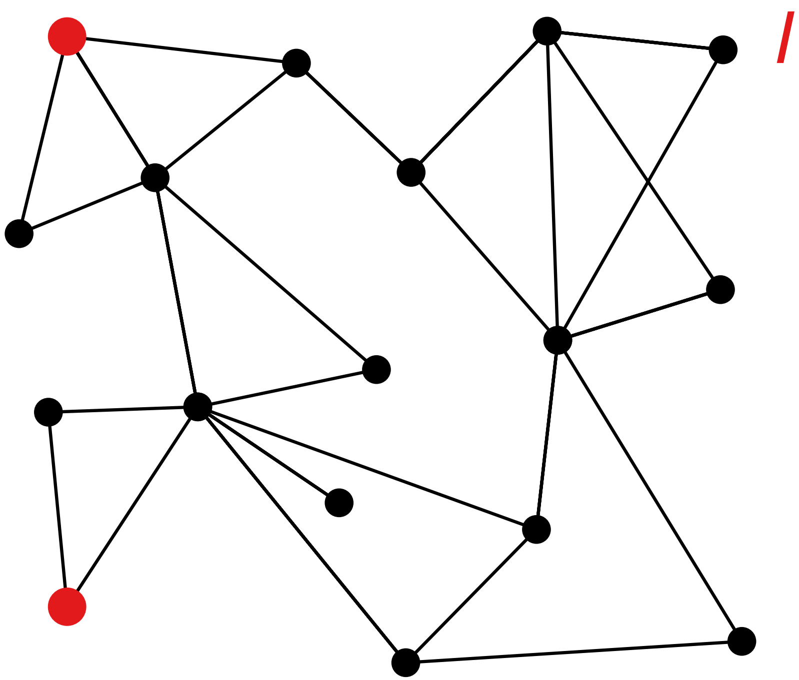
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



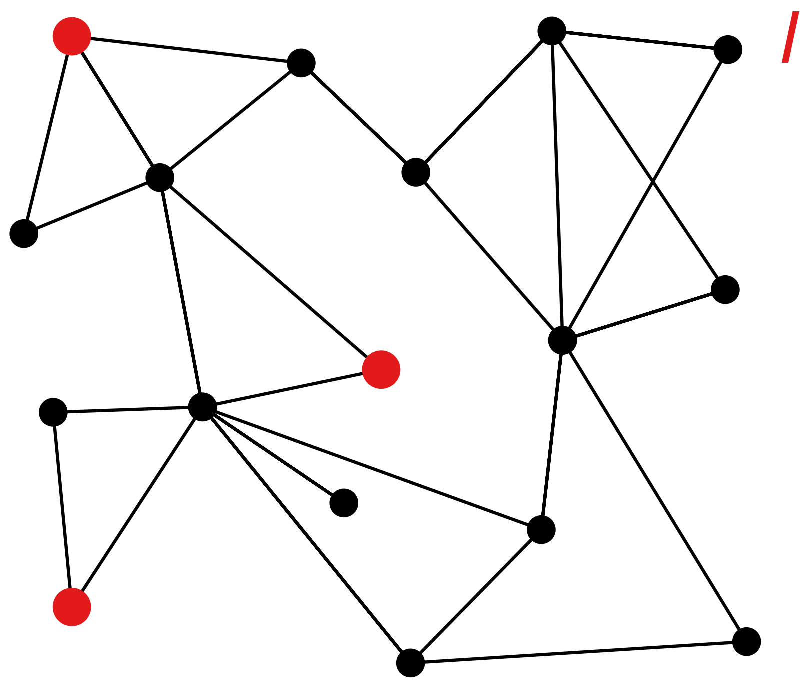
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



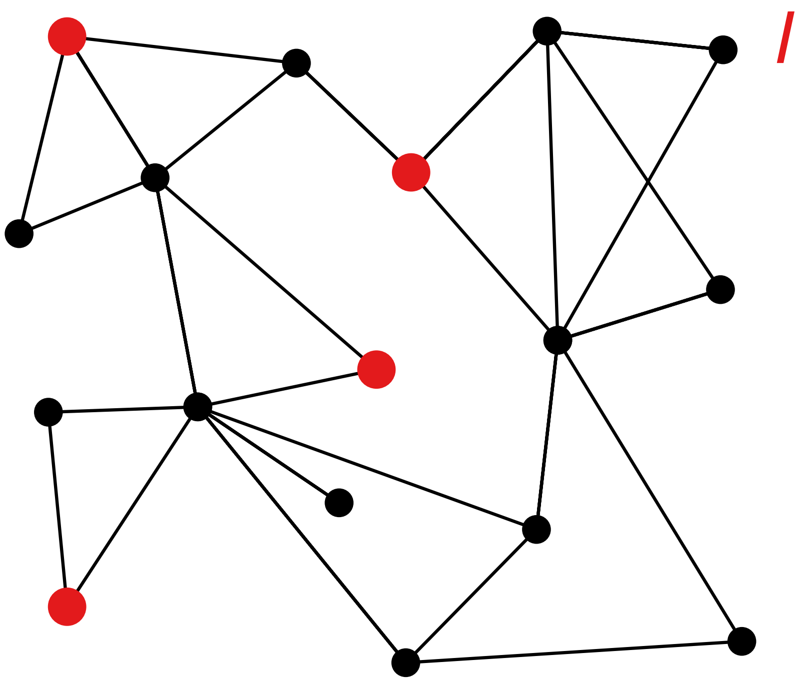
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



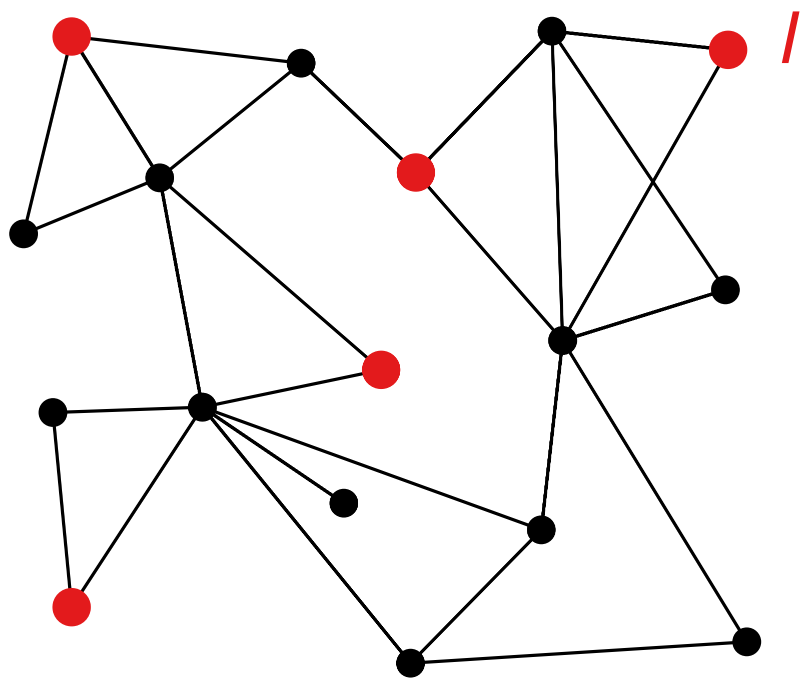
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



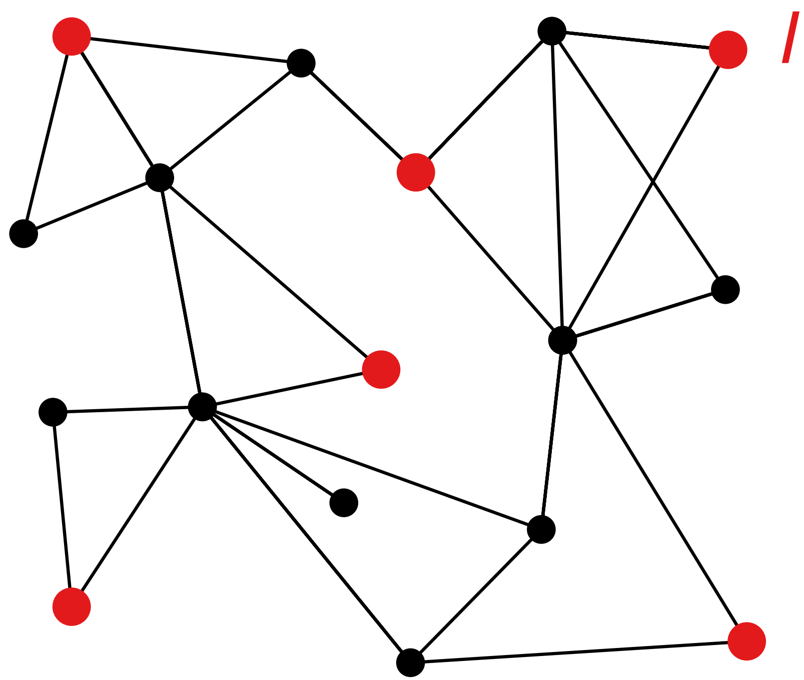
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



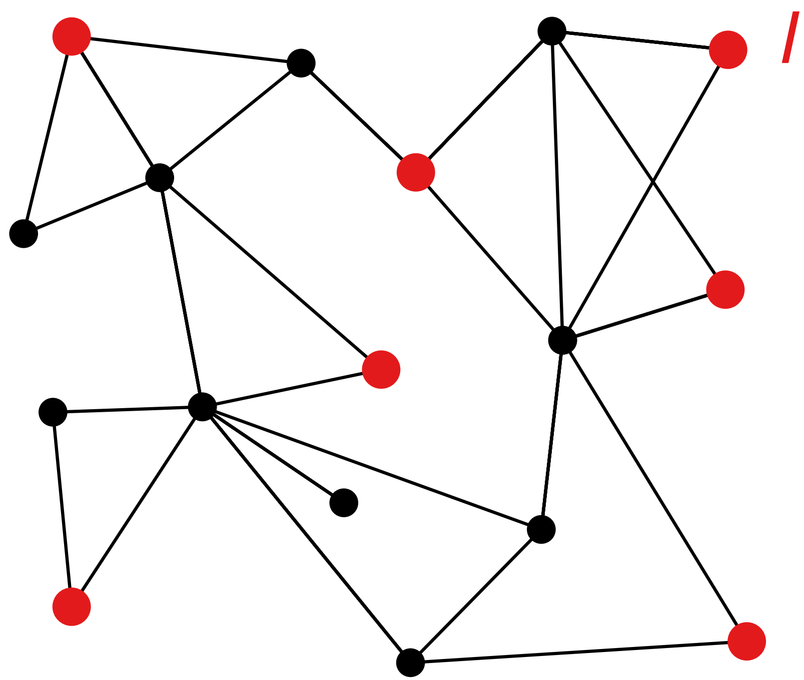
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



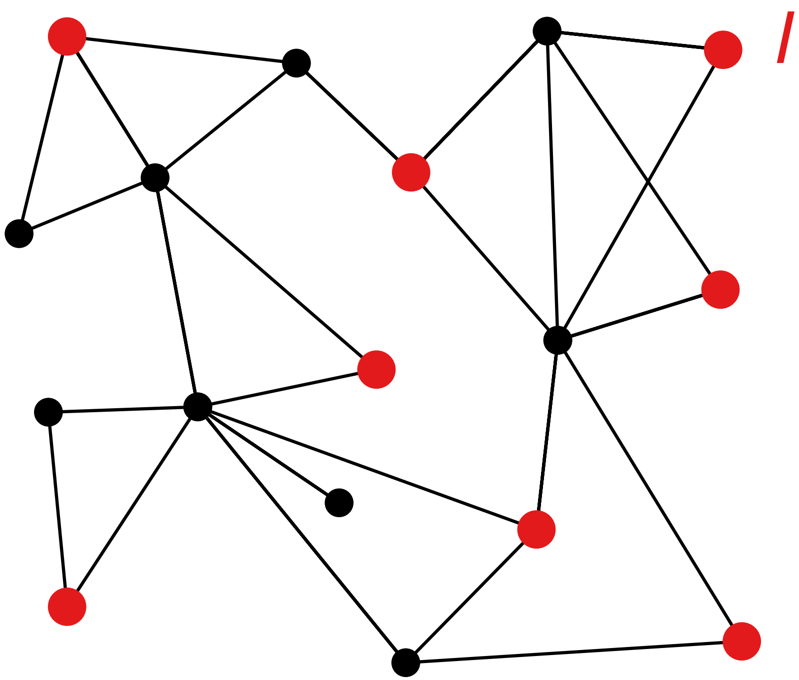
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



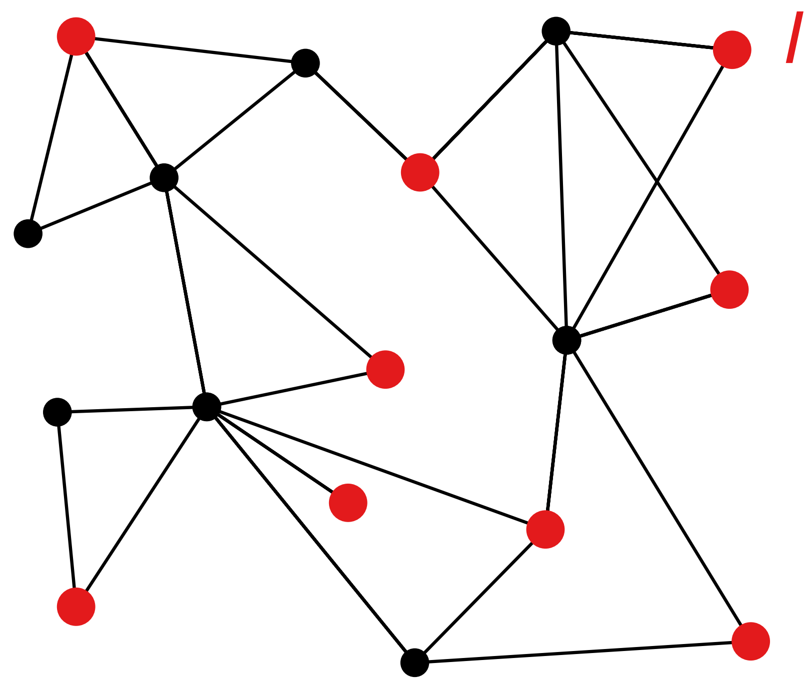
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



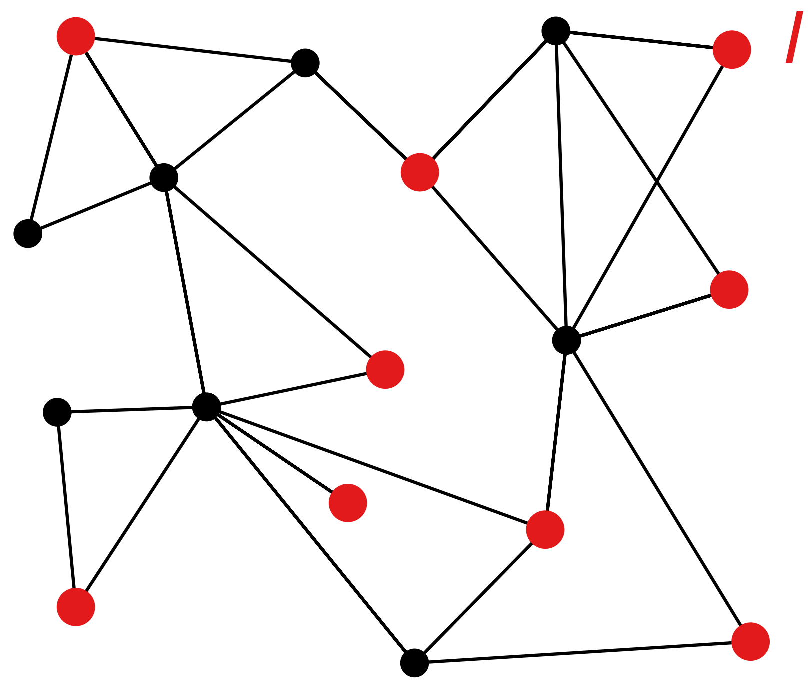
Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



Independent Sets

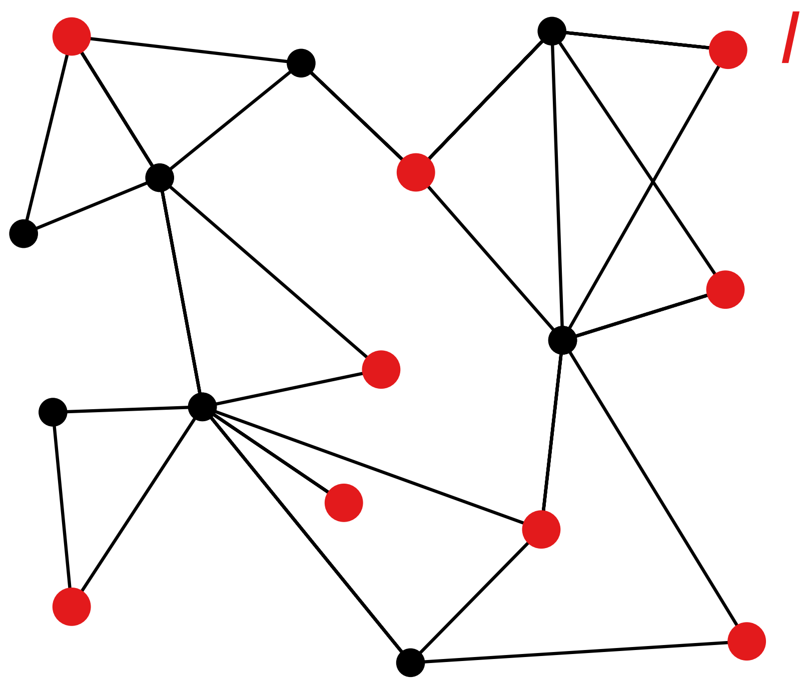
Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



Obs. Maximal independent sets are

Independent Sets

Def. A vertex set I in a graph is called **independent** (or **stable**) if no pair of vertices in I forms an edge.
An independent set is called **maximal** if it does not have an independent superset.



Obs. Maximal independent sets are dominating. :-)

Independent Sets in H^2

Lemma. For a graph H and an independent set I in H^2,
$|I| \leq$

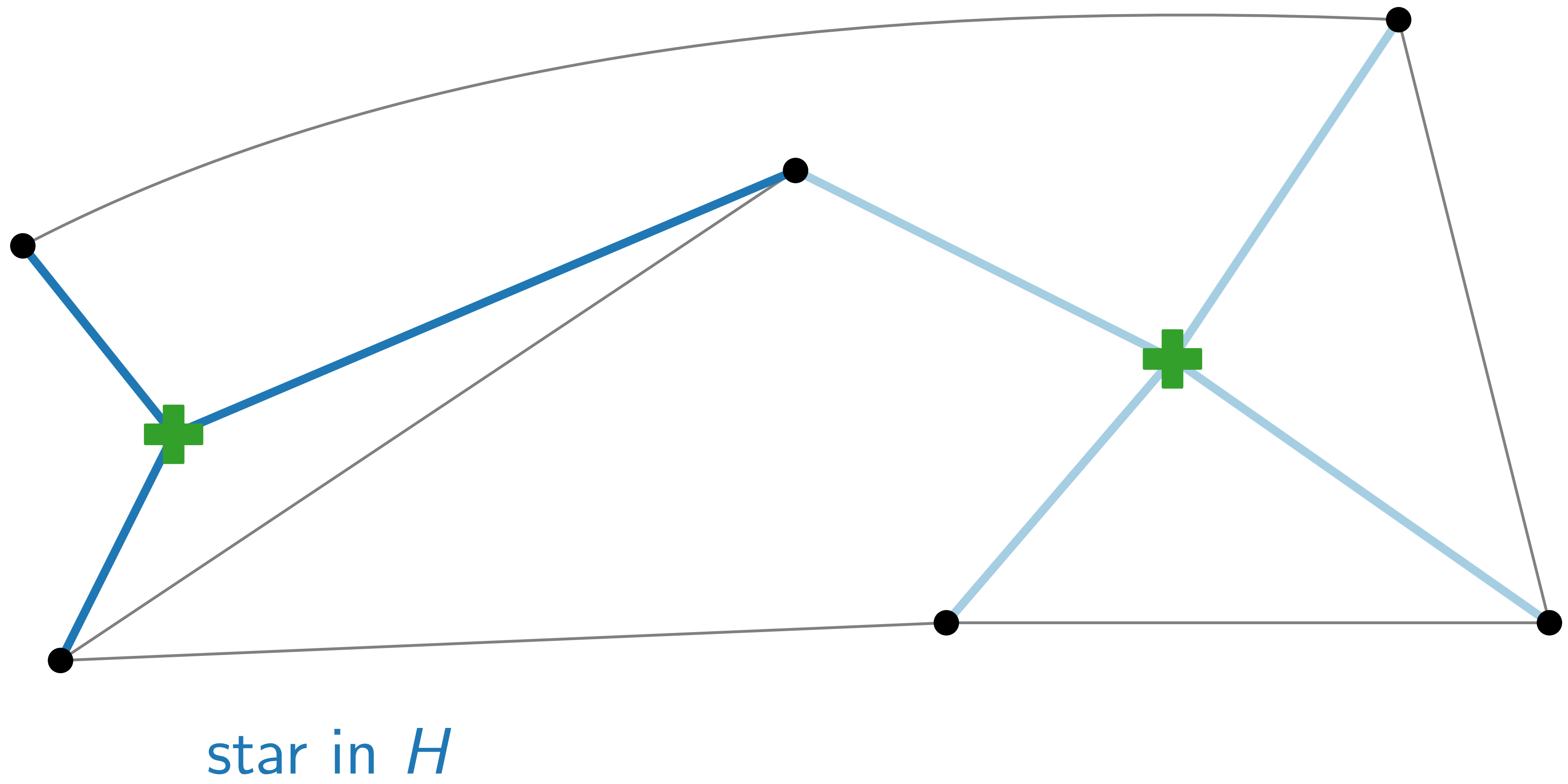
Independent Sets in H^2

Lemma. For a graph H and an independent set I in H^2, $|I| \leq \text{dom}(H)$.

Independent Sets in H^2

Lemma. For a graph H and an independent set I in H^2, $|I| \leq \mathrm{dom}(H)$.

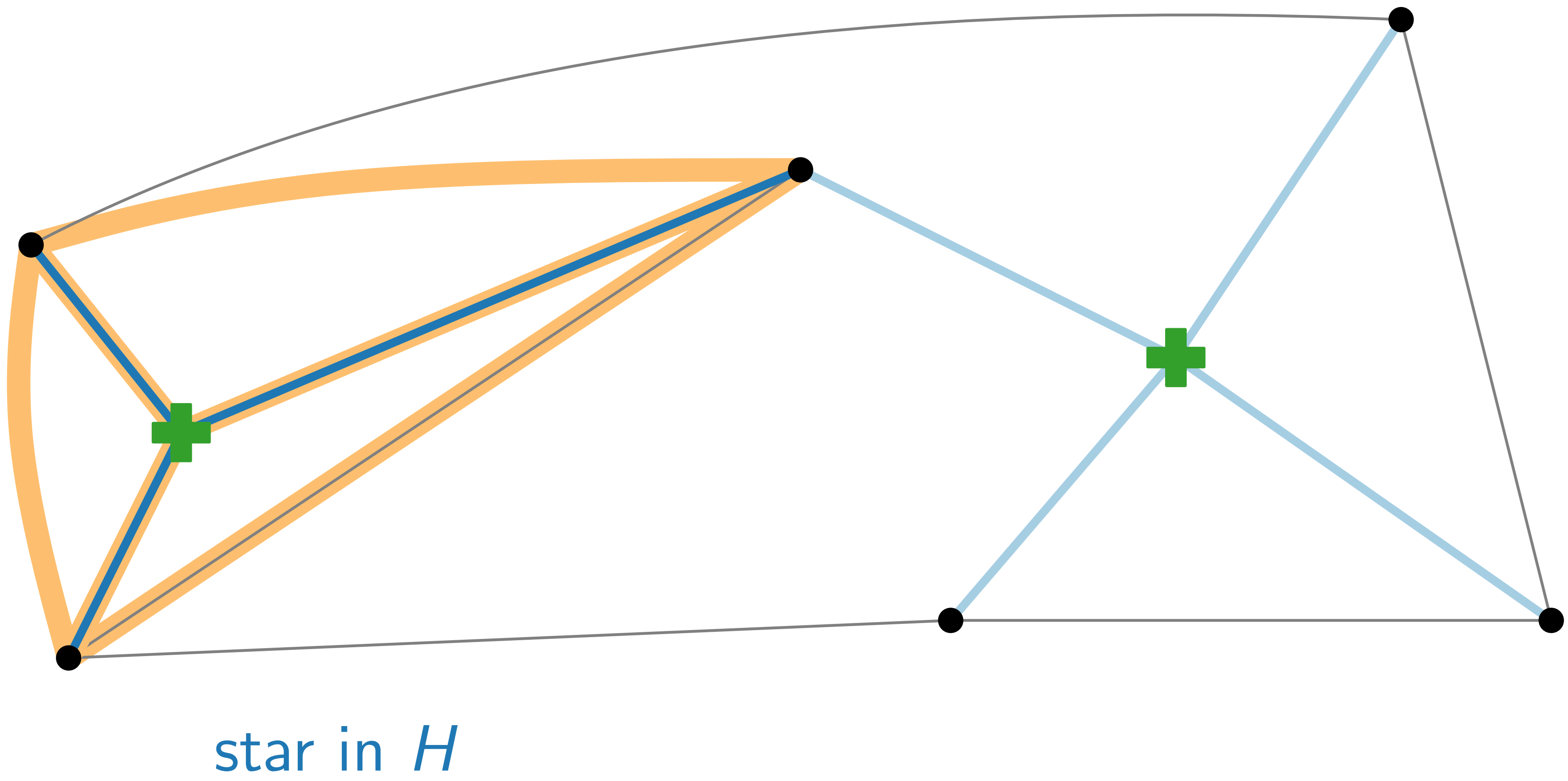
Proof. What does a dominating set of H look like in H^2?



Independent Sets in H^2

Lemma. For a graph H and an independent set I in H^2, $|I| \leq \mathrm{dom}(H)$.

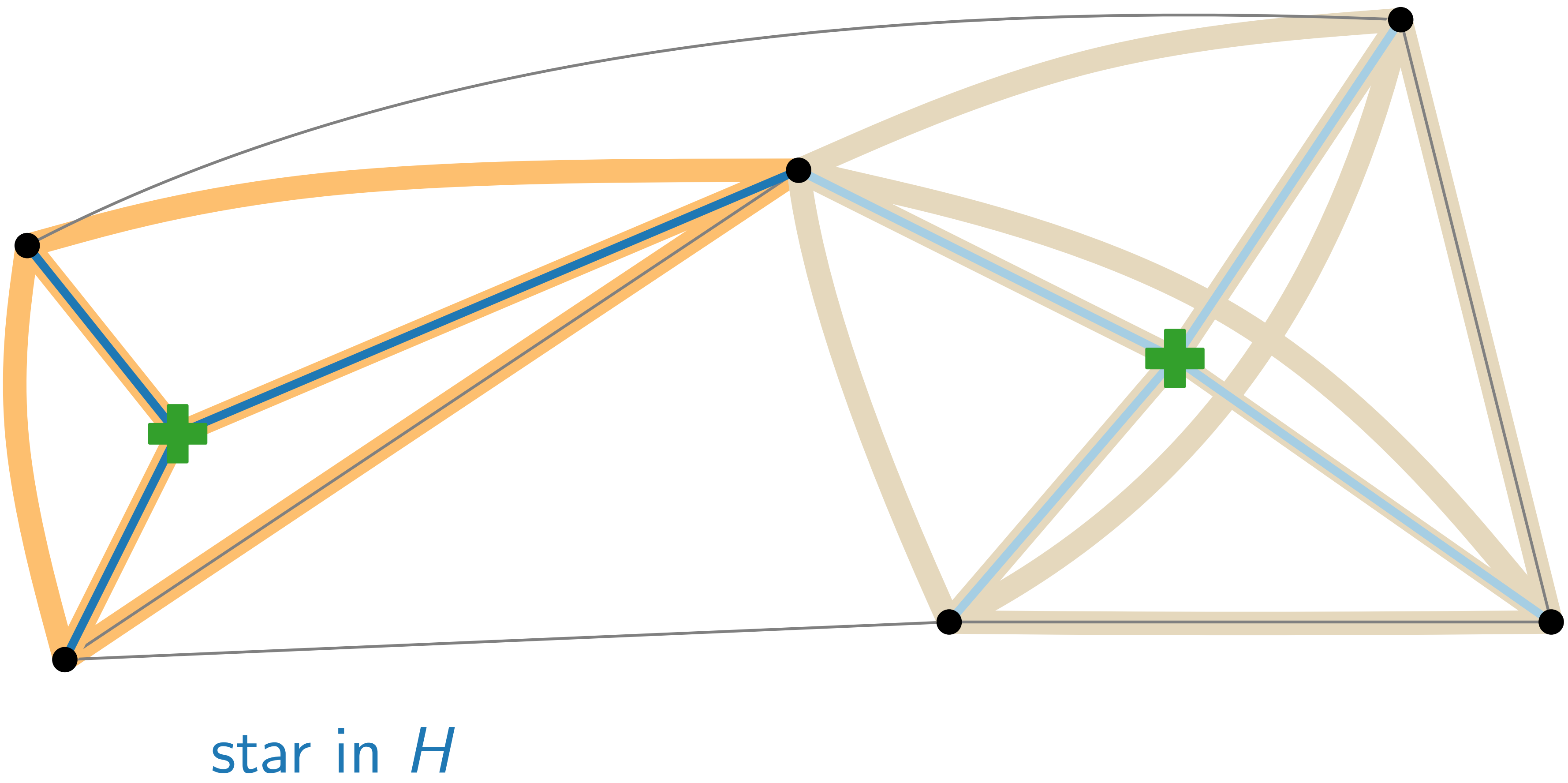
Proof. What does a dominating set of H look like in H^2?



Independent Sets in H^2

Lemma. For a graph H and an independent set I in H^2, $|I| \leq \text{dom}(H)$.

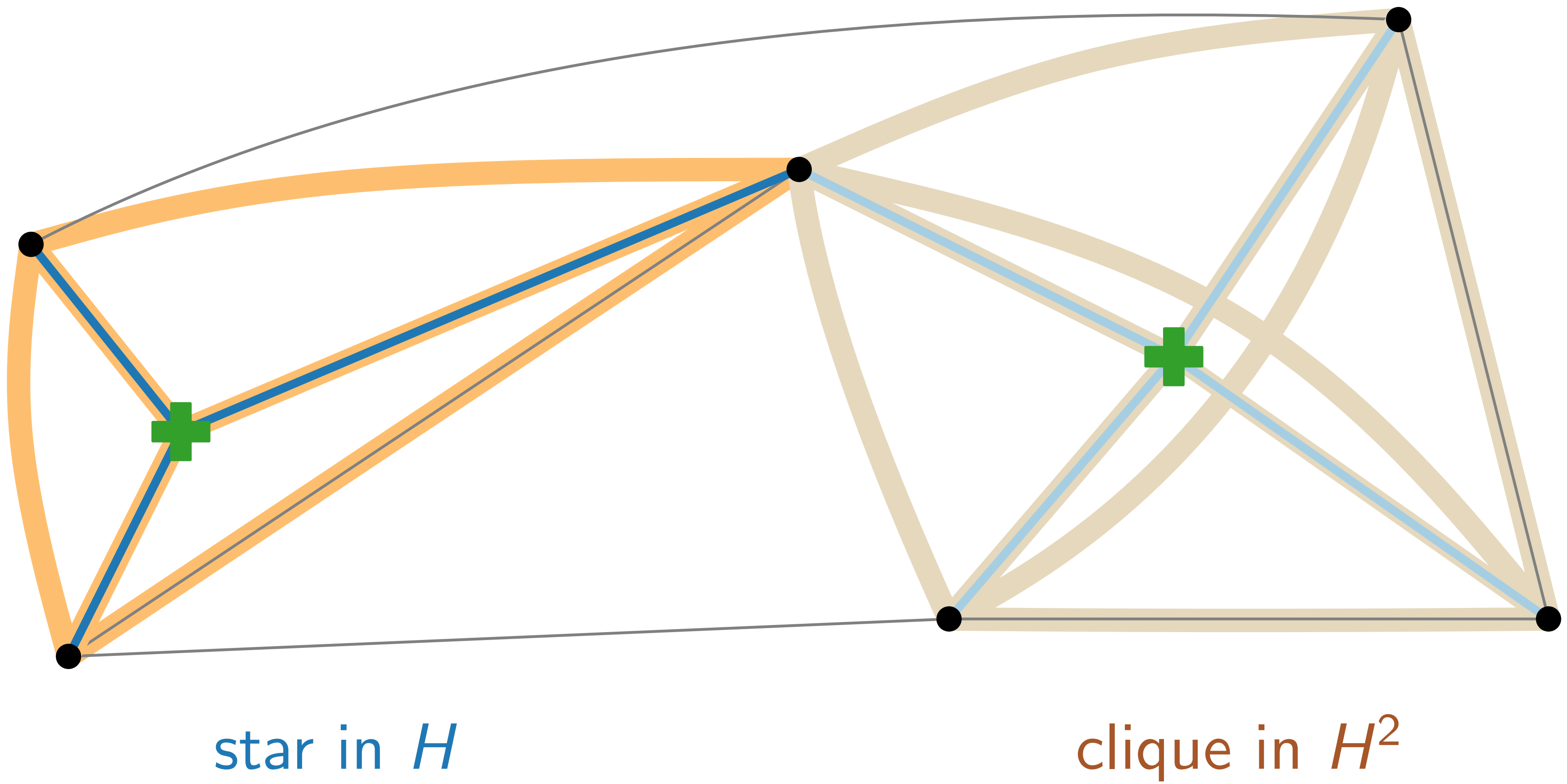
Proof. What does a dominating set of H look like in H^2?



Independent Sets in H^2

Lemma. For a graph H and an independent set I in H^2, $|I| \leq \mathrm{dom}(H)$.

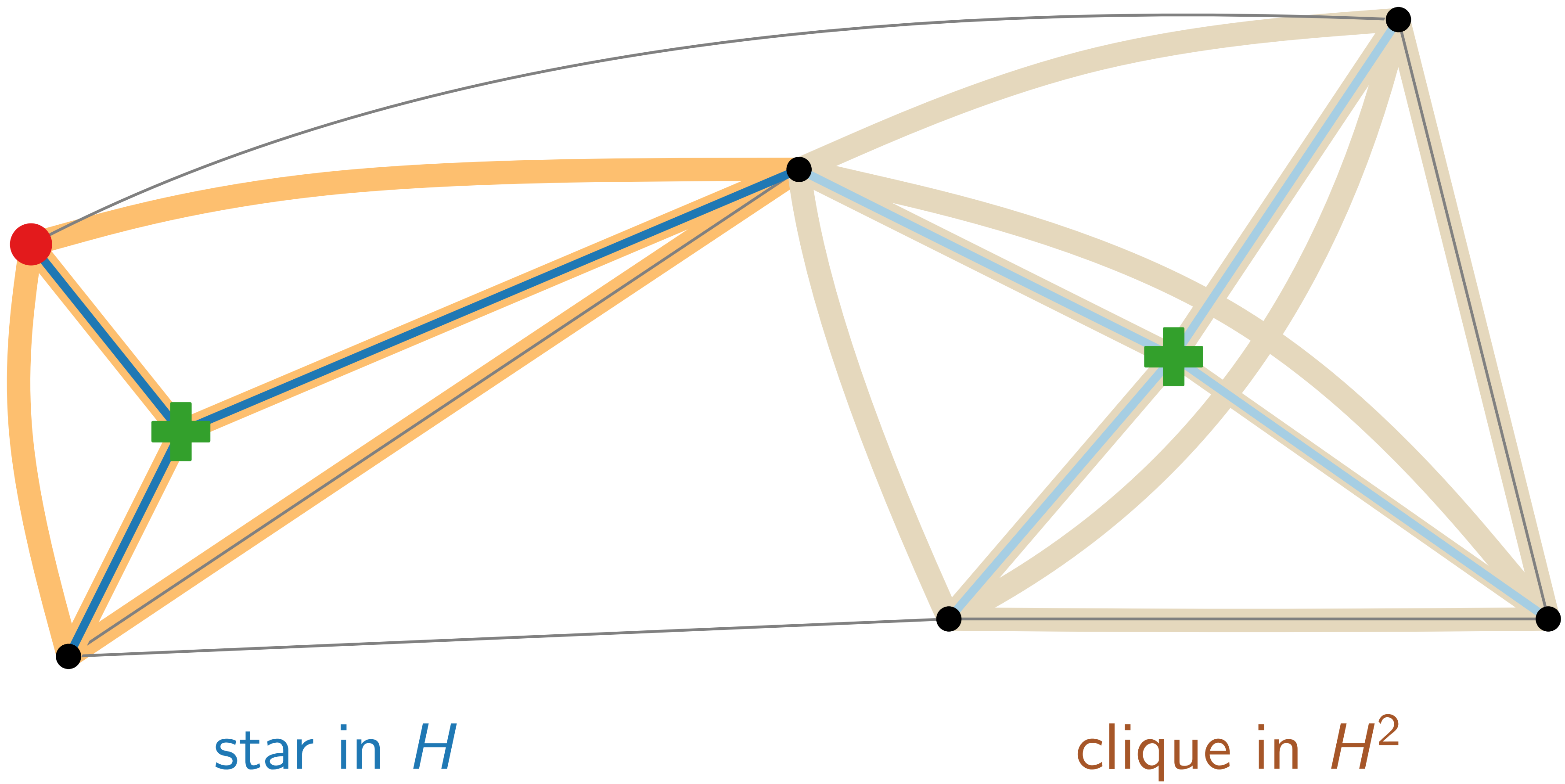
Proof. What does a dominating set of H look like in H^2?



Independent Sets in H^2

Lemma. For a graph H and an independent set I in H^2, $|I| \leq \mathrm{dom}(H)$.

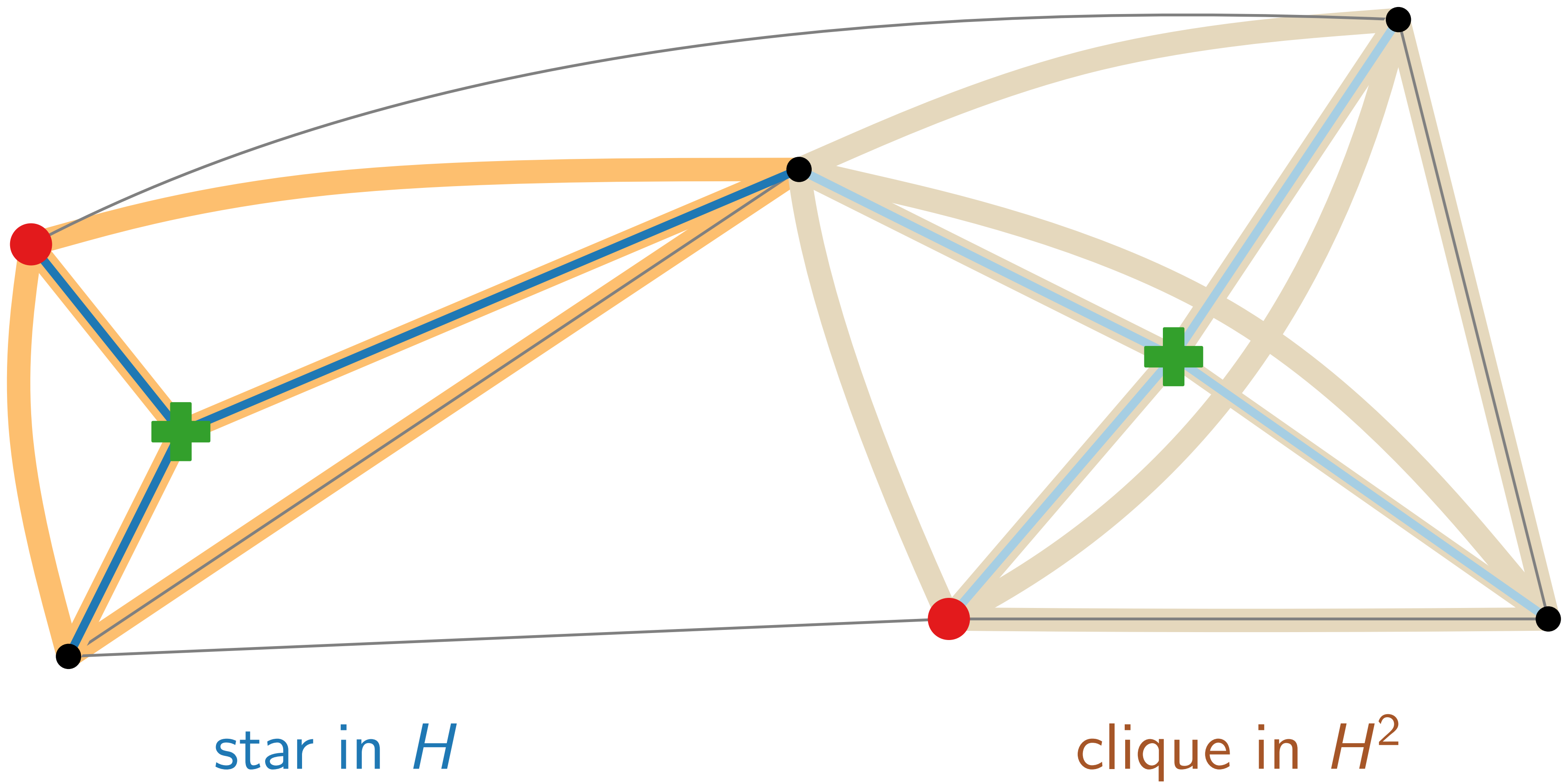
Proof. What does a dominating set of H look like in H^2?



Independent Sets in H^2

Lemma. For a graph H and an independent set I in H^2, $|I| \leq \mathrm{dom}(H)$.

Proof. What does a dominating set of H look like in H^2?



Approximation Algorithms

Lecture 6:
k-Center via Parametric Pruning

Part IV:
Factor-2 Approximation for Metric-k-Center

Factor-2 Approx. for Metric k-Center

Metric-k-Center-Approx(G, c, k)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$.

Factor-2 Approx. for Metric k-Center

Metric-k-Center-Approx(G, c, k)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$.

for $j = 1$ **to** m **do**

Factor-2 Approx. for Metric k-Center

Metric-k-Center-Approx(G, c, k)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$.

for $j = 1$ **to** m **do**

Construct G_j^2.

Factor-2 Approx. for Metric k-CENTER

Metric-k-CENTER-Approx(G, c, k)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$.

for $j = 1$ **to** m **do**

Construct G_j^2.

Find a maximal independent set I_j in G_j^2.

Factor-2 Approx. for Metric k-CENTER

Metric-k-CENTER-Approx(G, c, k)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$.
for $j = 1$ **to** m **do**
  Construct G_j^2.
  Find a maximal independent set I_j in G_j^2.
  if $|I_j| \leq k$ **then**
    return I_j

Factor-2 Approx. for Metric k-CENTER

Metric-k-CENTER-Approx(G, c, k)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$.
for $j = 1$ **to** m **do**
- Construct G_j^2.
- Find a maximal independent set I_j in G_j^2.
- **if** $|I_j| \leq k$ **then**
  - **return** I_j

Lemma. For j provided by the algorithm, it holds that $c(e_j) \leq \mathrm{OPT}$.

Factor-2 Approx. for Metric k-Center

Metric-k-Center-Approx(G, c, k)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$.
for $j = 1$ **to** m **do**
    Construct G_j^2.
    Find a maximal independent set I_j in G_j^2.
    if $|I_j| \leq k$ **then**
        return I_j

Lemma. For j provided by the algorithm, it holds that $c(e_j) \leq \mathrm{OPT}$.

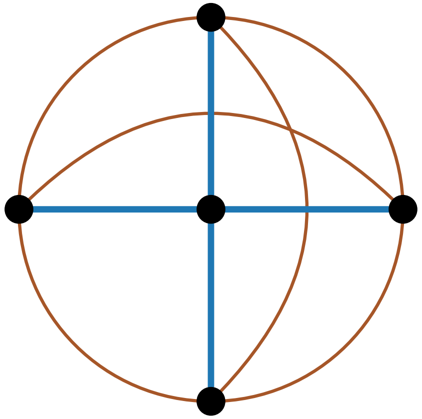
Theorem. The above algorithm is a factor-2 approximation algorithm for the metric k-Center problem.

Can we do better . . . ?

Can we do better . . . ?

What about a tight example?

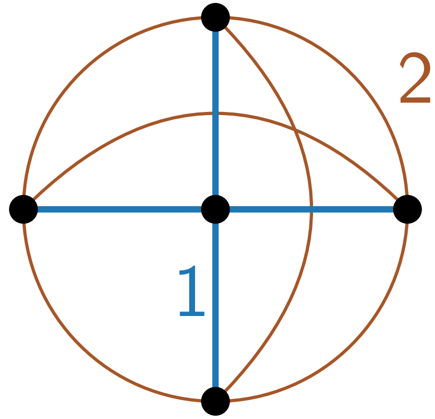
Can we do better . . . ?



What about a tight example?

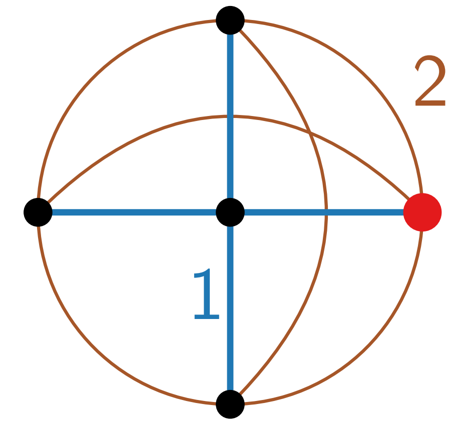
Can we do better . . . ?

What about a tight example?

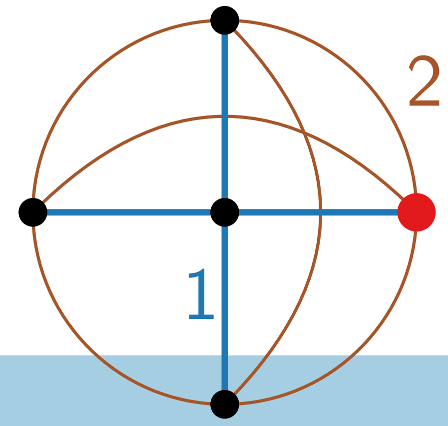


Can we do better . . . ?

What about a tight example?



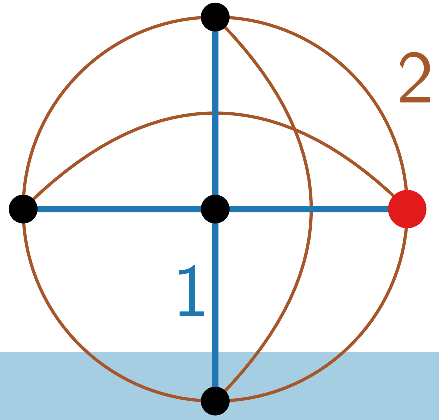
Can we do better . . . ?



What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2 - \varepsilon)$-approximation algorithm for the metric k-Center problem.

Can we do better . . . ?

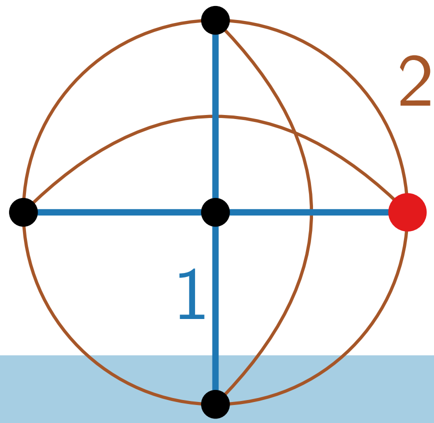


What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2 - \varepsilon)$-approximation algorithm for the metric k-Center problem.

Proof. Reduce from Dominating Set to metric k-Center.

Can we do better . . . ?

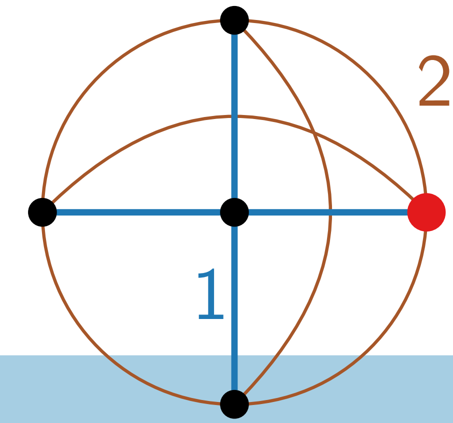


What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2 - \varepsilon)$-approximation algorithm for the metric k-Center problem.

Proof. Reduce from Dominating Set to metric k-Center. Given graph G and integer k,

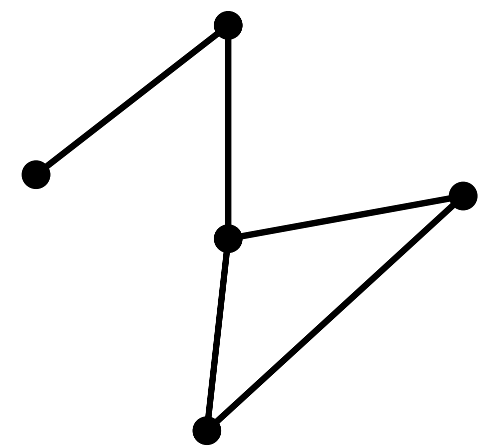
Can we do better . . . ?



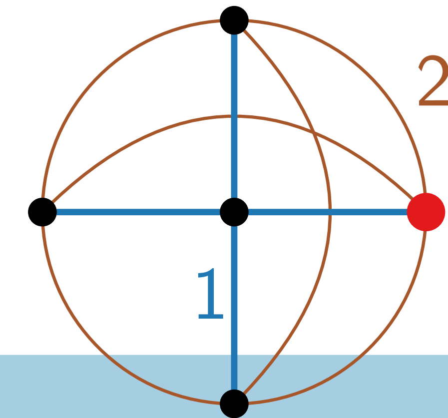
What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2 - \varepsilon)$-approximation algorithm for the metric k-CENTER problem.

Proof. Reduce from DOMINATING SET to metric k-CENTER.
Given graph G and integer k,



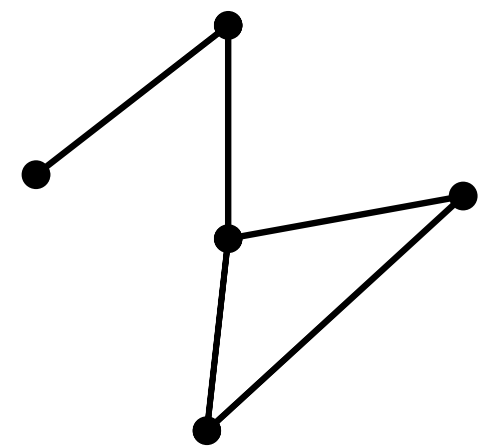
Can we do better . . . ?



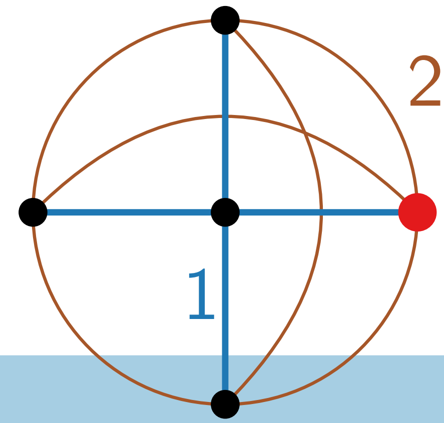
What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2 - \varepsilon)$-approximation algorithm for the metric k-CENTER problem.

Proof. Reduce from DOMINATING SET to metric k-CENTER. Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.



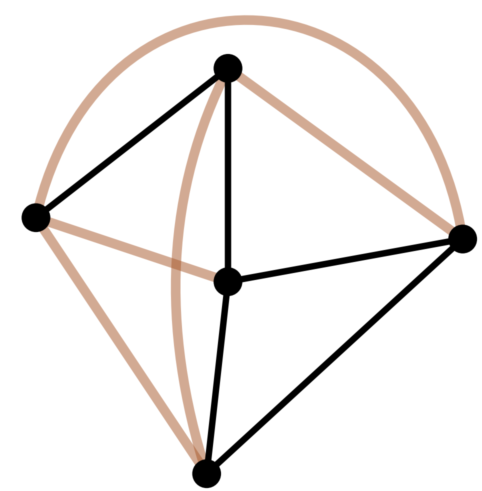
Can we do better . . . ?



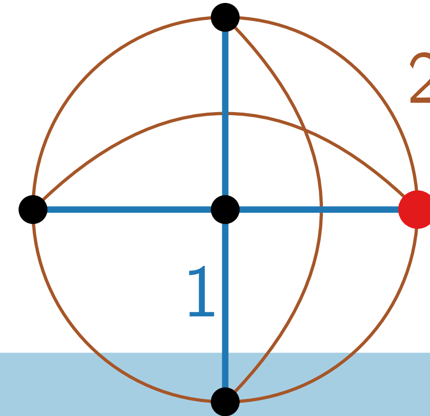
What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2 - \varepsilon)$-approximation algorithm for the metric k-Center problem.

Proof. Reduce from Dominating Set to metric k-Center. Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.



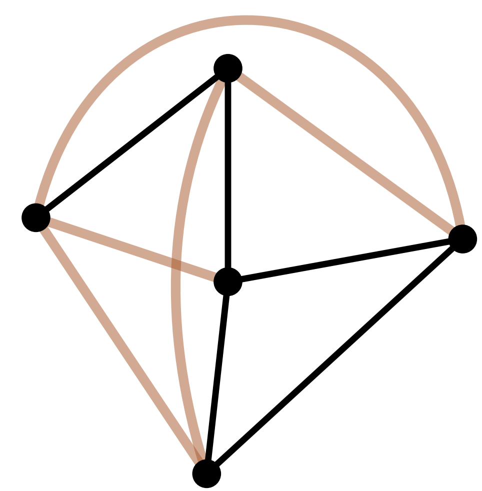
Can we do better . . . ?



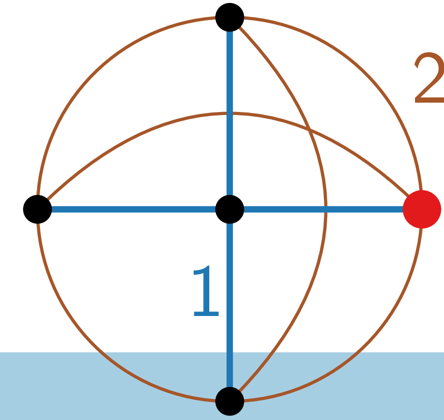
What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2-\varepsilon)$-approximation algorithm for the metric k-CENTER problem.

Proof. Reduce from DOMINATING SET to metric k-CENTER.
Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.
with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$



Can we do better . . . ?

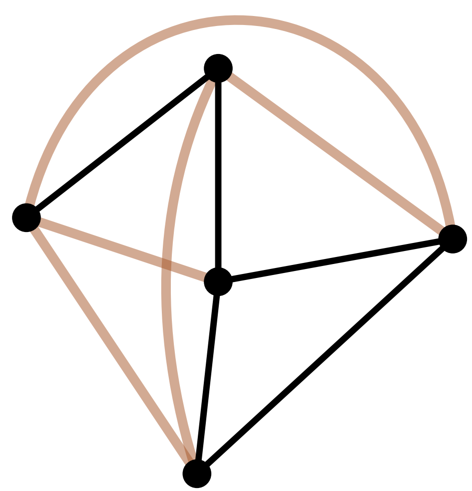


What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2 - \varepsilon)$-approximation algorithm for the metric k-Center problem.

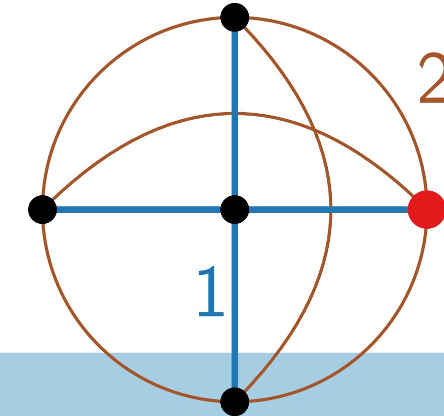
Proof. Reduce from Dominating Set to metric k-Center.
Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.

with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$



Let S be a metric k-center of G'.

Can we do better . . . ?

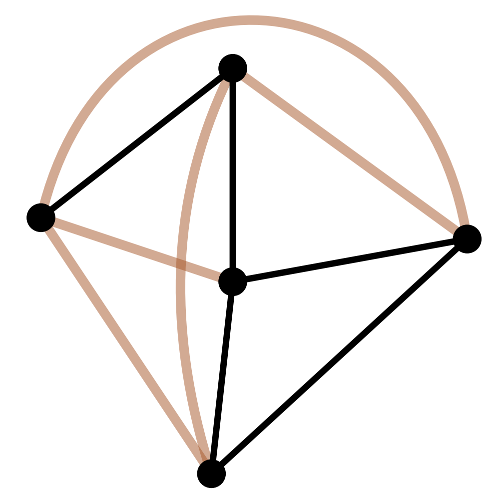


What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2-\varepsilon)$-approximation algorithm for the metric k-CENTER problem.

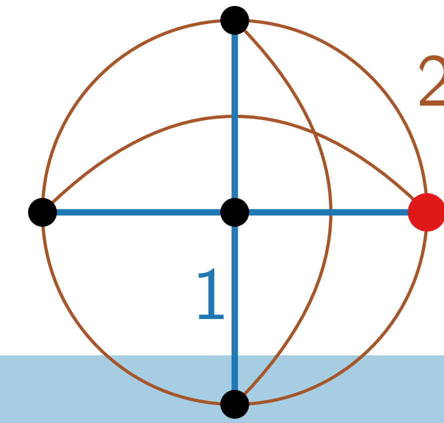
Proof. Reduce from DOMINATING SET to metric k-CENTER. Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.

with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$



Let S be a metric k-center of G'.
If $\text{dom}(G) \leq k$, then $\text{cost}(S) = 1$.

Can we do better . . . ?

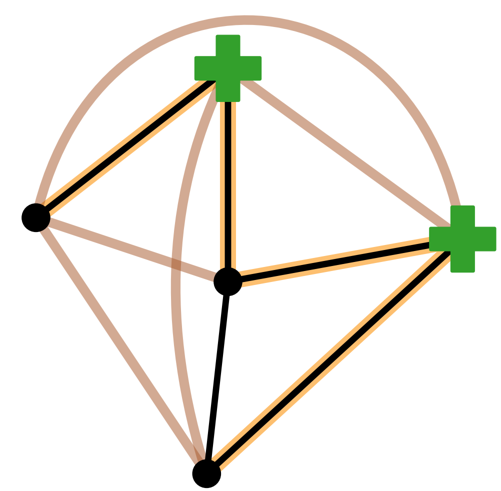


What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2 - \varepsilon)$-approximation algorithm for the metric k-Center problem.

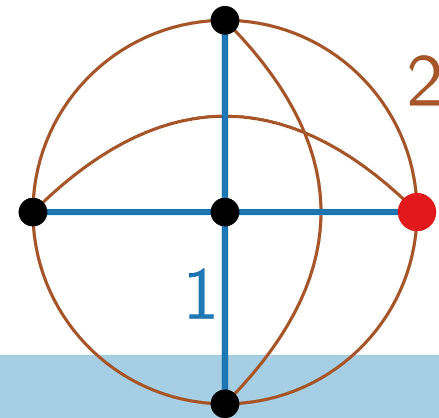
Proof. Reduce from Dominating Set to metric k-Center. Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.

with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$



Let S be a metric k-center of G'.
If $\text{dom}(G) \leq k$, then $\text{cost}(S) = 1$.

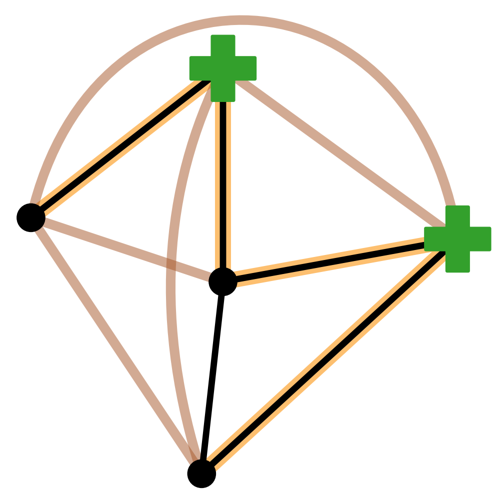
Can we do better . . . ?



What about a tight example?

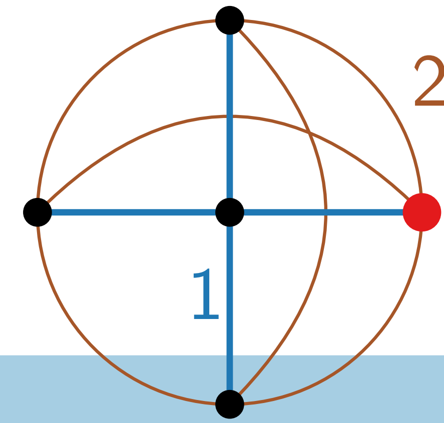
Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2-\varepsilon)$-approximation algorithm for the metric k-CENTER problem.

Proof. Reduce from DOMINATING SET to metric k-CENTER.
Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.
with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$



Let S be a metric k-center of G'.
If $\mathrm{dom}(G) \leq k$, then $\mathrm{cost}(S) = 1$.
If $\mathrm{dom}(G) > k$, then $\mathrm{cost}(S) = 2$.

Can we do better . . . ?



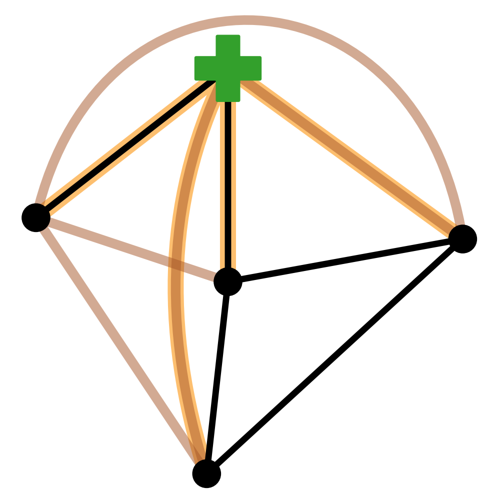
What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2-\varepsilon)$-approximation algorithm for the metric k-CENTER problem.

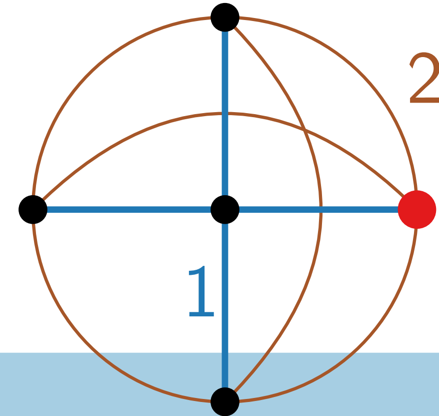
Proof. Reduce from DOMINATING SET to metric k-CENTER.
Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.

with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$

Let S be a metric k-center of G'.
If $\mathrm{dom}(G) \leq k$, then $\mathrm{cost}(S) = 1$.
If $\mathrm{dom}(G) > k$, then $\mathrm{cost}(S) = 2$.



Can we do better . . . ?

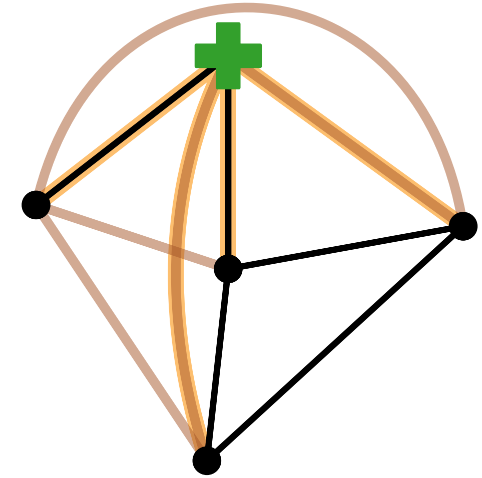


What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2-\varepsilon)$-approximation algorithm for the metric k-CENTER problem.

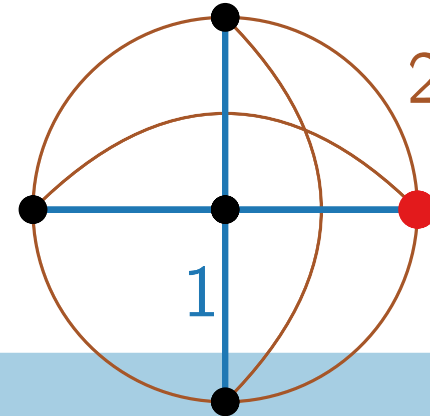
Proof. Reduce from DOMINATING SET to metric k-CENTER. Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.

with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$



Let S be a metric k-center of G'.
If $\text{dom}(G) \leq k$, then $\text{cost}(S) = 1$.
If $\text{dom}(G) > k$, then $\text{cost}(S) = 2$.

Can we do better . . . ?



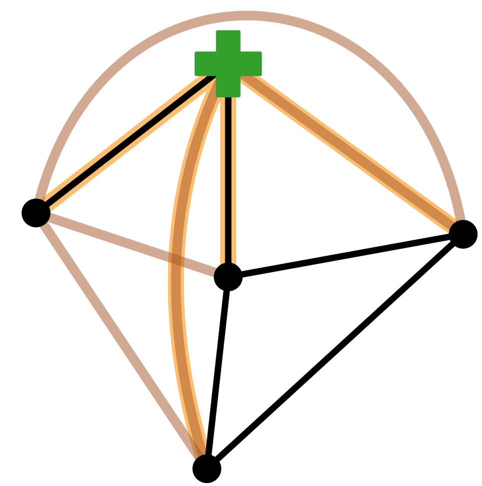
What about a tight example?

Theorem. Assuming P $\neq$ NP, for no $\varepsilon > 0$, there is a $(2 - \varepsilon)$-approximation algorithm for the metric k-CENTER problem.

Proof. Reduce from DOMINATING SET to metric k-CENTER. Given graph G and integer k, construct complete graph G' with $V(G') = V(G)$, $E(G') = E(G) \cup E'$.

with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$

$\triangle$-inequality holds



Let S be a metric k-center of G'.
If $\text{dom}(G) \leq k$, then $\text{cost}(S) = 1$.
If $\text{dom}(G) > k$, then $\text{cost}(S) = 2$.

Approximation Algorithms

Lecture 6:
k-Center via Parametric Pruning

Part V:
Metric-Weighted-Center

Metric-k-Center

Given: A complete graph G with metric edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ and an integer $k \leq |V|$.

For $S \subseteq V(G)$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

Find: A k-element vertex set S such that $\text{cost}(S) := \max_{v \in V(G)} c(v, S)$ is minimized.

Metric-k-Center

Weighted

Given: A complete graph G with metric edge costs $c \colon E(G) \to \mathbb{Q}_{\geq 0}$ and an integer $k \leq |V|$.

For $S \subseteq V(G)$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

Find: A k-element vertex set S such that $\mathrm{cost}(S) := \max_{v \in V(G)} c(v, S)$ is minimized.

Metric-~~k~~-Center

Weighted

Given: A complete graph G with metric edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ and ~~an integer $k \leq |V|$~~, vertex weights $w\colon V \to \mathbb{Q}_{\geq 0}$, and a budget $W \in \mathbb{Q}_{+}$

For $S \subseteq V(G)$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

Find: A k-element vertex set S such that $\text{cost}(S) := \max_{v \in V(G)} c(v, S)$ is minimized.

Metric-~~k~~-Center

Weighted

Given: A complete graph G with metric edge costs $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ and ~~an integer $k \leq |V|$~~, vertex weights $w\colon V \to \mathbb{Q}_{\geq 0}$, and a budget $W \in \mathbb{Q}_+$

For $S \subseteq V(G)$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.

vertex set S of weight at most W

Find: A ~~k-element vertex set S~~ such that $\text{cost}(S) := \max_{v \in V(G)} c(v, S)$ is minimized.

Algorithm for the Weighted Version

Algorithm Metric-k CENTER-Approx(G, c, k)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

  Construct G_j^2

  Find a maximal independent set I_j in G_j^2

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

    return I_j

Algorithm for the Weighted Version

Algorithm Metric-Weighted-CENTER-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

Construct G_j^2

Find a maximal independent set I_j in G_j^2

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

return I_j

Algorithm for the Weighted Version

Algorithm Metric-Weighted-Center-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

Construct G_j^2

Find a maximal independent set I_j in G_j^2

what about the weights?

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

return I_j

Algorithm for the Weighted Version

Algorithm Metric-Weighted-CENTER-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

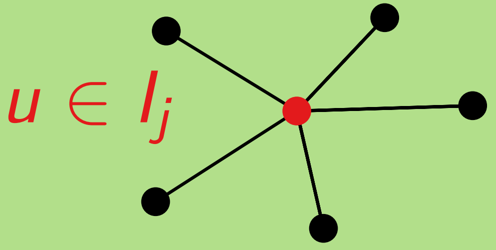
Construct G_j^2

Find a maximal independent set I_j in G_j^2

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

return I_j

what about the weights?



Algorithm for the Weighted Version

Algorithm Metric-Weighted-Center-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

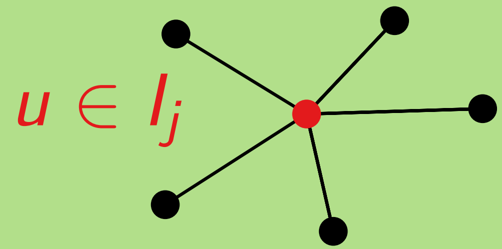
  Construct G_j^2

  Find a maximal independent set I_j in G_j^2

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

    return I_j

what about the weights?



$s_j(u) :=$ lightest node in $N_{G_j}(u) \cup \{u\}$

Algorithm for the Weighted Version

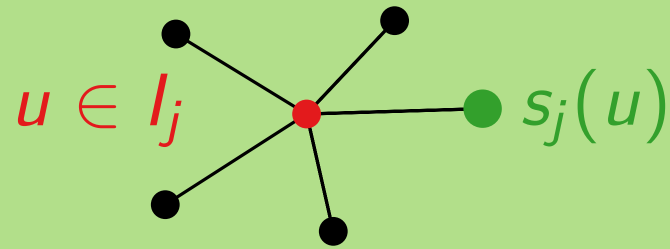
Algorithm Metric-Weighted-Center-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

- Construct G_j^2
- Find a maximal independent set I_j in G_j^2
- **if** $|I_j| \leq k$ **then**
  - **return** I_j

what about the weights?



$s_j(u) :=$ lightest node in $N_{G_j}(u) \cup \{u\}$

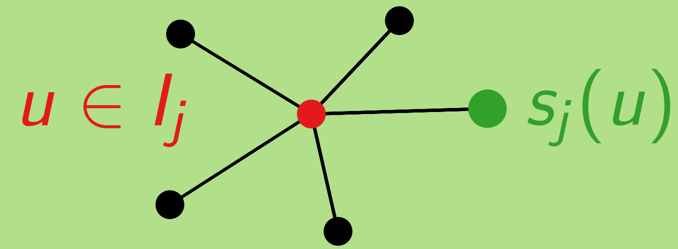
Algorithm for the Weighted Version

Algorithm Metric-Weighted-Center-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

- Construct G_j^2
- Find a maximal independent set I_j in G_j^2
- Compute $S_j := \{ s_j(u) \mid u \in I_j \}$
- **if** $|I_j| \leq k$ **then**
  - **return** I_j



$s_j(u)$:= lightest node in $N_{G_j}(u) \cup \{u\}$

Algorithm for the Weighted Version

Algorithm Metric-Weighted-Center-Approx(G, c, w, W)

  Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

  for $j = 1$ **to** m **do**

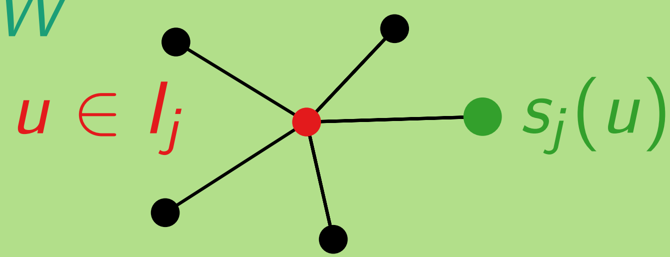
    Construct G_j^2

    Find a maximal independent set I_j in G_j^2

    Compute $S_j := \{ s_j(u) \mid u \in I_j \}$

    if ~~$|I_j| \leq k$~~ $w(S_j) \leq W$ **then**

      return I_j



$s_j(u) :=$ lightest node in $N_{G_j}(u) \cup \{u\}$

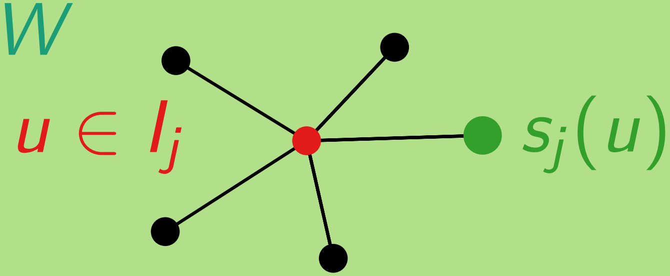
Algorithm for the Weighted Version

Algorithm Metric-Weighted-Center-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \le \cdots \le c(e_m)$

for $j = 1$ **to** m **do**

- Construct G_j^2
- Find a maximal independent set I_j in G_j^2
- Compute $S_j := \{ s_j(u) \mid u \in I_j \}$
- **if** ~~$|I_j| \le k$~~ $w(S_j) \le W$ **then**
  - **return** ~~I_j~~ S_j



$s_j(u) :=$ lightest node in $N_{G_j}(u) \cup \{u\}$

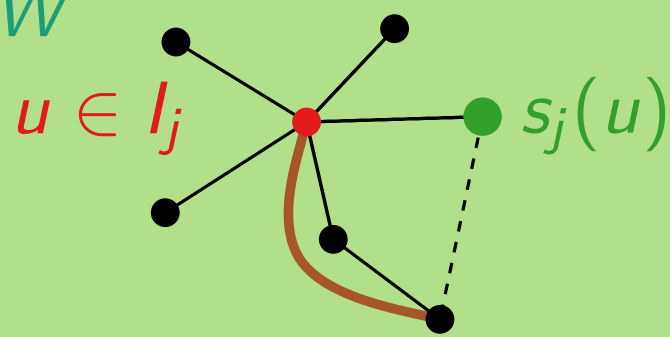
Algorithm for the Weighted Version

Algorithm Metric-Weighted-CENTER-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

- Construct G_j^2
- Find a maximal independent set I_j in G_j^2
- Compute $S_j := \{ s_j(u) \mid u \in I_j \}$
- **if** ~~$|I_j| \leq k$~~ $w(S_j) \leq W$ **then**
  - **return** ~~I_j~~ S_j



$s_j(u) :=$ lightest node in $N_{G_j}(u) \cup \{u\}$

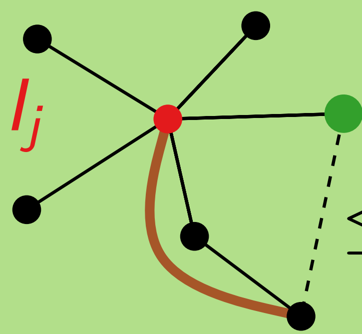
Algorithm for the Weighted Version

Algorithm Metric-Weighted-Center-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

- Construct G_j^2
- Find a maximal independent set I_j in G_j^2
- Compute $S_j := \{ s_j(u) \mid u \in I_j \}$
- **if** ~~$|I_j| \leq k$~~ $w(S_j) \leq W$ **then**
  - **return** ~~I_j~~ S_j



$s_j(u) :=$ lightest node in $N_{G_j}(u) \cup \{u\}$

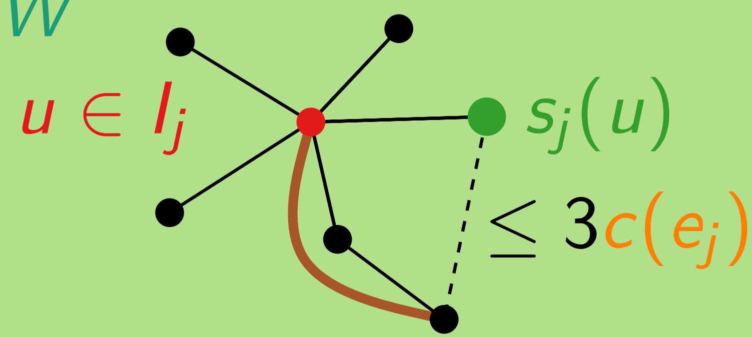
Algorithm for the Weighted Version

Algorithm Metric-Weighted-CENTER-Approx(G, c, w, W)

Sort the edges of G by cost: $c(e_1) \leq \cdots \leq c(e_m)$

for $j = 1$ **to** m **do**

- Construct G_j^2
- Find a maximal independent set I_j in G_j^2
- Compute $S_j := \{\, s_j(u) \mid u \in I_j \,\}$
- **if** ~~$|I_j| \leq k$~~ $w(S_j) \leq W$ **then**
  - **return** ~~I_j~~ S_j



$s_j(u) :=$ lightest node in $N_{G_j}(u) \cup \{u\}$

Theorem. The above is a factor-3 approximation algorithm for METRIC-WEIGHTED-CENTER.

Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

Tight Example. . . ?

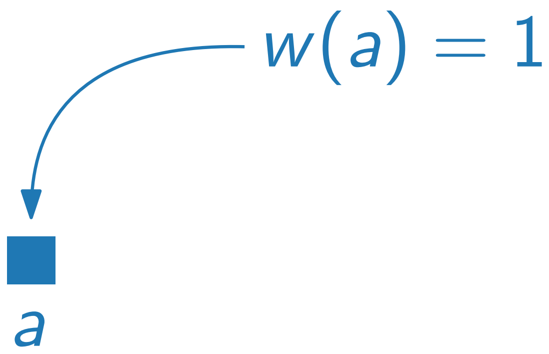
Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

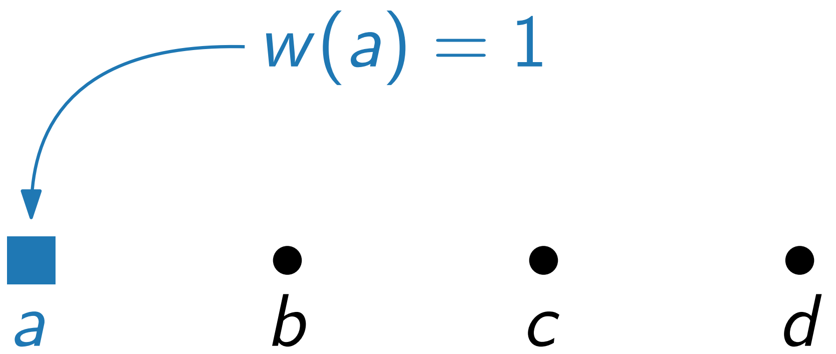
Consider $W = 3$.



Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

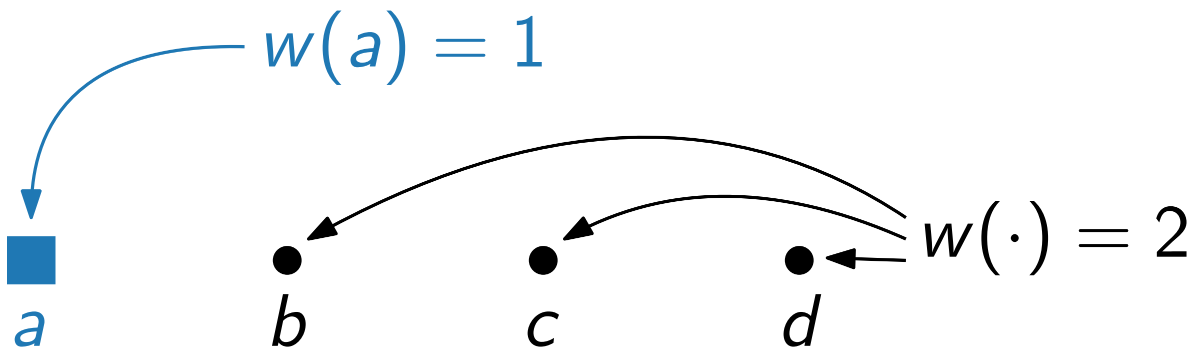
Consider $W = 3$.



Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

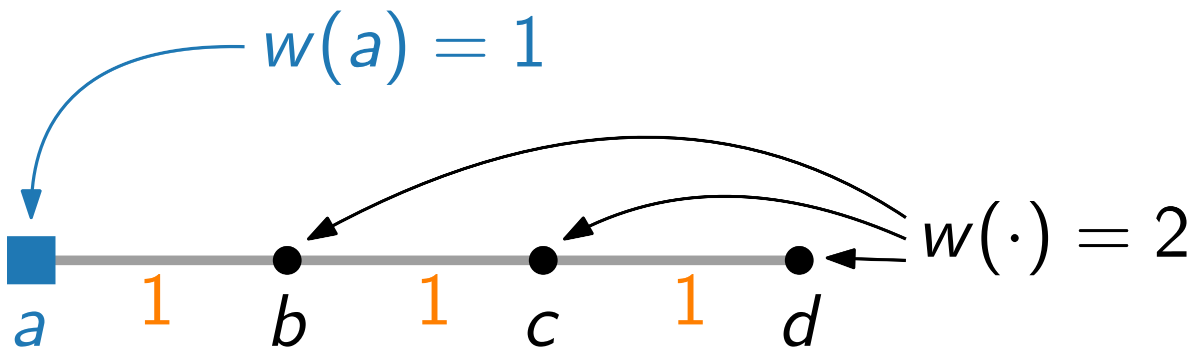
Consider $W = 3$.



Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

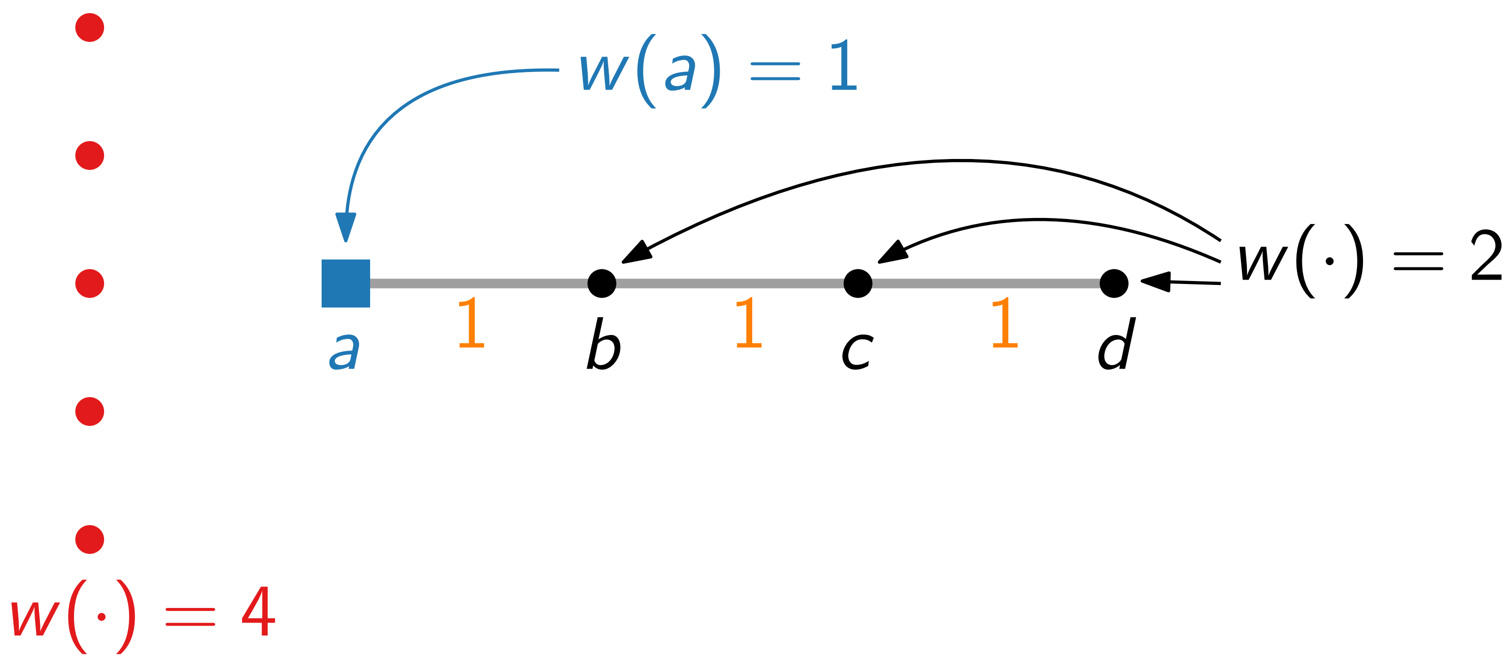
Consider $W = 3$.



Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

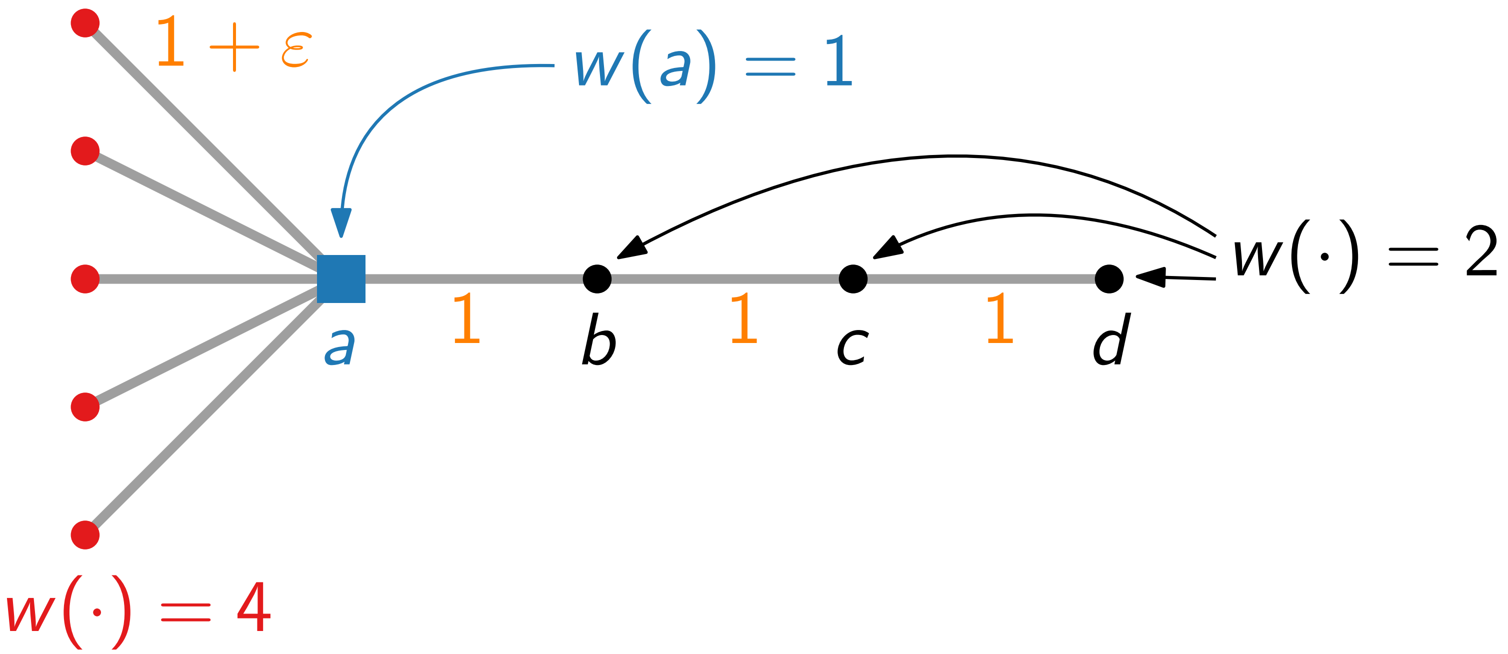
Consider $W = 3$.



Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

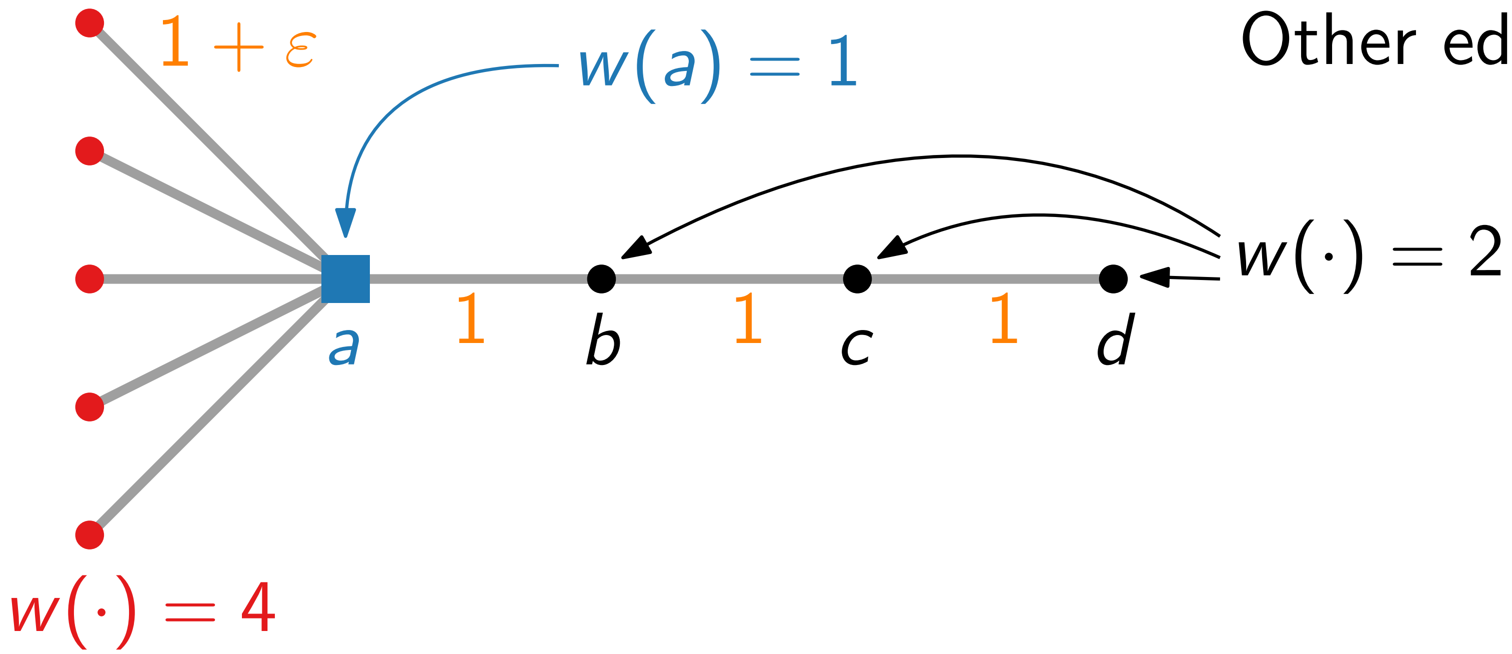


Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

Other edge costs?

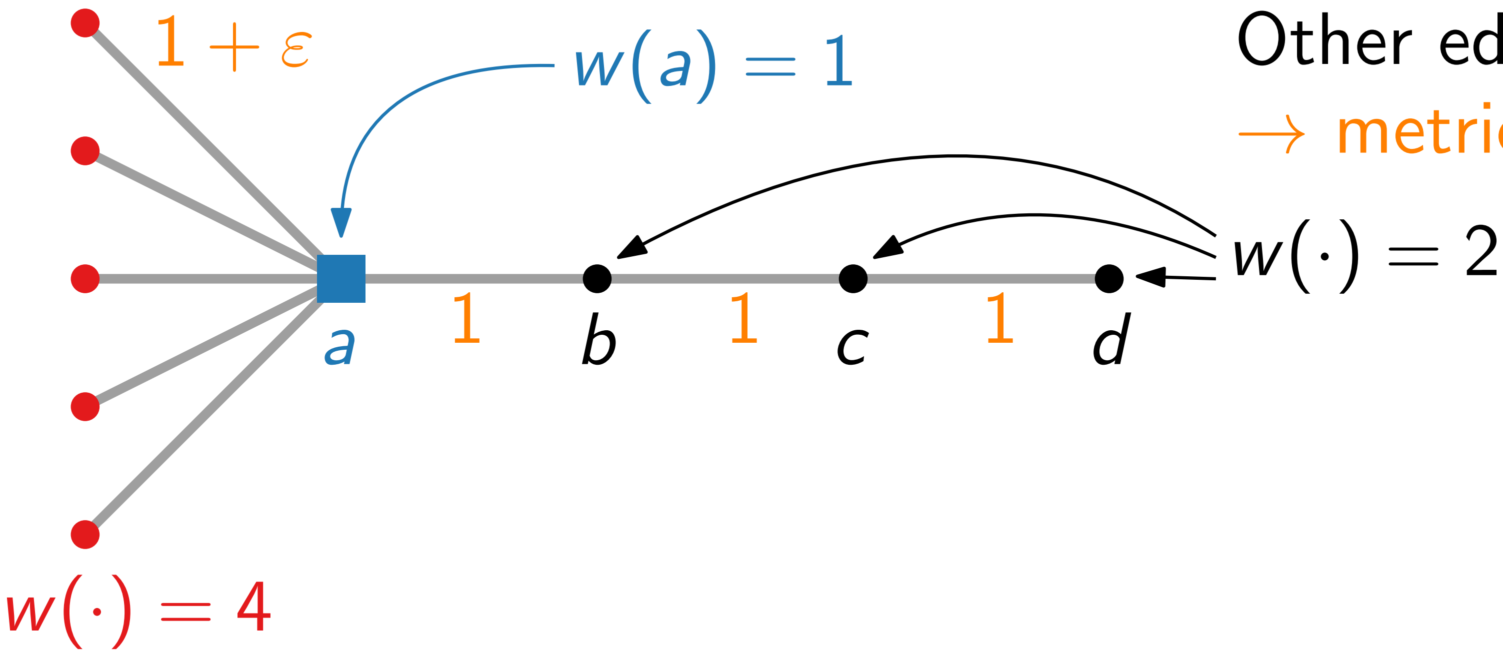


Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

Other edge costs?
$\rightarrow$ metric completion!

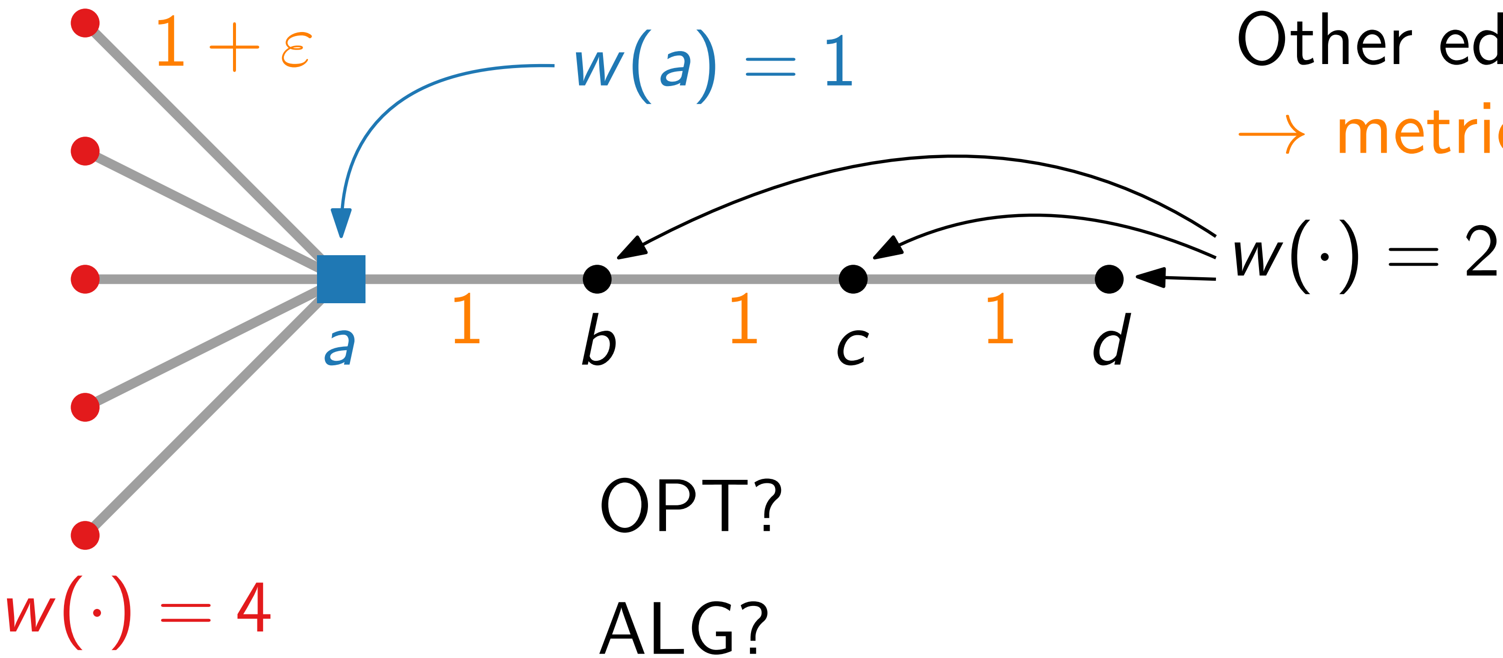


Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

Other edge costs?
$\rightarrow$ metric completion!

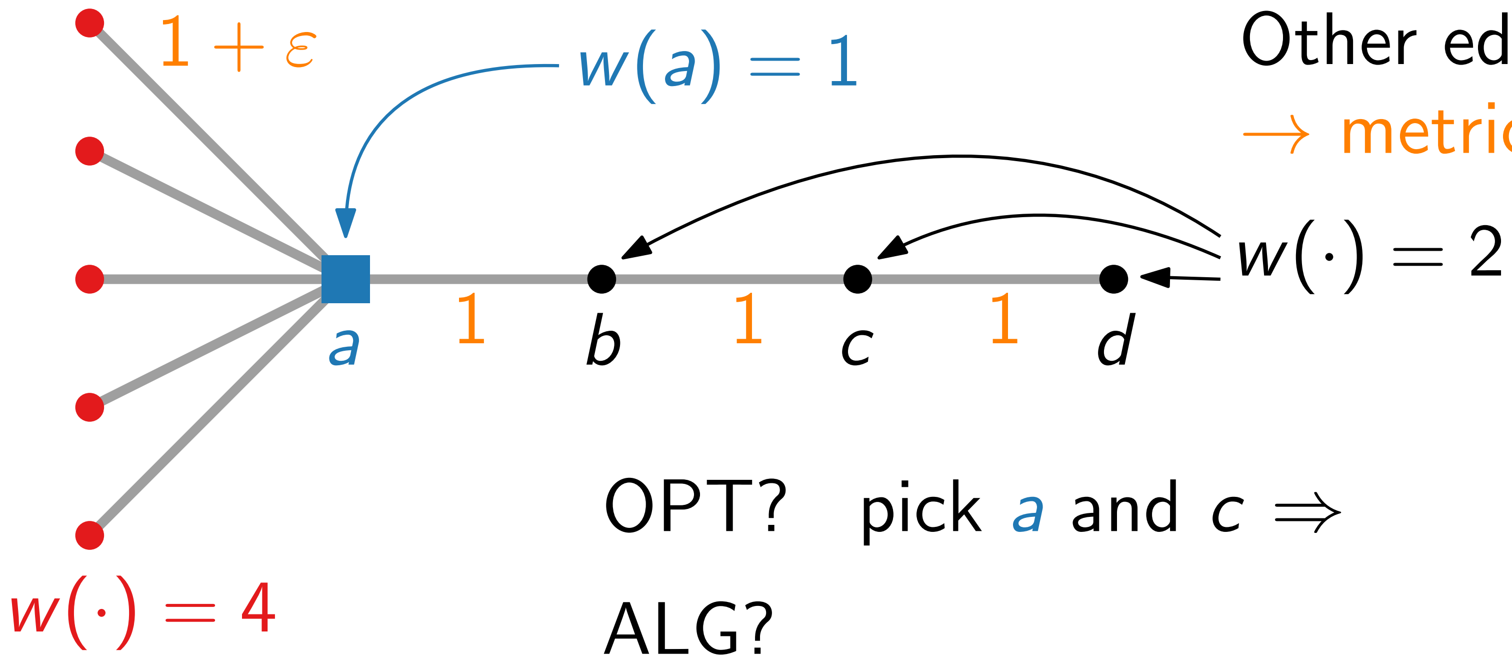


OPT?

ALG?

Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.



Consider $W=3$.

Other edge costs?

$\rightarrow$ metric completion!

OPT? pick a and c $\Rightarrow$

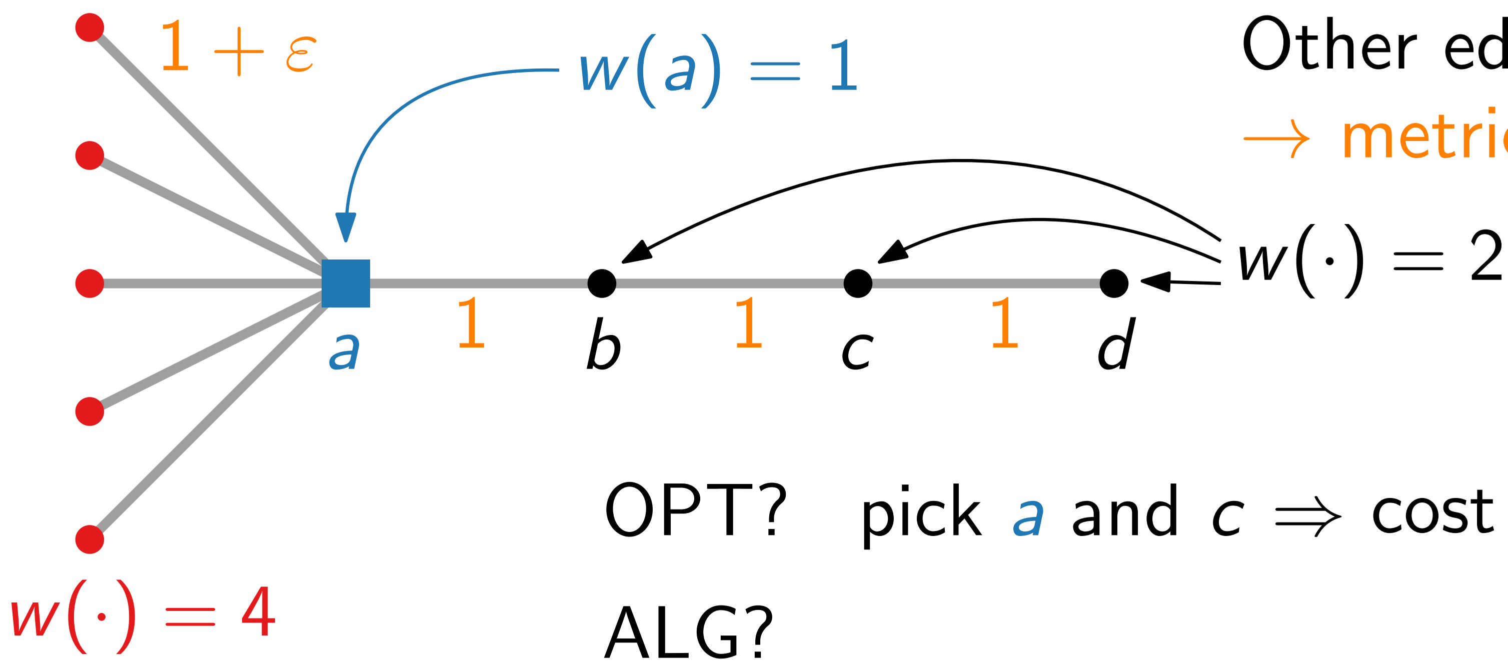
ALG?

Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

Other edge costs?
$\rightarrow$ metric completion!



OPT? pick a and c $\Rightarrow$ cost $1 + \varepsilon$.

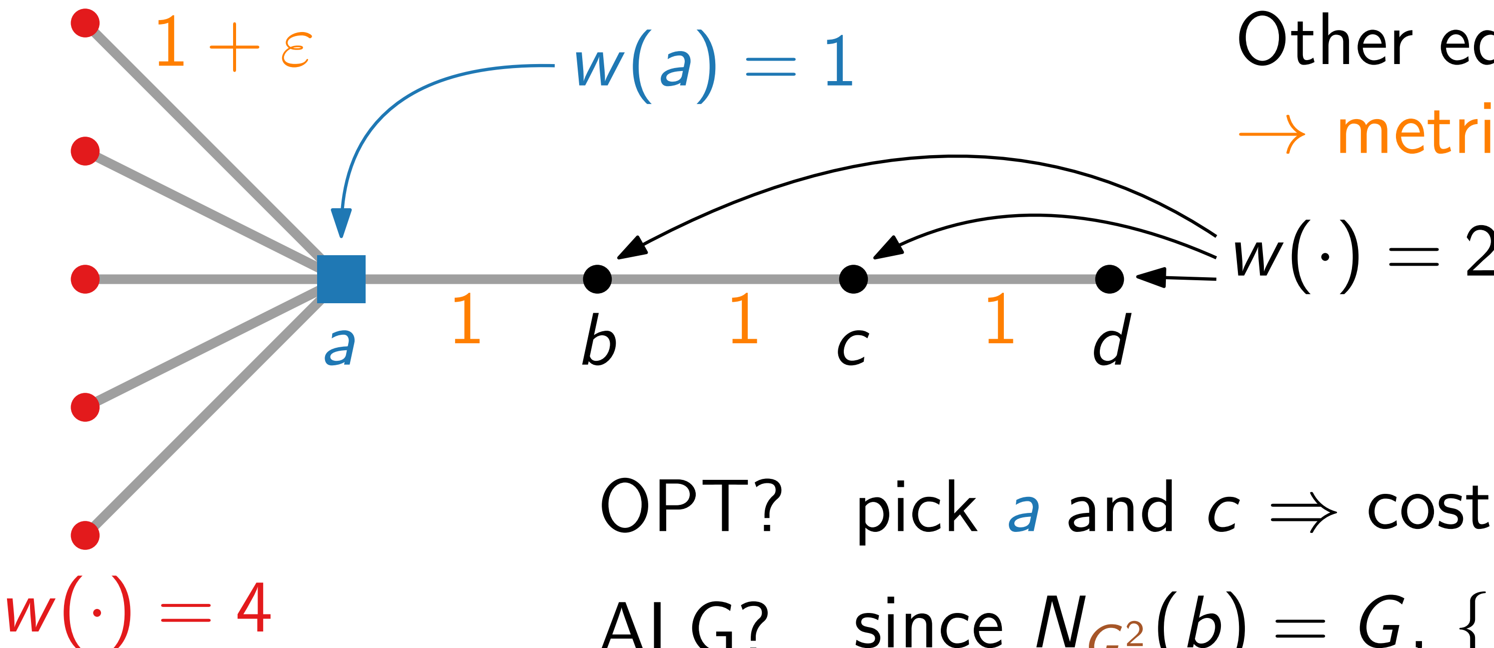
ALG?

Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

Other edge costs?
$\rightarrow$ metric completion!



OPT? pick a and $c \Rightarrow$ cost $1 + \varepsilon$.

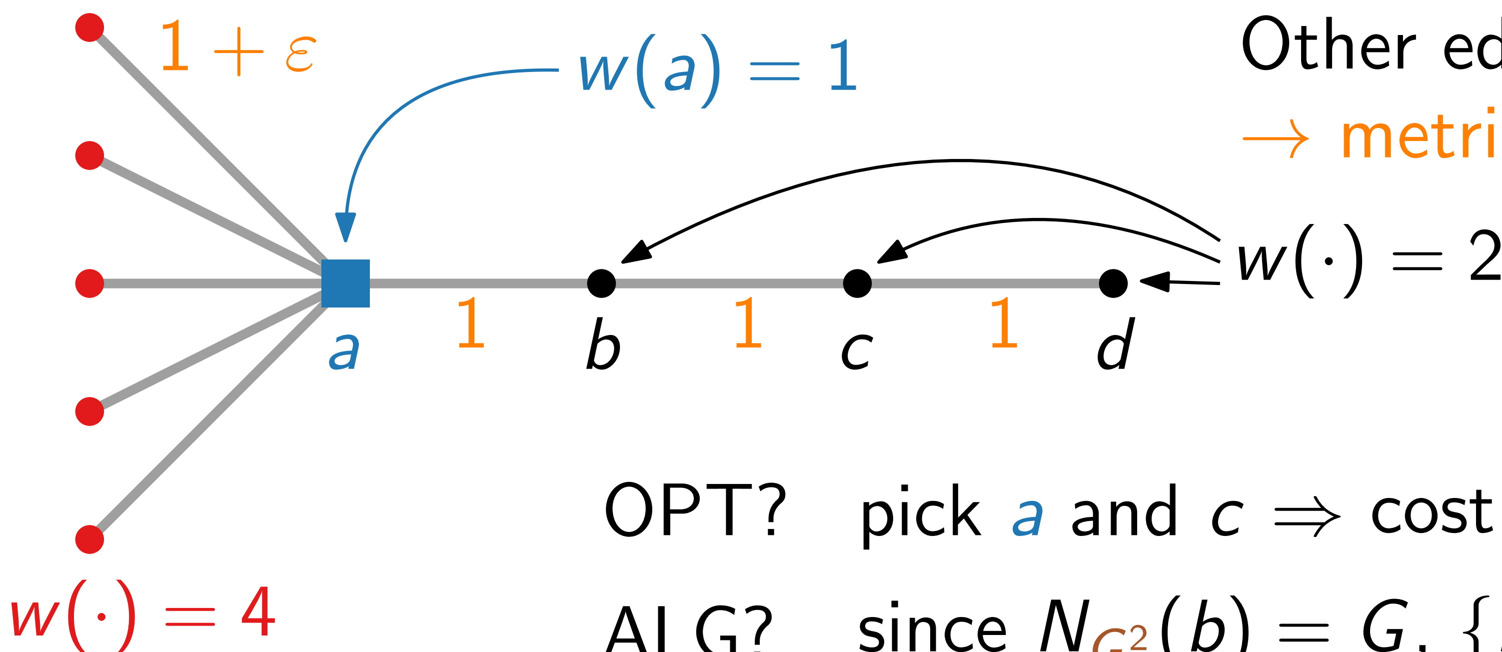
ALG? since $N_{G^2}(b) = G$, $\{b\}$ is a maximal independent set in G^2

Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

Other edge costs?
$\rightarrow$ metric completion!

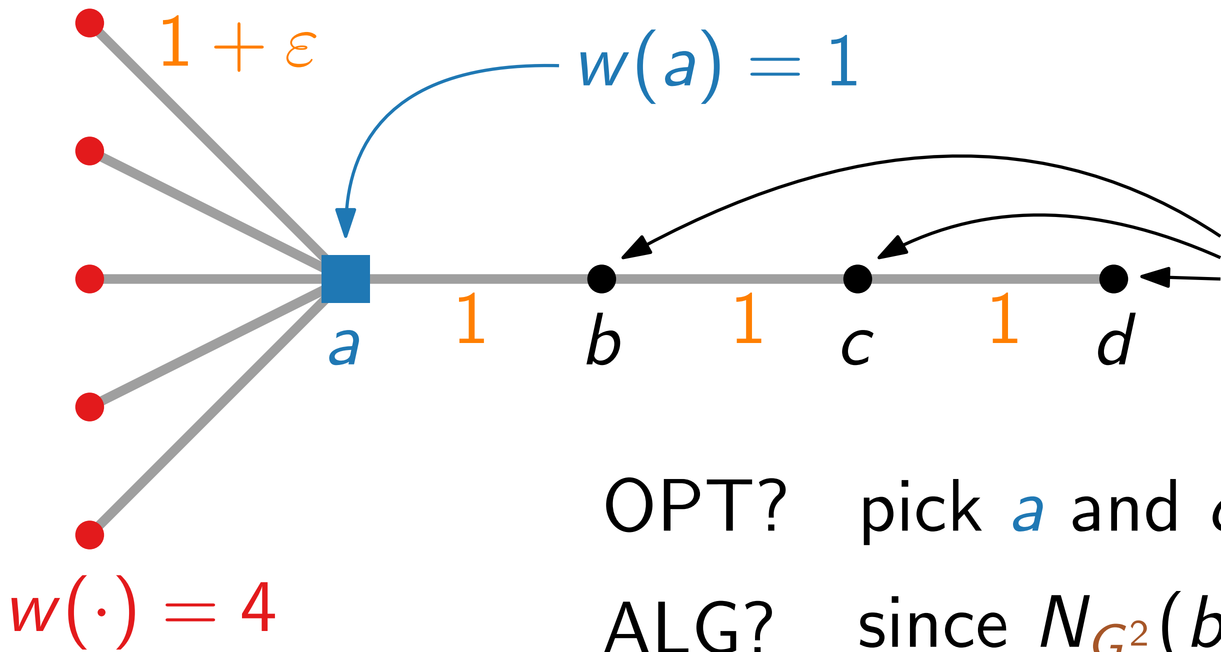


OPT? pick a and $c \Rightarrow$ cost $1 + \varepsilon$.

ALG? since $N_{G^2}(b) = G$, $\{b\}$ is a maximal independent set in G^2
Thus, alg. picks only $a \Rightarrow$ cost 3.

Tight Example...?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.



~~Consider $W = 3$.~~

Other edge costs?

$\rightarrow$ metric completion!

OPT? pick a and c $\Rightarrow$ cost $1+\varepsilon$.

ALG? since $N_{G^2}(b) = G$, $\{b\}$ is a maximal independent set in G^2

Thus, alg. picks only a $\Rightarrow$ cost 3.

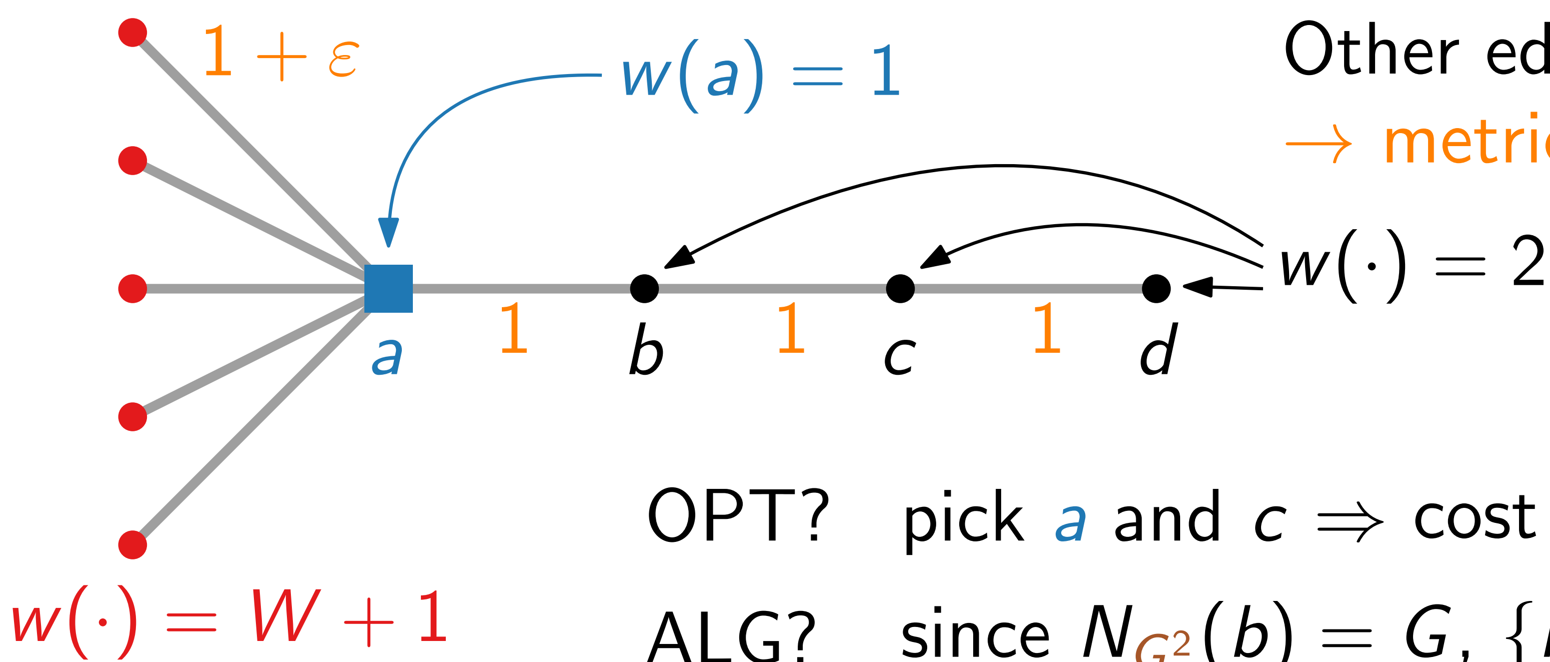
How can we generalize this to larger W?

Tight Example. . . ?

Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

~~Consider $W = 3$.~~

Other edge costs?
$\rightarrow$ metric completion!



OPT? pick a and $c \Rightarrow$ cost $1 + \varepsilon$.

ALG? since $N_{G^2}(b) = G$, $\{b\}$ is a maximal independent set in G^2.
Thus, alg. picks only $a \Rightarrow$ cost 3.

How can we generalize this to larger W?