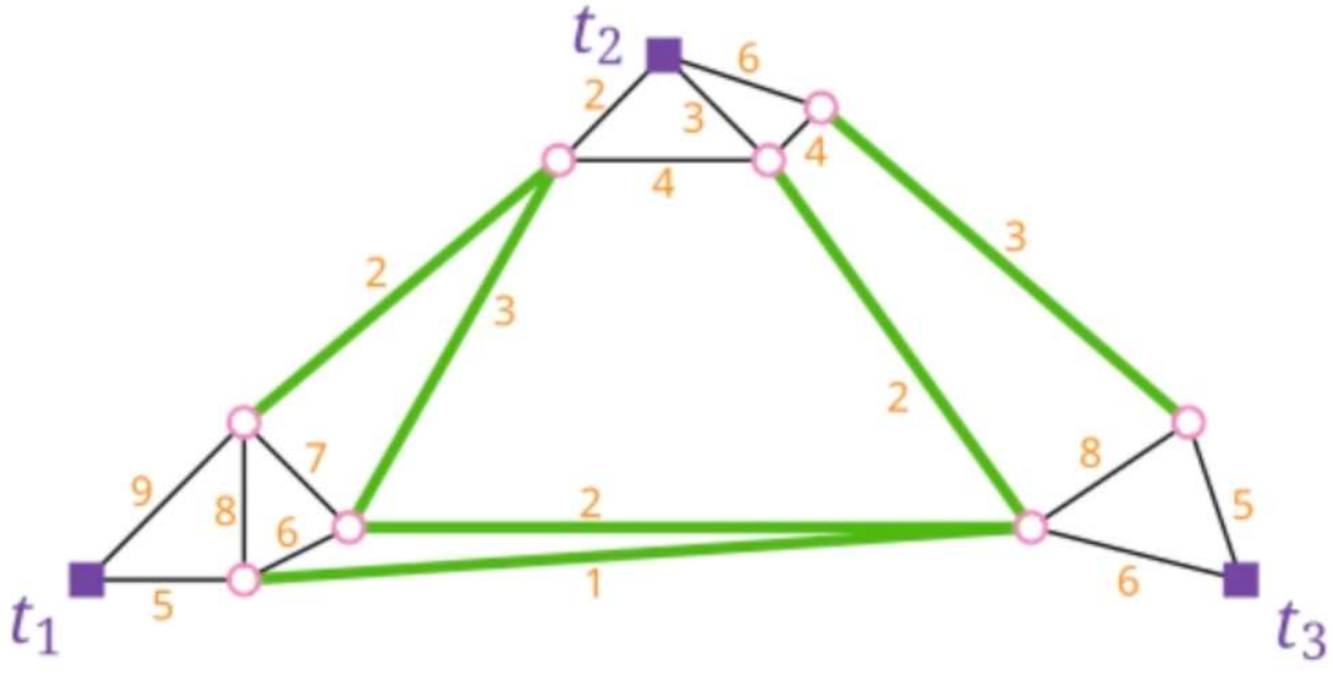
MULTIWAYCUT

Given: A connected graph G with edge costs $c\colon E(G) \to \mathbb{Q}^+$ and a set $T = \{t_1, \ldots, t_k\} \subseteq V(G)$ of **terminals**.

A multiway cut of T is a subset E' of edges such that no two terminals in the graph $(V(G), E(G) - E')$ are connected.

Find: A minimum-cost multiway cut of T.

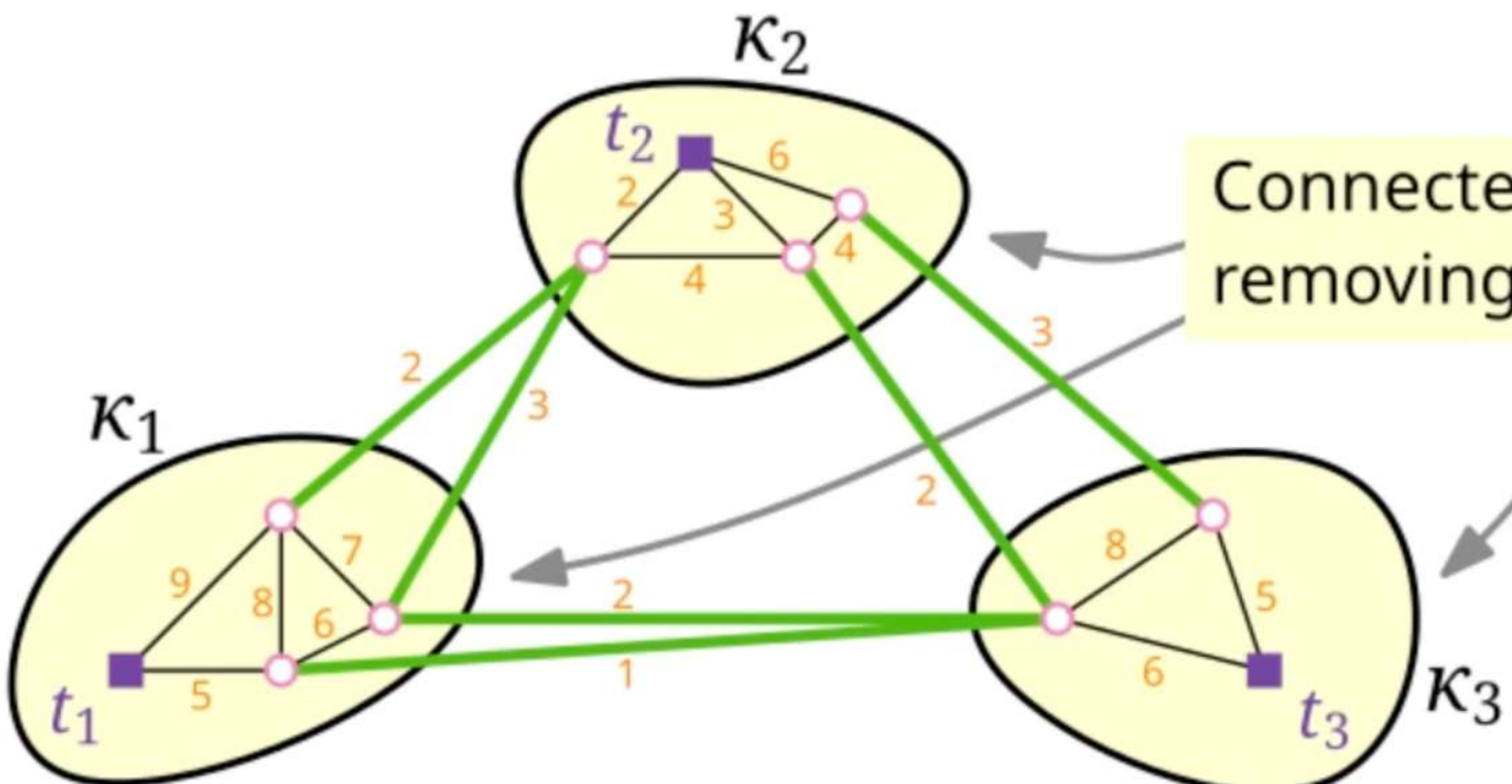


MULTIWAYCUT

Given: A connected graph G with edge costs $c\colon E(G) \to \mathbb{Q}^+$ and a set $T = \{t_1, \ldots, t_k\} \subseteq V(G)$ of **terminals**.

A multiway cut of T is a subset E' of edges such that no two terminals in the graph $(V(G), E(G) - E')$ are connected.

Find: A minimum-cost multiway cut of T.



NP complete?

Our problem is an optimisation problem. Can we reduce it to a decision problem?

Yes

The decision problem for the multiway cut is:
Can we partition the given graph vertices into k subsets such that each subset has exactly one terminal and the sum of weight of the cuts is = Min Weight(given).

An instance of this problem is if we also give the relaxation that cost of each terminal cut is equal. Now the problem instance becomes

Can we partition the given graph vertices into k subsets(here k is the number of terminals in the graph) such that each subset has exactly one terminal and the cost of each cut-set is M.

Now this problem is similar to bin packing where the number of bins is fixed(=k here).
Given the graph we need to put all the vertices into k bins such that the cost of each cut is equal to M.

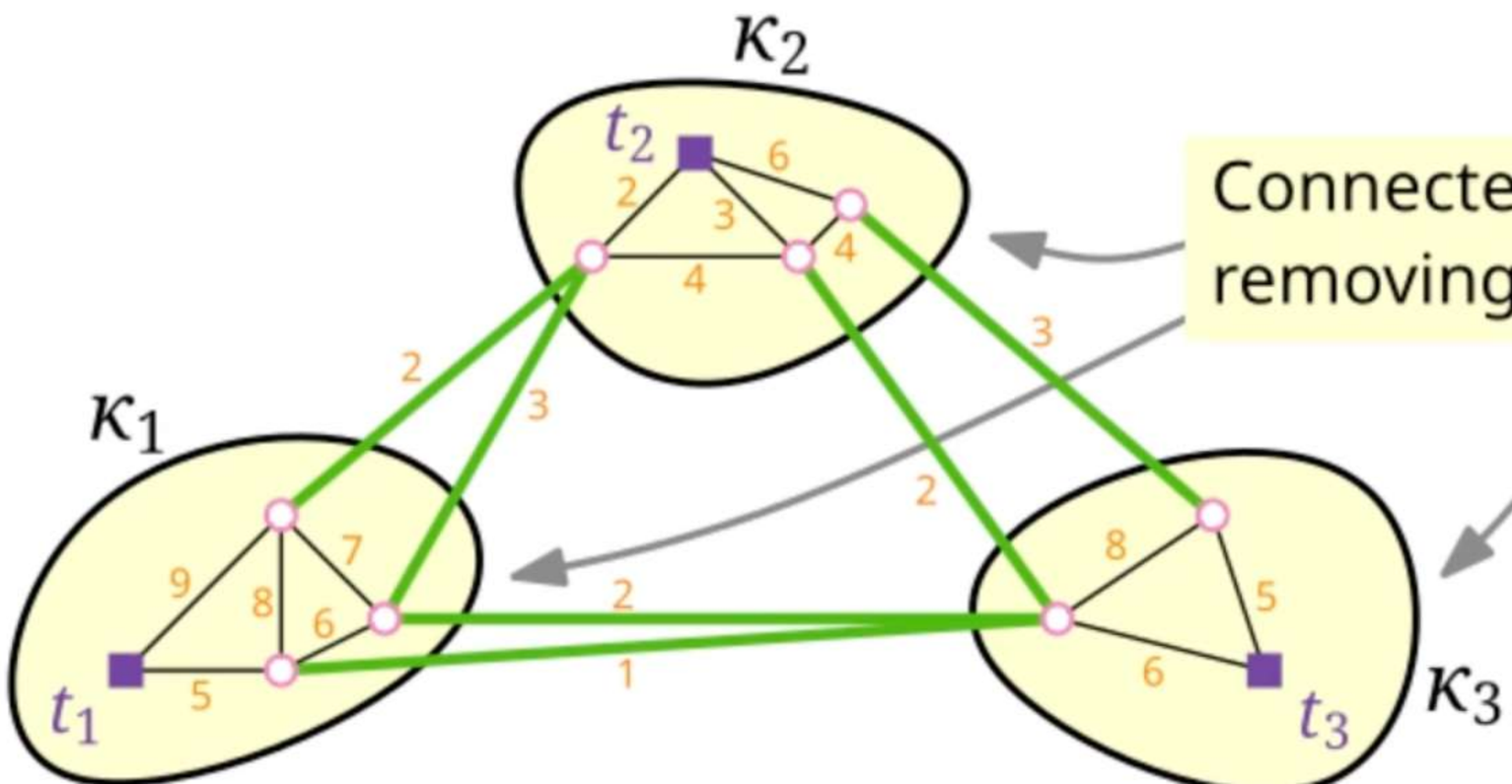
As Bin packing is a well known NP complete problem and our problem instance is reduced to it in polynomial time we can say that this instance of multiway cut is NP complete.

MULTIWAYCUT

Given: A connected graph G with edge costs $c\colon E(G) \to \mathbb{Q}^+$ and a set $T = \{t_1, \ldots, t_k\} \subseteq V(G)$ of **terminals**.

A multiway cut of T is a subset E' of edges such that no two terminals in the graph $(V(G), E(G) - E')$ are connected.

Find: A minimum-cost multiway cut of T.



Special cases:

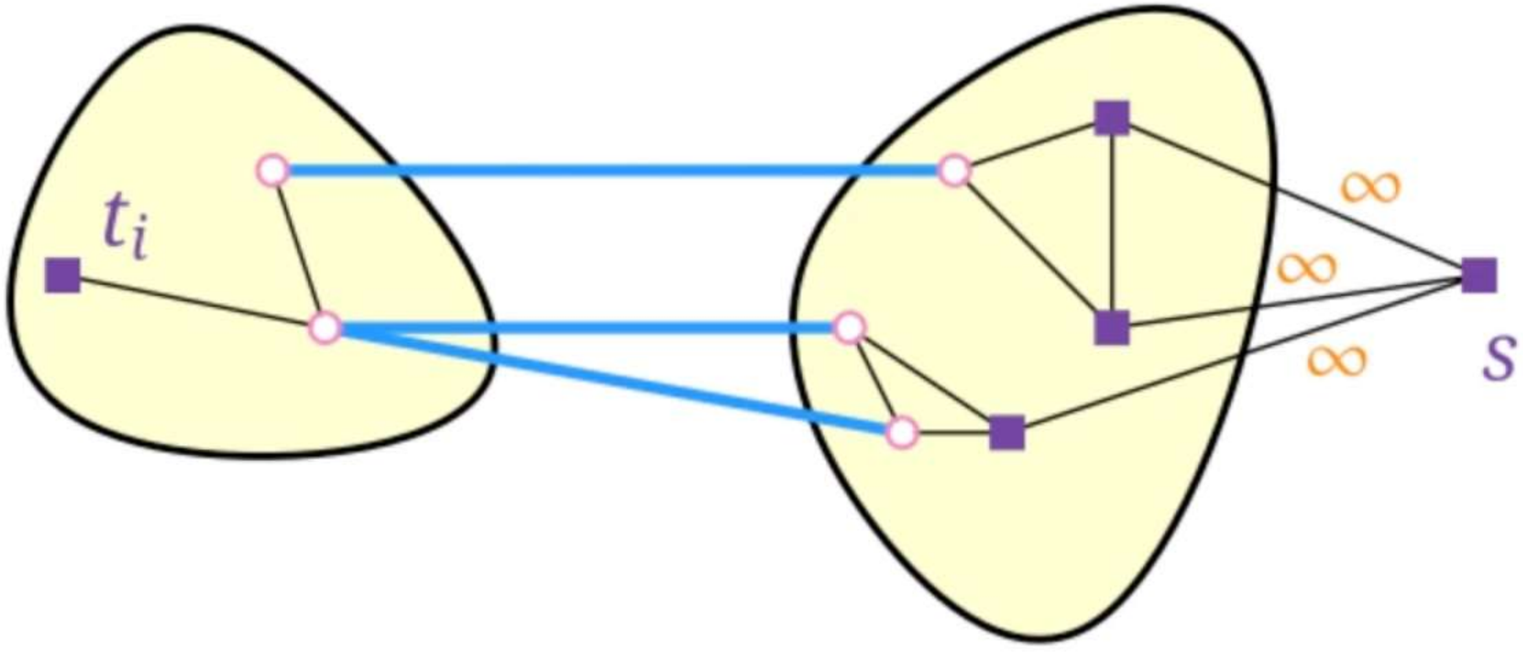
$k = 2$: Min s–t cut in P

$k \geq 3$: NP-hard

Isolating Cuts

An **isolating cut** for a terminal t_i is a set of edges
that disconnects t_i from all other terminals.

A minimum-cost isolating cut for t_i can be computed efficiently:

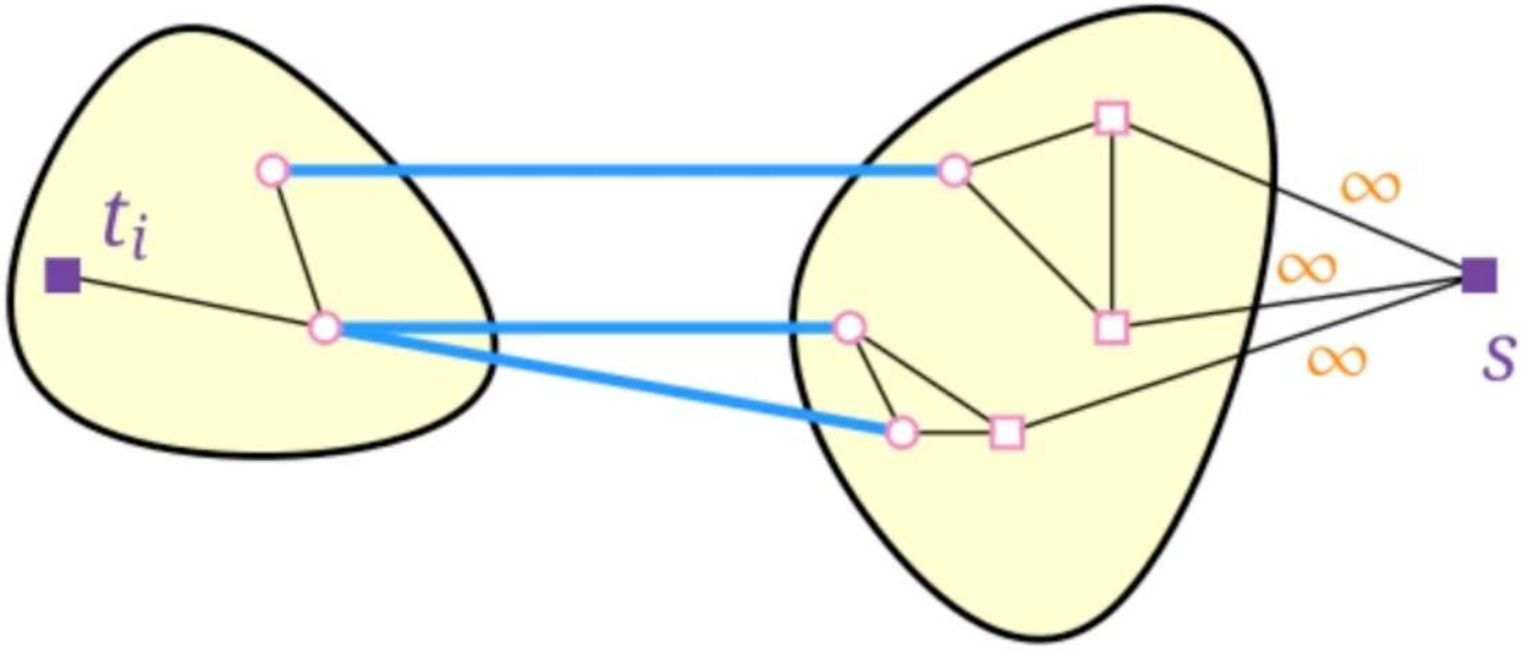


Add dummy terminal s

Isolating Cuts

An **isolating cut** for a terminal t_i is a set of edges
that disconnects t_i from all other terminals.

A minimum-cost isolating cut for t_i can be computed efficiently:



Add dummy terminal s and find a minimum-cost s–t_i cut.

Algorithm for MULTIWAYCUT

Algorithm MultiwayCut

For $i = 1, \ldots, k$:

- Compute a minimum-cost isolating cut C_i for t_i.
- Return the union C of the $k - 1$ cheapest such isolating cuts.

In other words:
Ignore the most expensive one of the isolating cuts $C_1, \ldots, C_k$.

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

Algorithm MultiwayCut

For $i = 1, \ldots, k$:

- Compute a minimum-cost isolating cut C_i for t_i.
- Return the union C of the $k - 1$ cheapest such isolating cuts.

In other words:
Ignore the most expensive one of the isolating cuts $C_1, \ldots, C_k$.

$$\Rightarrow c(C) \leq \left(1 - \frac{1}{k}\right) \sum_{i=1}^{k} c(C_i) \qquad \text{because:}$$

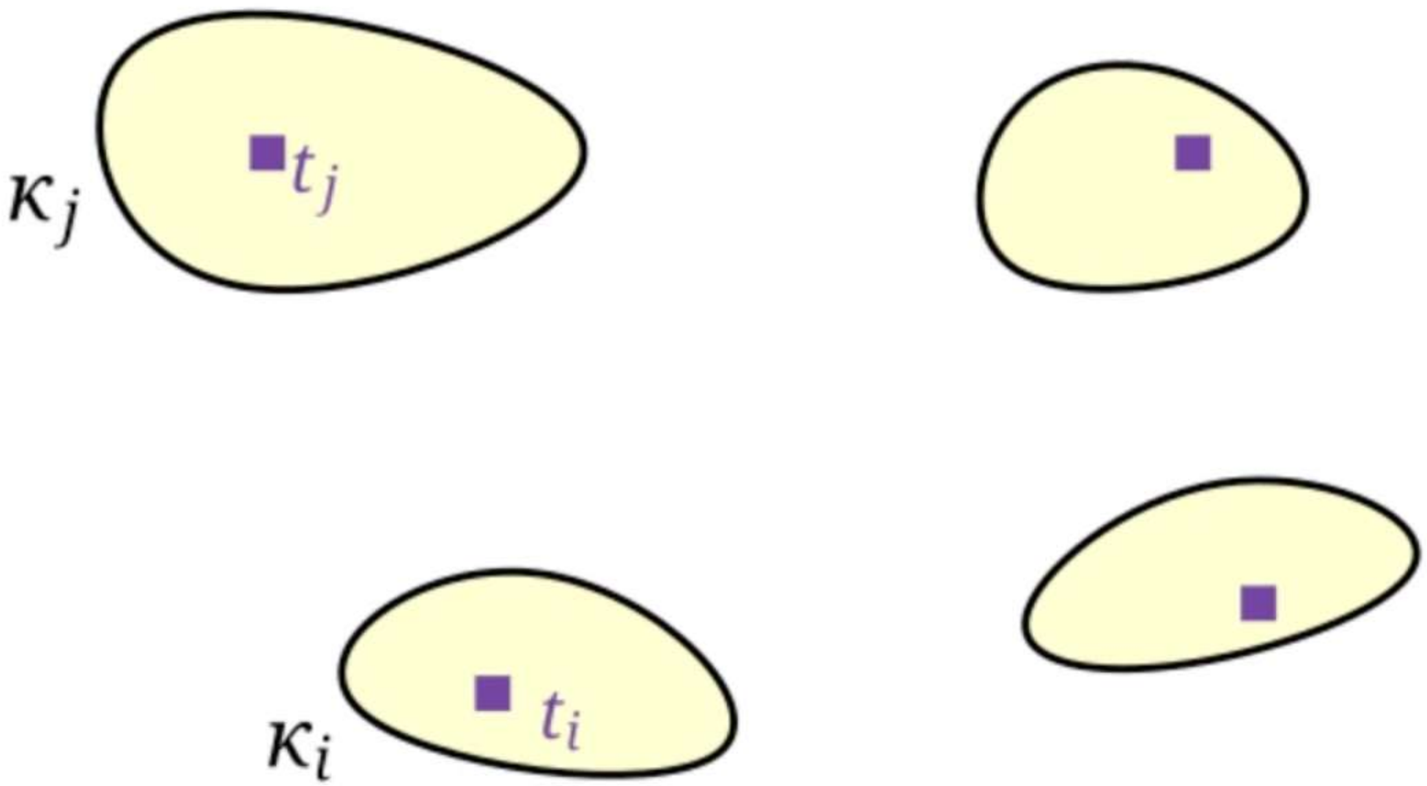
for the most expensive cut of $C_1, \ldots, C_k$, say C_1, we have

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

Approximation Factor

Theorem. This algorithm is a factor-$(2 - 2/k)$ approximation algorithm for MULTIWAYCUT.

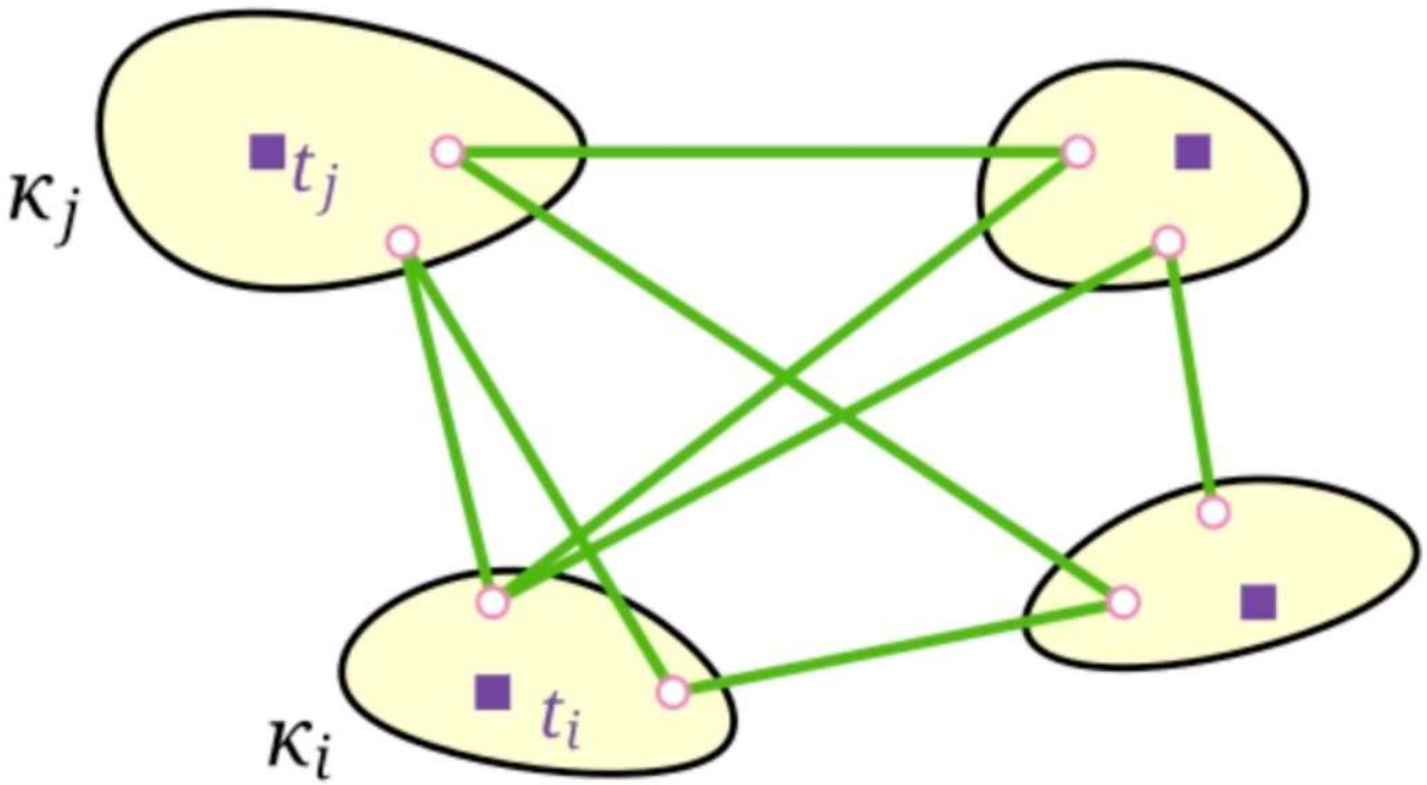
Proof. Consider an opt. multiway cut $\mathcal{A}$:



Approximation Factor

Theorem. This algorithm is a factor-$(2 - 2/k)$ approximation algorithm for MULTIWAYCUT.

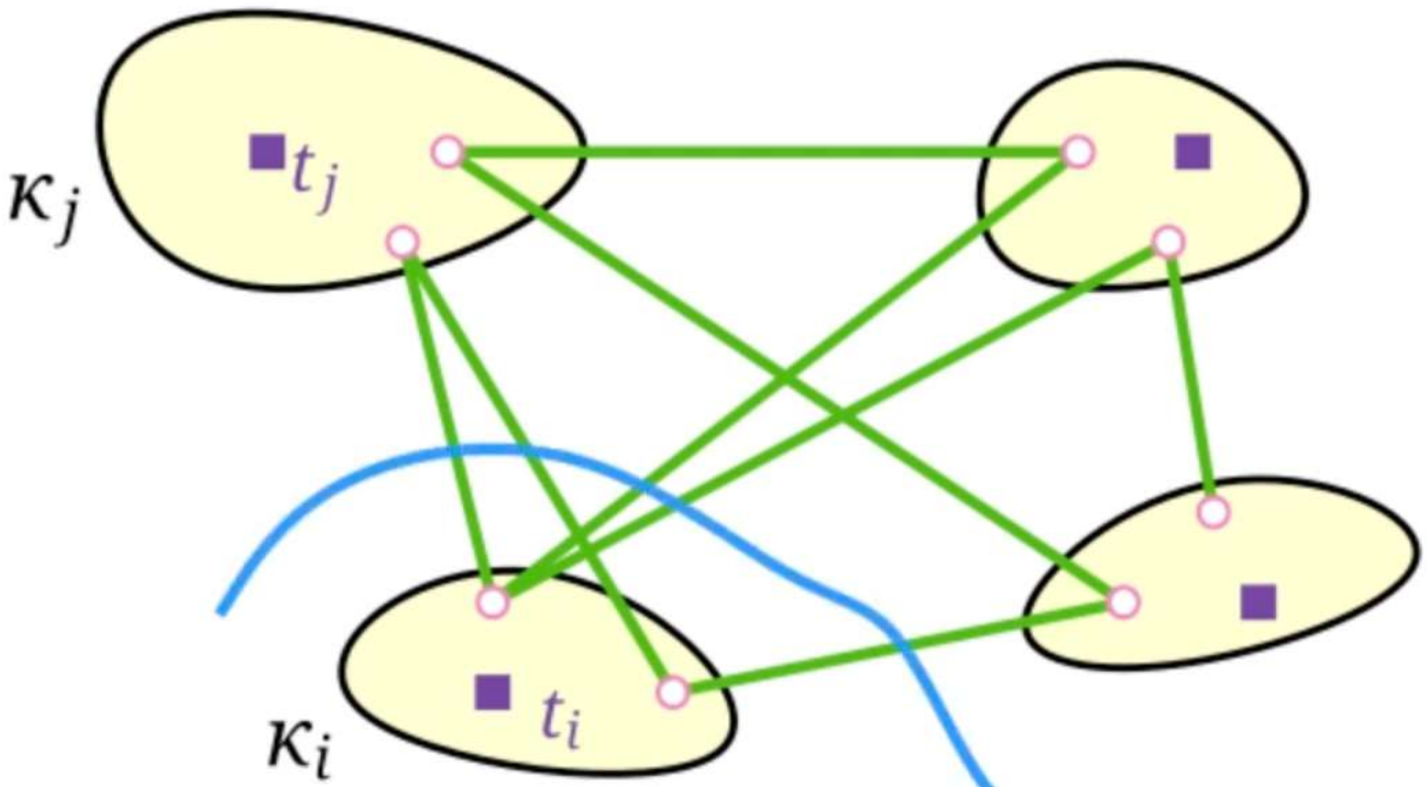
Proof. Consider an opt. multiway cut $\mathcal{A}$:



Approximation Factor

Theorem. This algorithm is a factor-$(2 - 2/k)$ approximation algorithm for MULTIWAYCUT.

Proof. Consider an opt. multiway cut $\mathcal{A}$:

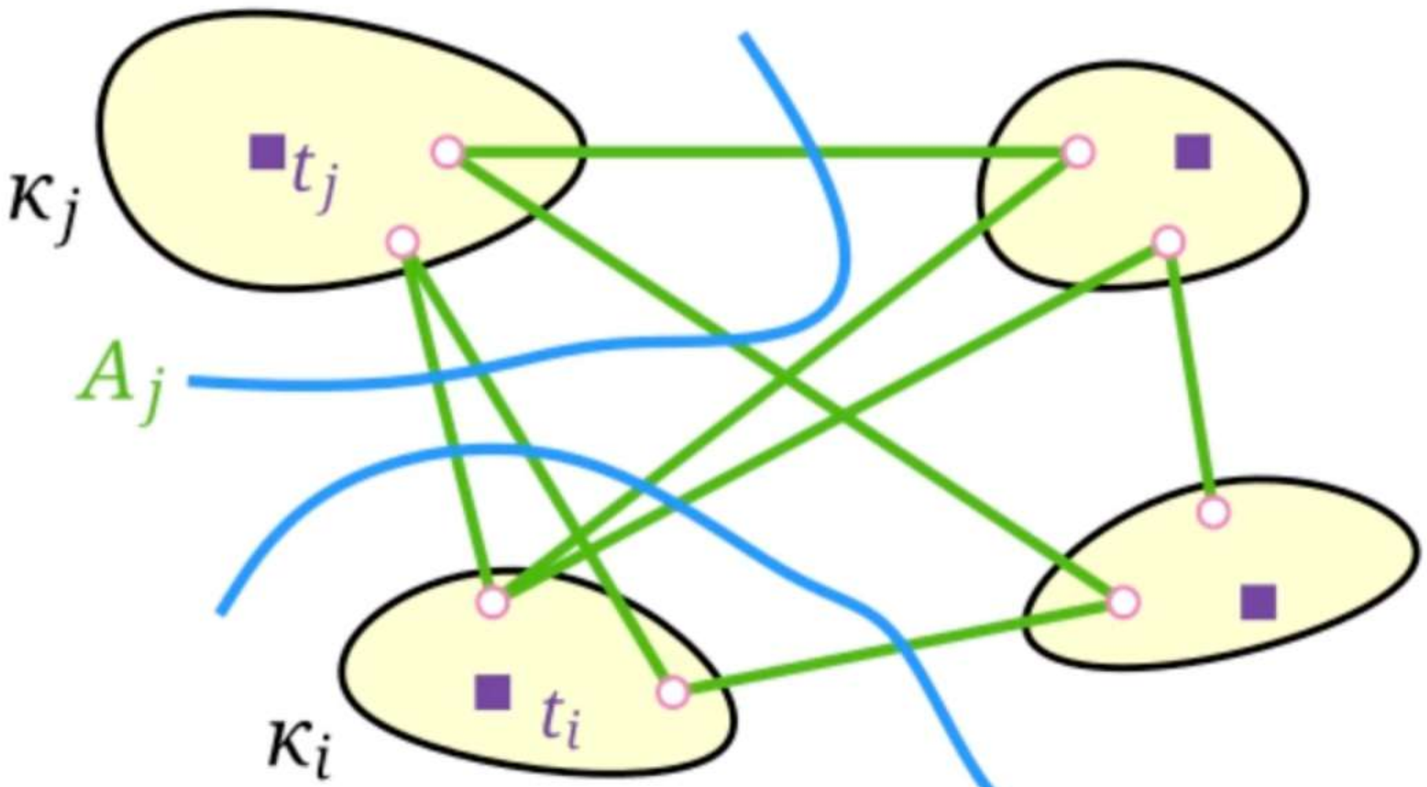


$A_i = \{uv \in \mathcal{A} : u \in \kappa_i, v \notin \kappa_i\}$

Approximation Factor

Theorem. This algorithm is a factor-$(2 - 2/k)$ approximation algorithm for MULTIWAYCUT.

Proof. Consider an opt. multiway cut $\mathcal{A}$:

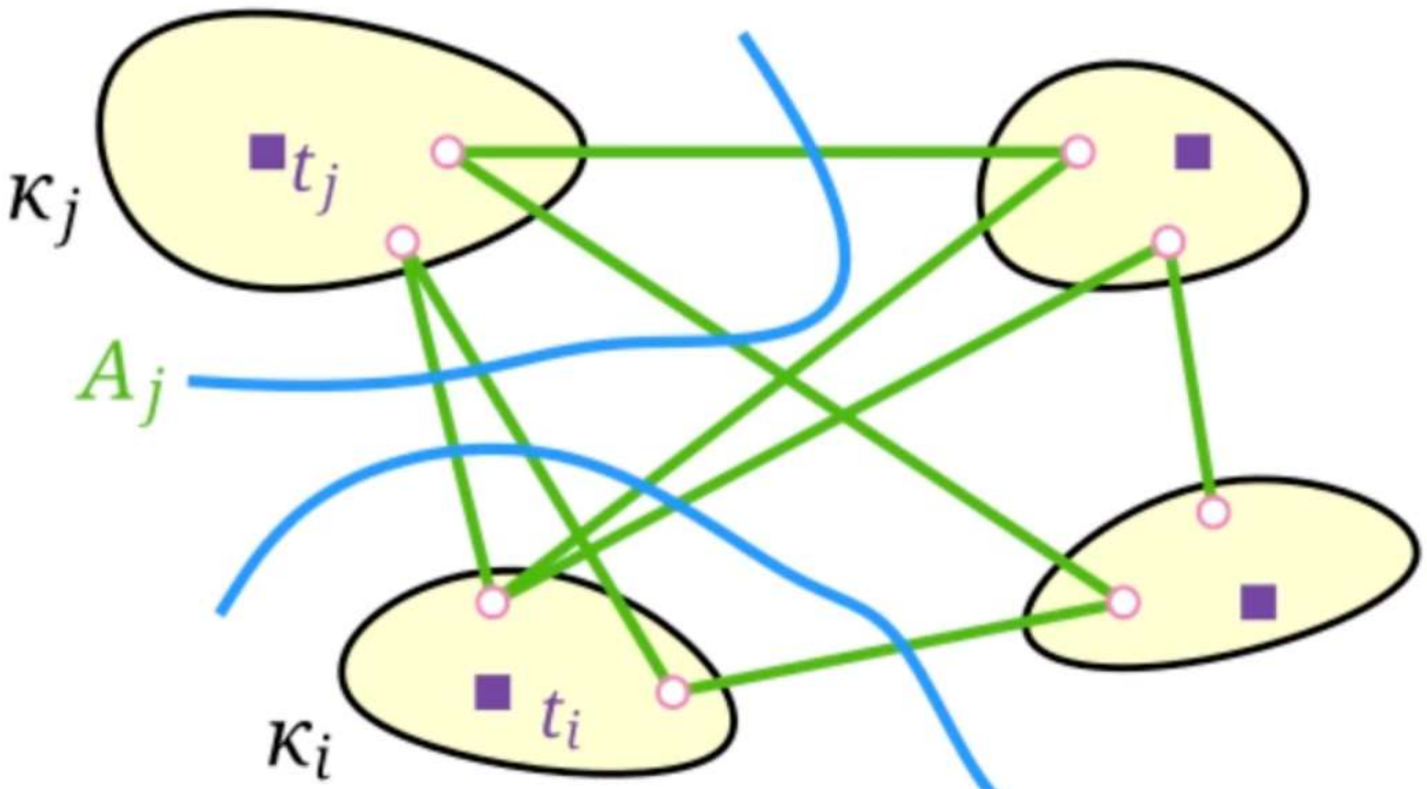


$$A_i = \{uv \in \mathcal{A} : u \in \kappa_i, v \notin \kappa_i\}$$

Approximation Factor

Theorem. This algorithm is a factor-$(2 - 2/k)$ approximation algorithm for MULTIWAYCUT.

Proof. Consider an opt. multiway cut $\mathcal{A}$:



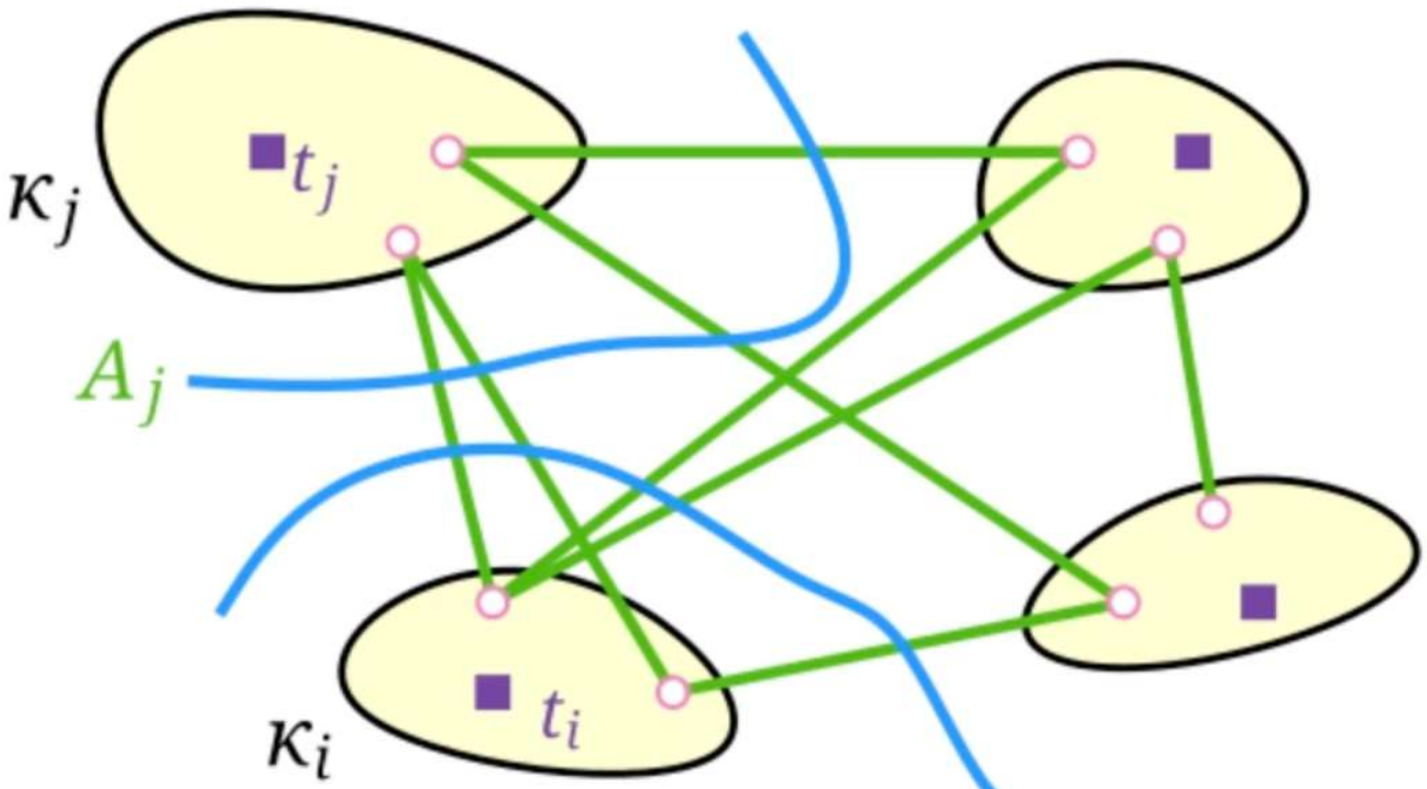
$$A_i = \{uv \in \mathcal{A} : u \in \kappa_i, v \notin \kappa_i\}$$

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

Approximation Factor

Theorem. This algorithm is a factor-$(2 - 2/k)$ approximation algorithm for MULTIWAYCUT.

Proof. Consider an opt. multiway cut $\mathcal{A}$:



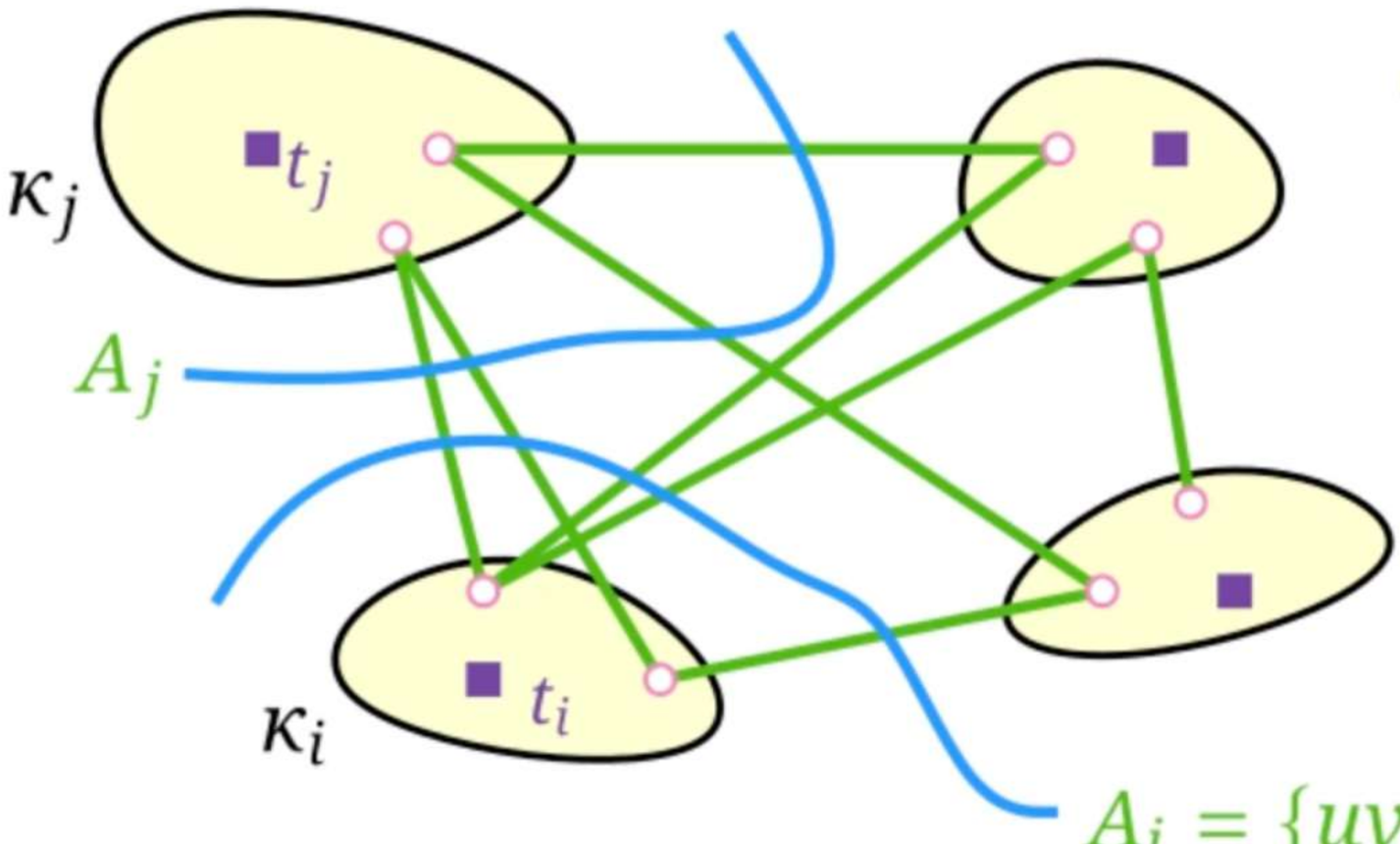
$A_i = \{uv \in \mathcal{A} : u \in \kappa_i, v \notin \kappa_i\}$

Observation. $\mathcal{A} = \bigcup_{i=1}^{k} A_i$ and $\sum_{i=1}^{k} c(A_i) = 2 \cdot c(\mathcal{A}) = 2 \cdot \mathrm{OPT}$.

Approximation Factor

Theorem. This algorithm is a factor-$(2 - 2/k)$ approximation algorithm for MULTIWAYCUT.

Proof. Consider an opt. multiway cut $\mathcal{A}$: Consider the alg.'s solution C:



$$c(C) \le \left(1 - \tfrac{1}{k}\right) \sum_{i=1}^{k} c(C_i)$$

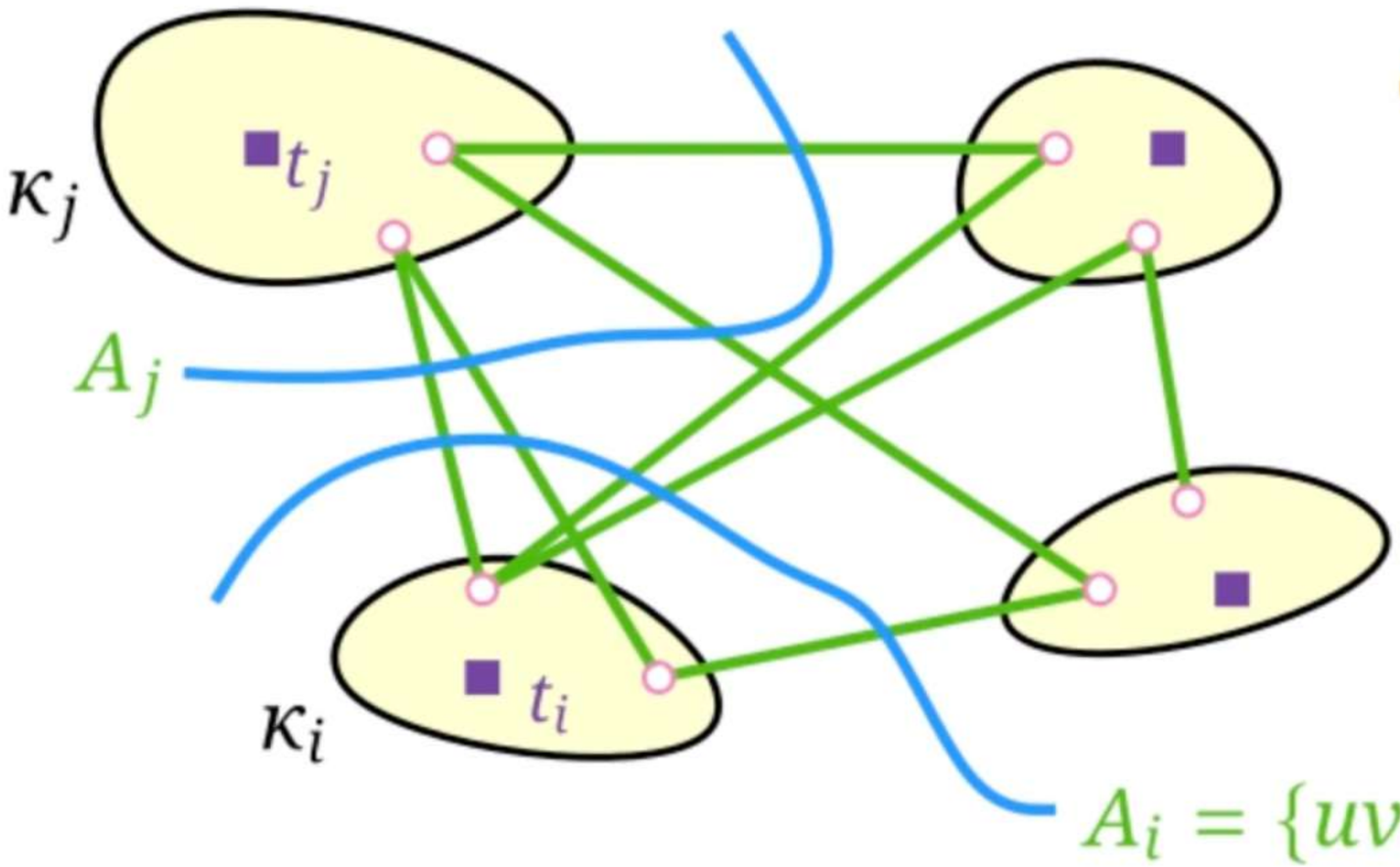
$$A_i = \{uv \in \mathcal{A} : u \in \kappa_i, v \notin \kappa_i\}$$

Observation. $\mathcal{A} = \bigcup_{i=1}^{k} A_i$ and $\sum_{i=1}^{k} c(A_i) = 2 \cdot c(\mathcal{A}) = 2 \cdot \mathrm{OPT}$.

Approximation Factor

Theorem. This algorithm is a factor-$(2 - 2/k)$ approximation algorithm for MULTIWAYCUT.

Proof. Consider an opt. multiway cut $\mathcal{A}$: Consider the alg.'s solution C:



$$c(C) \le \left(1 - \tfrac{1}{k}\right) \sum_{i=1}^{k} c(C_i)$$

$$\le \left(1 - \tfrac{1}{k}\right) \sum_{i=1}^{k} c(A_i)$$

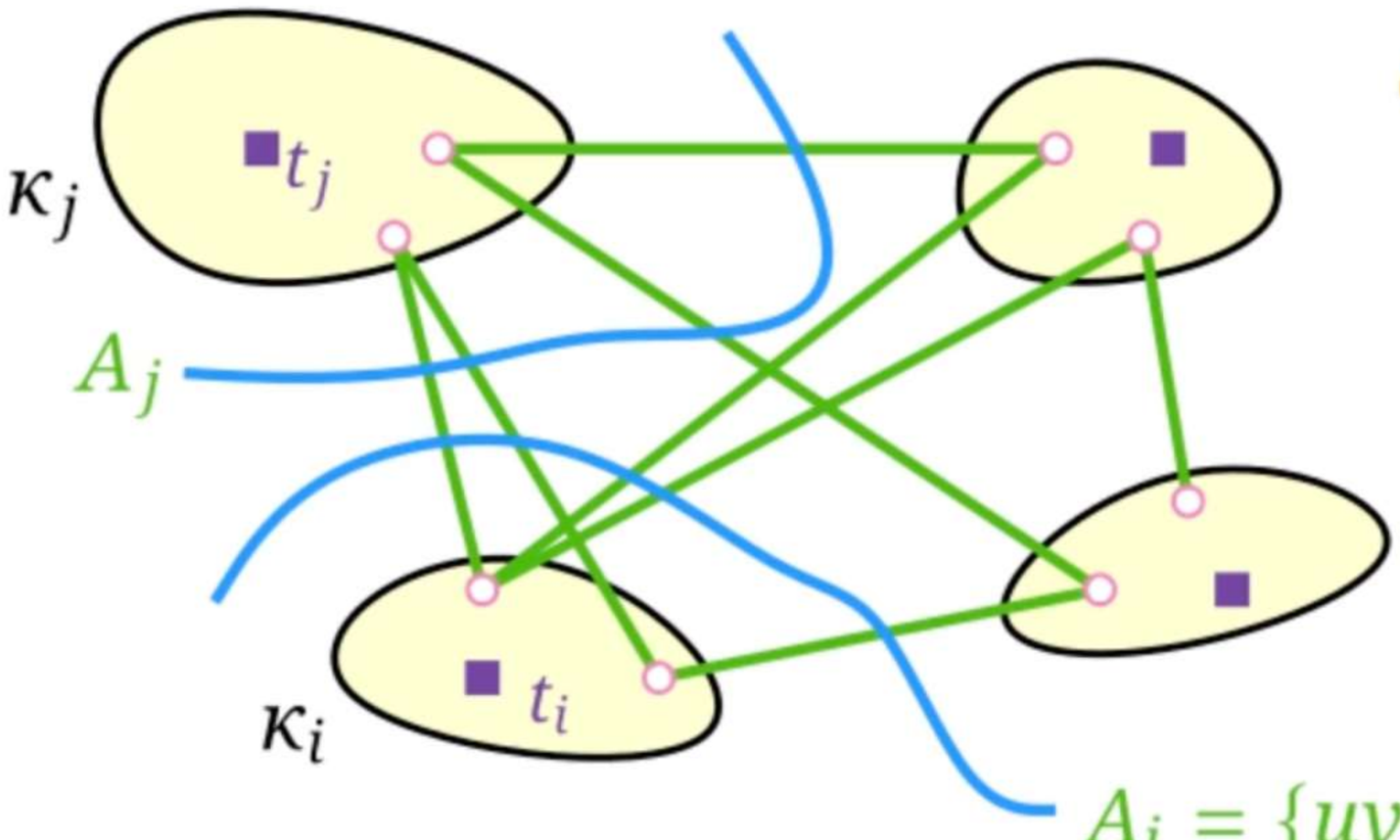
$$A_i = \{uv \in \mathcal{A} : u \in \kappa_i, v \notin \kappa_i\}$$

Observation. $\mathcal{A} = \bigcup_{i=1}^{k} A_i$ and $\sum_{i=1}^{k} c(A_i) = 2 \cdot c(\mathcal{A}) = 2 \cdot \mathrm{OPT}$.

Approximation Factor

Theorem. This algorithm is a factor-$(2 - 2/k)$ approximation algorithm for MULTIWAYCUT.

Proof. Consider an opt. multiway cut $\mathcal{A}$: Consider the alg.'s solution C:



$$c(C) \le \left(1 - \tfrac{1}{k}\right) \sum_{i=1}^{k} c(C_i)$$

$$\le \left(1 - \tfrac{1}{k}\right) \sum_{i=1}^{k} c(A_i)$$

$$= \left(1 - \tfrac{1}{k}\right) \cdot 2 \cdot c(\mathcal{A})$$

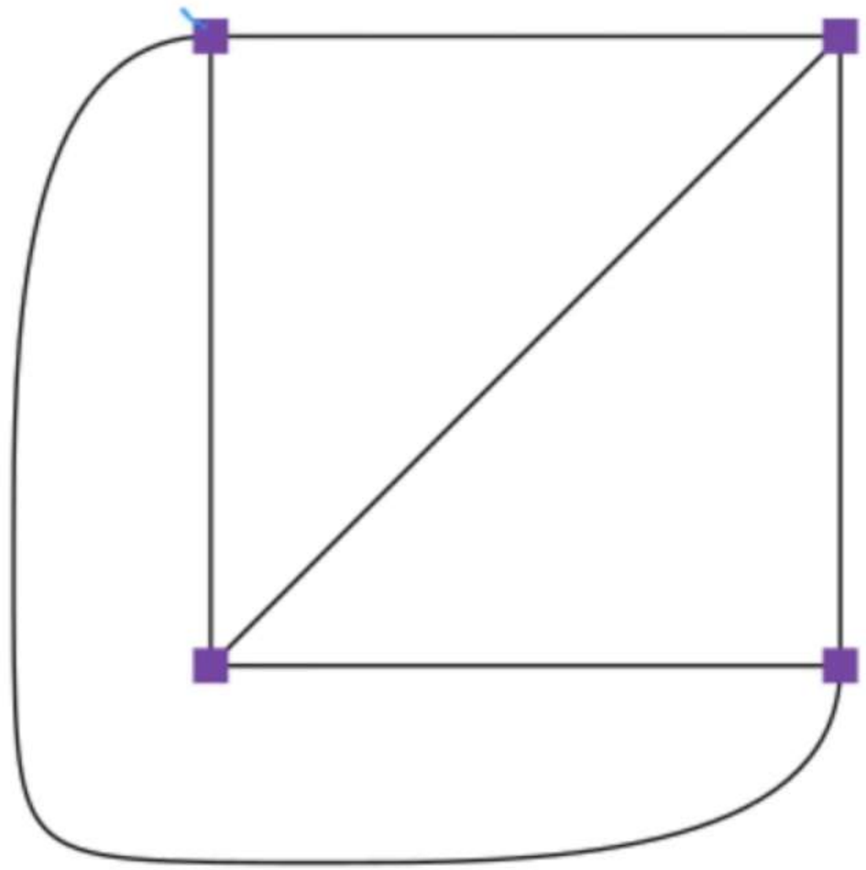
$$= \left(2 - \tfrac{2}{k}\right) \cdot \mathrm{OPT}$$

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

Observation. $\mathcal{A} = \bigcup_{i=1}^{k} A_i$ and $\sum_{i=1}^{k} c(A_i) = 2 \cdot c(\mathcal{A}) = 2 \cdot \mathrm{OPT}$.

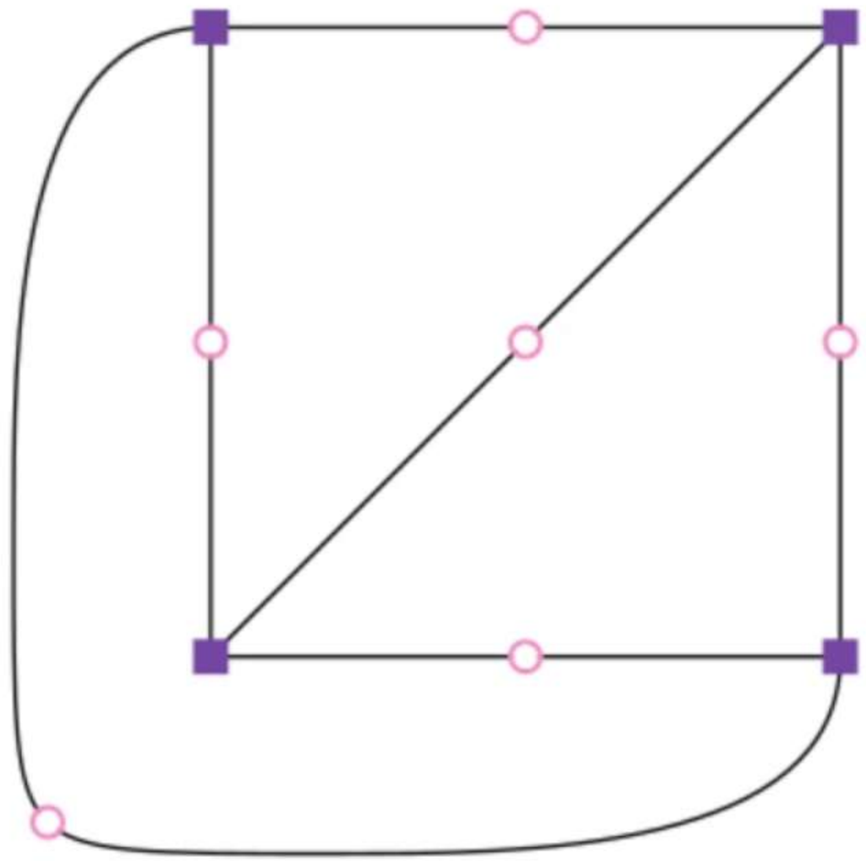
Analysis Tight?

K_k



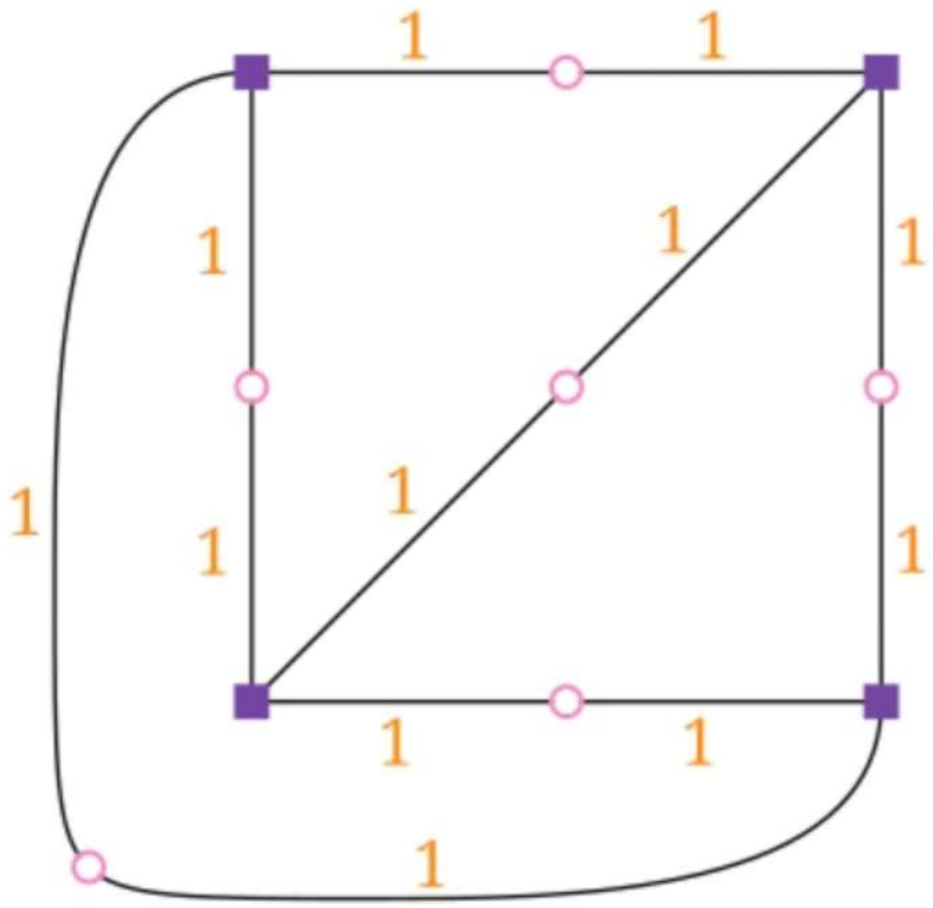
Analysis Tight?

K_k

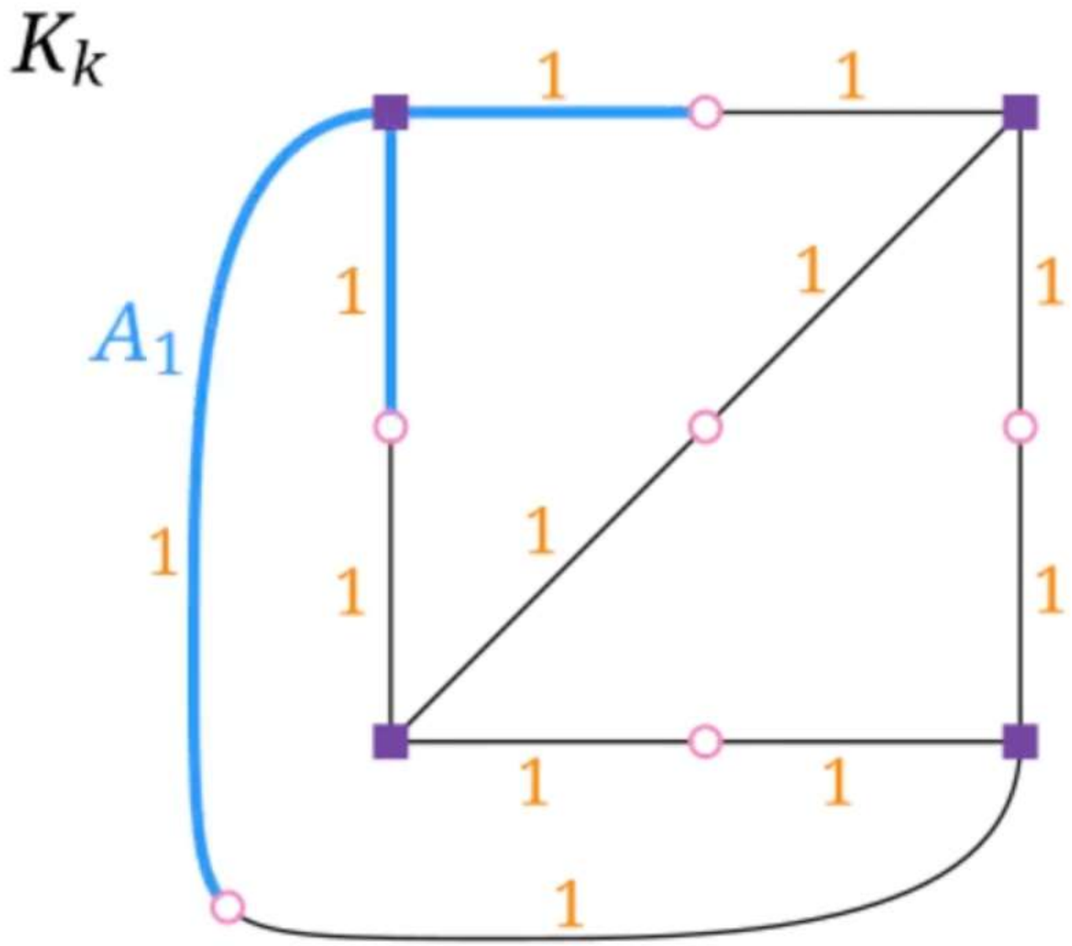


Analysis Tight?

K_k

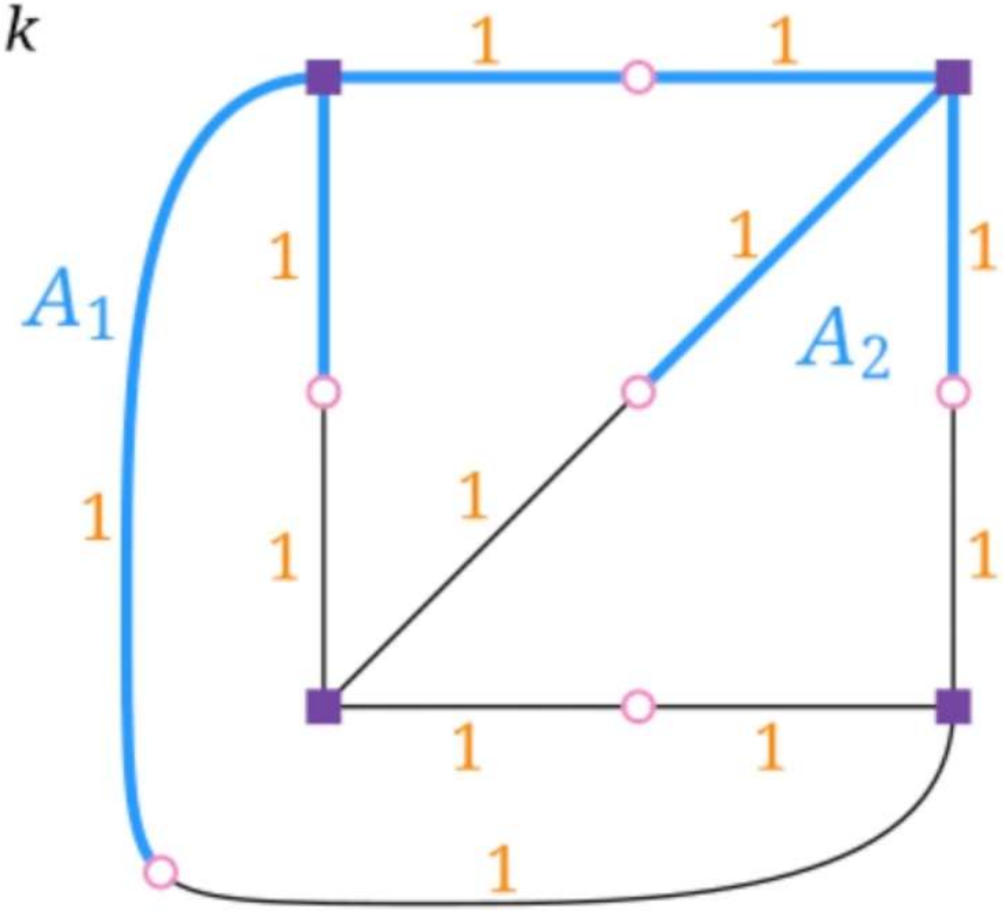


Analysis Tight?

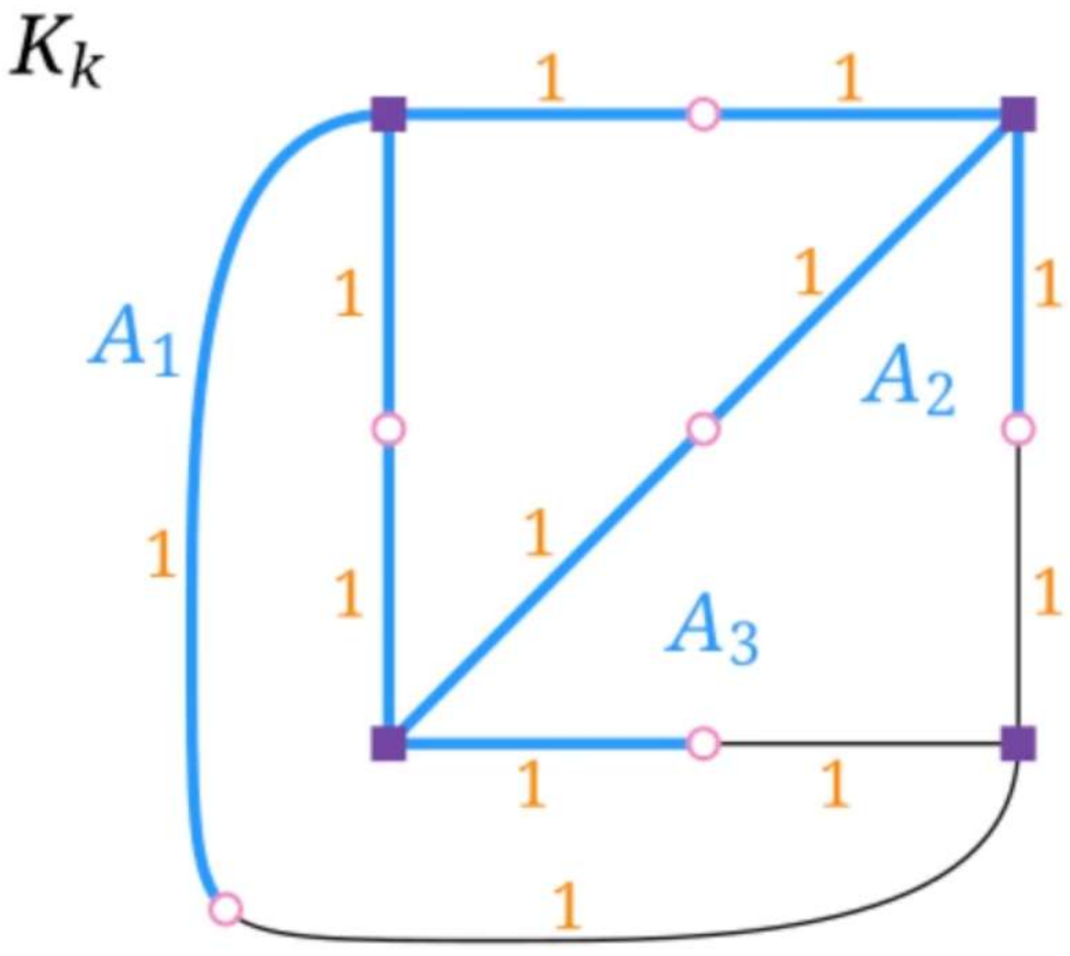


Analysis Tight?

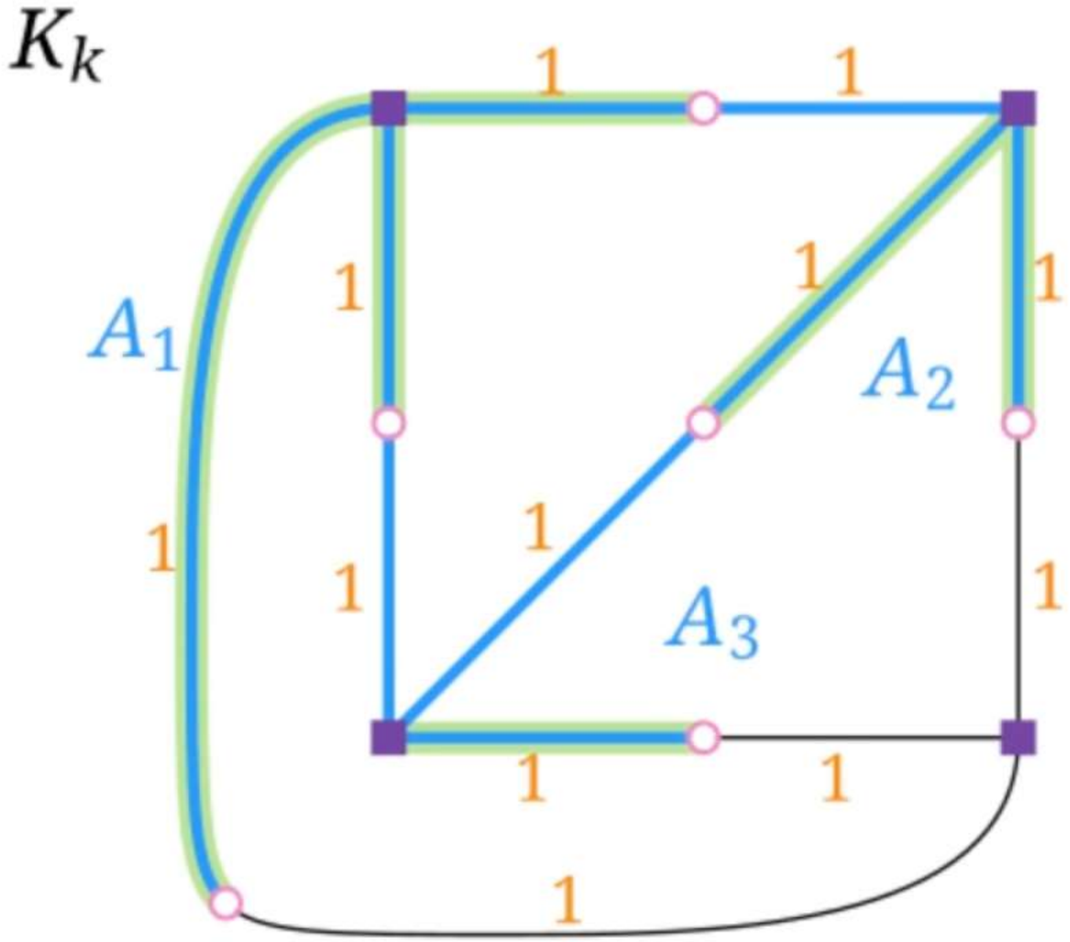
K_k



Analysis Tight?



Analysis Tight?



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

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

$$\text{ALG}/\text{OPT} = \frac{2(k-1)}{k} = 2 - \frac{2}{k}$$

Can we do better?