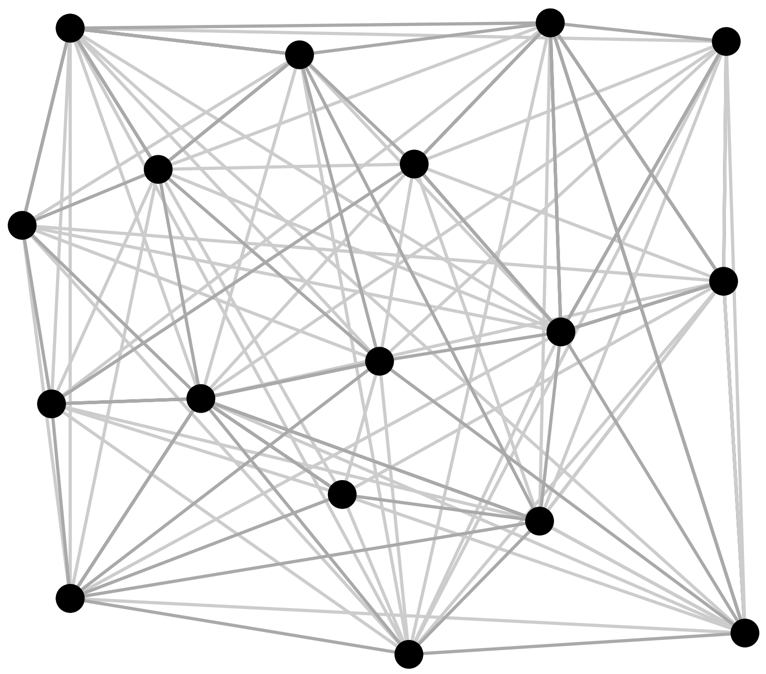
k-CENTER via Parametric Pruning

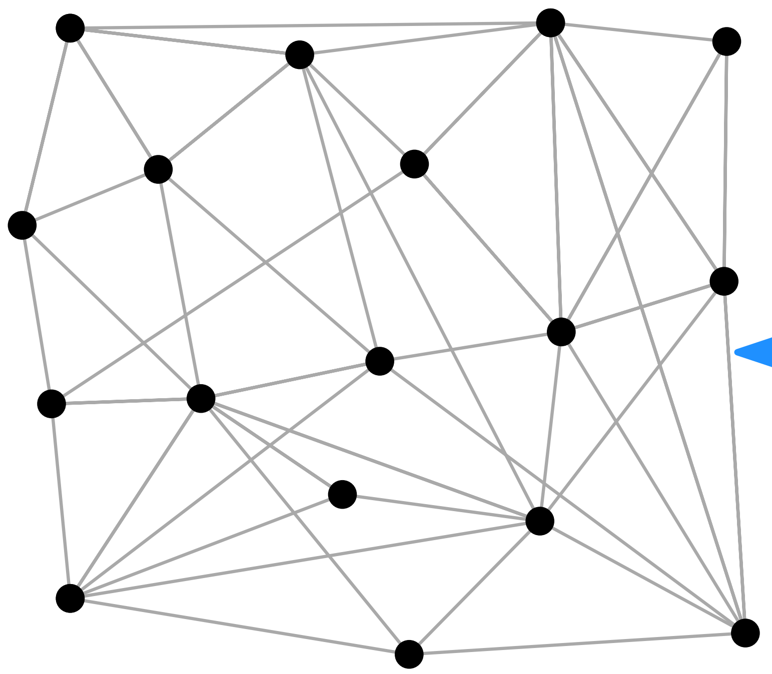
Metric k-CENTER

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



Metric k-Center

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

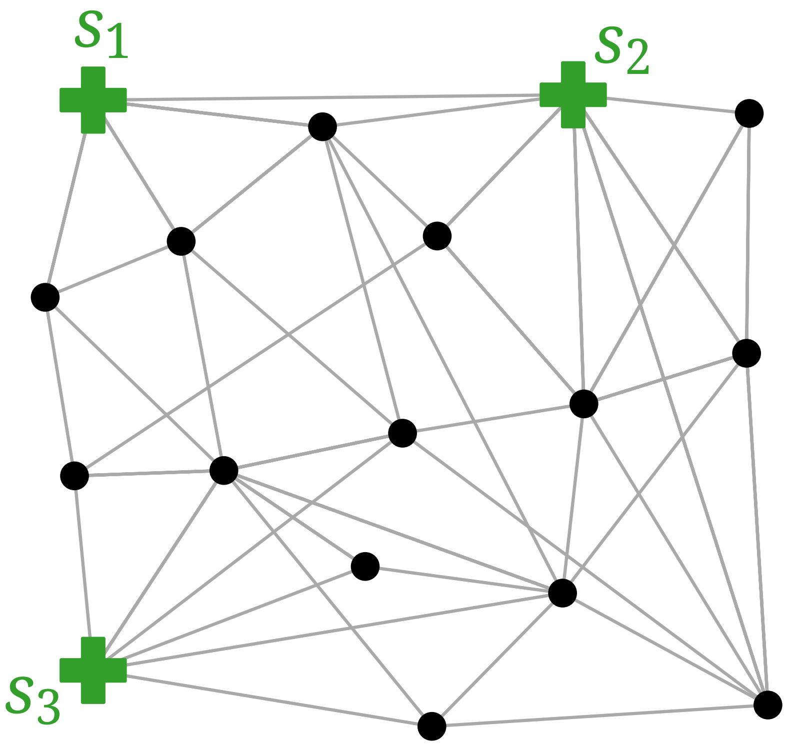


for readability:
not all edges shown

Metric k-Center

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

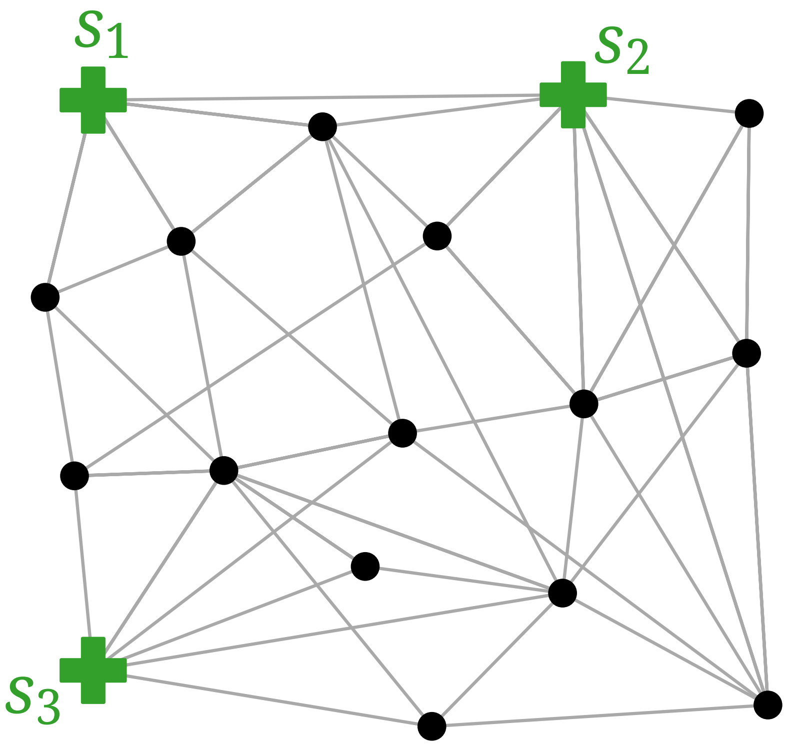
For each vertex set $S \subseteq V$



Metric k-CENTER

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

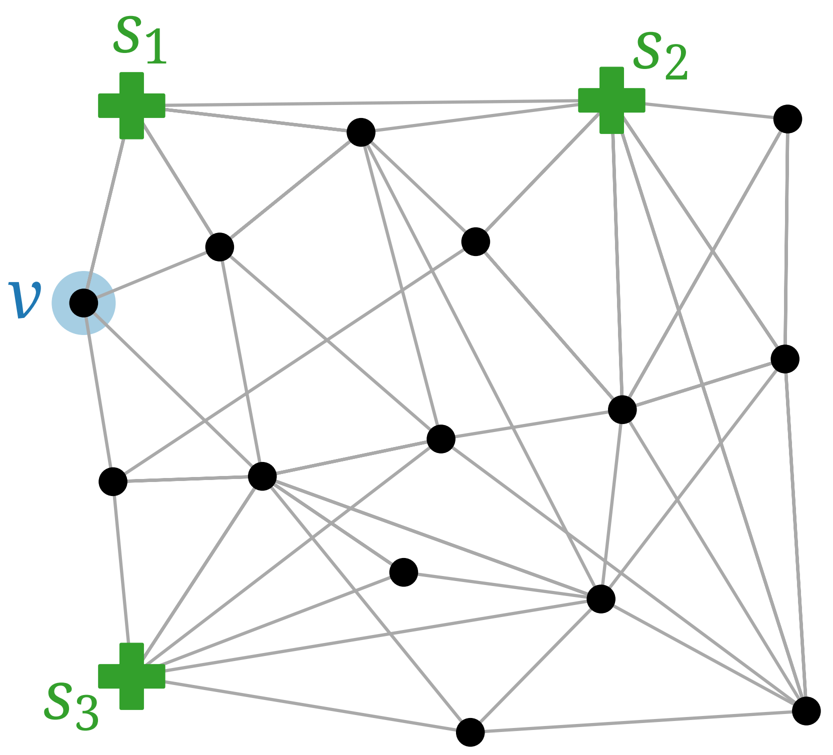
For each vertex set $S \subseteq V$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-CENTER

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

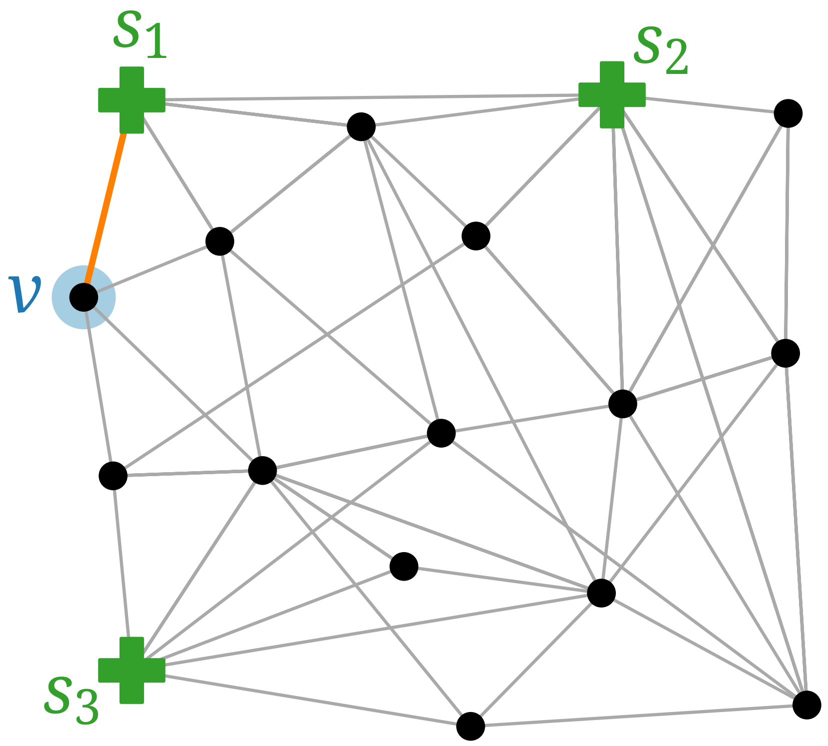
For each vertex set $S \subseteq V$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-CENTER

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

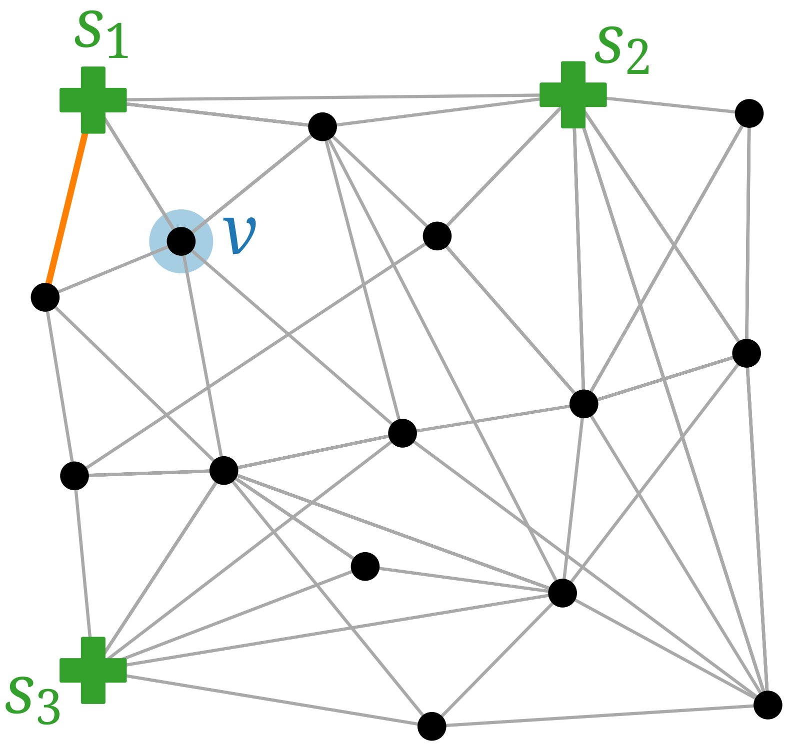
For each vertex set $S \subseteq V$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-CENTER

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

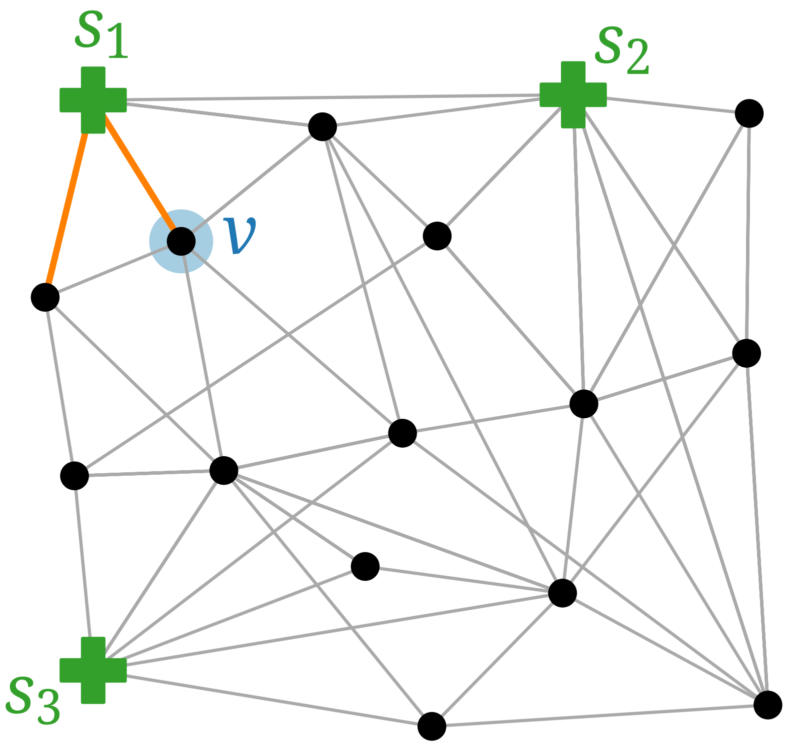
For each vertex set $S \subseteq V$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-Center

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

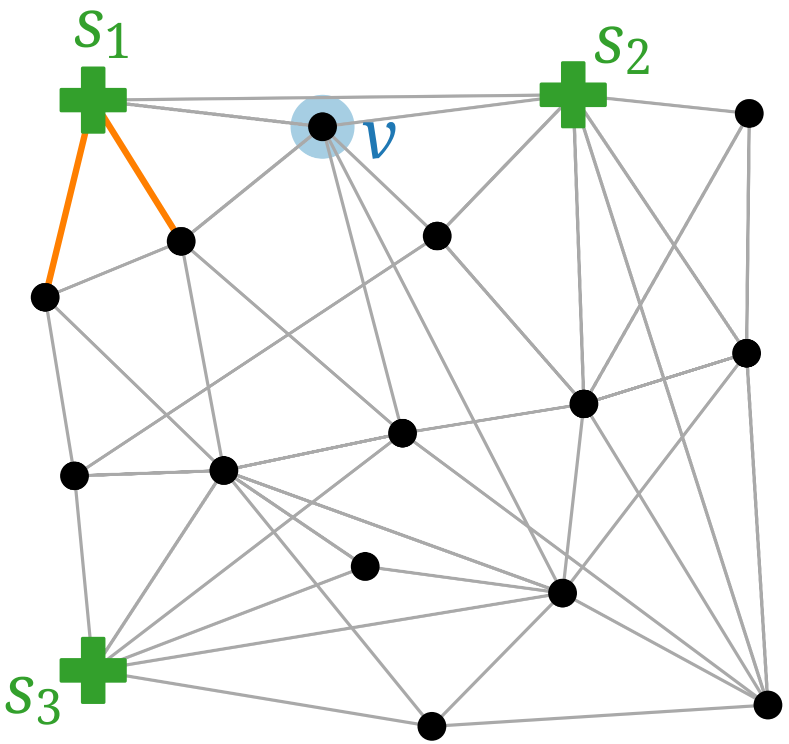
For each vertex set $S \subseteq V$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-CENTER

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

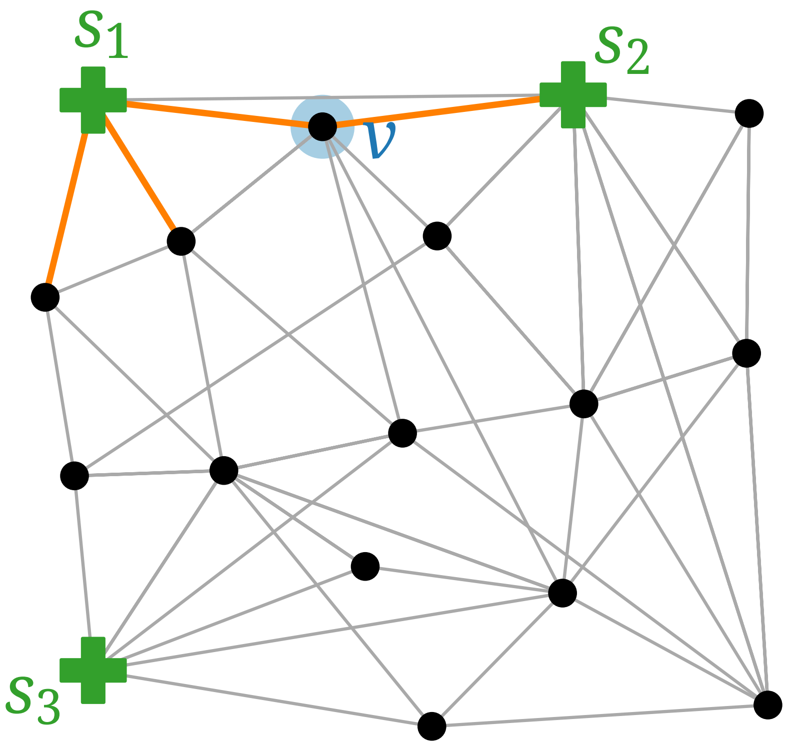
For each vertex set $S \subseteq V$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-CENTER

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

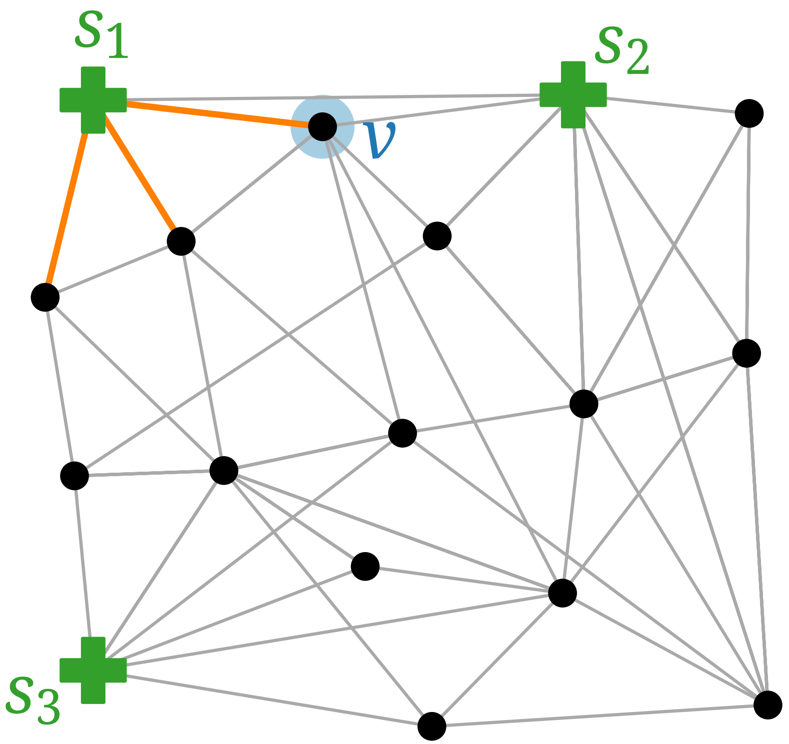
For each vertex set $S \subseteq V$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-CENTER

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

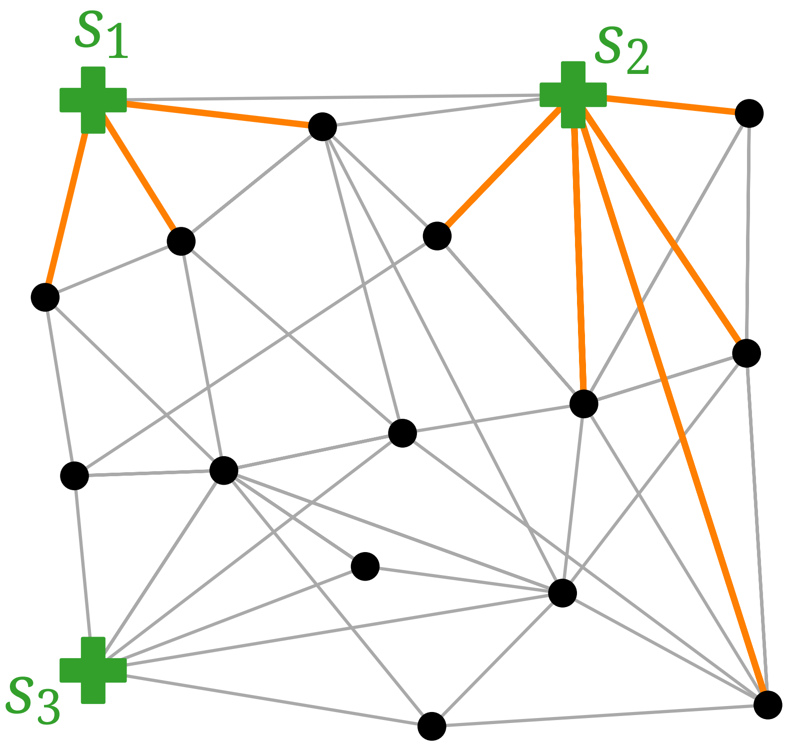
For each vertex set $S \subseteq V$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-CENTER

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

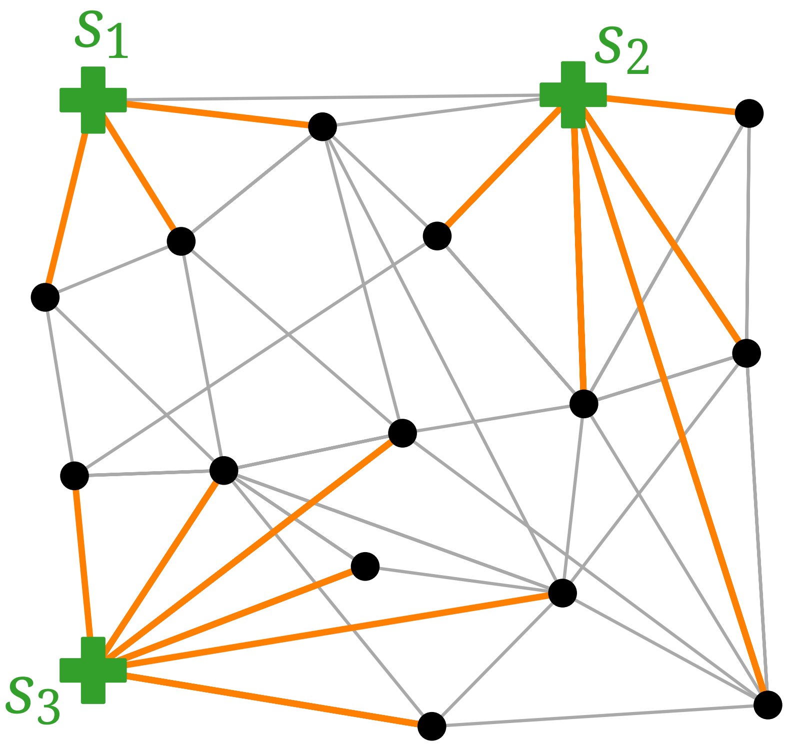
For each vertex set $S \subseteq V$,
$c(v, S)$ is the cost of the cheapest edge from v to a vertex in S.



Metric k-CENTER

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

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

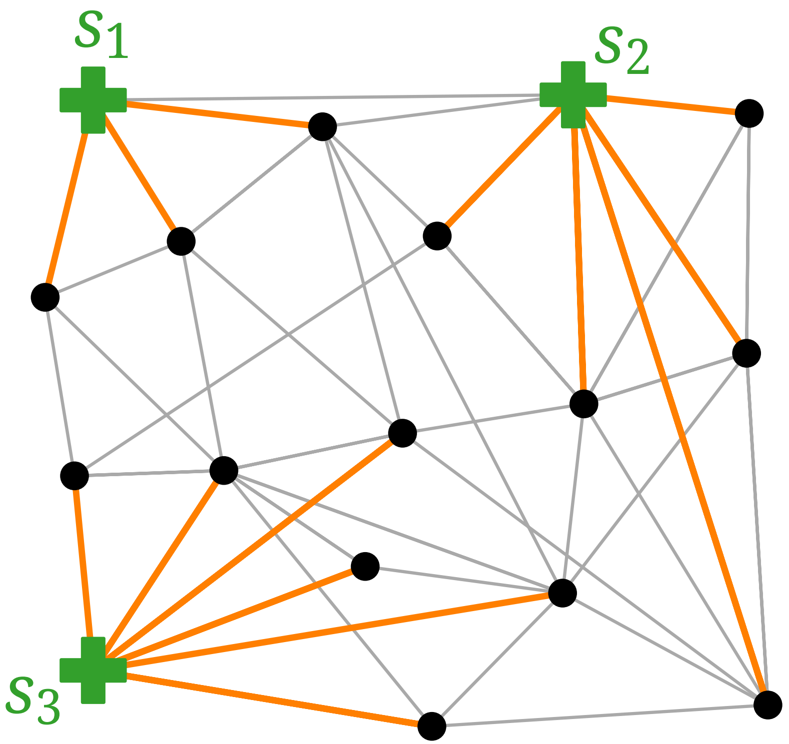


Metric k-Center

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

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

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

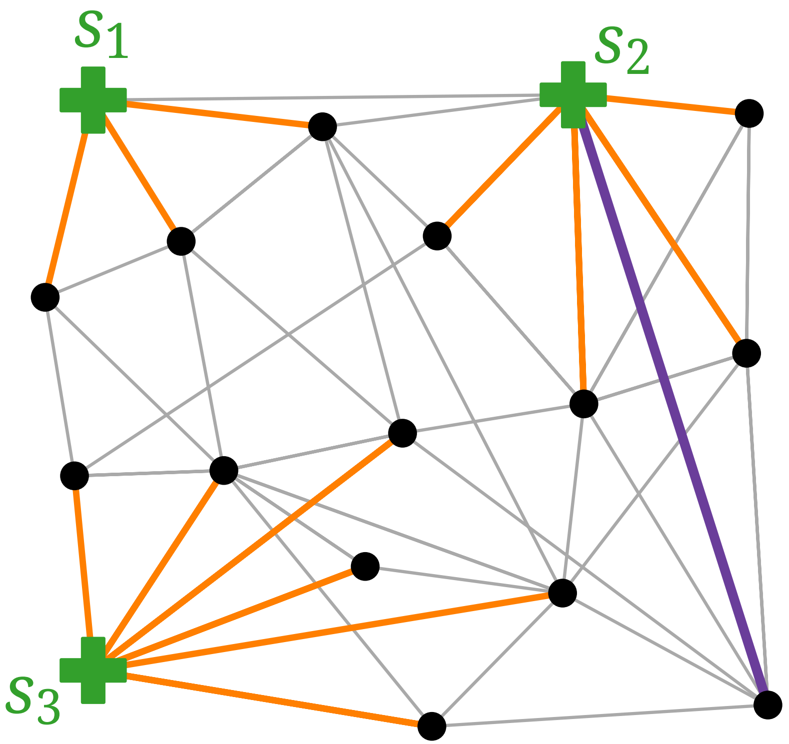


Metric k-CENTER

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

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

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

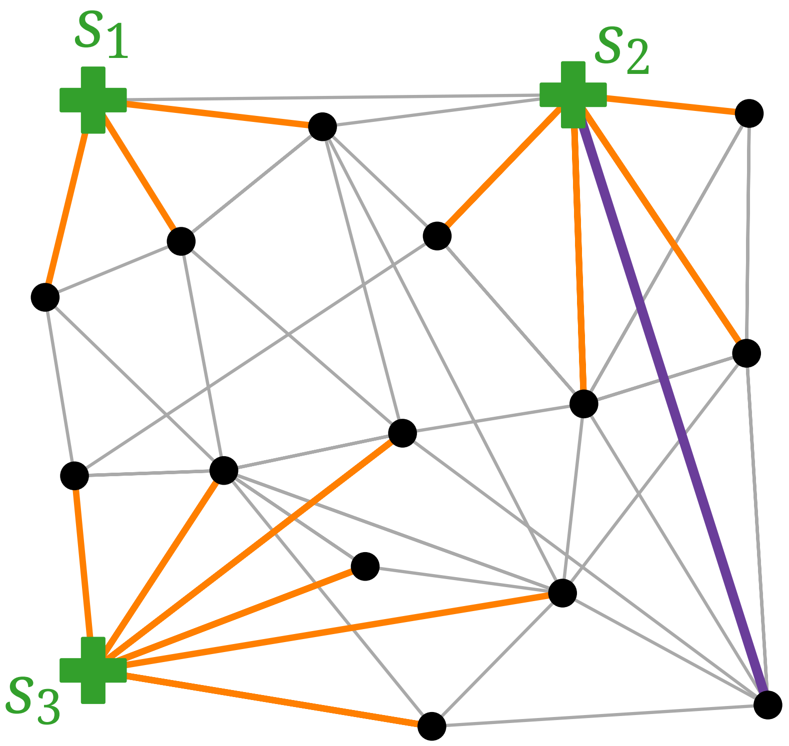


Metric k-CENTER

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

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

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

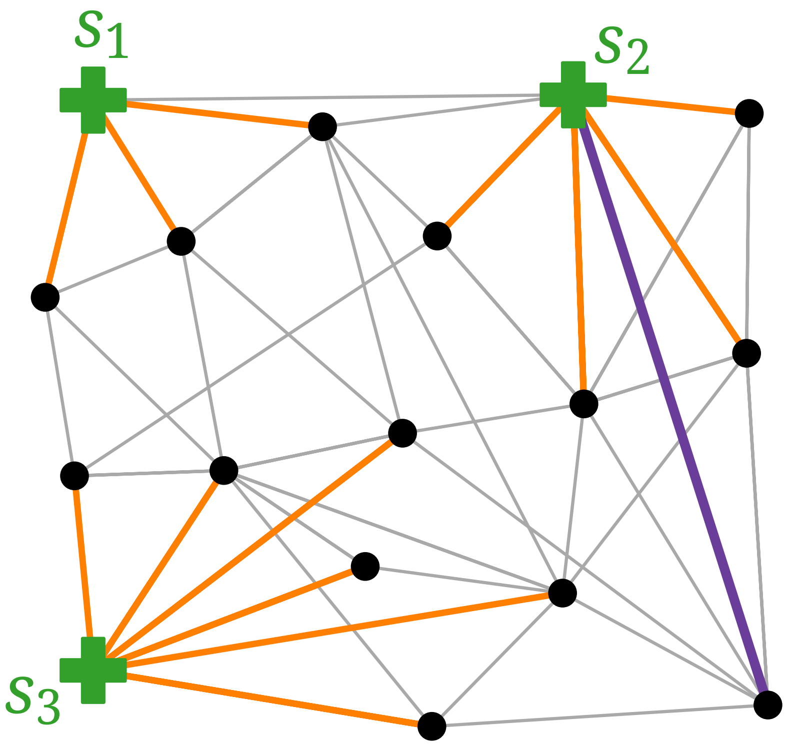


Metric k-Center

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

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

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

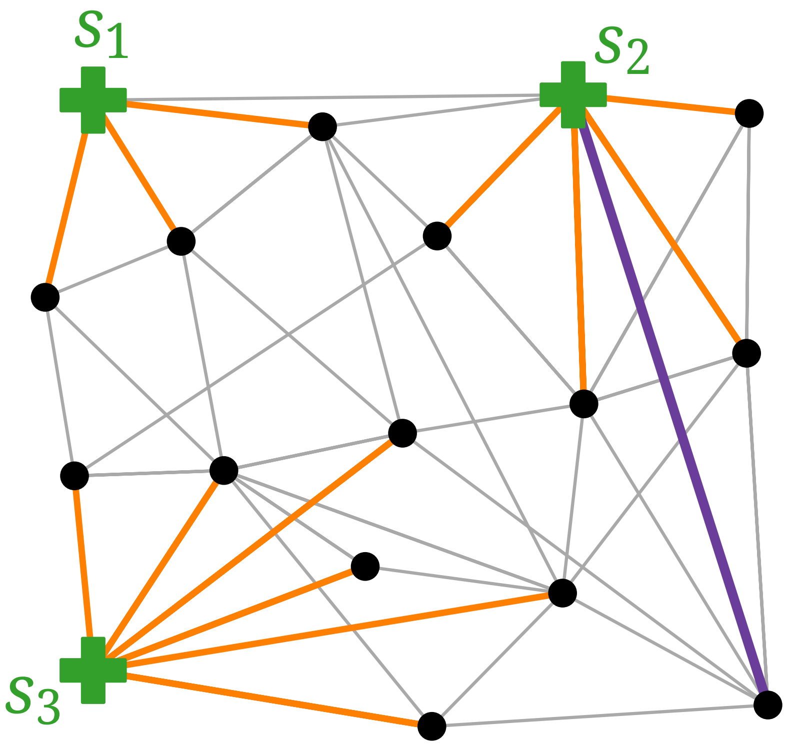


Metric k-CENTER

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

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

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

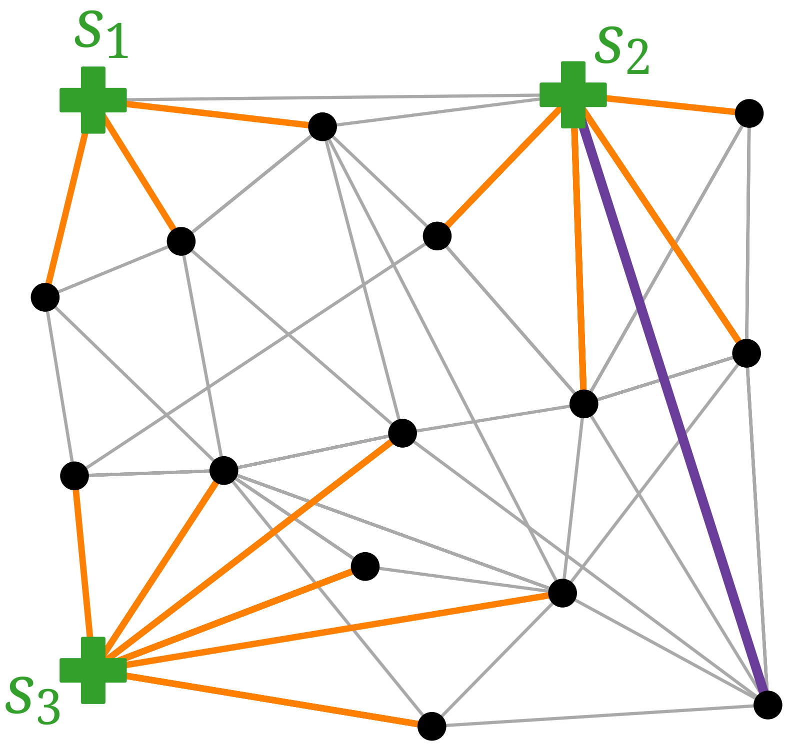


Metric k-CENTER

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

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

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



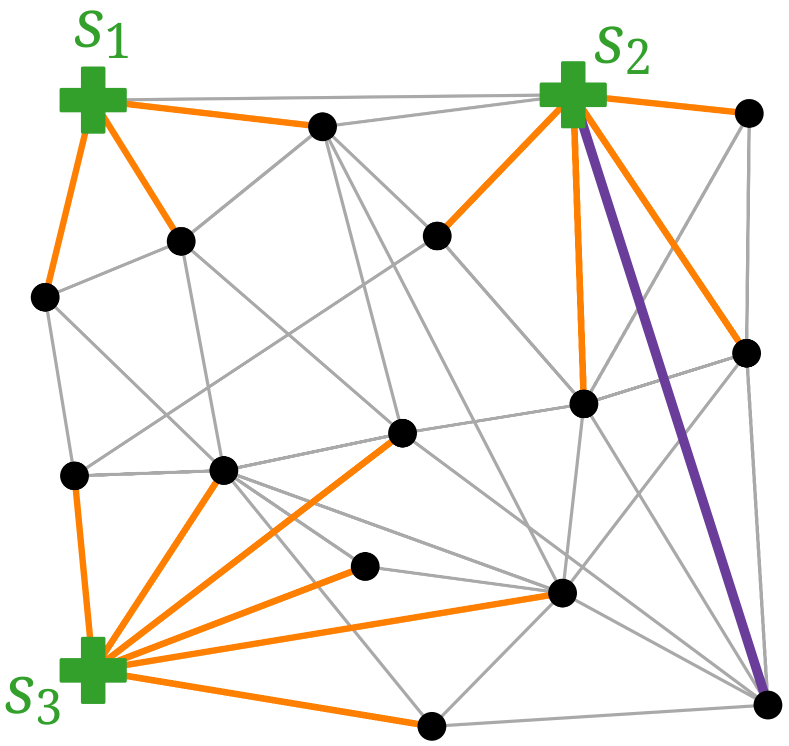
optimal for $k = 3$?   better solution?

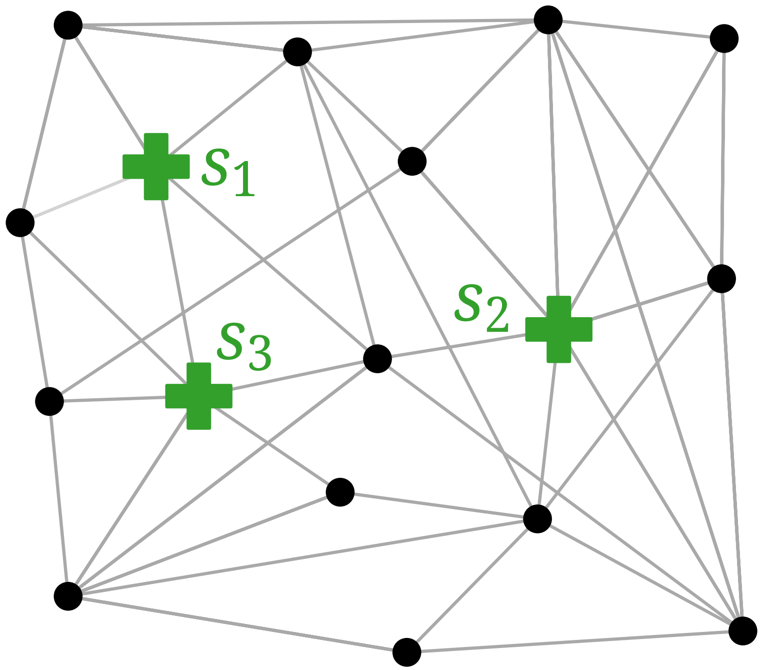
Metric k-CENTER

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

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

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



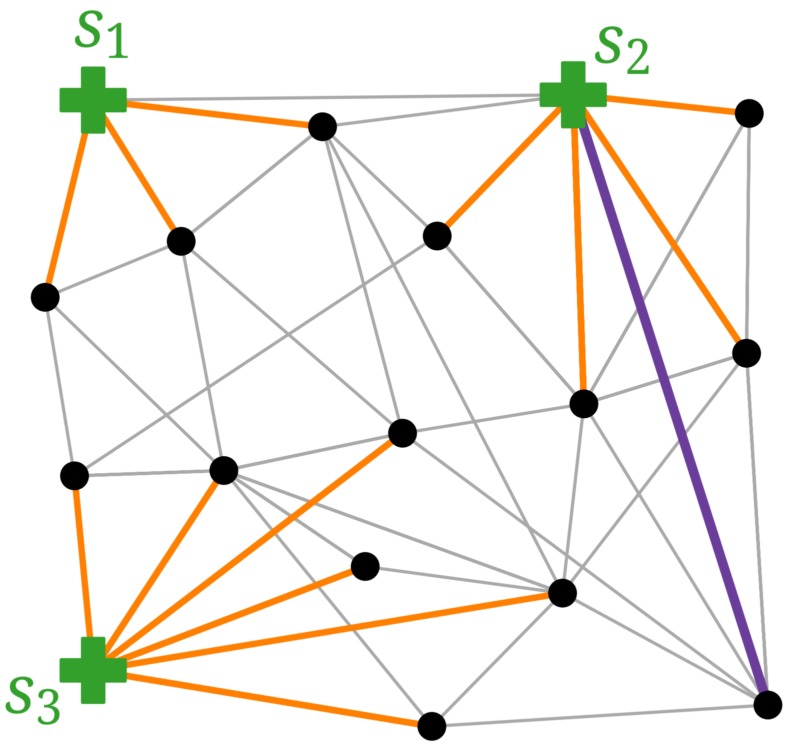


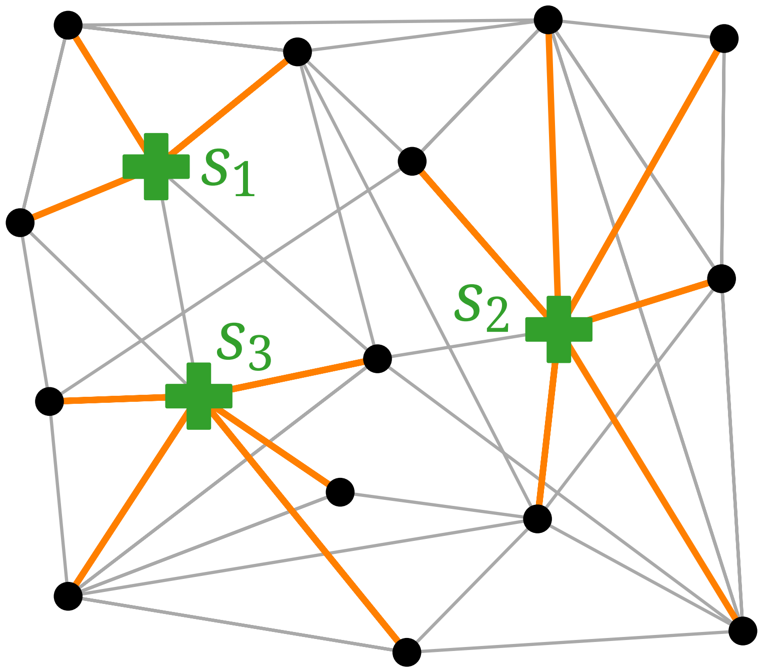
Metric k-CENTER

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

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

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



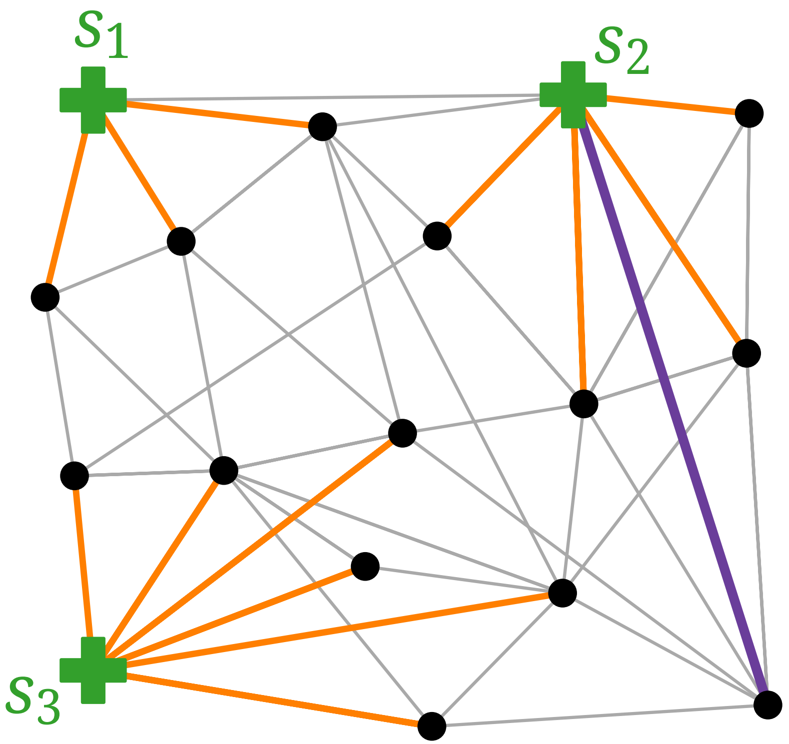


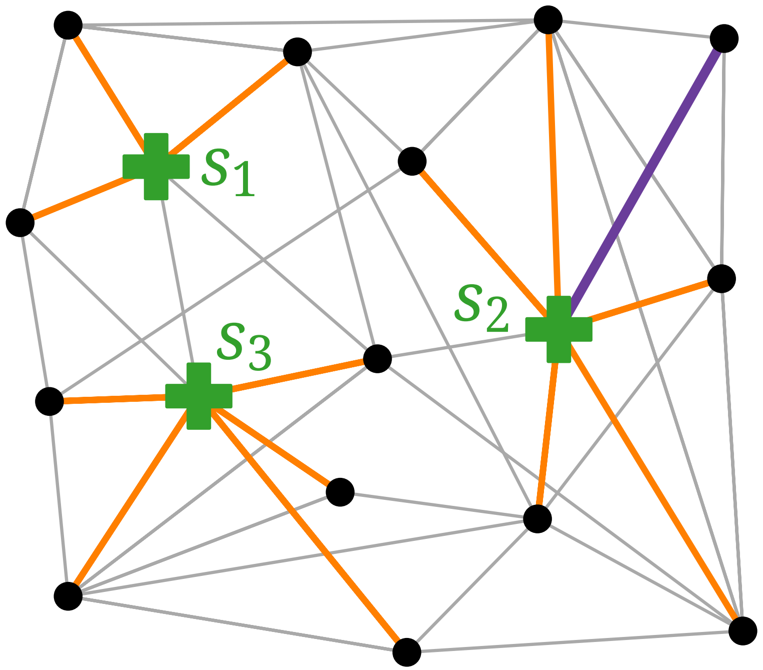
Metric k-CENTER

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

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

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





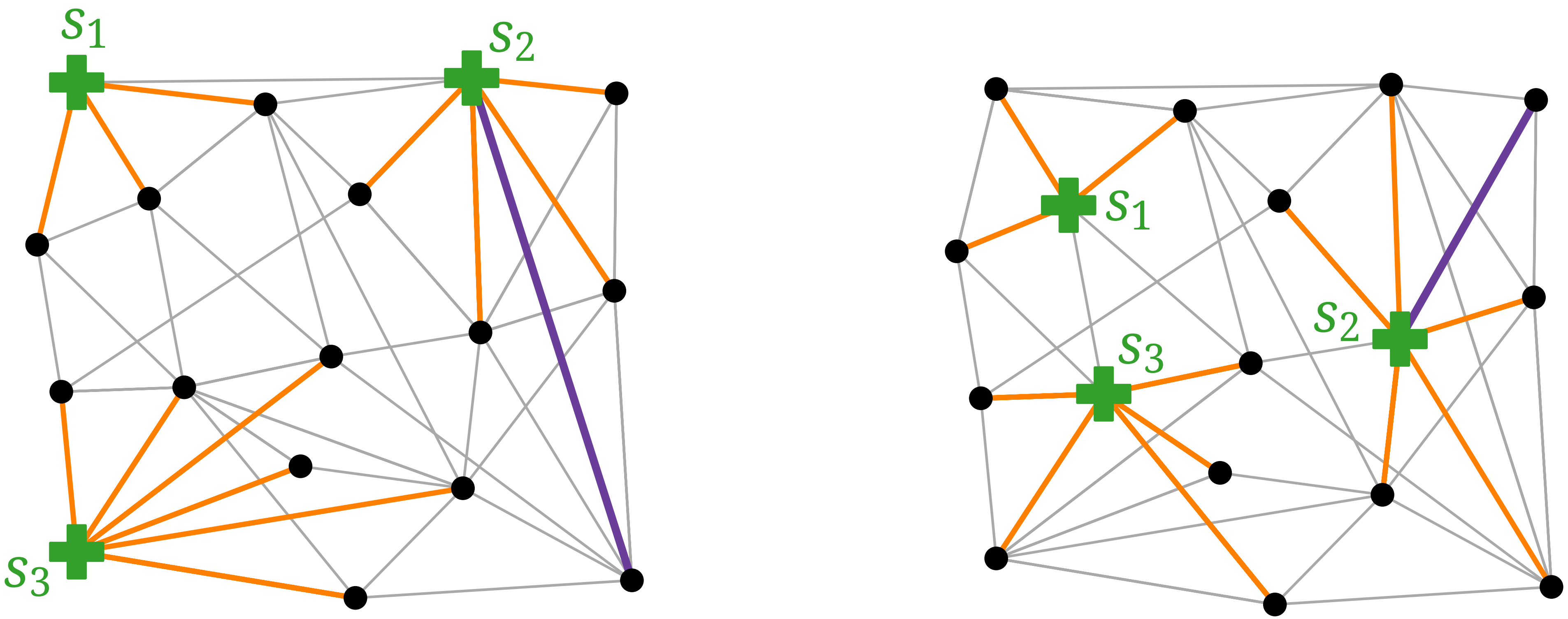
Metric k-CENTER

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

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

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

today: 2-approximation of metric k-center using *parametric pruning.*



Technique: Parametric Pruning

Idea 1: Reduce optimization problem to decision problem

Technique: Parametric Pruning

Idea 1: Reduce optimization problem to decision problem

optimization: minimize cost of solution (or maximize)

decision (t): find answer to: Is there a feasible solution with cost $\leq t$?

Technique: Parametric Pruning

Idea 1: Reduce optimization problem to decision problem
by solving decision(t) for increasing t until answer is yes

optimization: minimize cost of solution (or maximize)

decision (t): find answer to: Is there a feasible solution with cost $\leq t$?

Technique: Parametric Pruning

Idea 1: Reduce optimization problem to decision problem
by solving decision(t) for increasing t until answer is yes

optimization: minimize cost of solution (or maximize)

decision (t): find answer to: Is there a feasible solution with cost $\leq t$?

Idea 2 (pruning): Given instance I of optimization problem, obtain instance $I(t)$
by removing parts that cannot be used in solution with cost $\leq t$

Technique: Parametric Pruning

Idea 1: Reduce optimization problem to decision problem
by solving decision(t) for increasing t until answer is yes

optimization: minimize cost of solution (or maximize)

decision (t): find answer to: Is there a feasible solution with cost $\leq t$?

Idea 2 (pruning): Given instance I of optimization problem, obtain instance $I(t)$
by removing parts that cannot be used in solution with cost $\leq t$

Idea 3 (lower bound): factor-α approximation algorithm consists of two steps

Technique: Parametric Pruning

Idea 1: Reduce optimization problem to decision problem
by solving decision(t) for increasing t until answer is yes

optimization: minimize cost of solution (or maximize)

decision (t): find answer to: Is there a feasible solution with cost $\leq t$?

Idea 2 (pruning): Given instance I of optimization problem, obtain instance $I(t)$
by removing parts that cannot be used in solution with cost $\leq t$

Idea 3 (lower bound): factor-α approximation algorithm consists of two steps

1.) use family of instances $I(t)$ to compute lower bound t^* for OPT

Technique: Parametric Pruning

Idea 1: Reduce optimization problem to decision problem
by solving decision(t) for increasing t until answer is yes

optimization: minimize cost of solution (or maximize)

decision (t): find answer to: Is there a feasible solution with cost $\leq t$?

Idea 2 (pruning): Given instance I of optimization problem, obtain instance $I(t)$
by removing parts that cannot be used in solution with cost $\leq t$

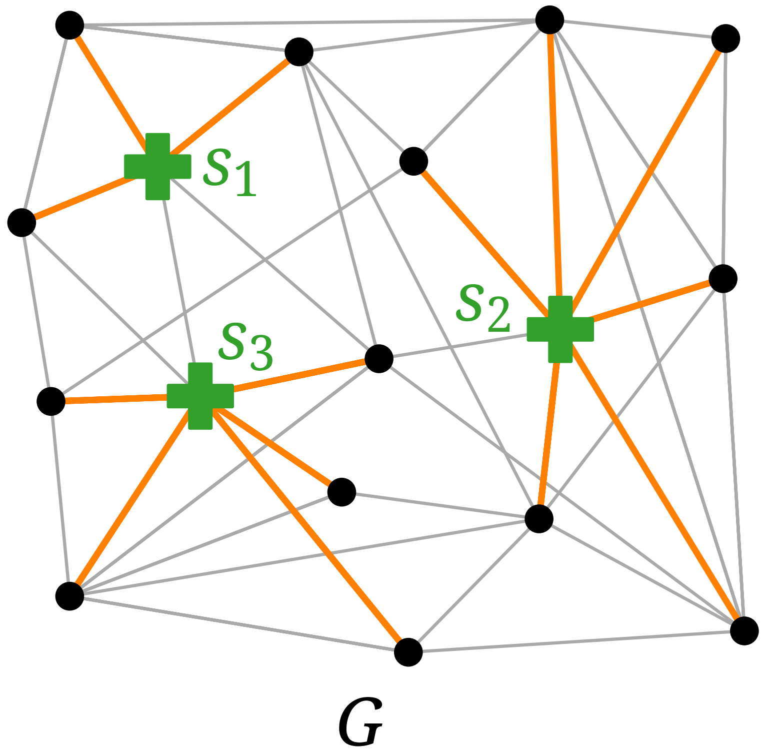
Idea 3 (lower bound): factor-α approximation algorithm consists of two steps

1.) use family of instances $I(t)$ to compute lower bound t^* for OPT

2.) find solution in instance $I(\alpha t^*)$

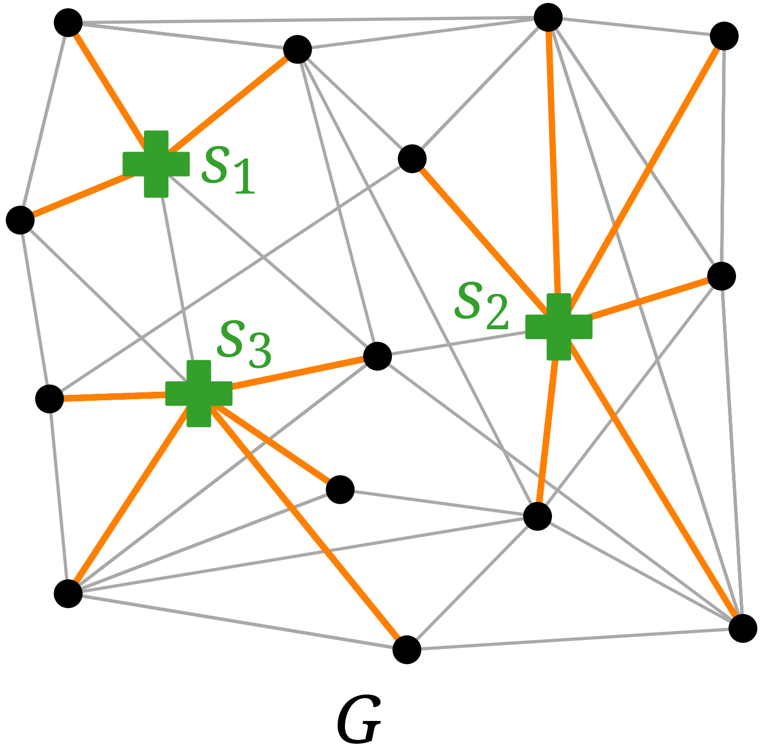
Parametric Pruning: Metric k-center

Metric k-center: What is the decision problem? How do the pruned instances $I(t)$ look like?



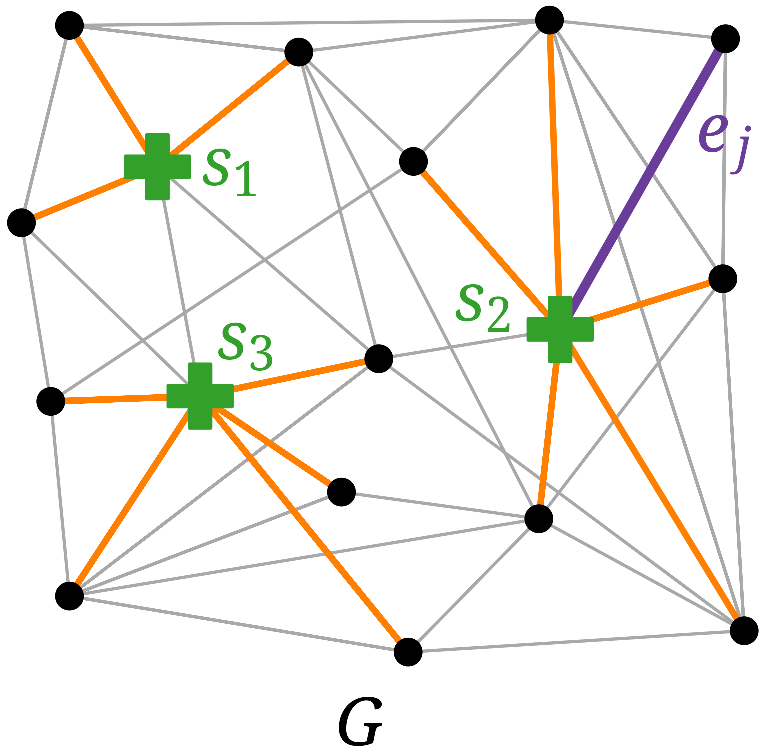
Parametric Pruning: Metric k-center

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



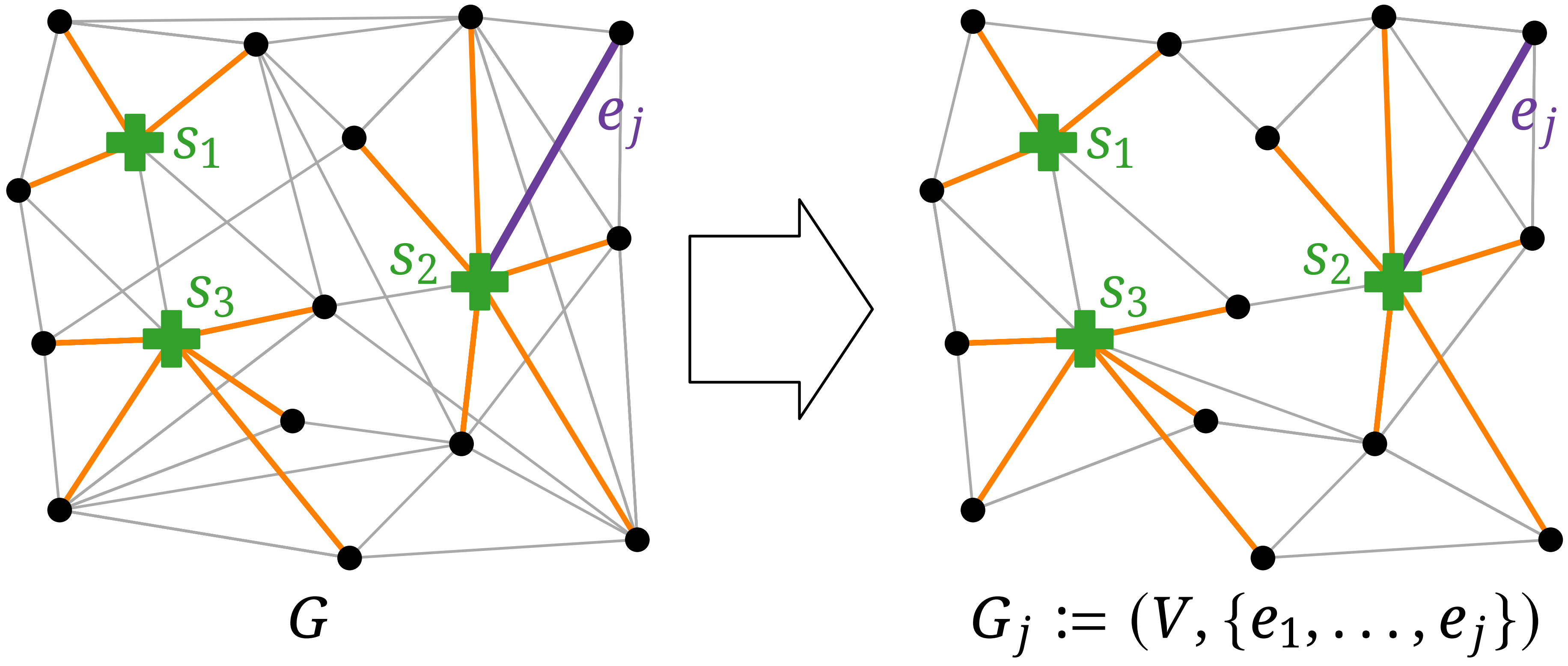
Parametric Pruning: Metric k-center

Let $E = \{e_1, \ldots, e_m\}$ with $c(e_1) \leq \ldots \leq c(e_m)$.
Suppose we want to decide whether $\mathrm{OPT} \leq c(e_j)$.



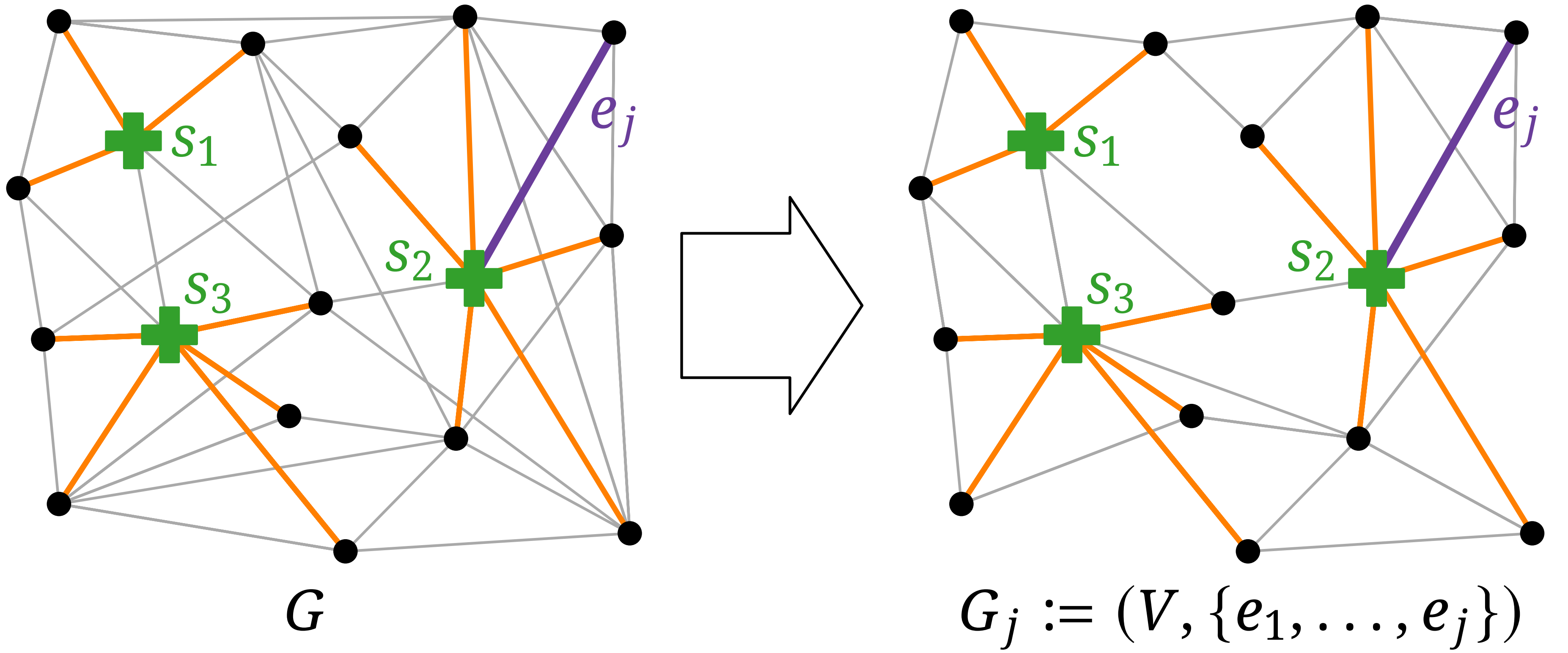
Parametric Pruning: Metric k-center

Let $E = \{e_1, \ldots, e_m\}$ with $c(e_1) \leq \ldots \leq c(e_m)$.
Suppose we want to decide whether $\mathrm{OPT} \leq c(e_j)$.



Parametric Pruning: Metric k-center

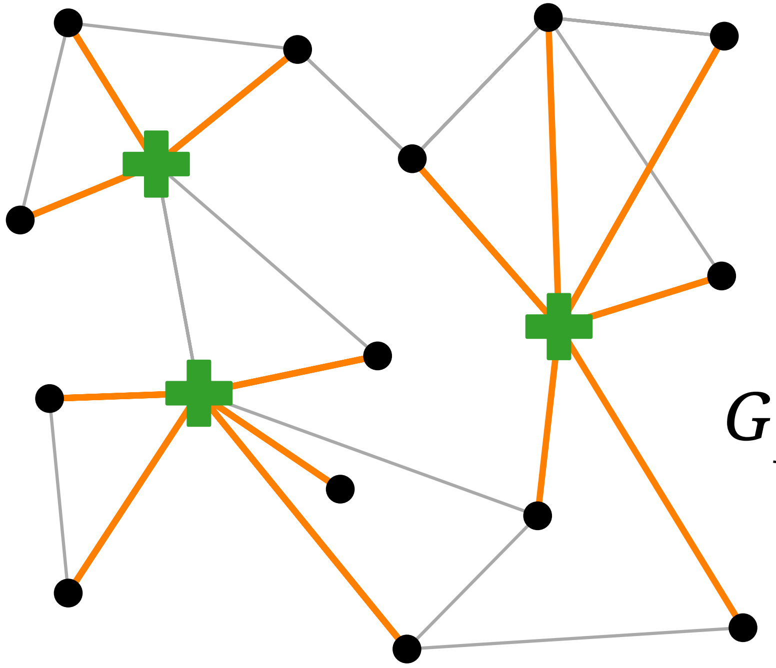
Let $E = \{e_1, \ldots, e_m\}$ with $c(e_1) \leq \ldots \leq c(e_m)$.
Suppose we want to decide whether $\mathrm{OPT} \leq c(e_j)$.



…try each pruned instance G_j.

…try each G_j.

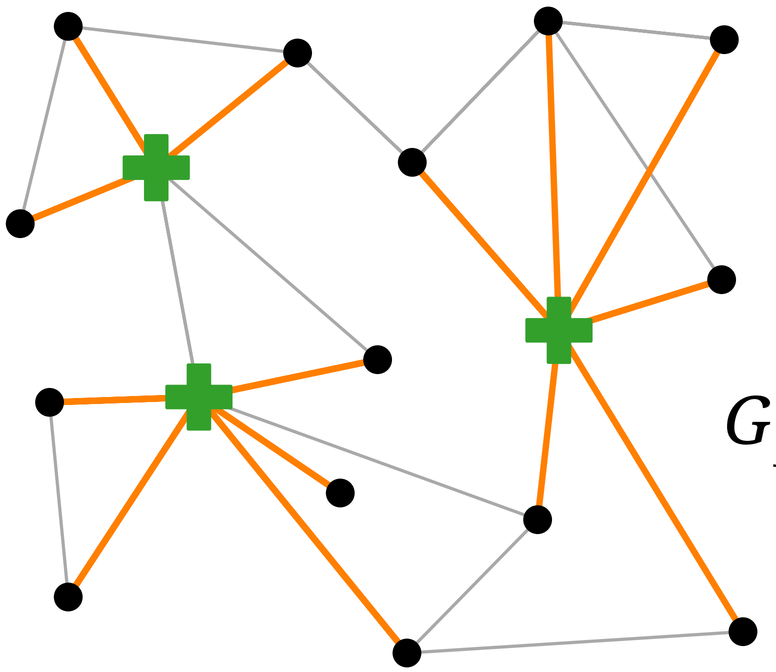
Def.



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

…try each G_j.

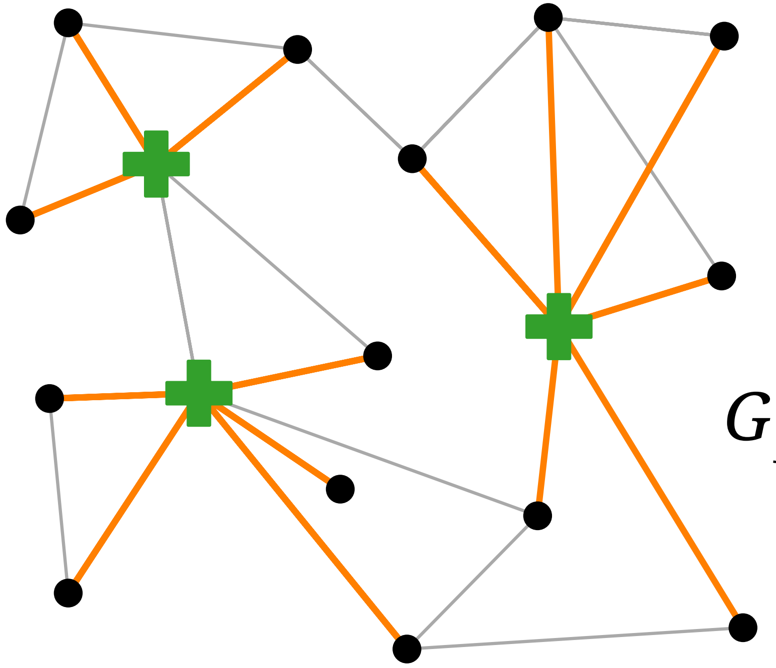
Def. A vertex set D of a graph H is dominating if each vertex is either in D or adjacent to a vertex in D.



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

…try each G_j.

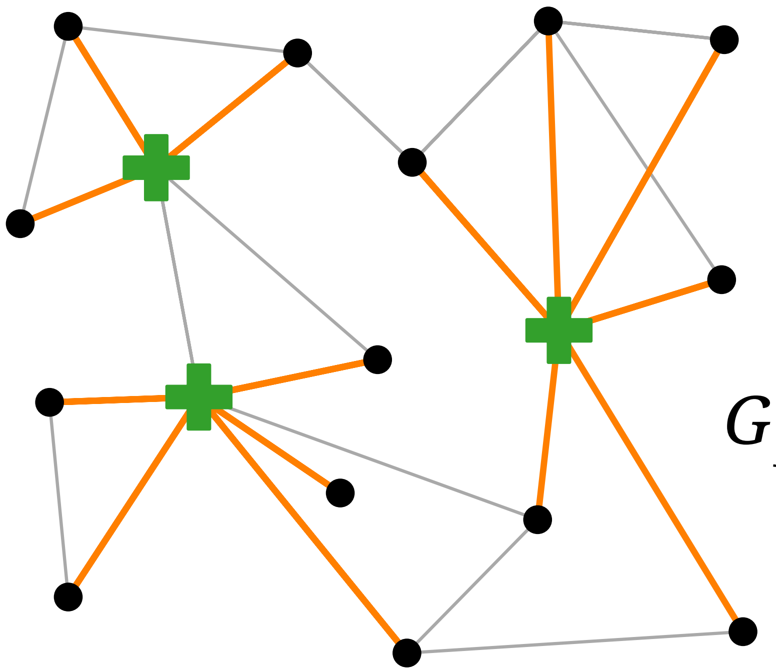
Def. A vertex set D of a graph H is dominating if each vertex is either in D or adjacent to a vertex in D. The cardinality of a smallest dominating set in H is denoted by $\mathrm{dom}(H)$.



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

…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 $\mathrm{dom}(H)$.

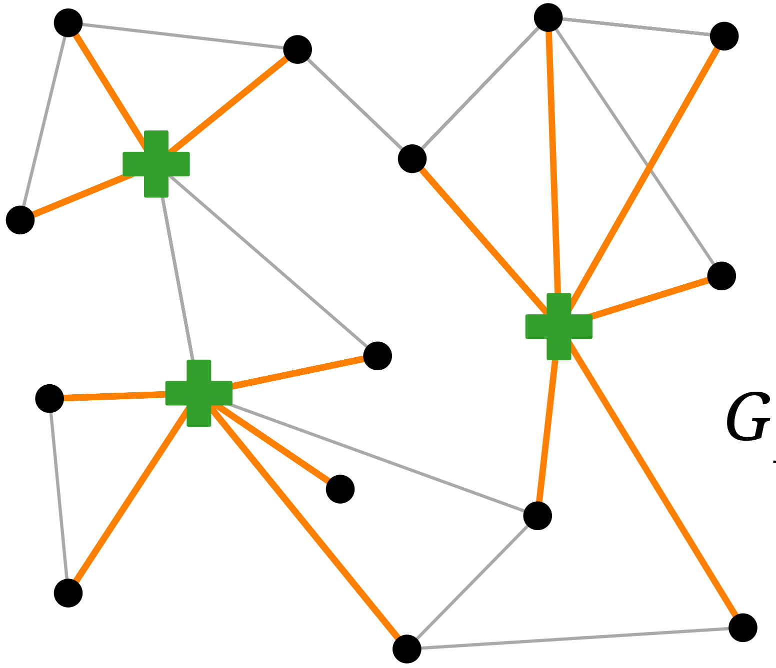


decision problem for k-Center:
$\mathrm{dom}(G_j) \leq k$?

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

…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 $\mathrm{dom}(H)$.



decision problem for k-Center:
$\mathrm{dom}(G_j) \leq k$?

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

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

Square of a Graph – Lower bounding k-Center

Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers

Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers

step 2.) show that this set gives an approximate solution

Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers

step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ “coarsened” G_j.

Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers

step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "coarsened" G_j.

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

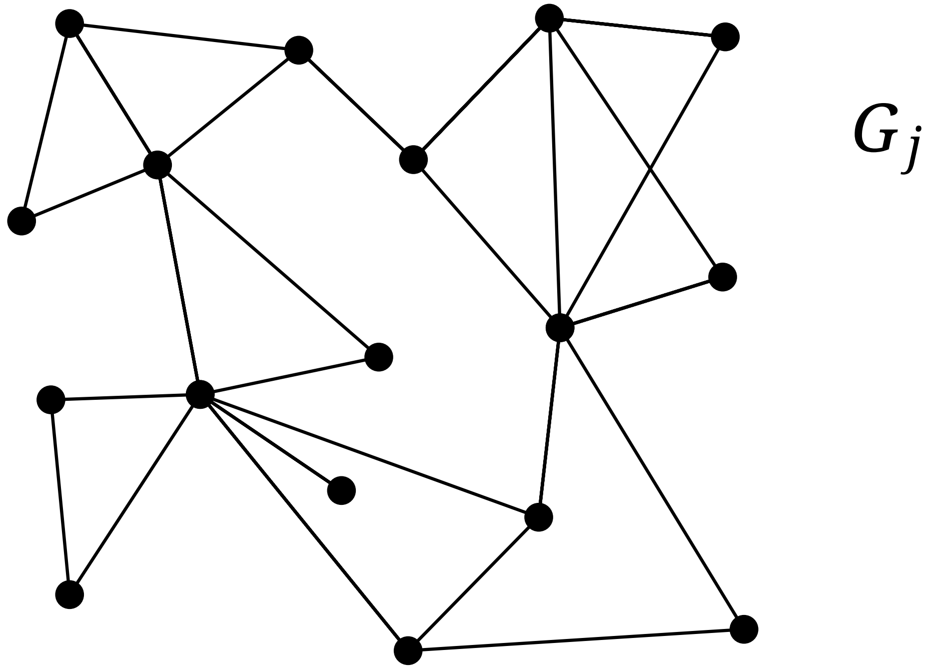
Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers

step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "coarsened" G_j.

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

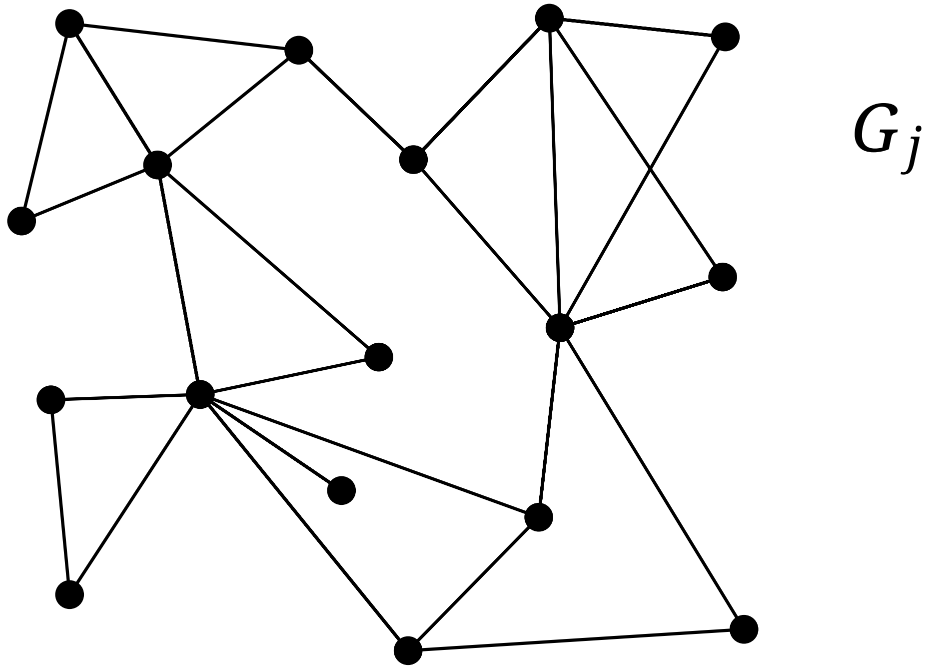


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "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 (graph-)distance at most two in H.

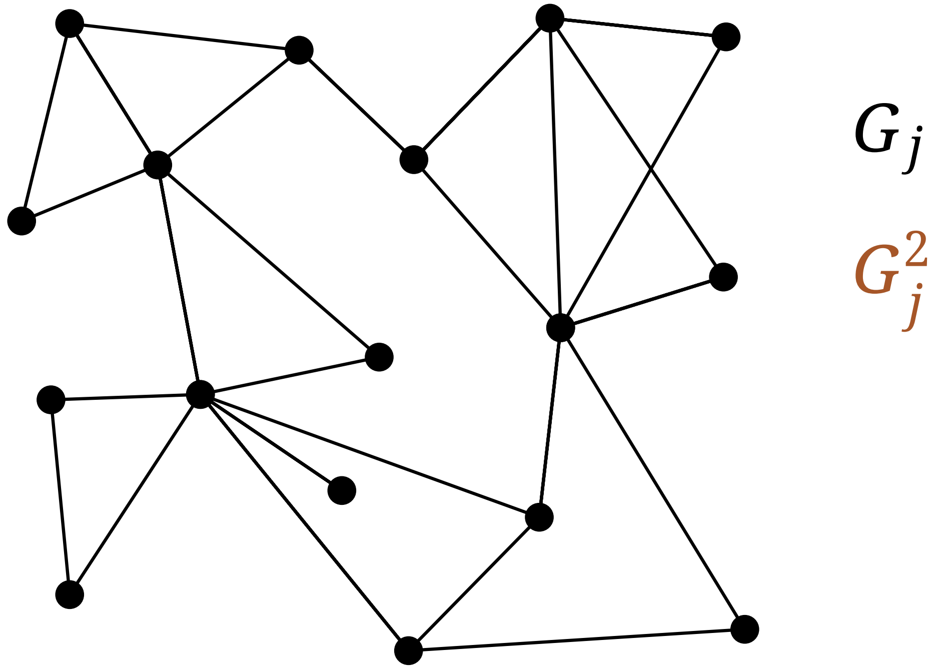


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "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 (graph-)distance at most two in H.

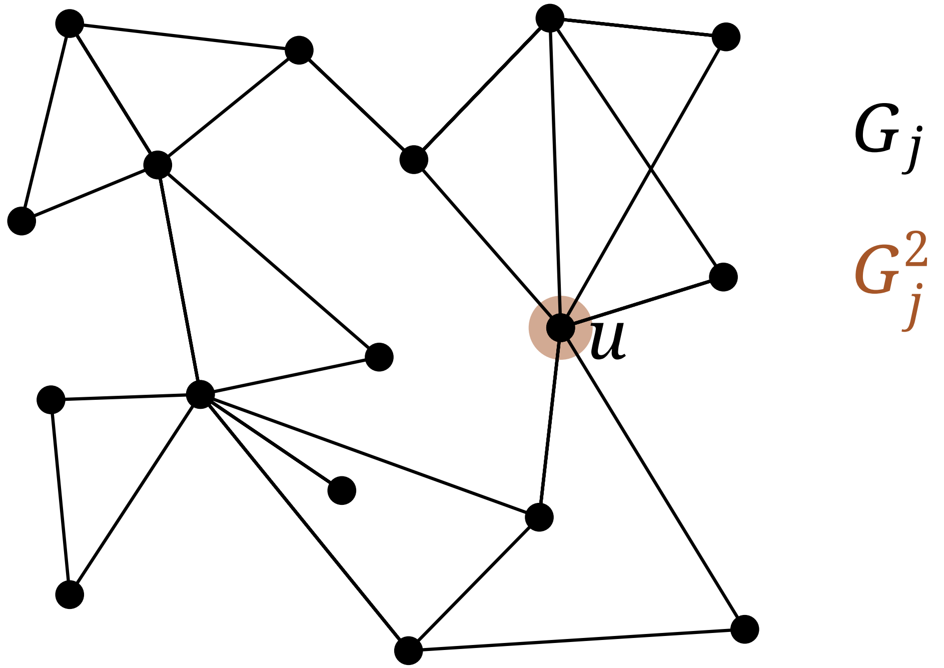


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "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 (graph-)distance at most two in H.

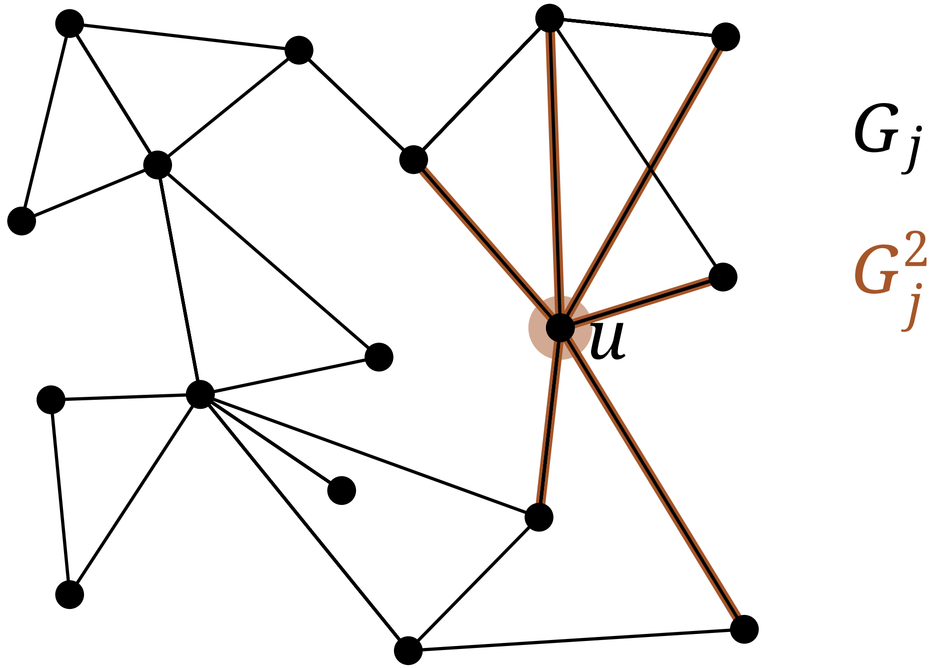


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ “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 (graph-)distance at most two in H.

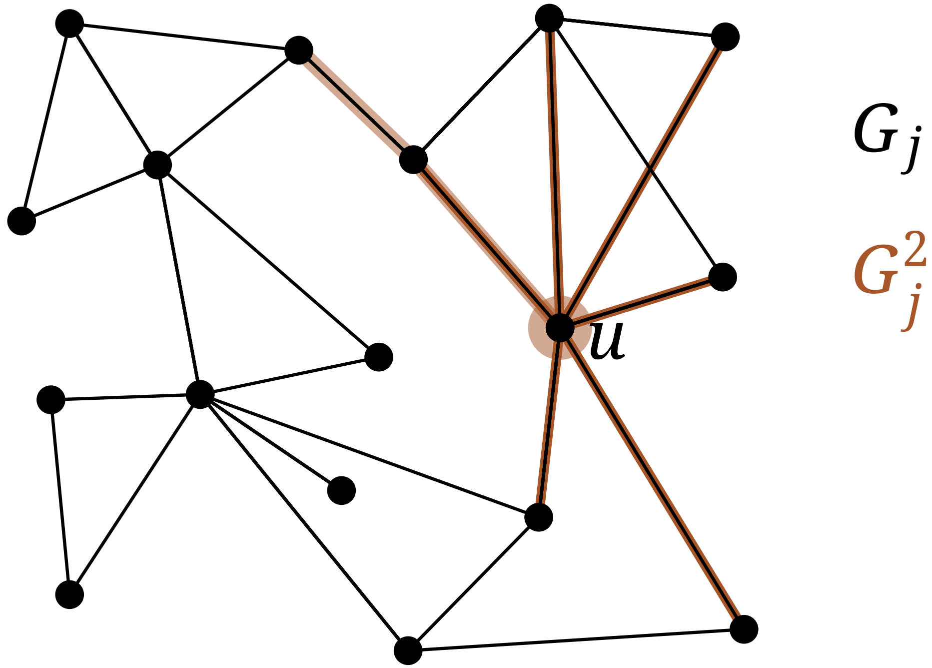


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "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 (graph-)distance at most two in H.

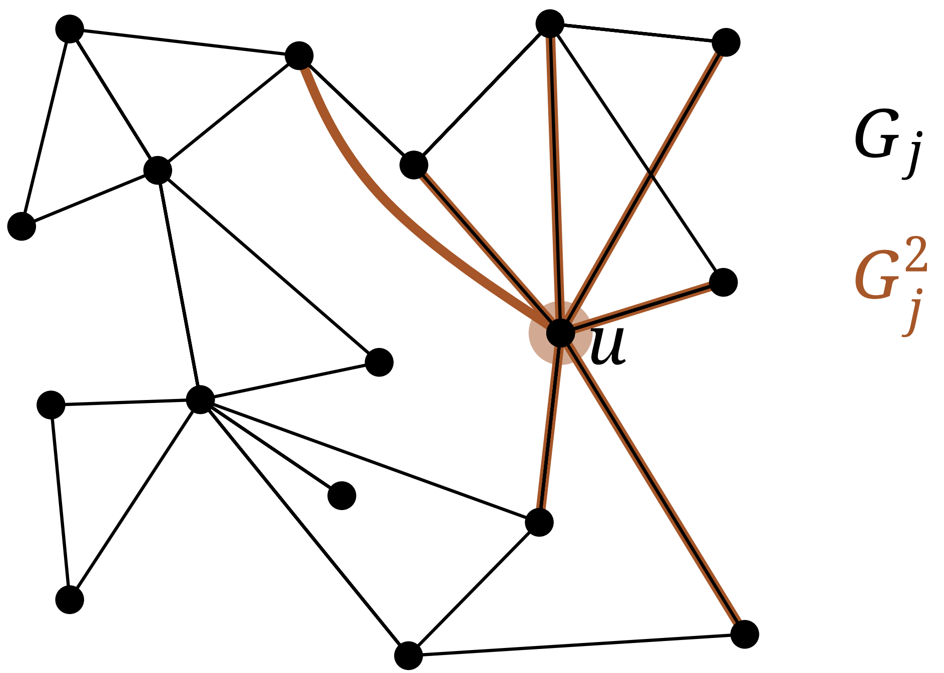


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "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 (graph-)distance at most two in H.



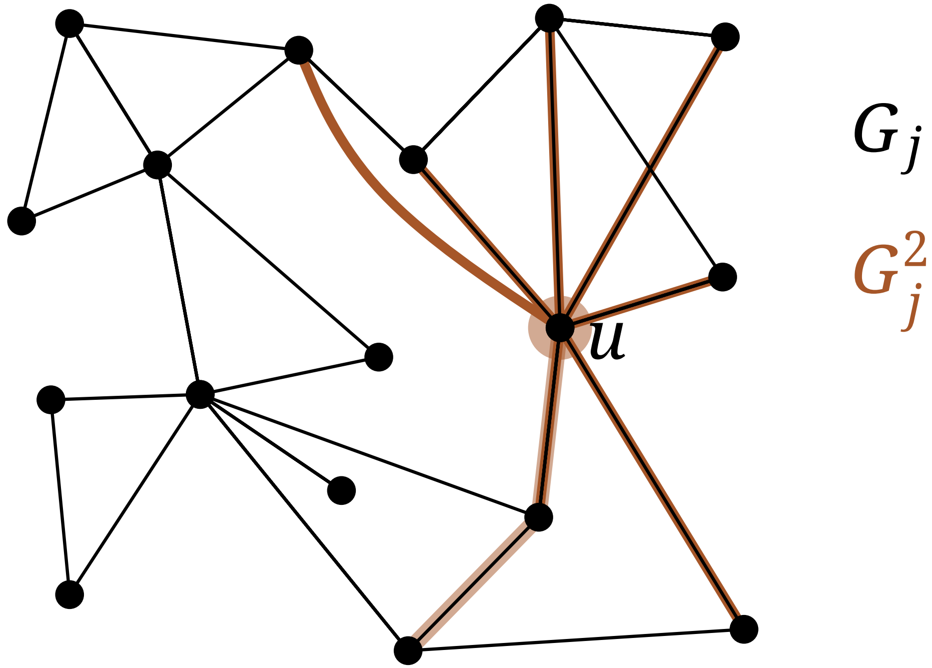
Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers

step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ “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 (graph-)distance at most two in H.

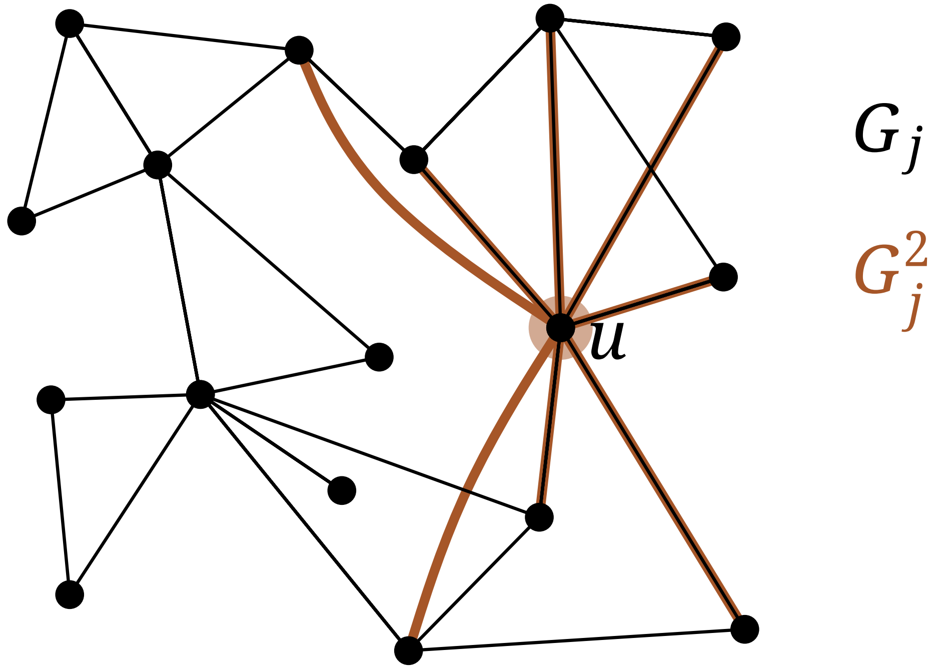


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ “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 (graph-)distance at most two in H.

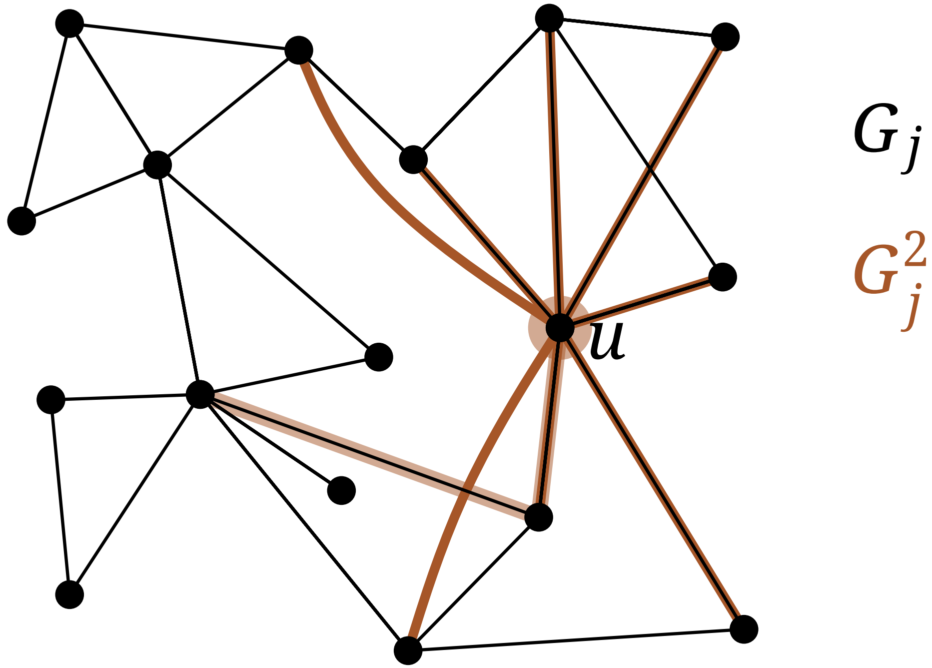


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ “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 (graph-)distance at most two in H.

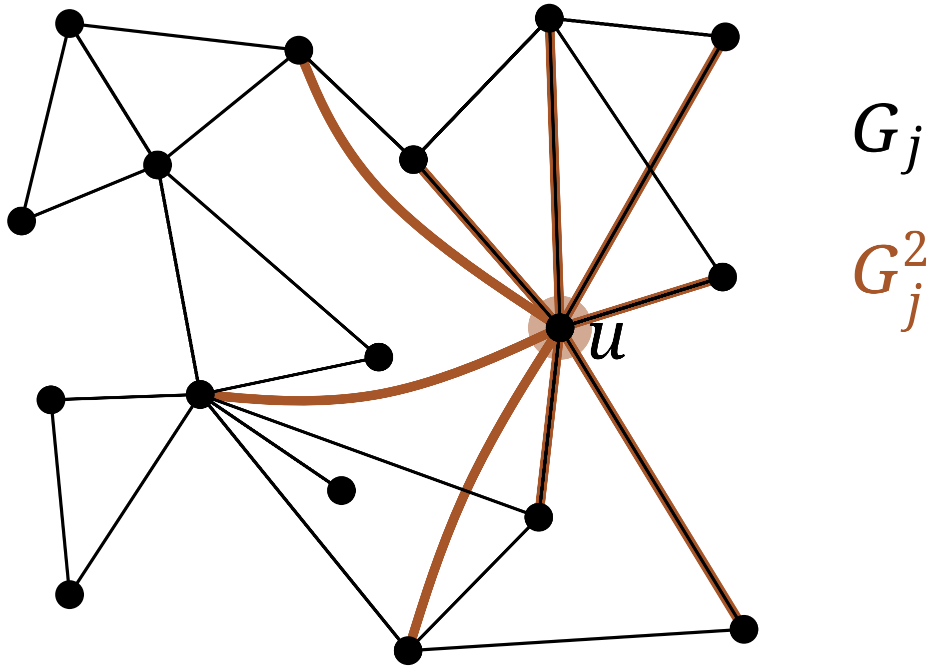


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "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 (graph-)distance at most two in H.

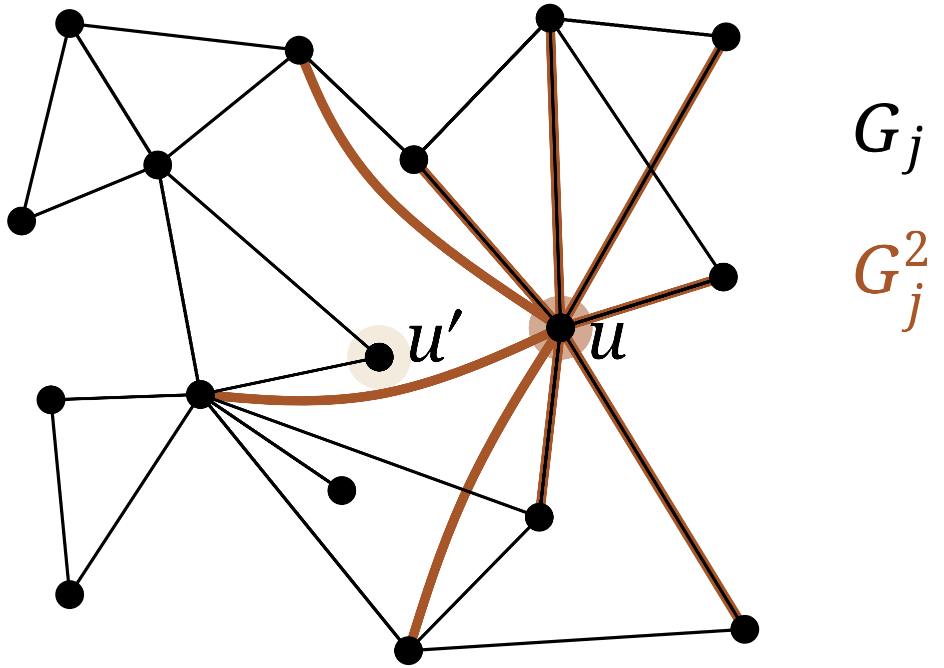


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ “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 (graph-)distance at most two in H.



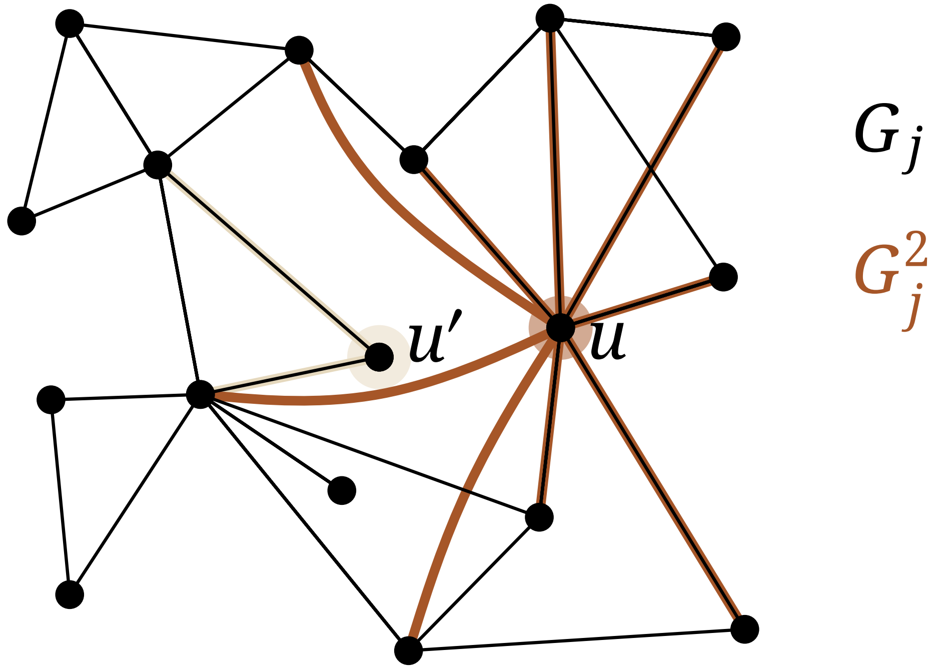
Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers

step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "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 (graph-)distance at most two in H.

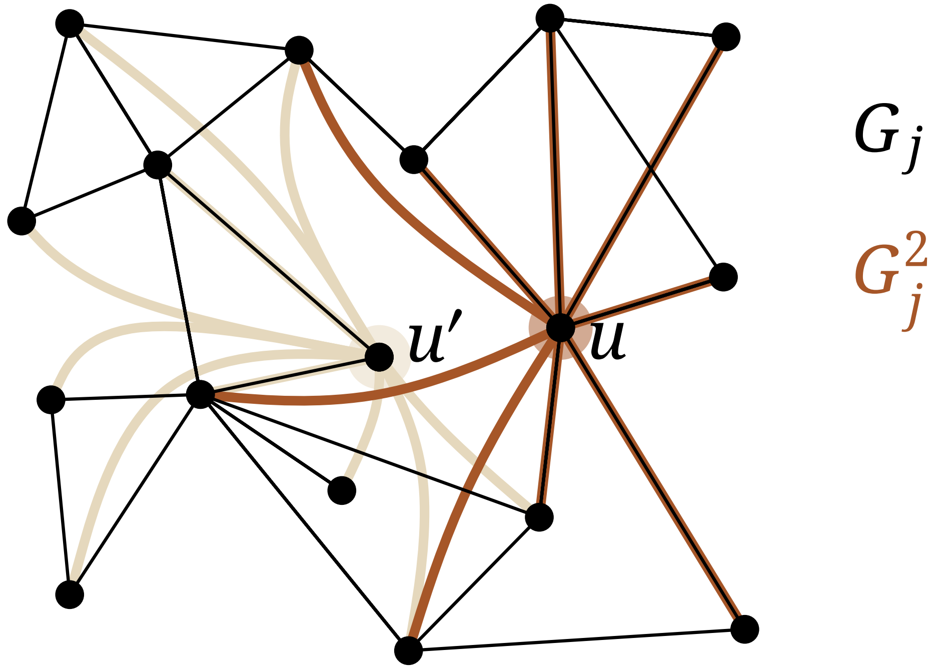


Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ “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 (graph-)distance at most two in H.



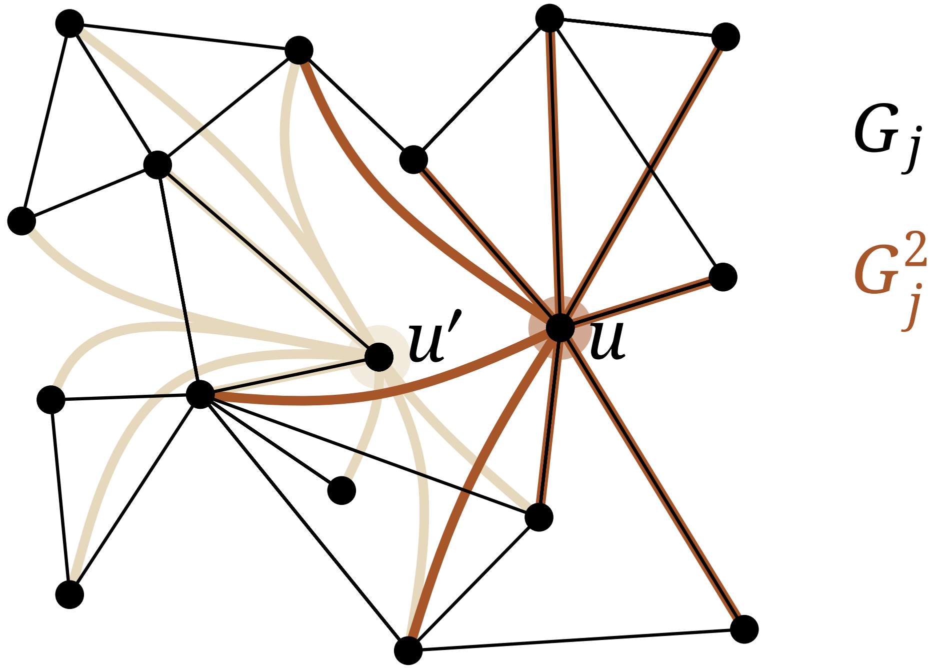
Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "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 (graph-)distance at most two in H.

Obs. If $\mathrm{OPT} \geq c(e_j)$ then a dominating set with at most k elements in G_j^2 is a 2-approximation for metric k-CENTER.



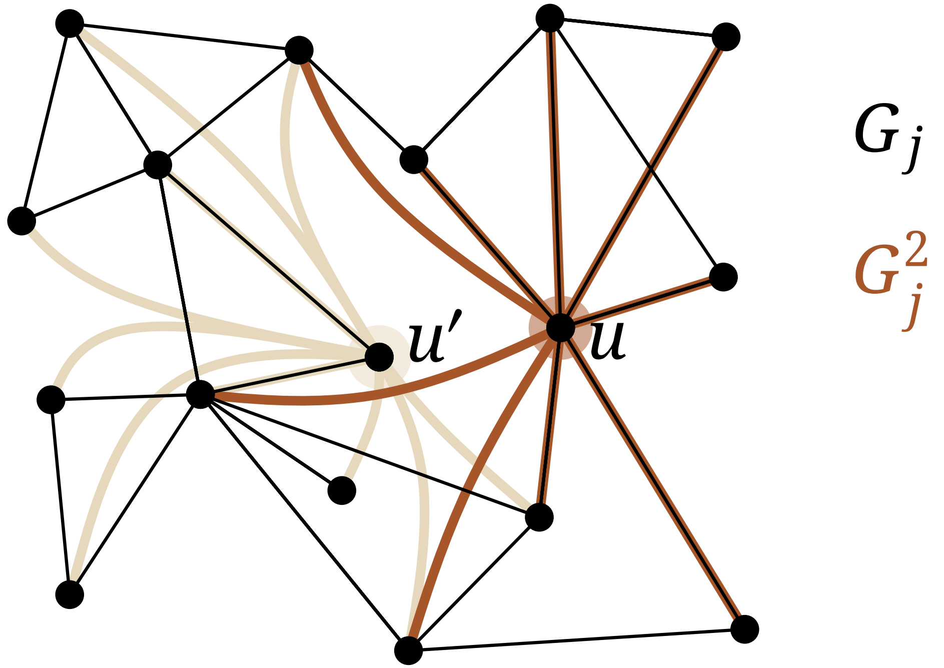
Square of a Graph

Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ “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 (graph-)distance at most two in H.

Obs. If $\mathrm{OPT} \geq c(e_j)$ then a dominating set with at most k elements in G_j^2 is a 2-approximation for metric k-CENTER.



Why?

Square of a Graph

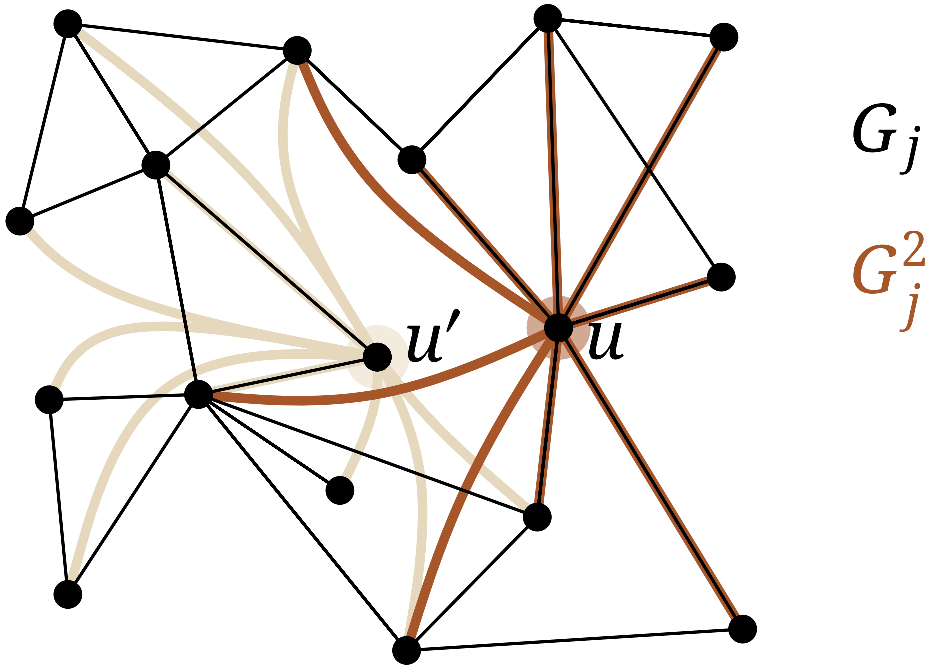
Idea: step 1.) Find a maximal set of vertices that have to connect to different centers
step 2.) show that this set gives an approximate solution

Ingredient 1: square of $G_j \approx$ "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 (graph-)distance at most two in H.

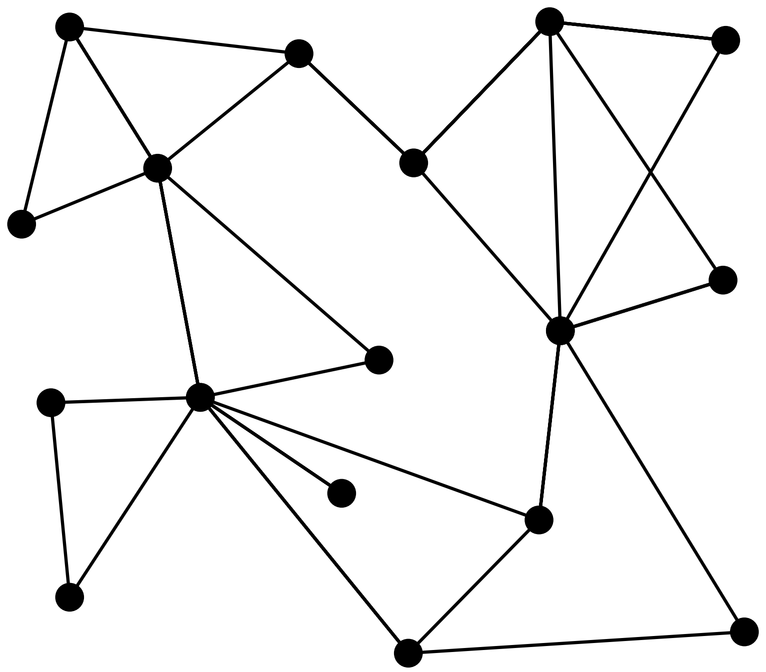
Obs. If $\mathrm{OPT} \geq c(e_j)$ then a dominating set with at most k elements in G_j^2 is a 2-approximation for metric k-CENTER.

Why? triangle inequality



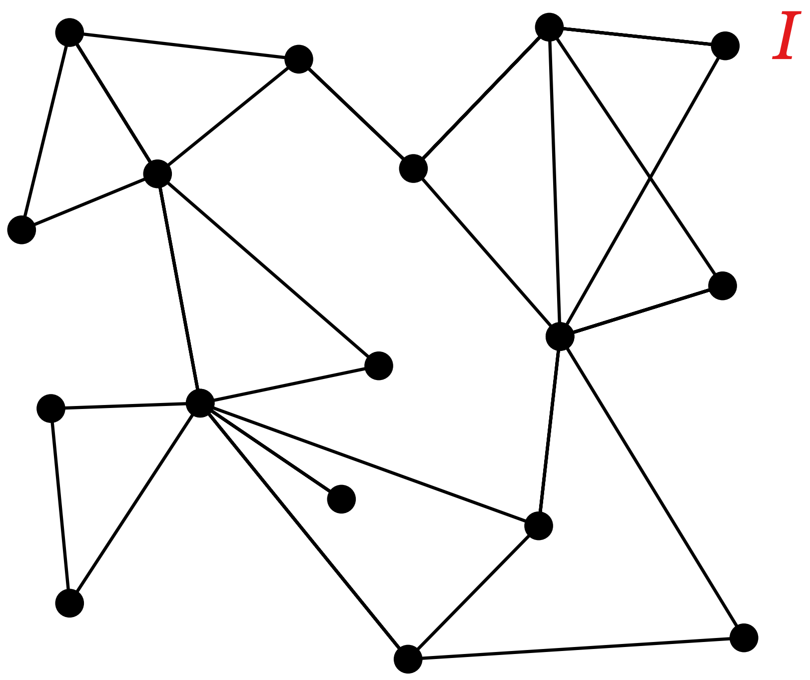
Ingredient 2: Independent Sets

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



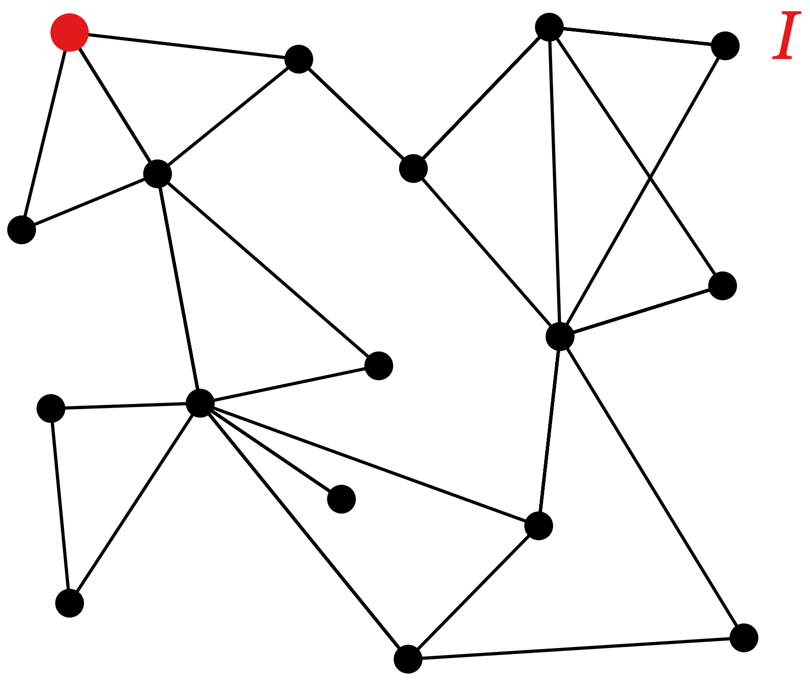
Ingredient 2: Independent Sets

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



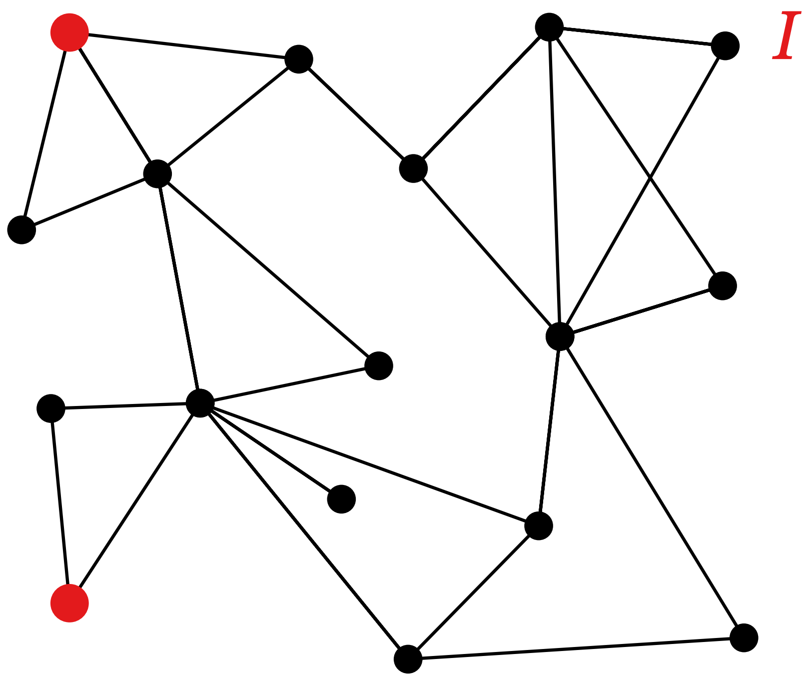
Ingredient 2: Independent Sets

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



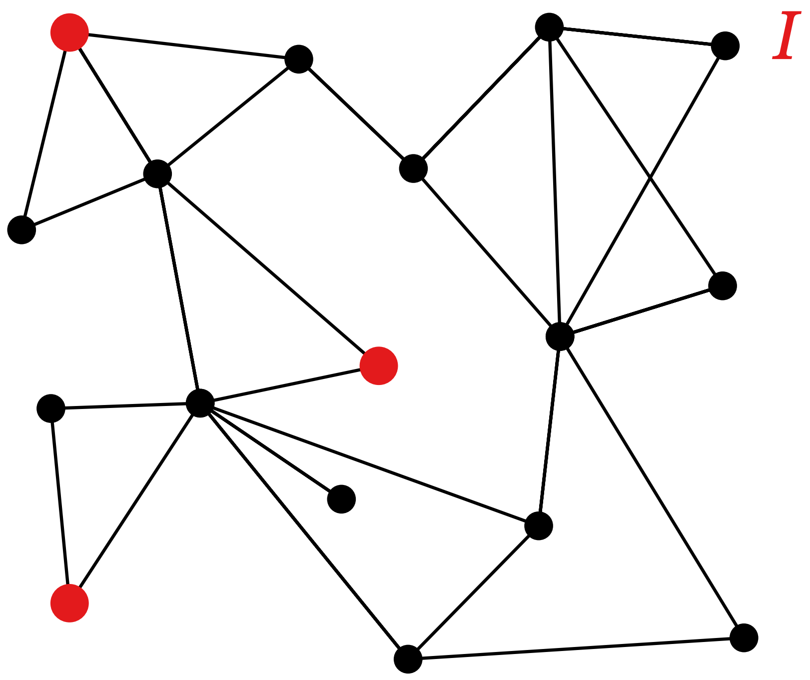
Ingredient 2: Independent Sets

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



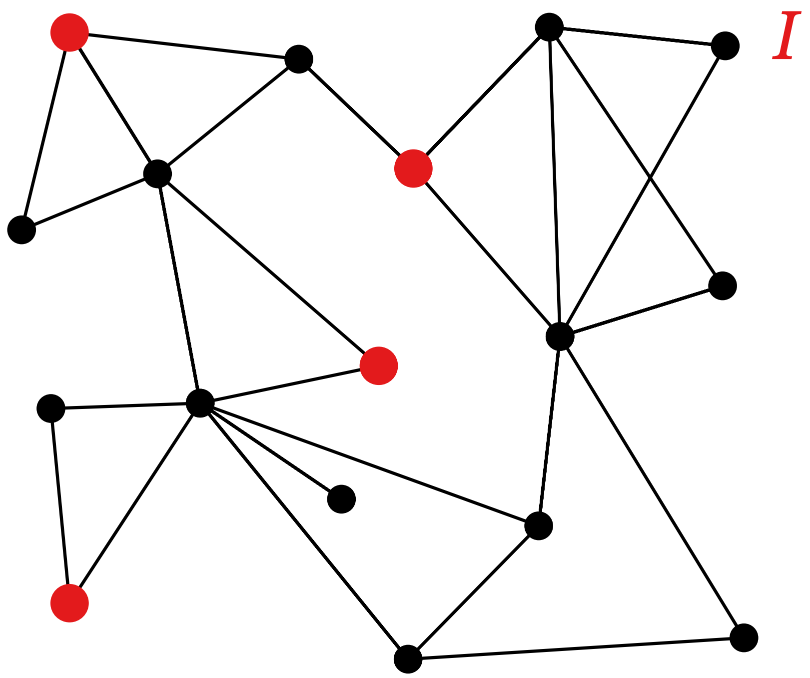
Ingredient 2: Independent Sets

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



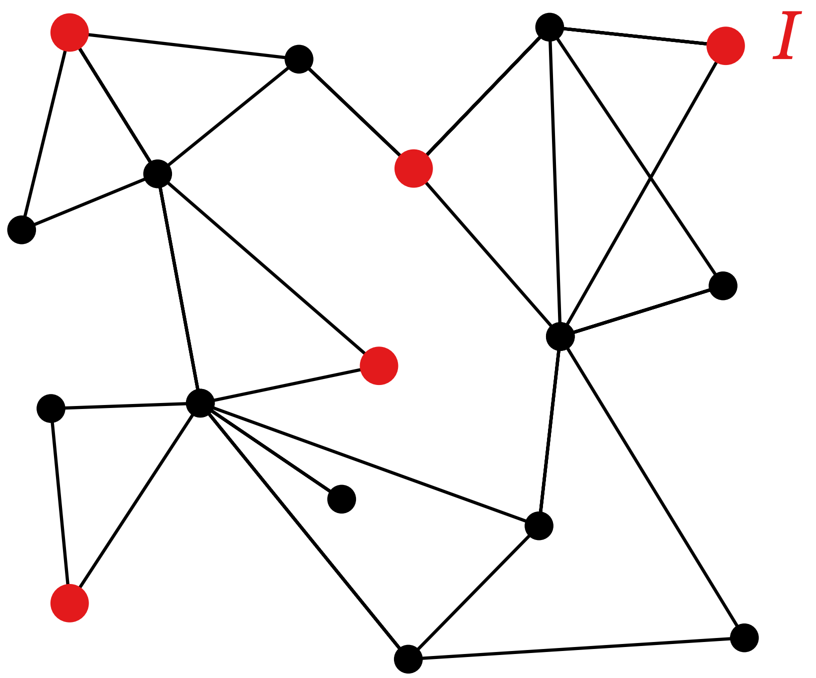
Ingredient 2: Independent Sets

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



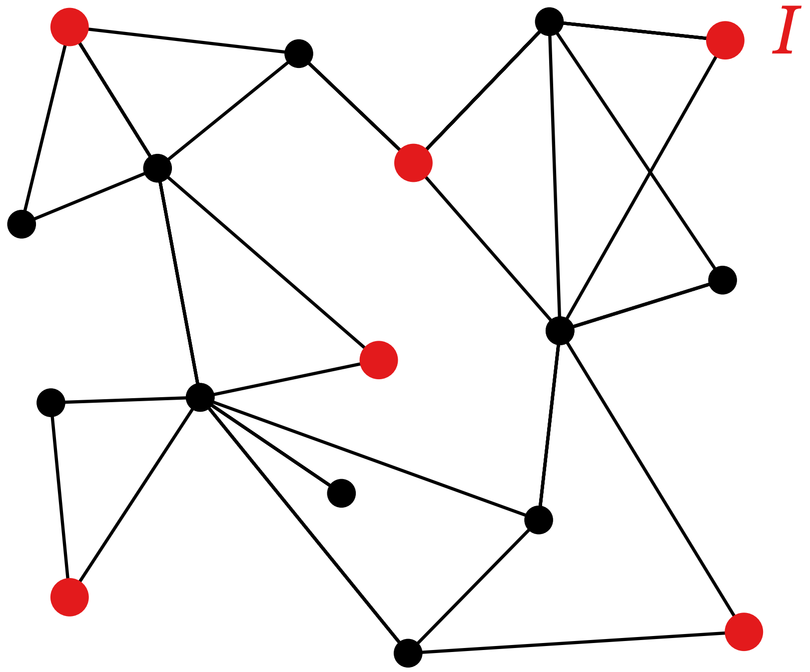
Ingredient 2: Independent Sets

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



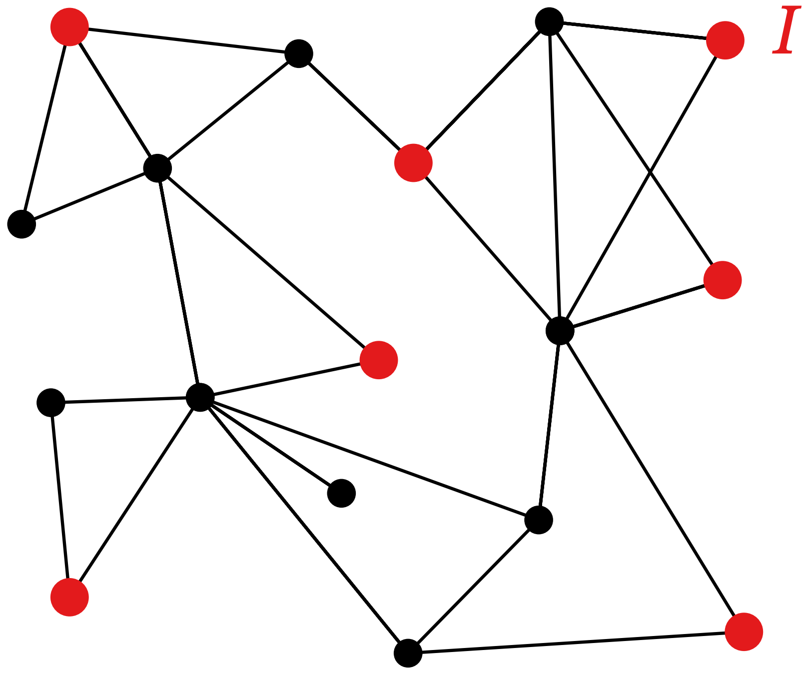
Ingredient 2: Independent Sets

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



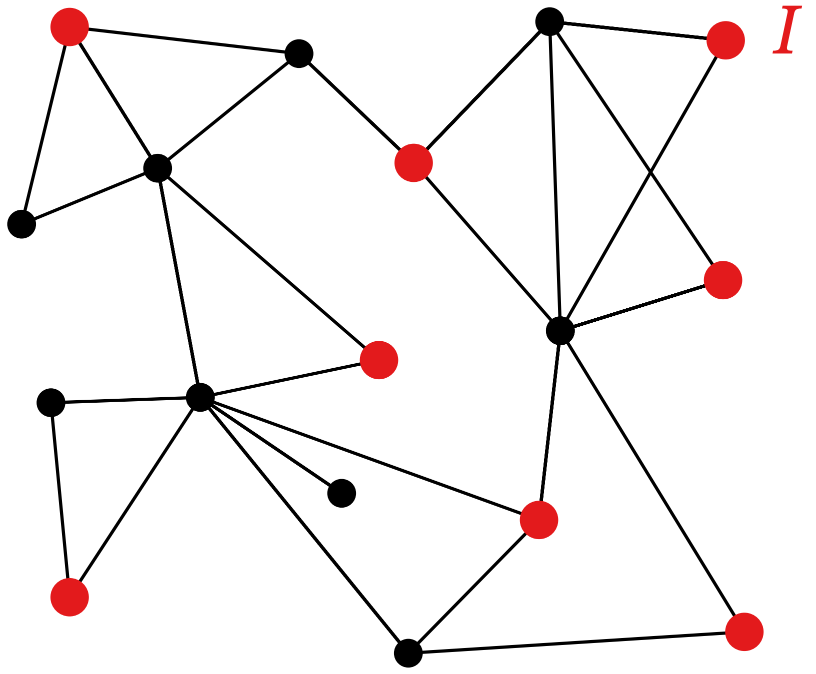
Ingredient 2: Independent Sets

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



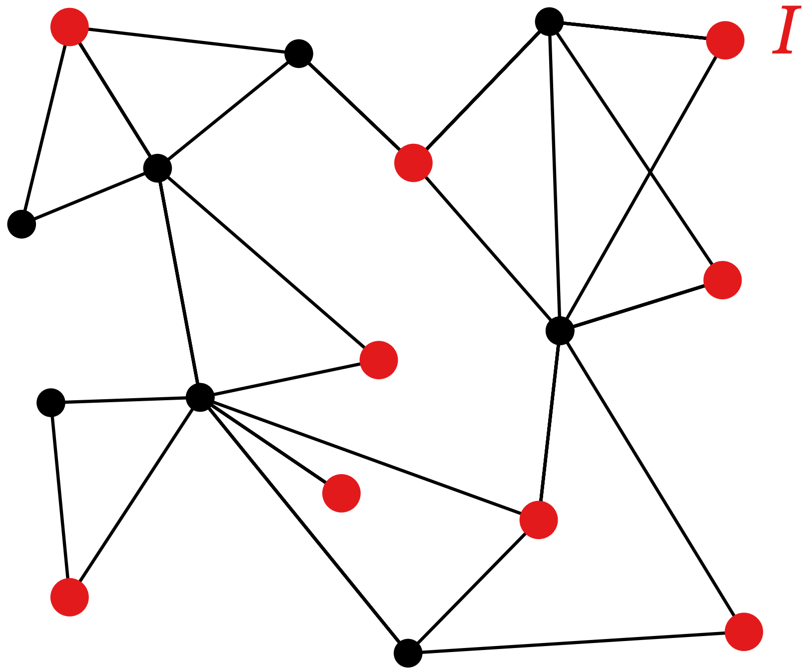
Ingredient 2: Independent Sets

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



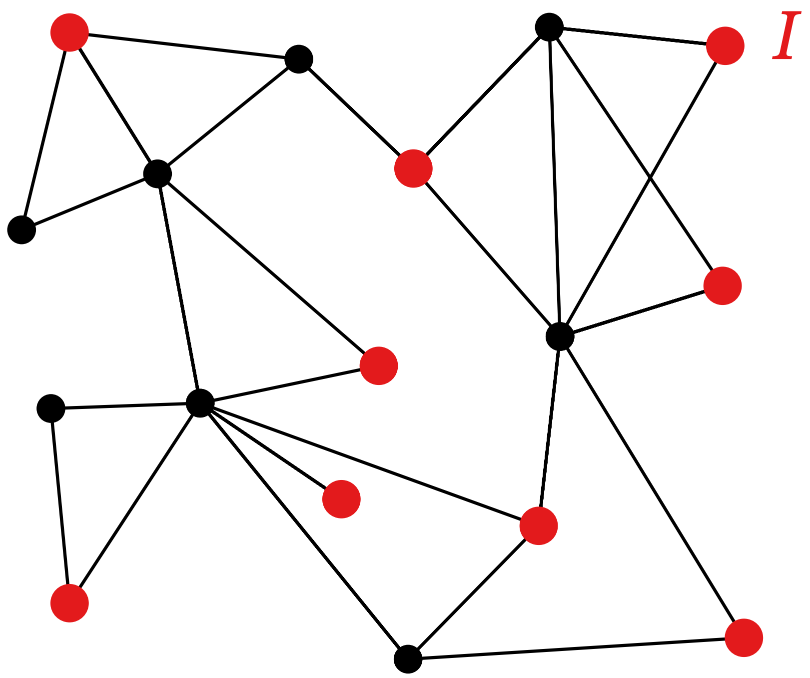
Ingredient 2: Independent Sets

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



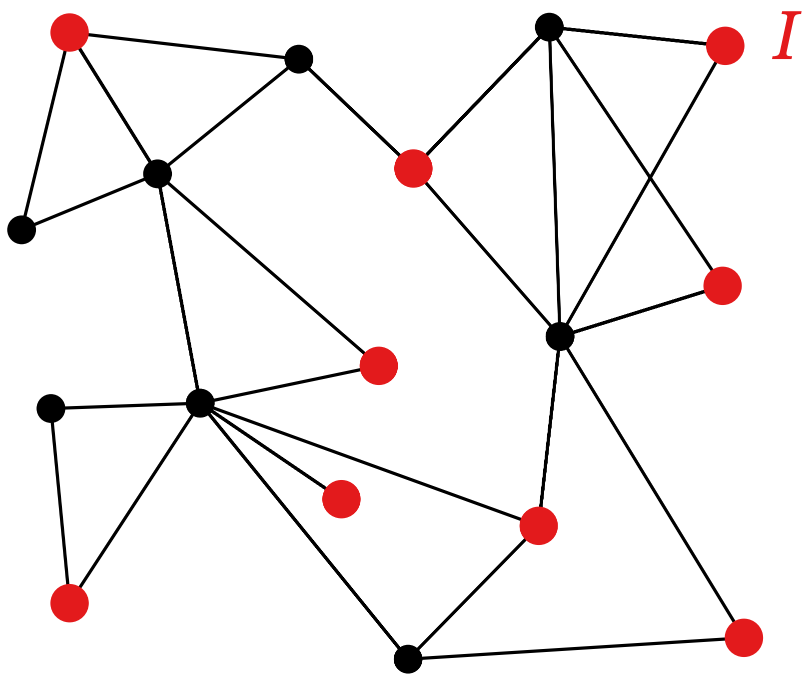
Ingredient 2: Independent Sets

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 no superset of it is independent.



Ingredient 2: 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 no superset of it is independent.



Obs. Maximal independent sets are dominating sets!

Independent Sets in H^2

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

Independent Sets in H^2

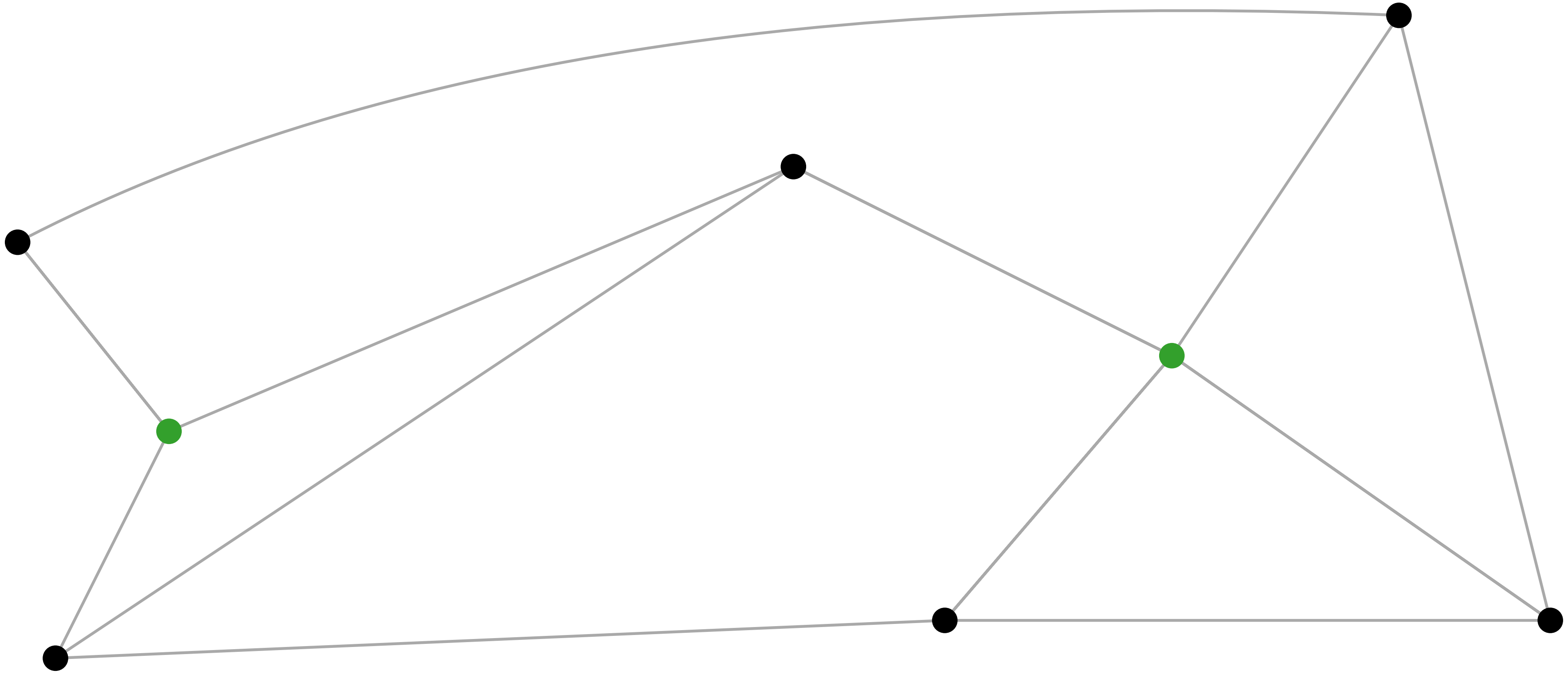
Lemma. For a graph H and an independent set I in H^2, $|I| \leq \text{dom}(H)$.

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, $|I| \leq \mathrm{dom}(H)$.

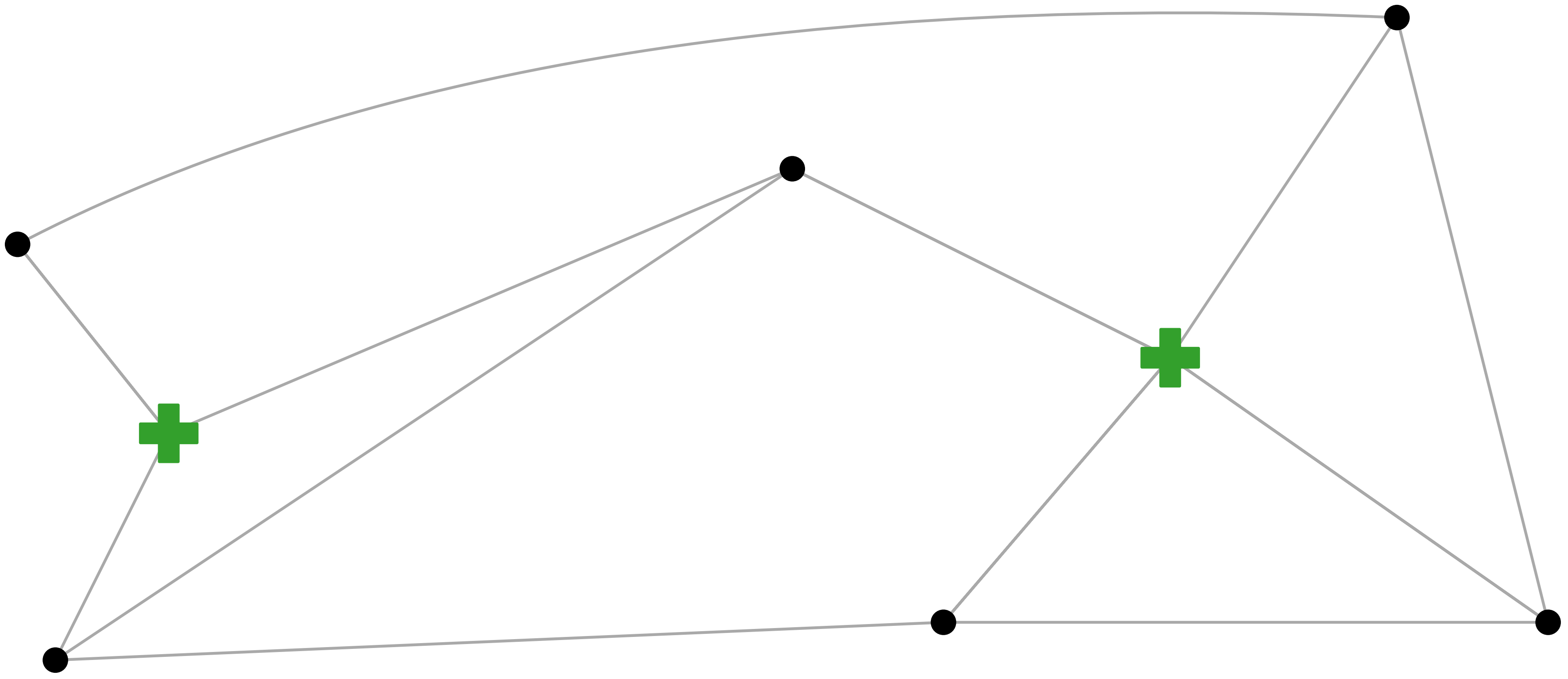
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

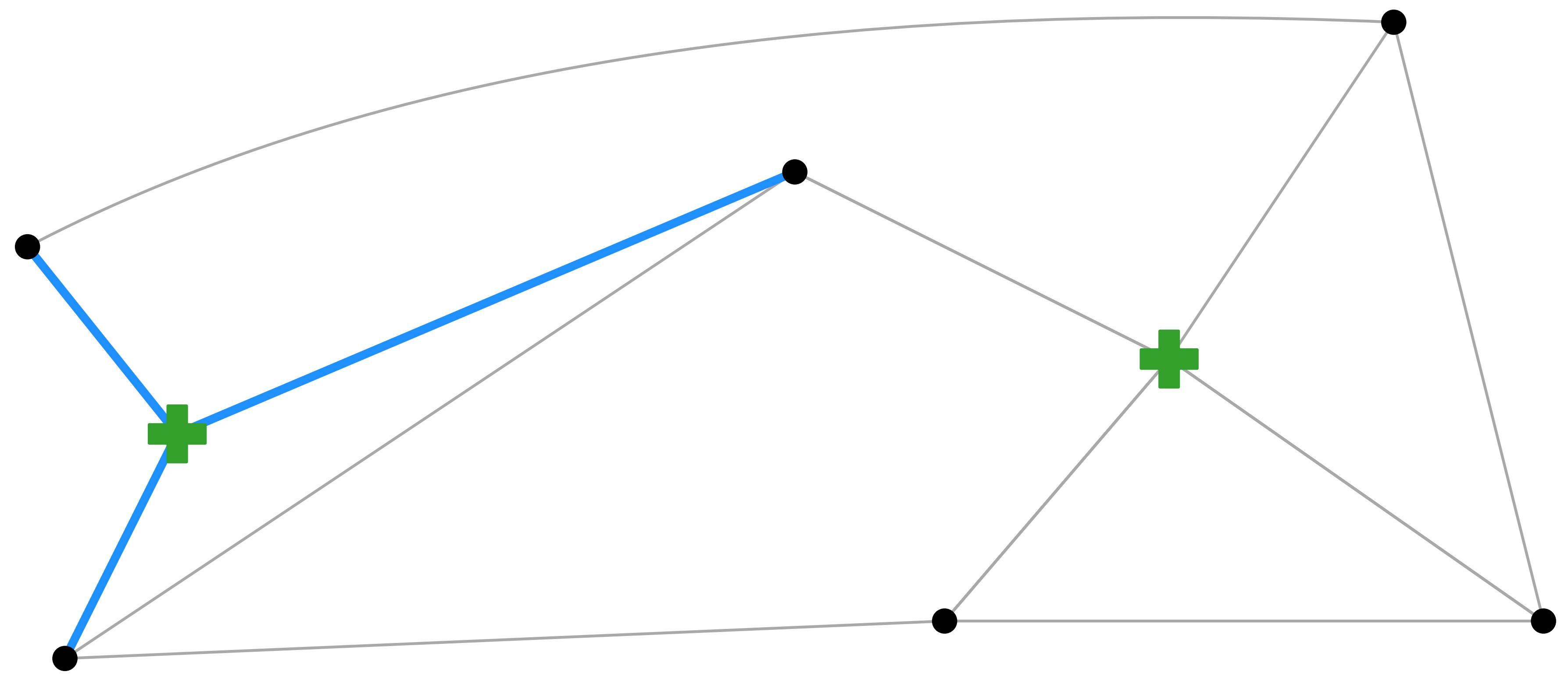
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

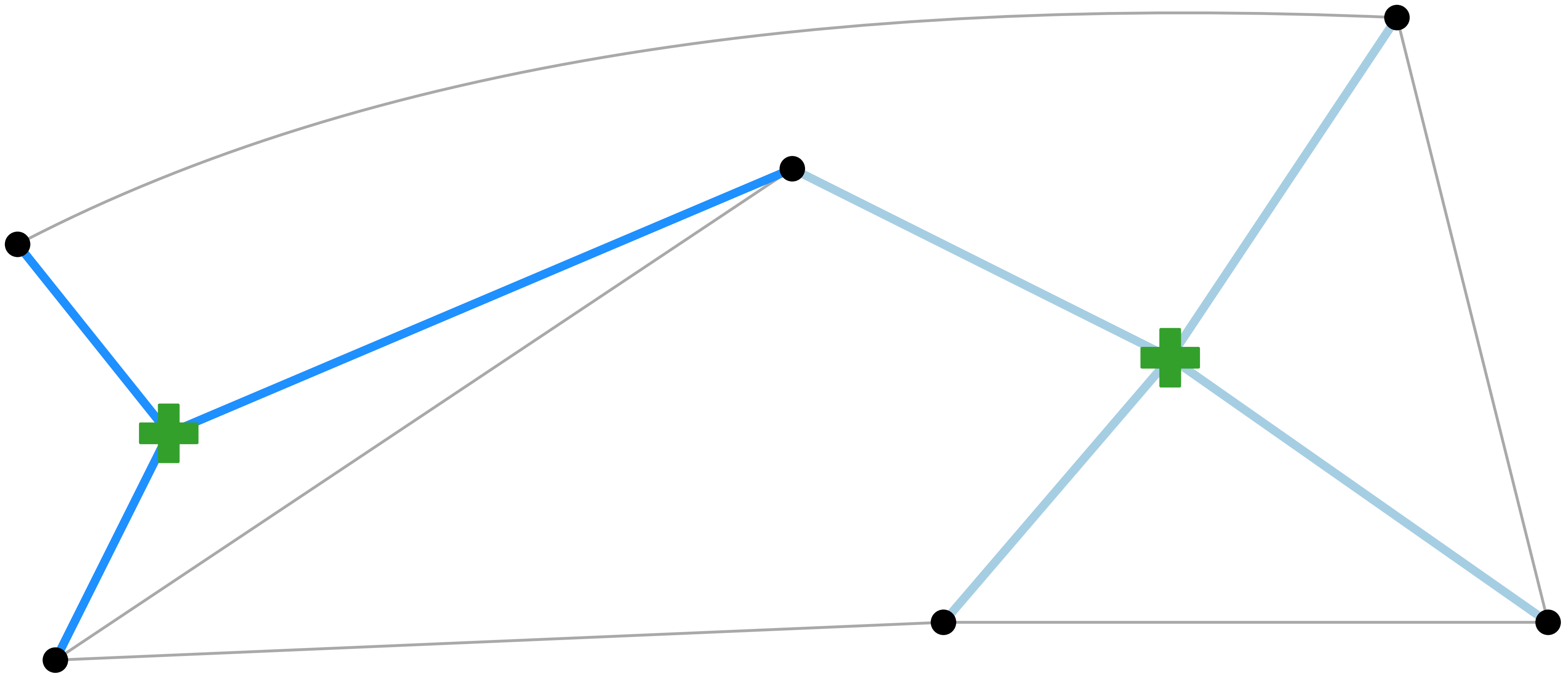
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

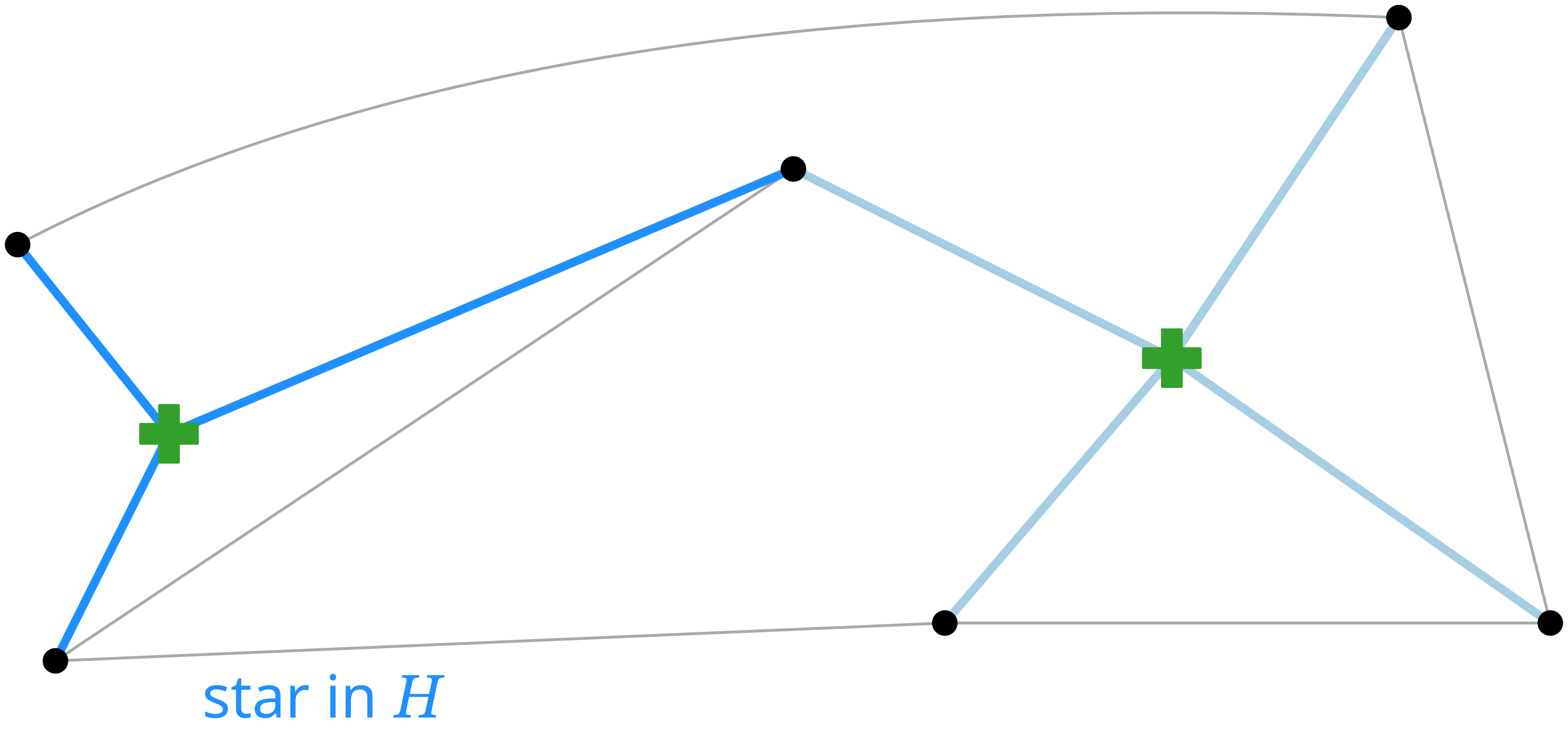
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

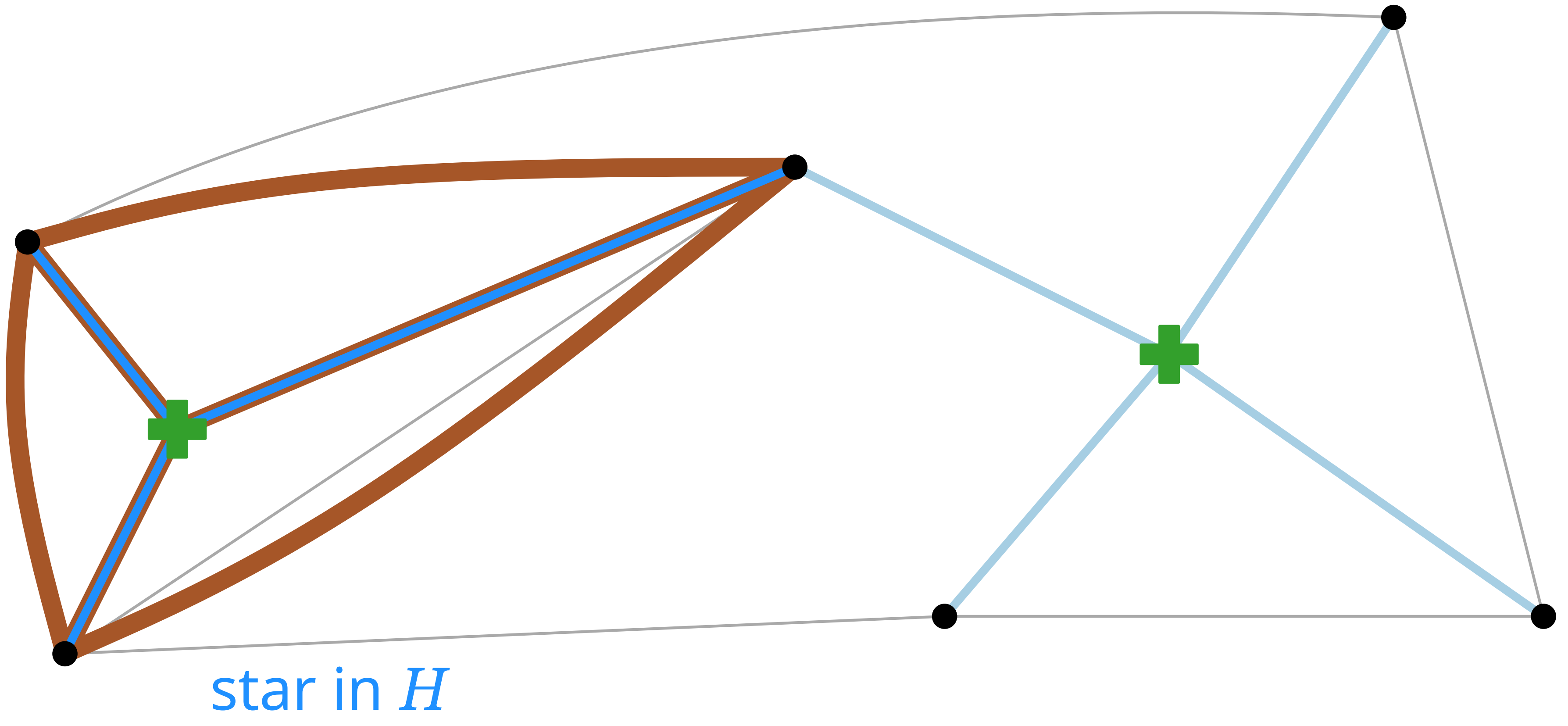
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

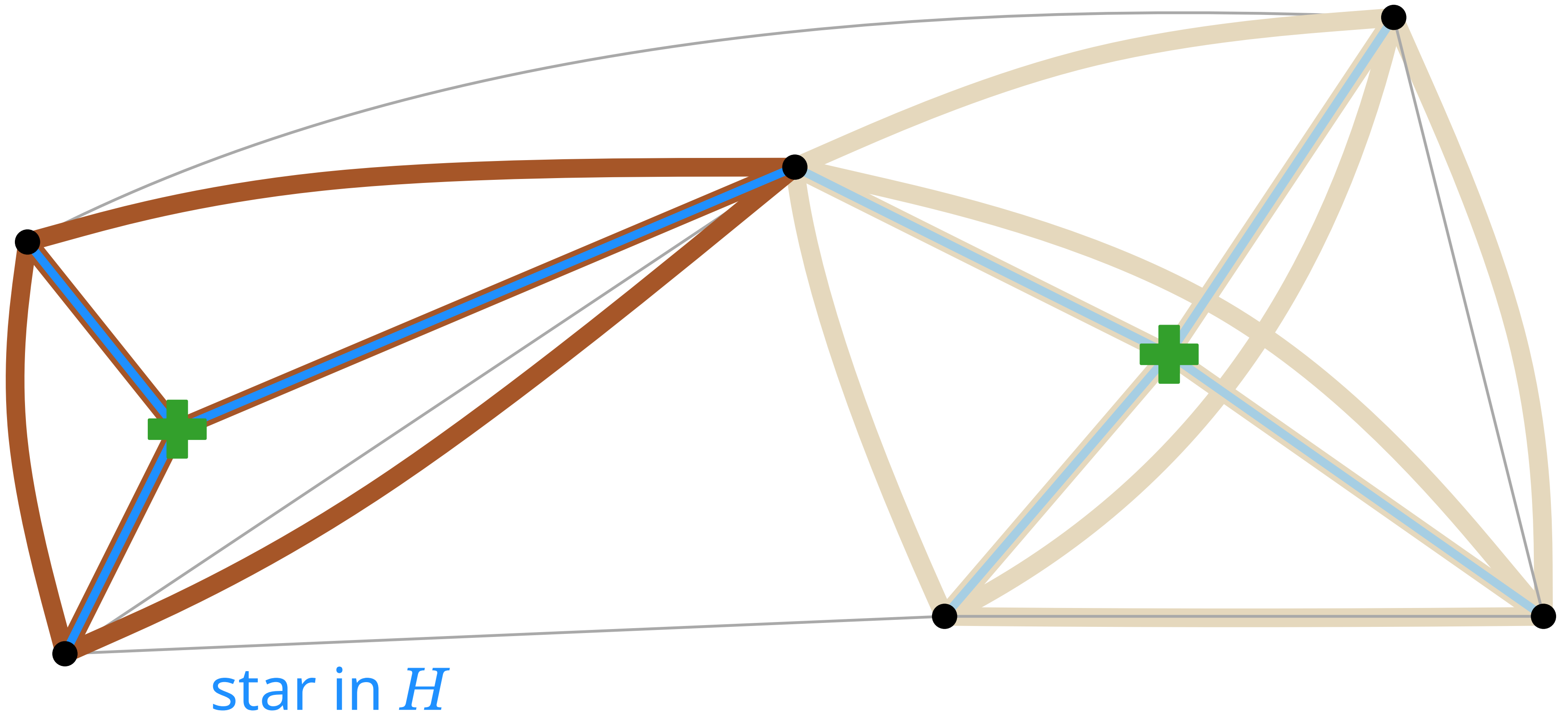
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

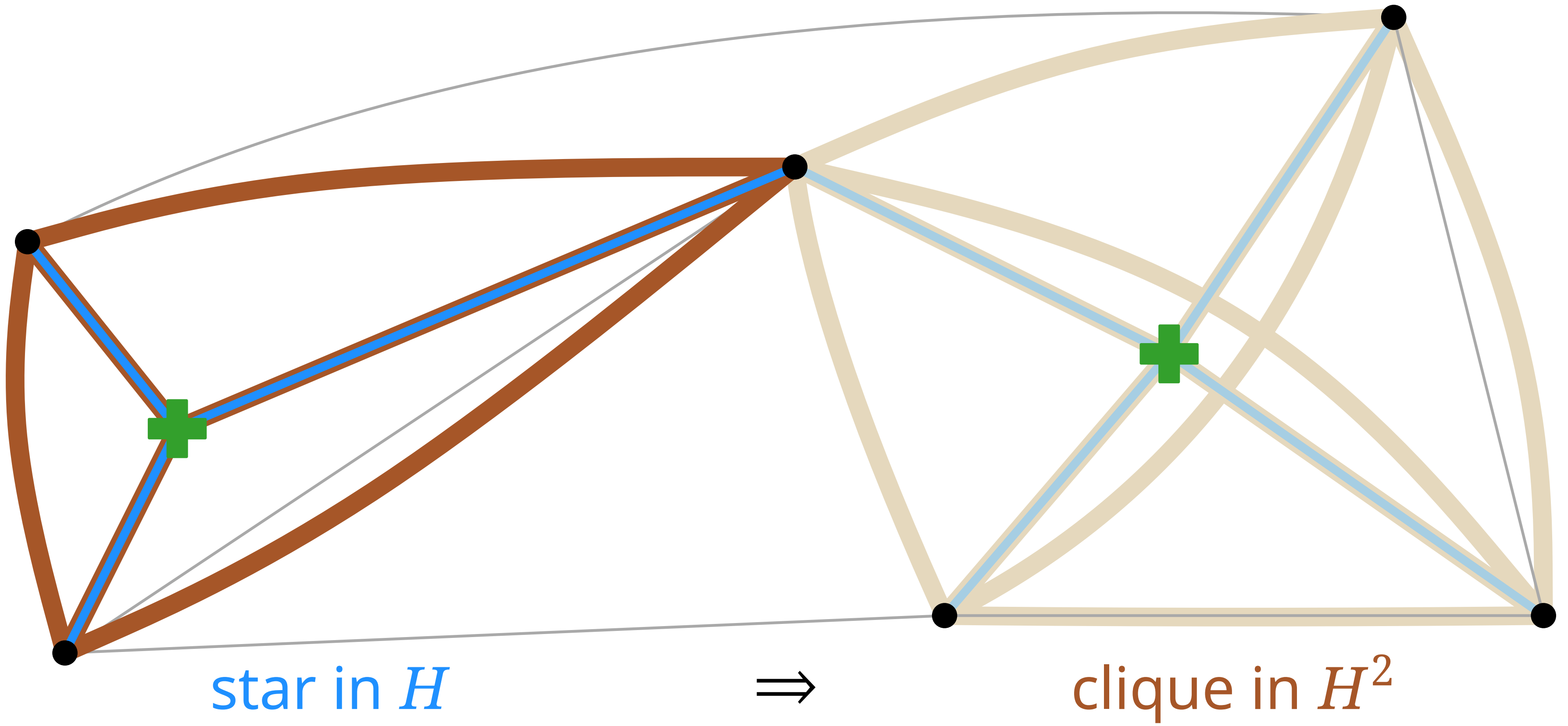
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

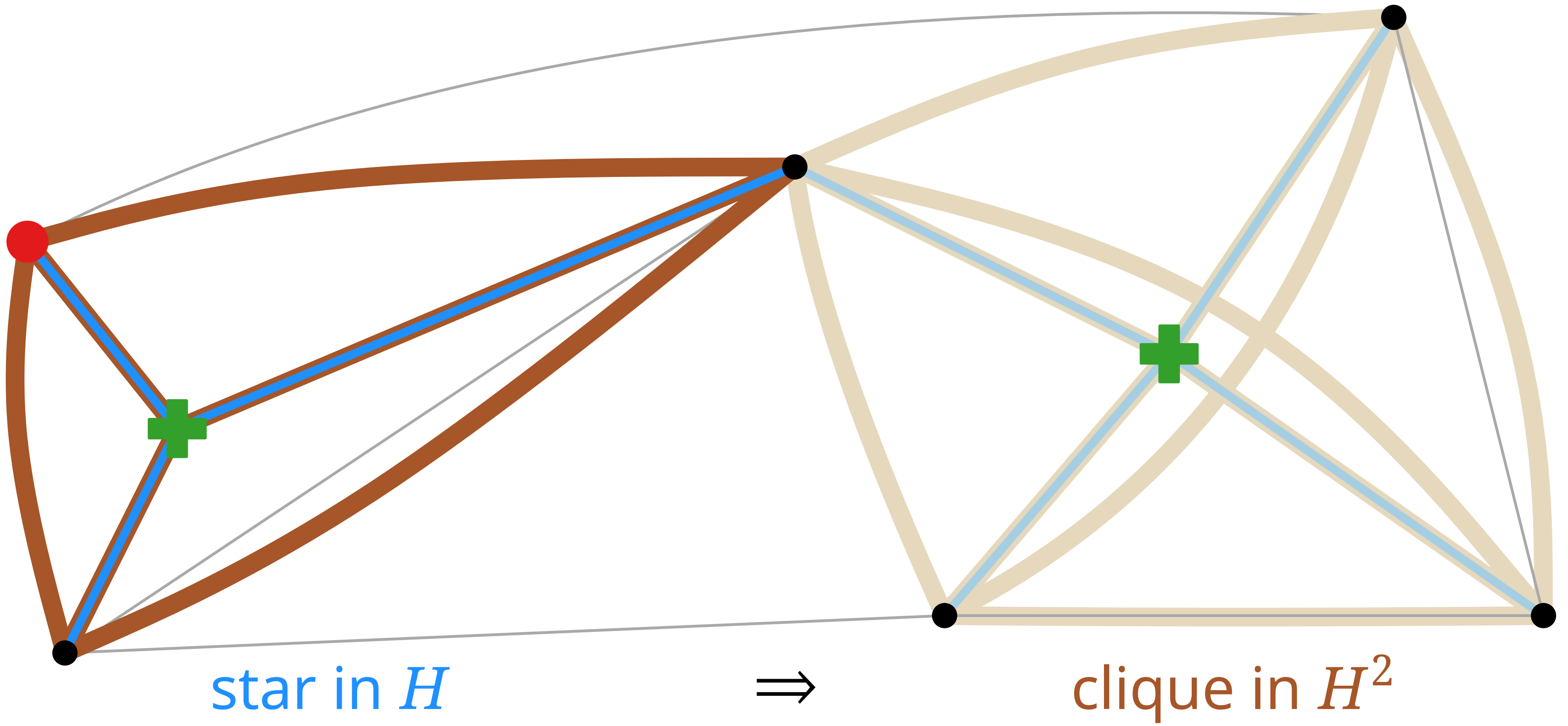
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

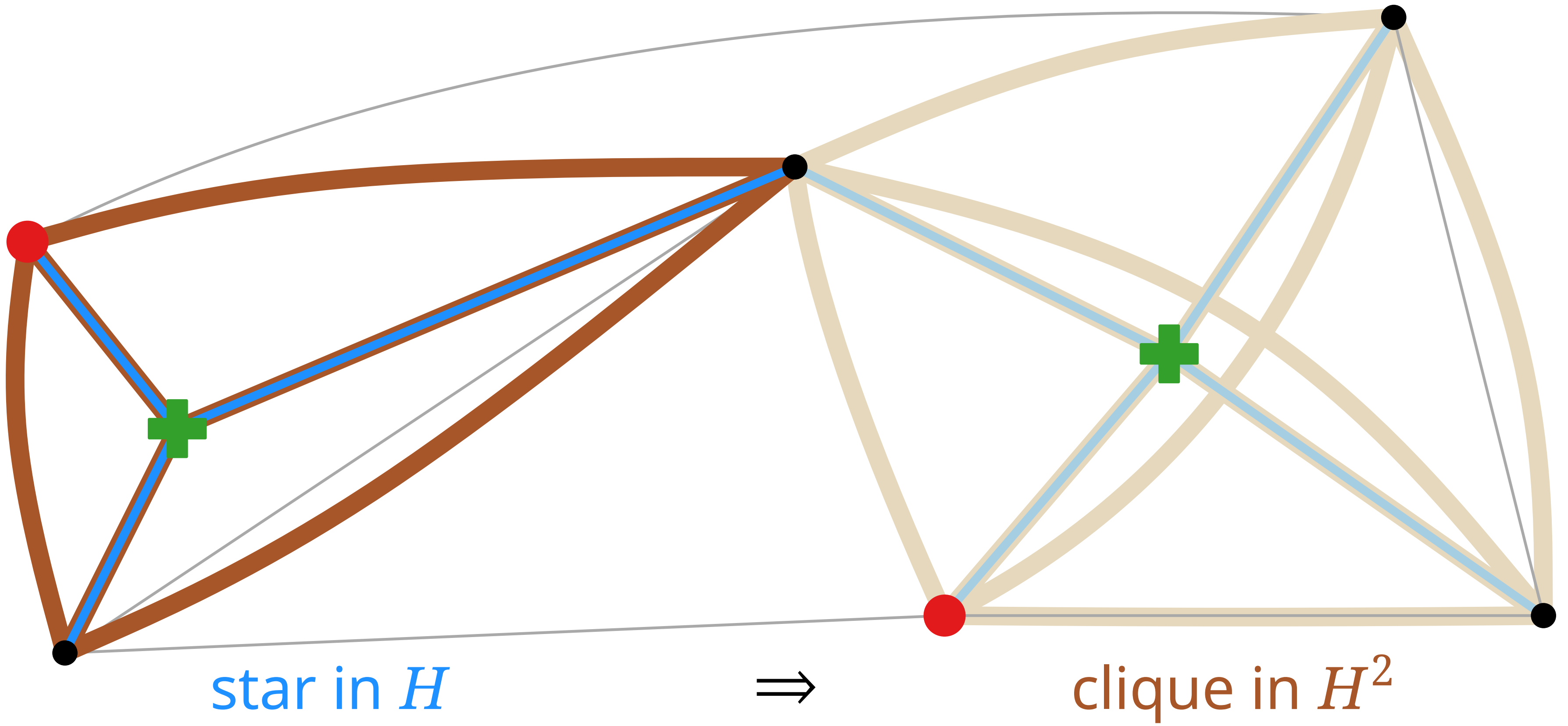
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

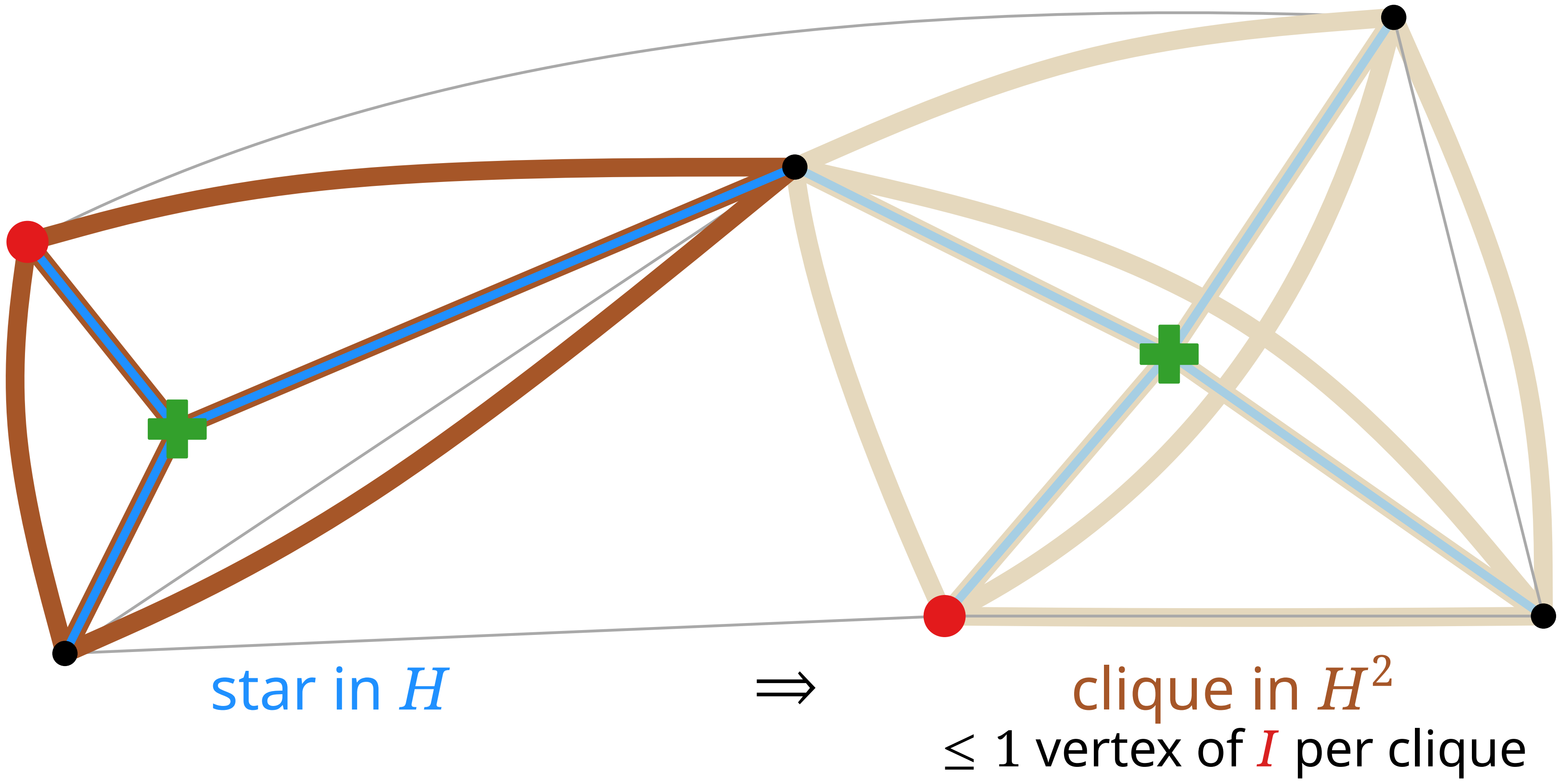
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

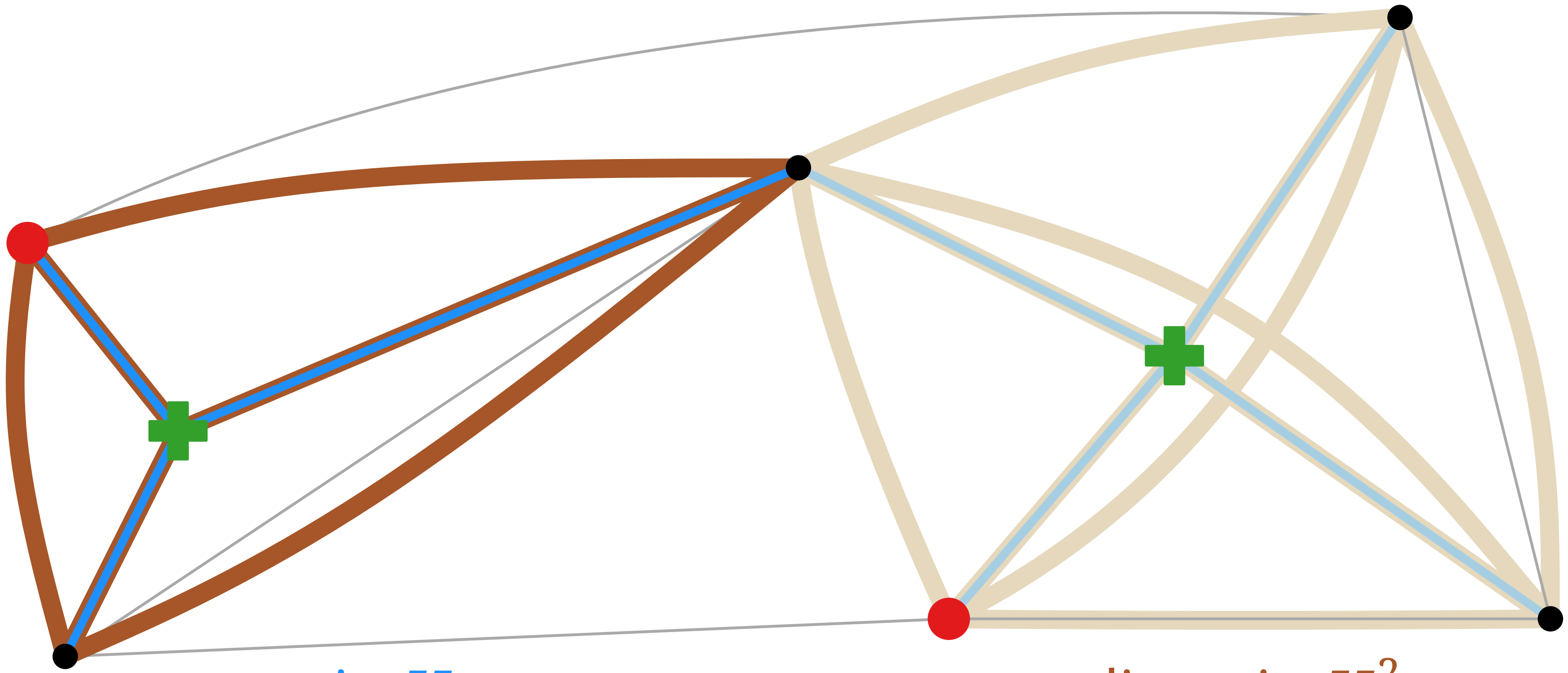
What does a dominating set of H look like in H^2?



Independent Sets in H^2

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

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



star in H $\Rightarrow$ clique in H^2

$\leq |I|$ stars $\Leftarrow$ ≤ 1 vertex of I per clique

Factor-2 Approx. for Metric k-Center

Metric-k-Center($G = (V, E; c), k$)

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

Factor-2 Approx. for Metric k-Center

Metric-k-Center$(G = (V, E; c), k)$

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

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

Factor-2 Approx. for Metric k-CENTER

Metric-k-CENTER($G = (V, E; c), k$)

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

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

Construct G_j^2

Factor-2 Approx. for Metric k-Center

Metric-k-Center$(G = (V, E; c), k)$

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

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($G = (V, E; c), k$)

Sort the edges of G by cost: $c(e_1) \leq \ldots \leq c(e_m)$
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($G = (V, E; c), k$)

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

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($G = (V, E; c), k$)

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

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 $c(e_j) \leq \mathrm{OPT}$.

because for $i < j$: $\mathrm{dom}(G_i) \geq |I_i| > k$.

Factor-2 Approx. for Metric k-CENTER

Metric-k-CENTER($G = (V, E; c), k$)

Sort the edges of G by cost: $c(e_1) \leq \ldots \leq c(e_m)$
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 $c(e_j) \leq \mathrm{OPT}$.

because for $i < j$: $\mathrm{dom}(G_i) \geq |I_i| > k$.

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

Factor-2 Approx. for Metric k-Center

Metric-k-Center($G = (V, E; c), k$)

Sort the edges of G by cost: $c(e_1) \leq \ldots \leq c(e_m)$
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 $c(e_j) \leq \mathrm{OPT}$.

because for $i < j$: $\mathrm{dom}(G_i) \geq |I_i| > k$.

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

I_j dominating in G_j^2: $\mathrm{cost}(I_j) \leq 2c(e_j) \leq 2\mathrm{OPT}$

Factor-2 Approx. for Metric k-CENTER

Metric-k-CENTER($G = (V, E; c), k$)

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

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

recall: parametric pruning:

- reduce to decision problem, work on pruned instance $I(t)$
- step 1: Use family of $I(t)$ to compute lower bound t^*
- step 2: find solution in $I(\alpha t^*)$

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

because for $i < j$:
$\mathrm{dom}(G_i) \geq |I_i| > k$.

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

I_j dominating in G_j^2:
$\mathrm{cost}(I_j) \leq 2c(e_j) \leq 2\mathrm{OPT}$

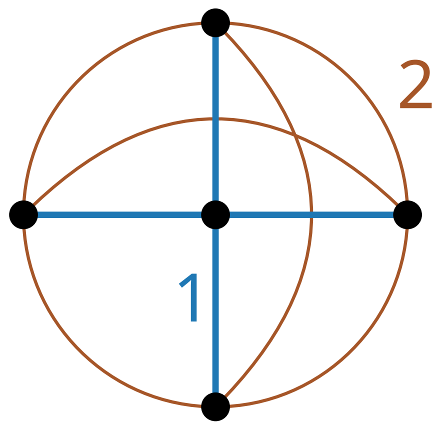
Can we do better . . . ?

Can we do better . . . ?

What about a tight example?

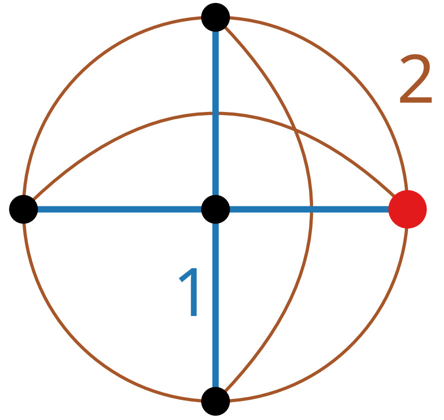
Can we do better . . . ?

What about a tight example?



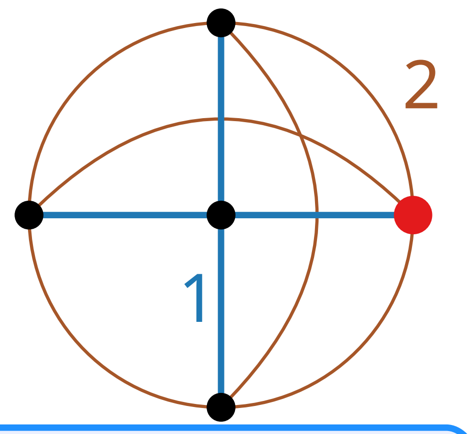
Can we do better . . . ?

What about a tight example?



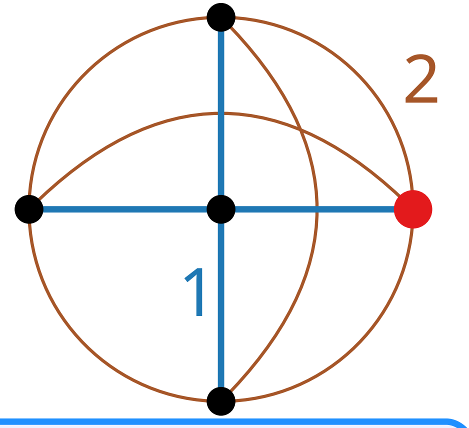
Can we do better . . . ?

What about a tight example?



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

Can we do better . . . ?

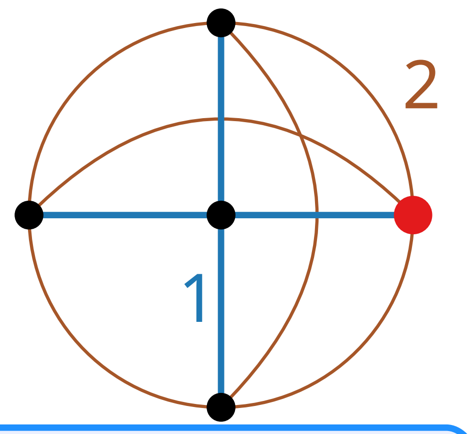


What about a tight example?

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

Proof. Reduce from dominating set to metric k-Center.

Can we do better . . . ?



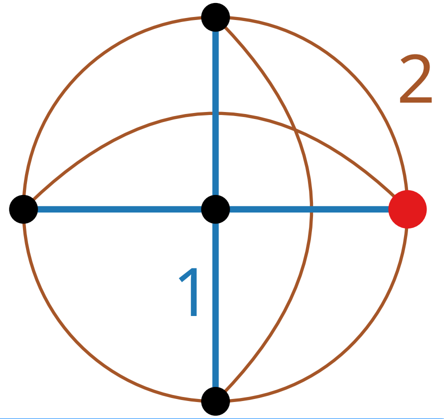
What about a tight example?

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

Proof. Reduce from dominating set to metric k-CENTER.
Given graph $G = (V, E)$ and integer k,

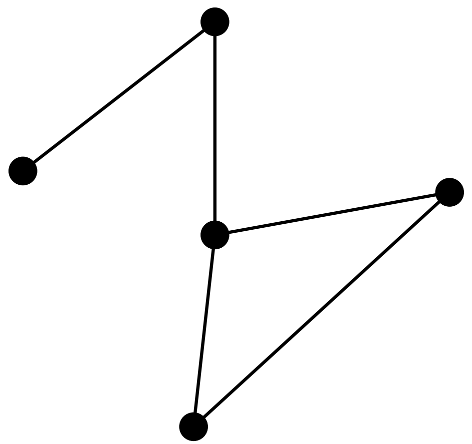
Can we do better . . . ?

What about a tight example?

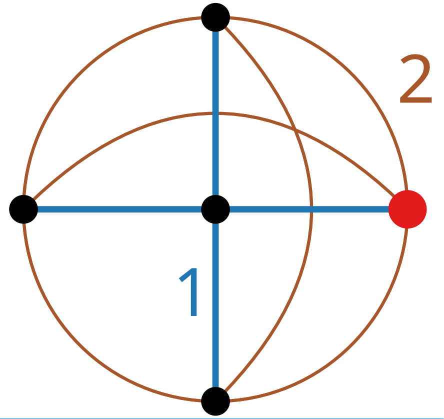


Theorem. Assuming P $\neq$ NP, there is no factor-$(2-\varepsilon)$ approximation algorithm for the metric k-Center problem, for any $\varepsilon > 0$.

Proof. Reduce from dominating set to metric k-Center.
Given graph $G = (V, E)$ and integer k,



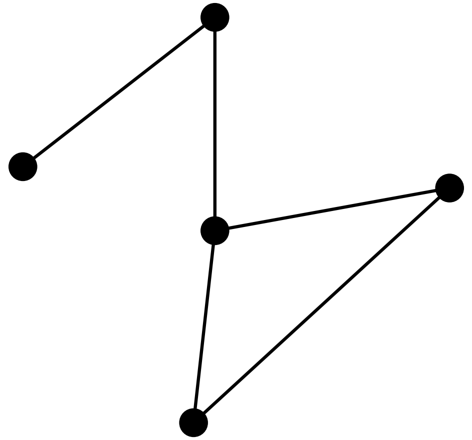
Can we do better . . . ?



What about a tight example?

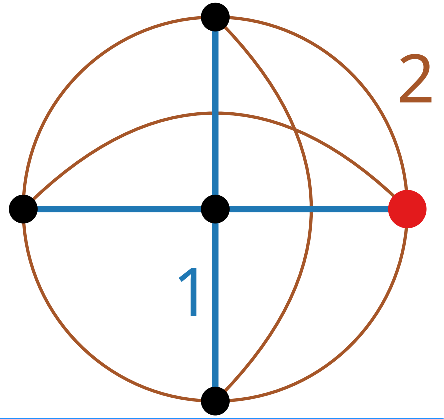
Theorem. Assuming P $\neq$ NP, there is no factor-$(2-\varepsilon)$ approximation algorithm for the metric k-Center problem, for any $\varepsilon > 0$.

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



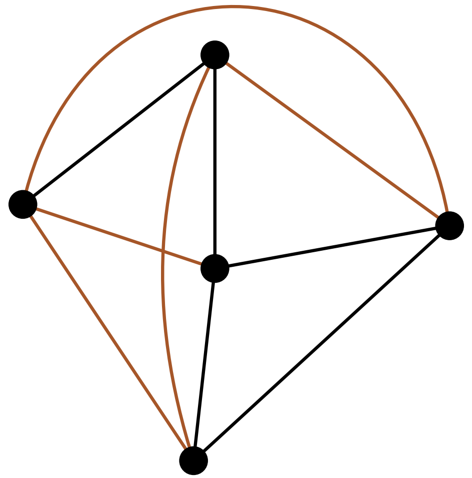
Can we do better . . . ?

What about a tight example?



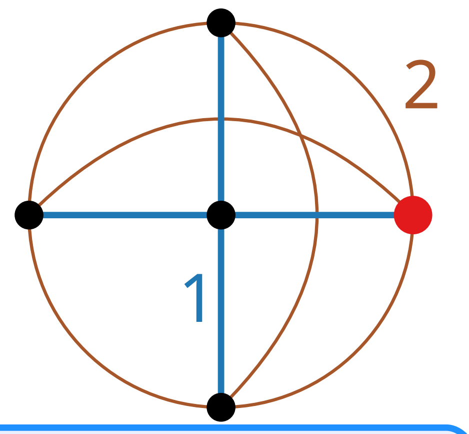
Theorem. Assuming P $\neq$ NP, there is no factor-$(2-\varepsilon)$ approximation algorithm for the metric k-CENTER problem, for any $\varepsilon > 0$.

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



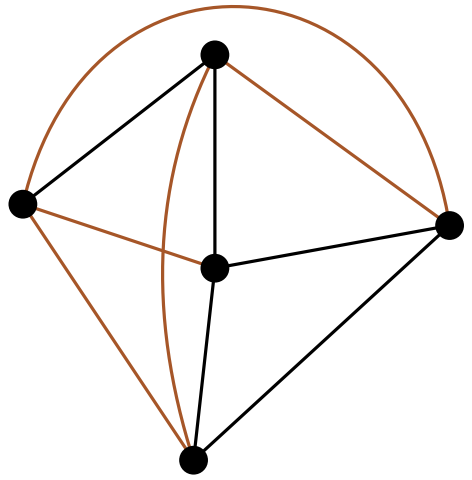
Can we do better ...?

What about a tight example?

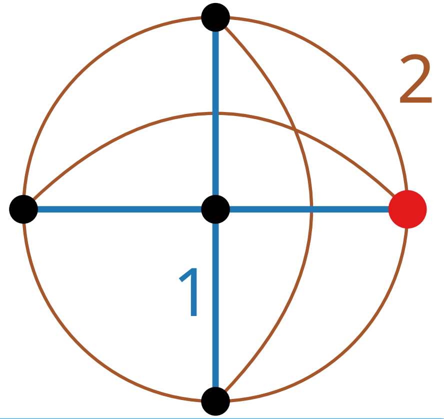


Theorem. Assuming P $\neq$ NP, there is no factor-$(2-\varepsilon)$ approximation algorithm for the metric k-CENTER problem, for any $\varepsilon > 0$.

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



Can we do better ...?

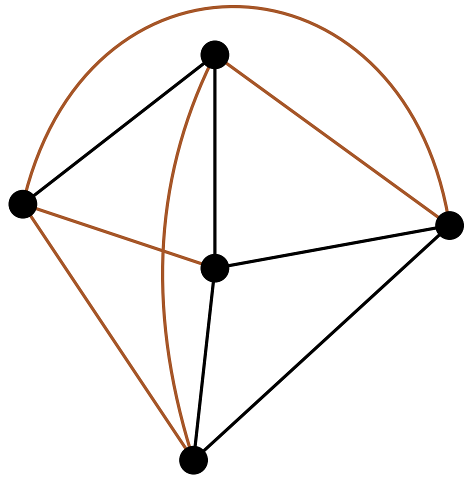


What about a tight example?

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

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

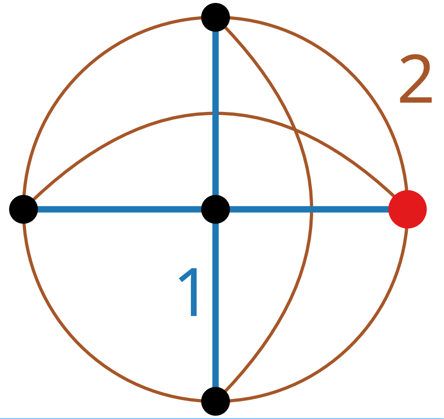
with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$



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

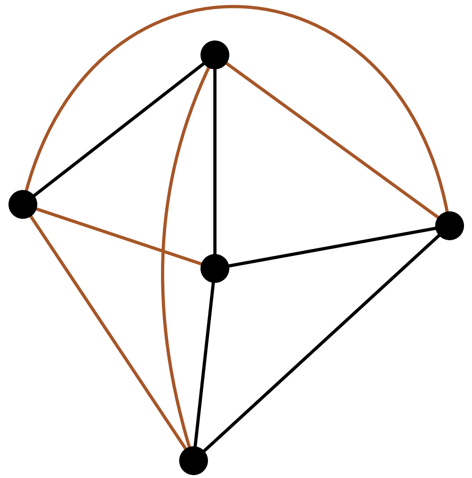
Can we do better . . . ?

What about a tight example?



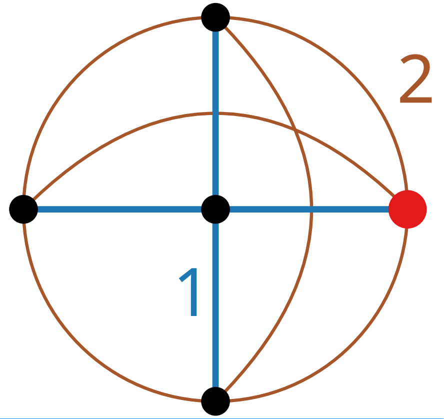
Theorem. Assuming P $\neq$ NP, there is no factor-$(2-\varepsilon)$ approximation algorithm for the metric k-CENTER problem, for any $\varepsilon > 0$.

Proof. Reduce from dominating set to metric k-CENTER.
Given graph $G = (V, E)$ and integer k,
construct complete graph $G' = (V, E \cup E')$
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 $\mathrm{cost}(S) = 1$.

Can we do better . . . ?

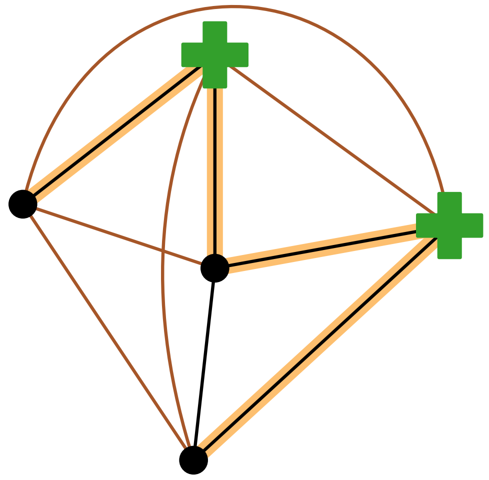


What about a tight example?

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

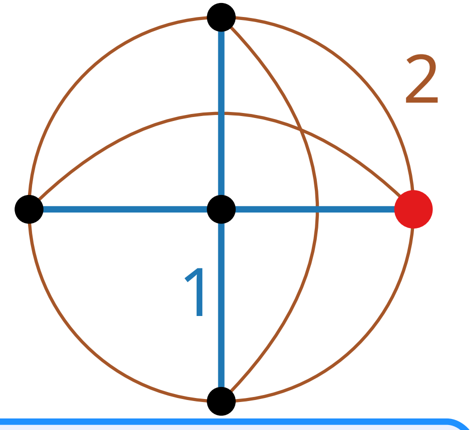
Proof. Reduce from dominating set to metric k-CENTER.
Given graph $G = (V, E)$ and integer k,
construct complete graph $G' = (V, E \cup E')$

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 $\mathrm{cost}(S) = 1$.

Can we do better . . . ?

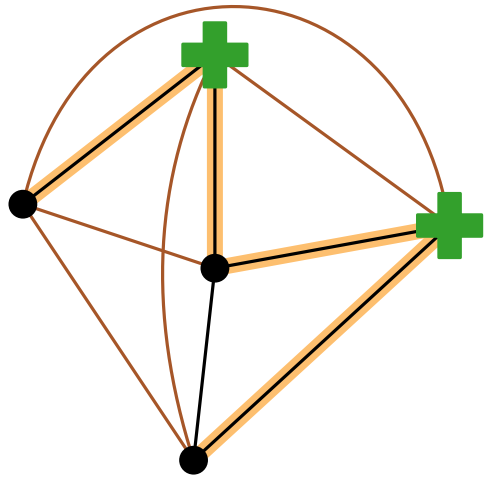


What about a tight example?

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

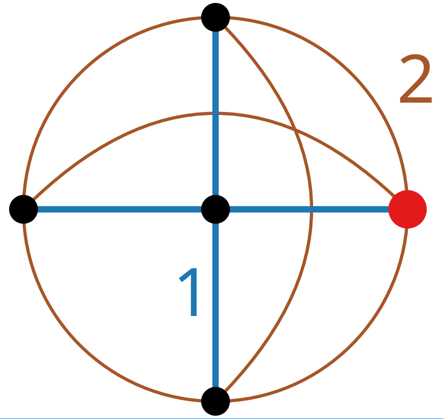
Proof. Reduce from dominating set to metric k-Center.
Given graph $G = (V, E)$ and integer k,
construct complete graph $G' = (V, E \cup E')$

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 . . . ?

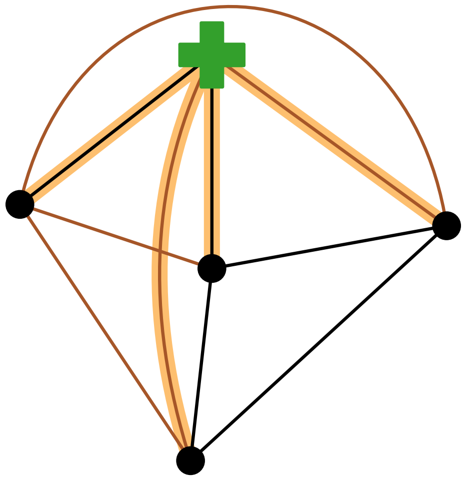


What about a tight example?

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

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

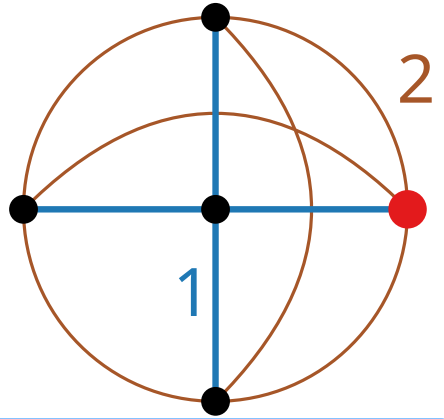
with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$



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 $\mathrm{cost}(S) = 2$.

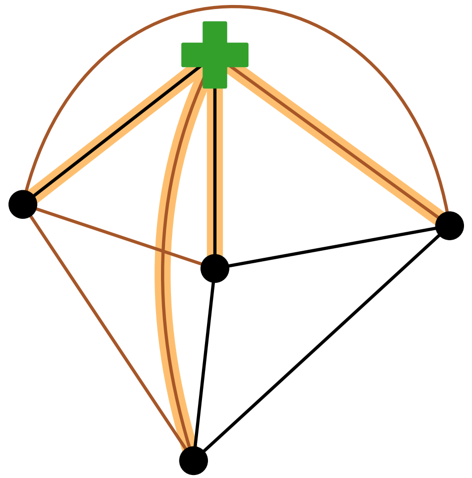
Can we do better . . . ?

What about a tight example?



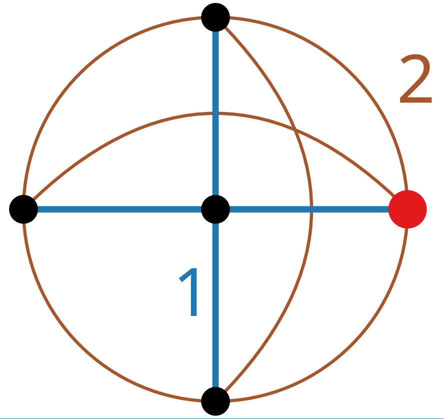
Theorem. Assuming P $\neq$ NP, there is no factor-$(2-\varepsilon)$ approximation algorithm for the metric k-CENTER problem, for any $\varepsilon > 0$.

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



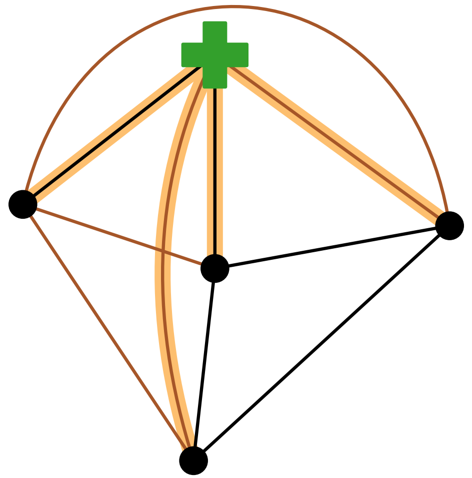
What about a tight example?

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

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

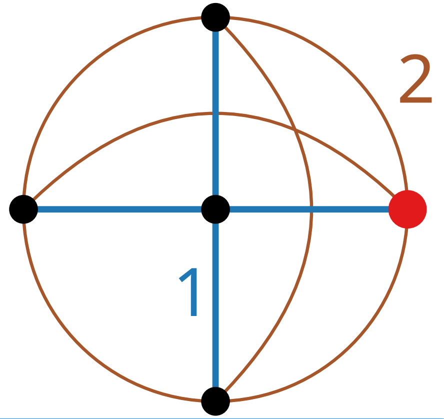
with $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$

$\triangle$-inequality holds



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

Can we do better . . . ?



What about a tight example?

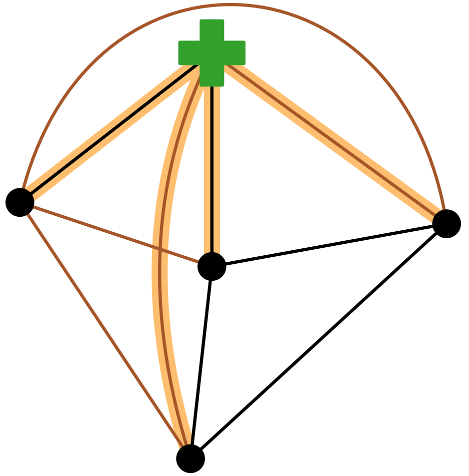
Theorem. Assuming P $\neq$ NP, there is no factor-$(2-\varepsilon)$ approximation algorithm for the metric k-CENTER problem, for any $\varepsilon > 0$.

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

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

$\triangle$-inequality holds

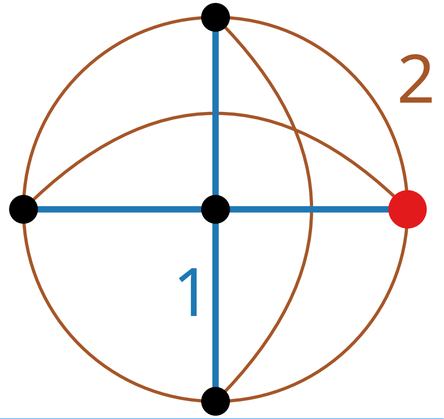
Let S be a metric k-center of G'.
If $\text{dom}(G) \leq k$, then $\text{cost}(S) = 1$.
If $\text{dom}(G) > k$, then $\text{cost}(S) = 2$.



Why does this prove that, e.g., 1.5-approximation is NP-hard?

Can we do better . . . ?

What about a tight example?



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

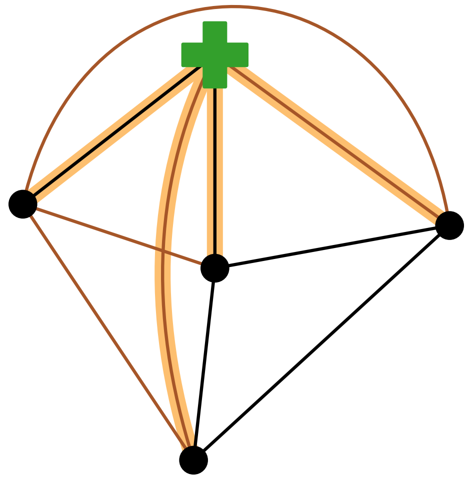
Proof. Reduce from dominating set to metric k-CENTER. Given graph $G = (V, E)$ and integer k, construct complete graph $G' = (V, E \cup E')$

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 $\mathrm{dom}(G) \leq k$, then $\mathrm{cost}(S) = 1$.
If $\mathrm{dom}(G) > k$, then $\mathrm{cost}(S) = 2$.

Why does this prove that, e.g., 1.5-approximation is NP-hard?



1.5-approx. allows to distinguish between 1 and 2, and these are the only possible answers.

3-approximation for METRIC-WEIGHTED-CENTER

Metric-k-Center

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

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

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

METRIC-~~k~~-CENTER

WEIGHTED

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

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

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

Metric-$\cancel{k}$-Center

Weighted

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

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

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

METRIC-~~k~~-CENTER

WEIGHTED

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

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

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

Algorithm for the Weighted Version

Algorithm Metric- -Center

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

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

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

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

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

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

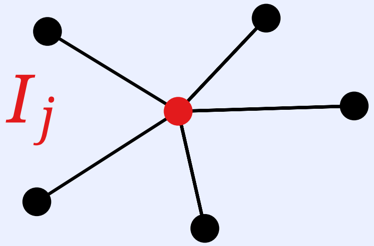
Sort the edges of G by cost : $c(e_1) \leq \ldots \leq c(e_m)$

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

$u \in I_j$



Algorithm for the Weighted Version

Algorithm Metric-Weighted-CENTER

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

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

Construct G_j^2

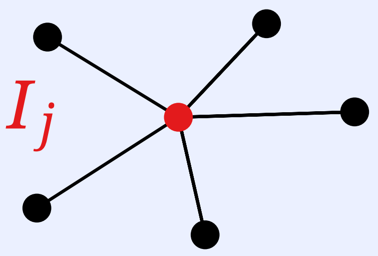
Find a maximal independent set I_j in G_j^2

what about the weights?

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

return I_j

$u \in I_j$



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

Algorithm for the Weighted Version

Algorithm Metric-Weighted-Center

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

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

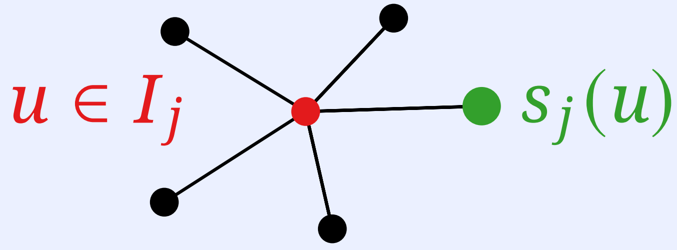
Construct G_j^2

Find a maximal independent set I_j in G_j^2

what about the weights?

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

return I_j



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

Algorithm for the Weighted Version

Algorithm Metric-Weighted-Center

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

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

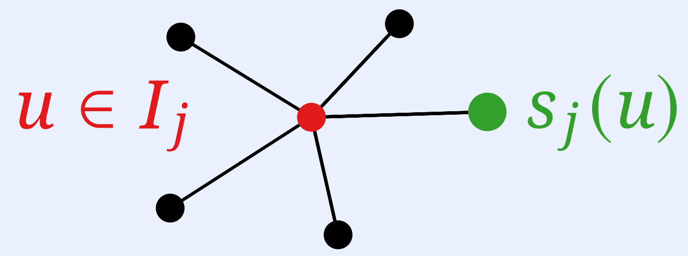
  Construct G_j^2

  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

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

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

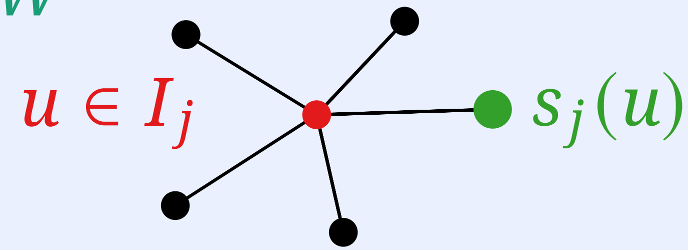
  Construct G_j^2

  Find a maximal independent set I_j in G_j^2

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

  if $|\cancel{I_j}| \leq k$ **then** $w(S_j) \leq W$

    return I_j



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

Algorithm for the Weighted Version

Algorithm Metric-Weighted-CENTER

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

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

Construct G_j^2

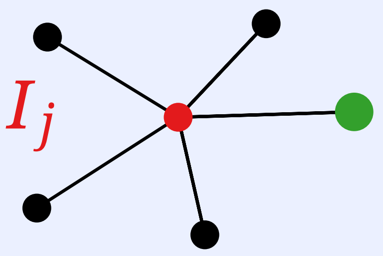
Find a maximal independent set I_j in G_j^2

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

if $\cancel{|I_j| \le k}$ $w(S_j) \le W$ **then**

return $\cancel{I_j}$ S_j

$u \in I_j$



$s_j(u)$

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

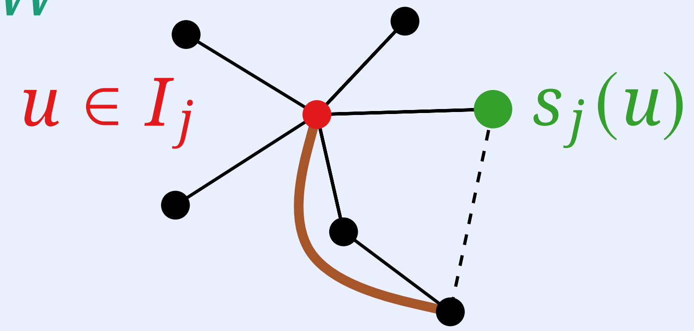
Algorithm for the Weighted Version

Algorithm Metric-Weighted-CENTER

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

for $j = 1, \ldots, m$ **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** $\cancel{|I_j| \leq k}$ $w(S_j) \leq W$ **then**
  - **return** $\cancel{I_j}$ S_j



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

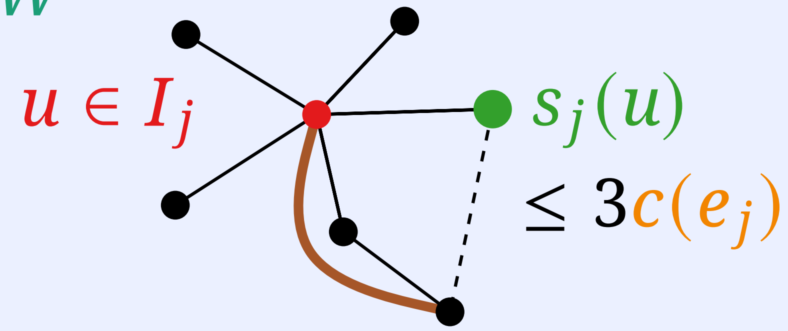
Algorithm for the Weighted Version

Algorithm Metric-Weighted-Center

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

for $j = 1, \ldots, m$ **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** $\cancel{|I_j| \leq k}$ $w(S_j) \leq W$ **then**
  - **return** $\cancel{I_j}$ S_j



because of triangle inequality

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

Algorithm for the Weighted Version

Algorithm Metric-Weighted-CENTER

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

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

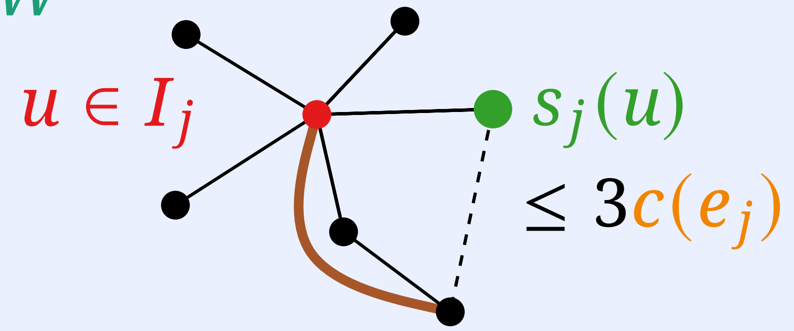
  Construct G_j^2

  Find a maximal independent set I_j in G_j^2

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

  if $\cancel{|I_j| \leq k}$ $w(S_j) \leq W$ **then**

    return $\cancel{I_j}$ S_j



because of triangle inequality

$s_j(u) :=$ 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… ?

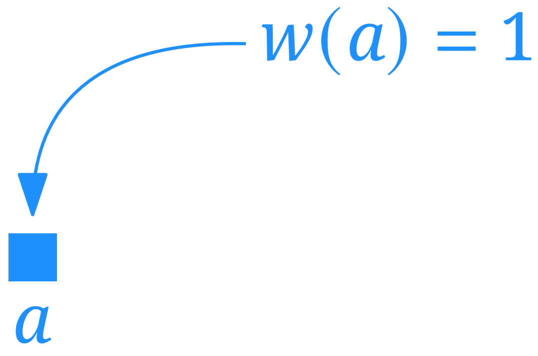
Here, we need to have a budget W,
and edge costs satisfying the triangle inequality.

Consider $W = 3$.

Tight Example... ?

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

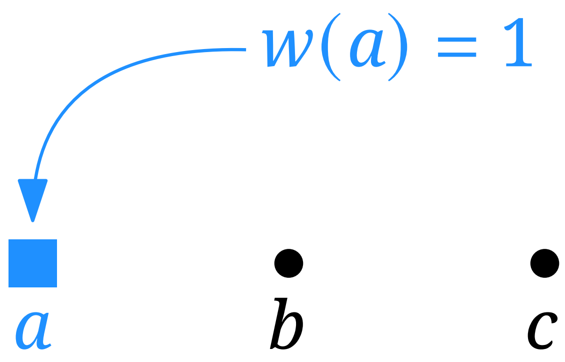
Consider $W = 3$.



Tight Example... ?

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

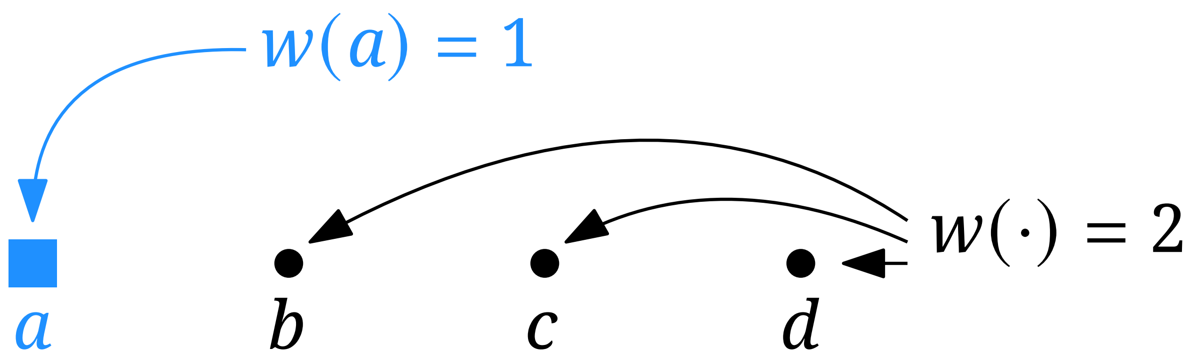
Consider $W = 3$.



Tight Example... ?

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

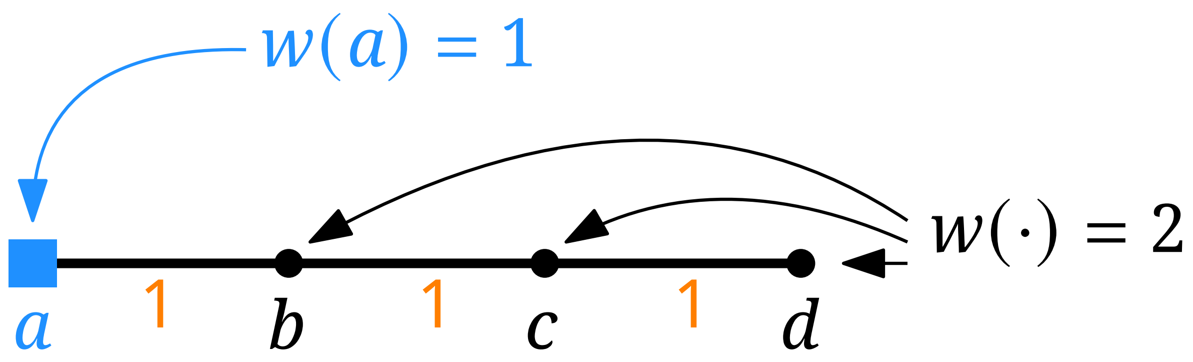
Consider $W = 3$.



Tight Example… ?

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

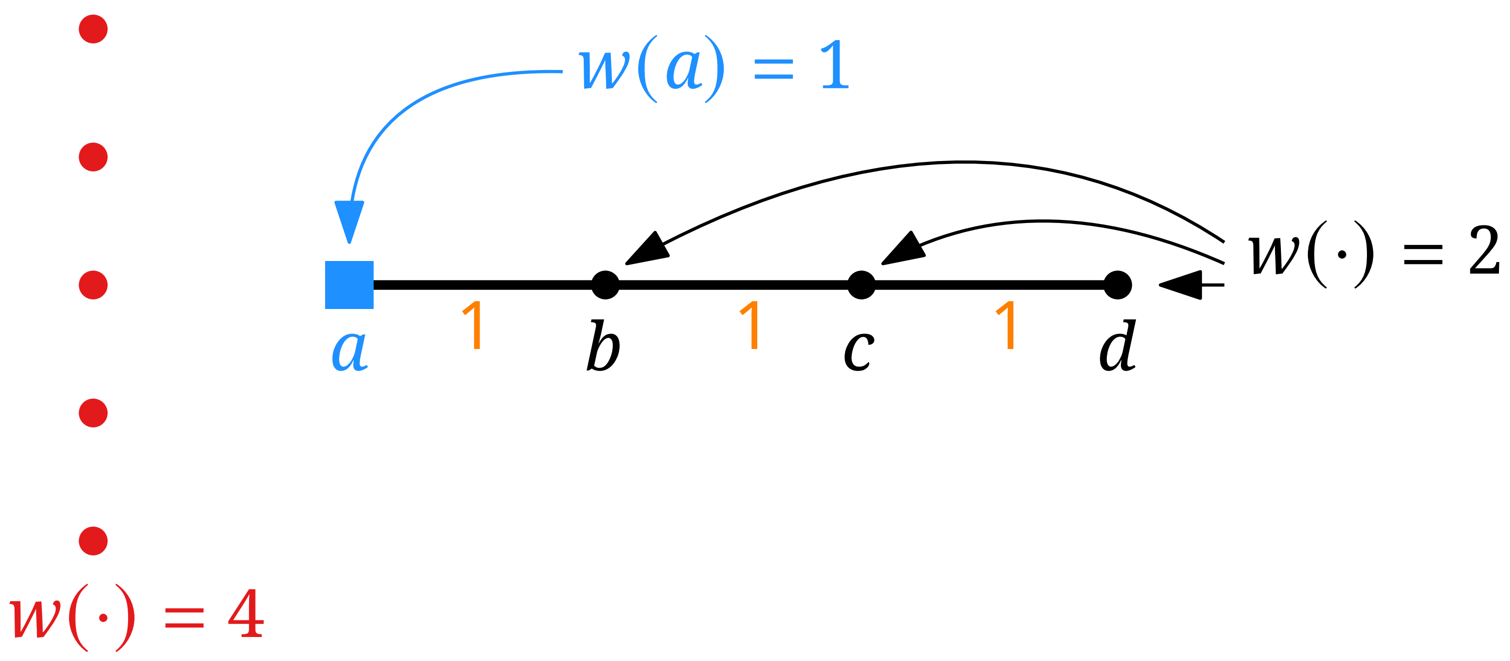
Consider $W = 3$.



Tight Example… ?

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

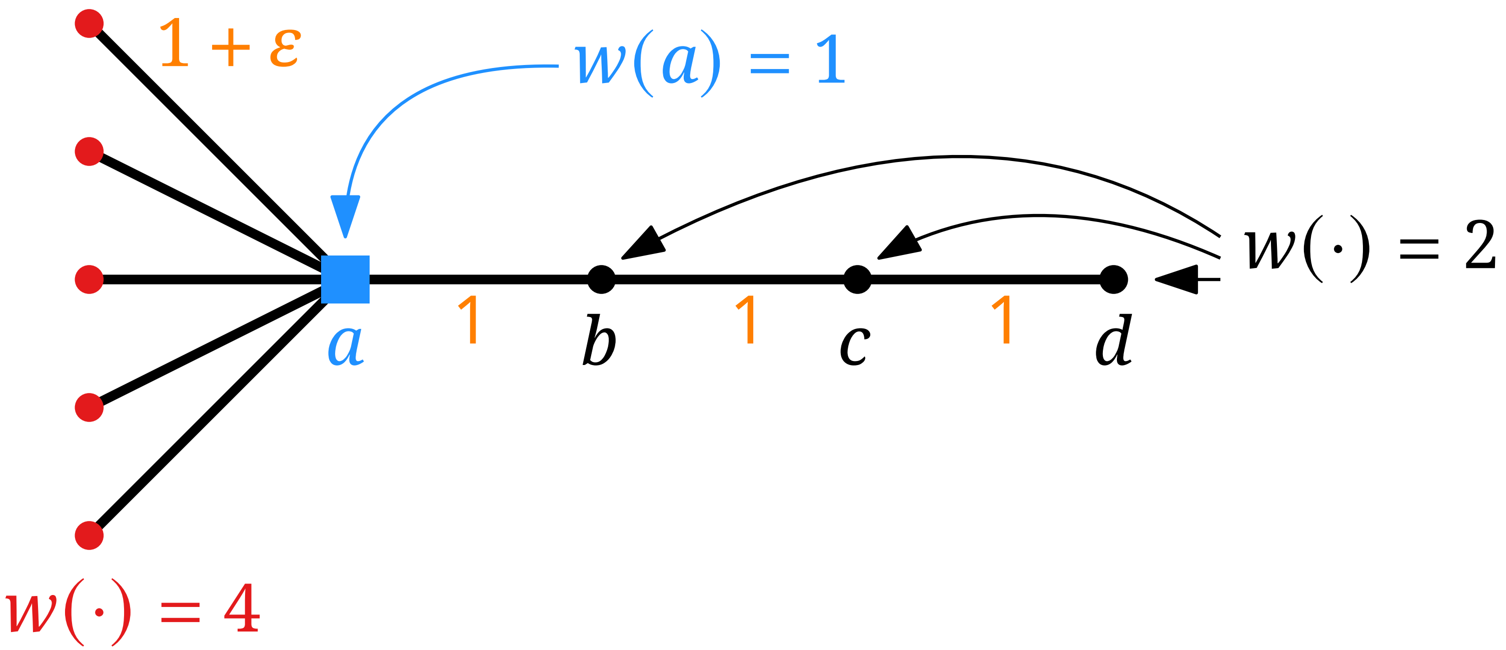
Consider $W = 3$.



Tight Example... ?

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

Consider $W = 3$.

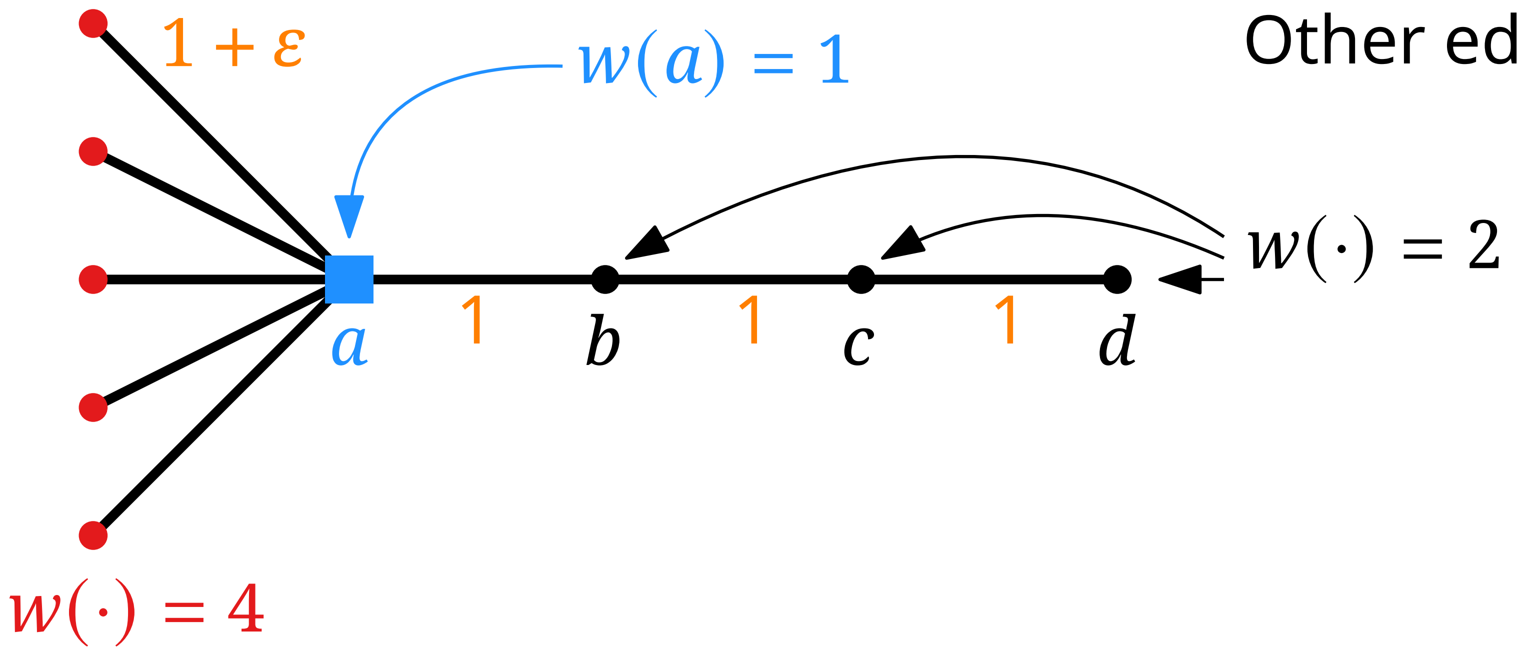


Tight Example... ?

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

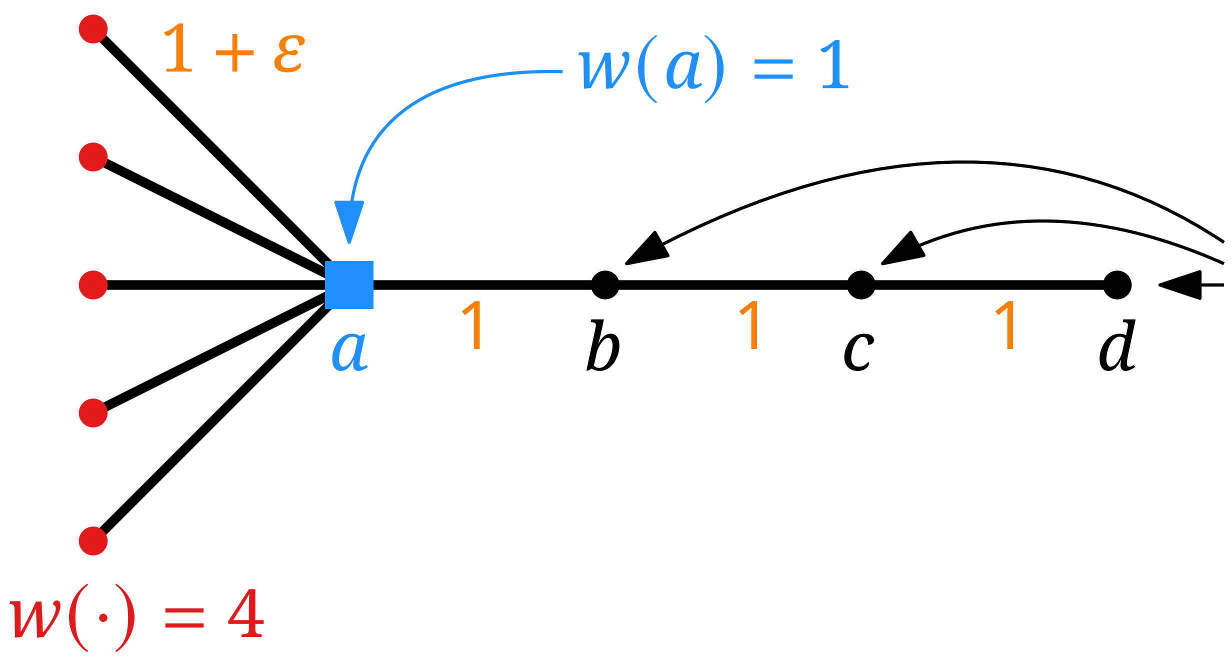
Consider $W = 3$.

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?

$\longrightarrow$ metric completion!

$w(\cdot) = 2$

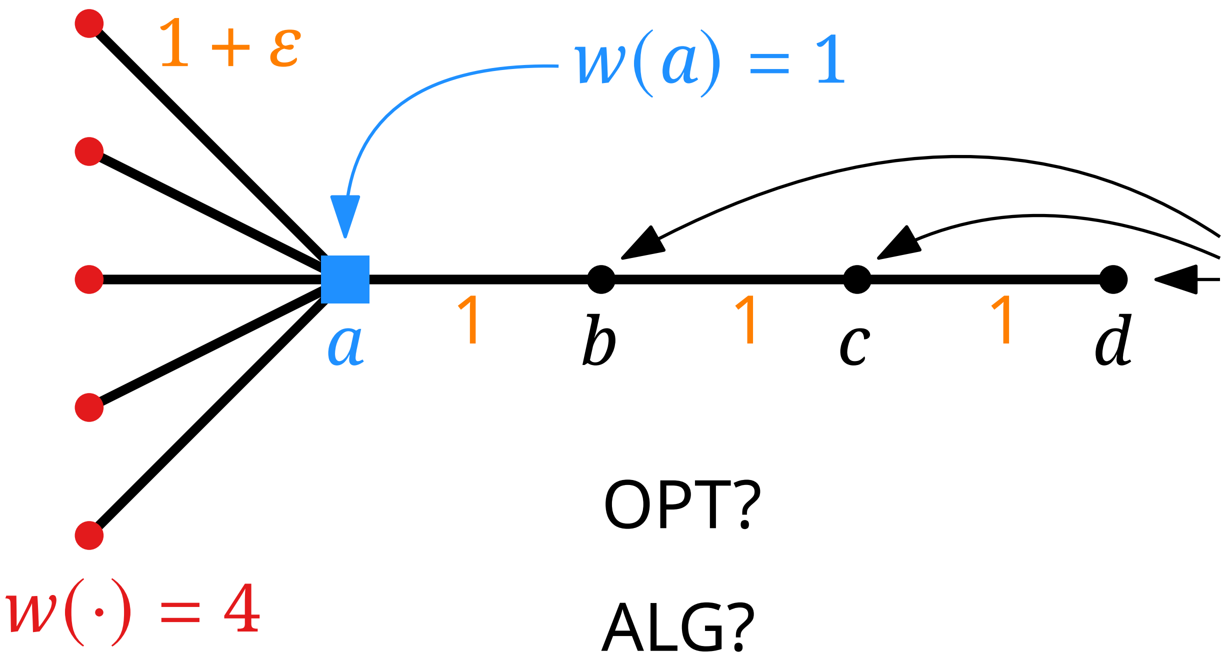
Tight Example... ?

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

Consider $W = 3$.

Other edge costs?

$\longrightarrow$ metric completion!



$w(\cdot) = 2$

OPT?

ALG?

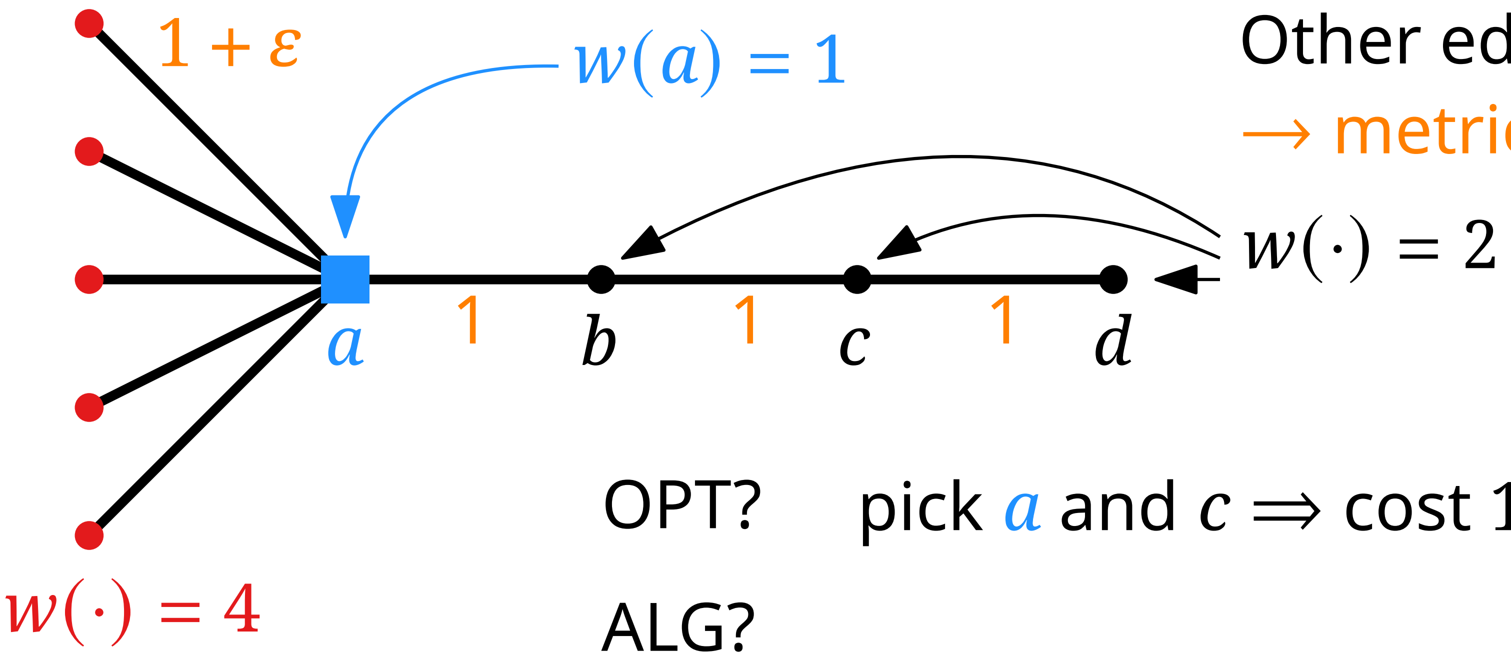
Tight Example... ?

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

Consider $W = 3$.

Other edge costs?

$\longrightarrow$ metric completion!

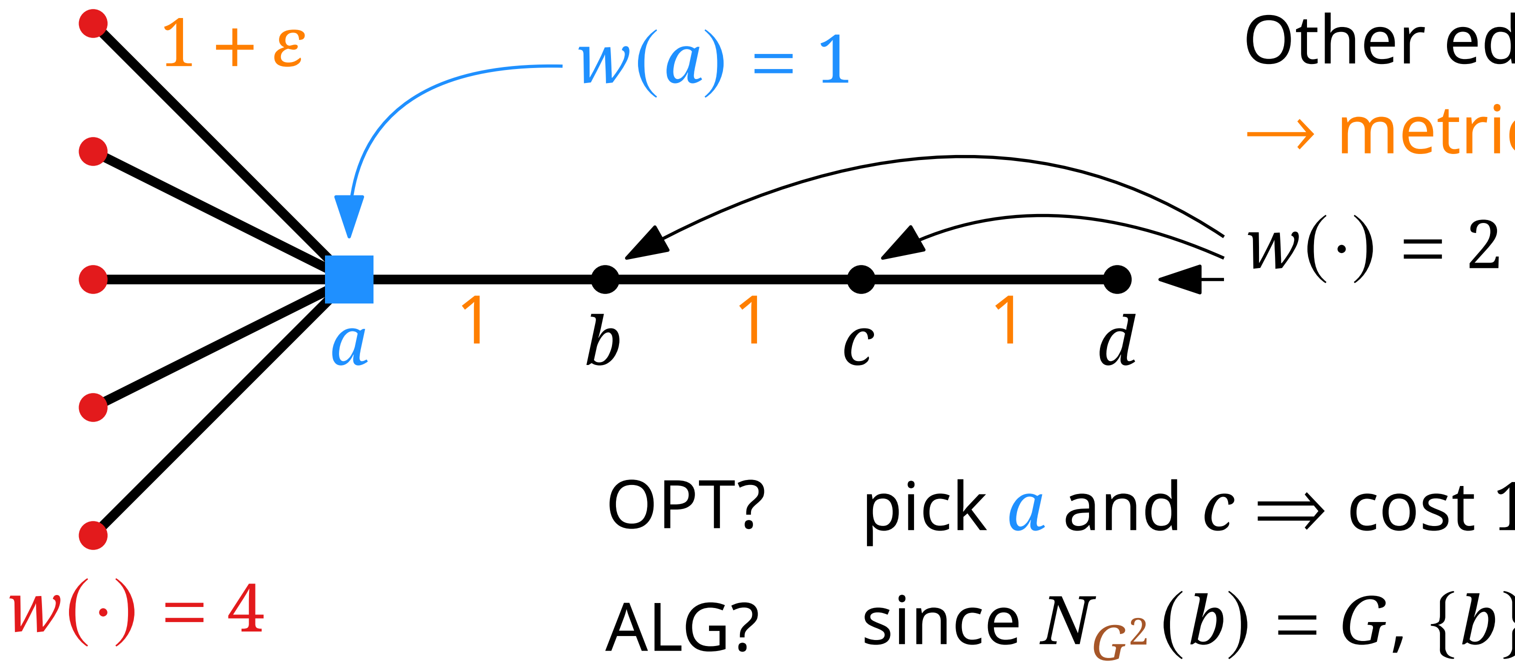


OPT? pick a and $c \Rightarrow$ cost $1 + \varepsilon$.

ALG?

Tight Example... ?

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



Consider $W = 3$.

Other edge costs?

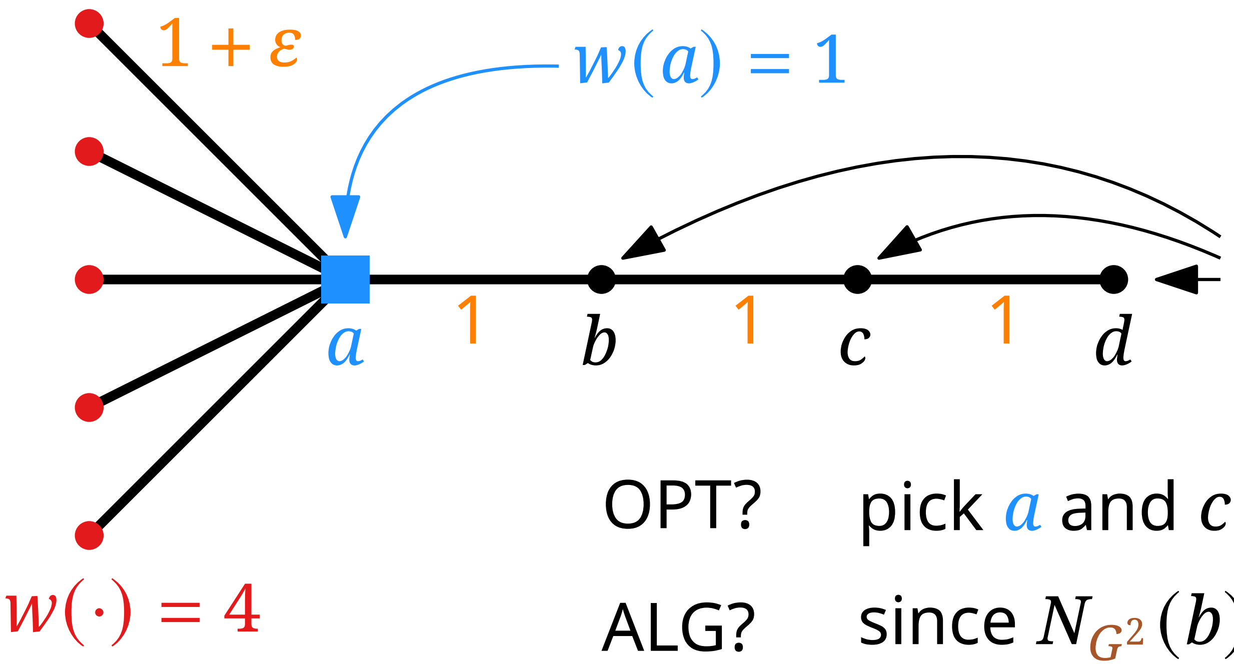
$\longrightarrow$ metric completion!

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?

$\longrightarrow$ metric completion!

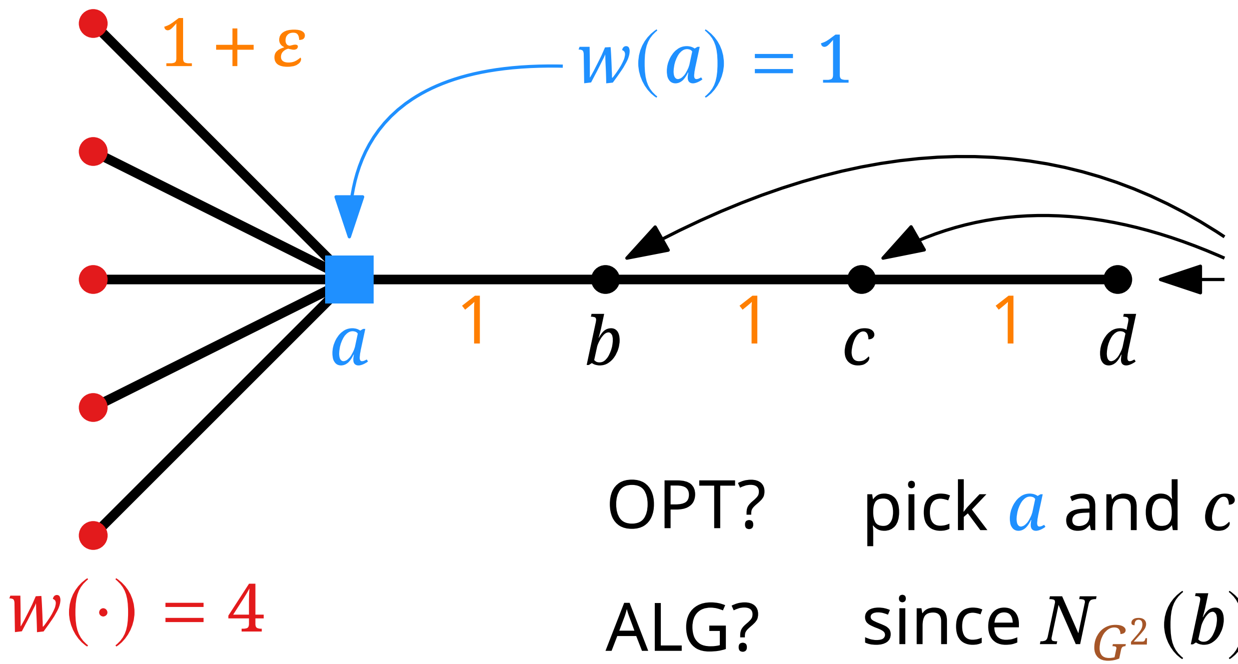
OPT? pick a and $c \Rightarrow$ cost $1 + \varepsilon$.

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?

$\longrightarrow$ metric completion!

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

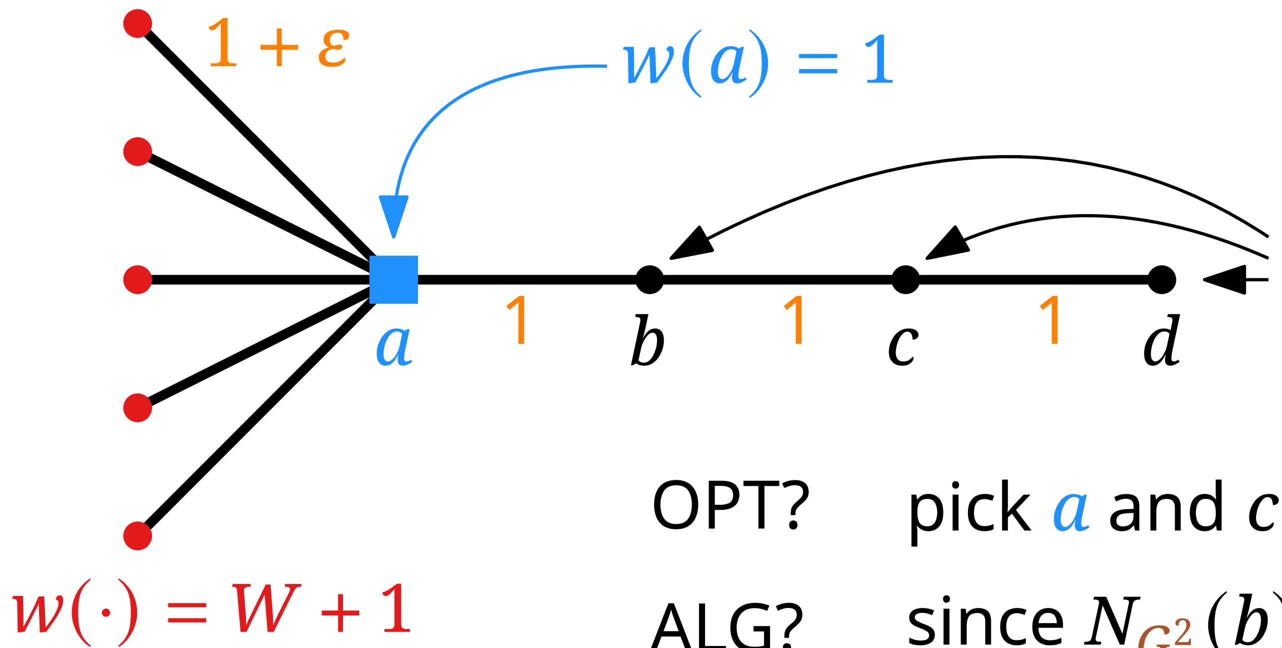
ALG? since $N_{G^2}(b) = G$, $\{b\}$ is a maximal independent set in G^2

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?

$\longrightarrow$ 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?

Summary

parametric pruning: decision problem $\longrightarrow$ pruned instance $I(t)$
$\longrightarrow$ lower bound t^* opt OPT & find solution in $I(\alpha t^*)$

Metric k-Center: 2-approximation using pruned instances G_j, and finding a maximal independent set in G_j^2 (square of G_j)

Metric Weighted-Center: 3-approximation by same algorithm but taking for each v in independent set, lowest weight vertex in neighborhood

Note Alternative 2-approximation algorithm (Gonzalez 1985):
Pick an arbitrary vertex as first center, then greedily add the vertex with largest distance (cost) to centers selected, until k centers have been selected.