An Improved LP-Based Approximation for Steiner Tree

Fabrizio Grandoni

Tor Vergata Rome

grandoni@disp.uniroma2.it

Joint work with

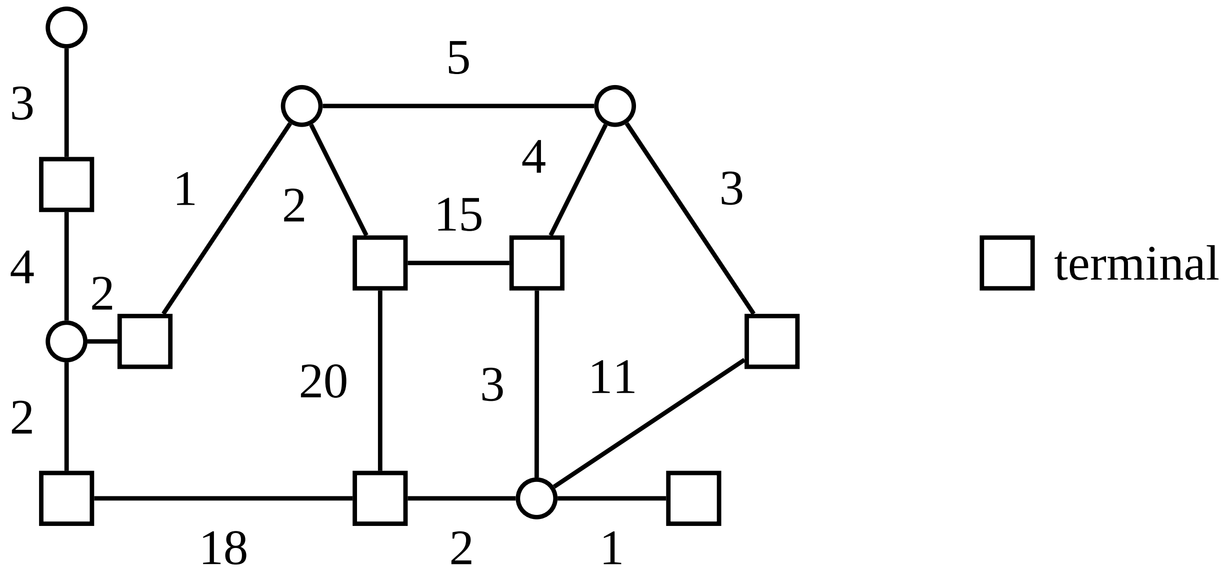
J. Byrka, T. Rothvoß, L. Sanità

The Steiner Tree Problem

Def (Steiner tree) Given an undirected graph $G = (V, E)$ with edge costs $c : E \longrightarrow \mathbb{R}_{>0}$, and a set of **terminal** nodes $R \subseteq V$, find the tree S spanning R of minimum cost $c(S) := \sum_{e \in S} c(e)$.

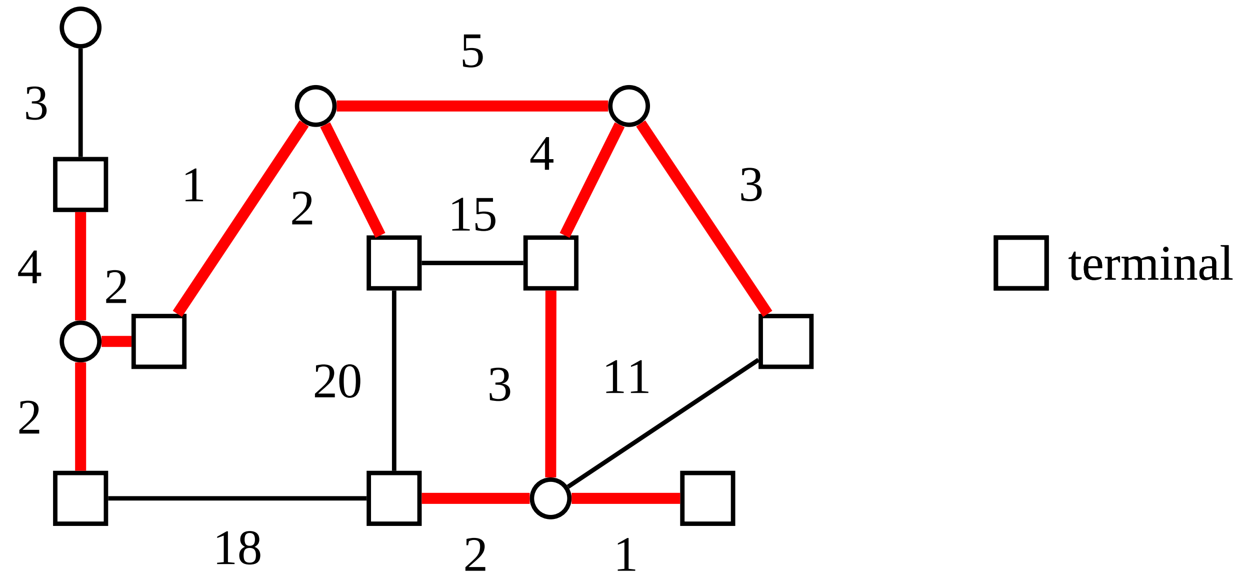
The Steiner Tree Problem

Def (Steiner tree) Given an undirected graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_{>0}$, and a set of **terminal** nodes $R \subseteq V$, find the tree S spanning R of minimum cost $c(S) := \sum_{e \in S} c(e)$.



The Steiner Tree Problem

Def (Steiner tree) Given an undirected graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_{>0}$, and a set of **terminal** nodes $R \subseteq V$, find the tree S spanning R of minimum cost $c(S) := \sum_{e \in S} c(e)$.



Known Results

Hardness:

- NP-hard even for edge costs in $\{1, 2\}$ [Bern&Plassmann'89]
- no $< \frac{96}{95}$-apx unless P=NP [Chlebik&Chlebikova'02]

Known Results

Hardness:

- NP-hard even for edge costs in $\{1, 2\}$ [Bern&Plassmann'89]
- no $< \frac{96}{95}$-apx unless P=NP [Chlebik&Chlebikova'02]

Approximation:

- 2-apx [minimum spanning tree heuristic]
- 1.83-apx [Zelikovsky'93]
- 1.67-apx [Prömel&Steger'97]
- 1.65-apx [Karpinski&Zelikovsky'97]
- 1.60-apx [Hougardy&Prömel'99]
- 1.55-apx [Robins&Zelikovsky'00]

Known Results

Hardness:

- NP-hard even for edge costs in $\{1, 2\}$ [Bern&Plassmann'89]
- no $< \frac{96}{95}$-apx unless P=NP [Chlebik&Chlebikova'02]

Approximation:

- 2-apx [minimum spanning tree heuristic]
- 1.83-apx [Zelikovsky'93]
- 1.67-apx [Prömel&Steger'97]
- 1.65-apx [Karpinski&Zelikovsky'97]
- 1.60-apx [Hougardy&Prömel'99]
- 1.55-apx [Robins&Zelikovsky'00]

Integrality gap:

- ≤ 2 [Goemans&Williamson'95, Jain'98]

Our Results and Techniques

Thr There is an (LP-based) deterministic $\ln 4 + \varepsilon < 1.39$ approximation for the Steiner tree problem

- Here we show an expected $1.5 + \varepsilon$ apx

Thr There is an LP-relaxation for Steiner tree with integrality gap at most $1 + \ln(3)/2 < 1.55$

- Here we show $1 + \ln 2 < 1.7$

Our Results and Techniques

Thr There is an (LP-based) deterministic $\ln 4 + \varepsilon < 1.39$ approximation for the Steiner tree problem

- Here we show an expected $1.5 + \varepsilon$ apx

Thr There is an LP-relaxation for Steiner tree with integrality gap at most $1 + \ln(3)/2 < 1.55$

- Here we show $1 + \ln 2 < 1.7$

- **Directed-Component Cut Relaxation**
  - bidirected cut relaxation
  - k-components
- **Iterative Randomized Rounding**
  - randomized rounding
  - iterative rounding

Directed-Component Cut Relaxation

Bidirected Cut Relaxation

- We select a **root** $r \in R$ and bi-direct the edges. Then

$$\min \sum_{e \in E} c(e) z_e \qquad \text{(BCR)}$$

$$\sum_{e \in \delta^+(U)} z_e \geq 1 \qquad \forall U \subseteq V - r, U \cap R \neq \emptyset$$

$$z_e \geq 0 \qquad \forall e \in E$$

- $\delta^+(U) = \{ab \in E : a \in U \text{ and } b \notin U\}$

Bidirected Cut Relaxation

- We select a **root** $r \in R$ and bi-direct the edges. Then

$$\min \sum_{e \in E} c(e) z_e \qquad \text{(BCR)}$$

$$\sum_{e \in \delta^+(U)} z_e \geq 1 \qquad \forall U \subseteq V - r, U \cap R \neq \emptyset$$

$$z_e \geq 0 \qquad \forall e \in E$$

- $\delta^+(U) = \{ab \in E : a \in U \text{ and } b \notin U\}$

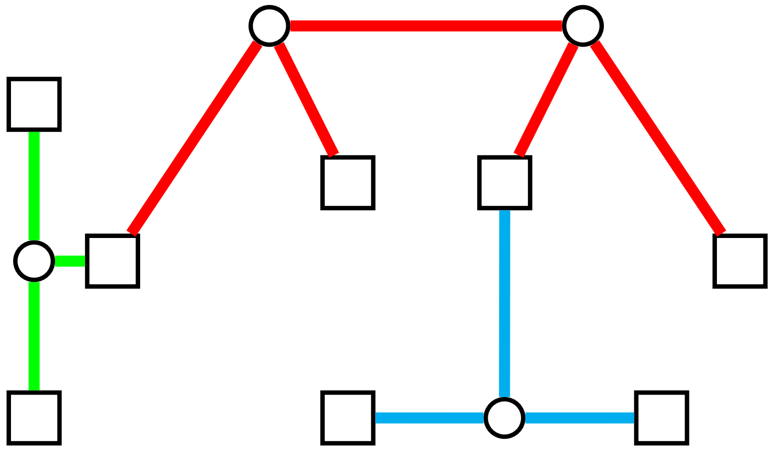
Thr [Edmonds'67] For $R = V$, BCR is integral

Rem the undirected version has integrality gap 2 even for $R = V$

Components

Def A **component** of a Steiner tree is a maximal subtree whose terminals coincide with its leaves

- A **k-component** is a component with at most k terminals
- A Steiner tree made of k-components is **k-restricted**.



Components

Def A **component** of a Steiner tree is a maximal subtree whose terminals coincide with its leaves

- A **k-component** is a component with at most k terminals
- A Steiner tree made of k-components is **k-restricted**.

Thr [Borchers & Du'97] If opt_k and opt are the costs of an optimal k-restricted Steiner tree and an optimal Steiner tree, respectively, then

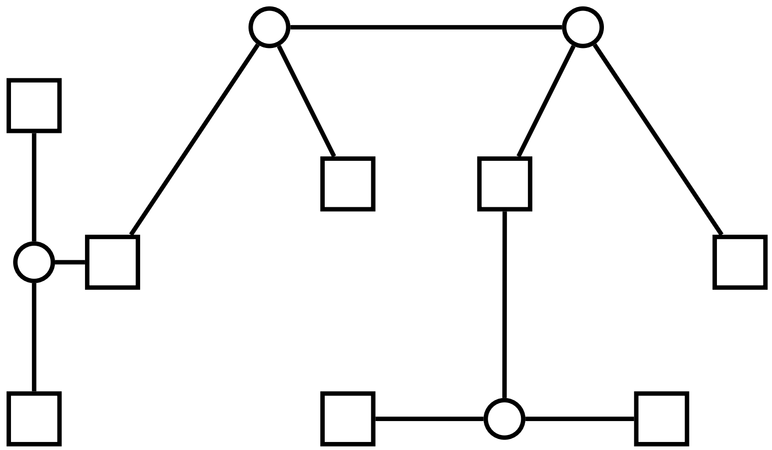
$$opt_k \leq \left(1 + \frac{1}{\lfloor \log_2 k \rfloor}\right) opt$$

Directing Components

- Direct the edges of an optimal Steiner tree towards a root terminal $r \in R$. This way we obtain **directed components**

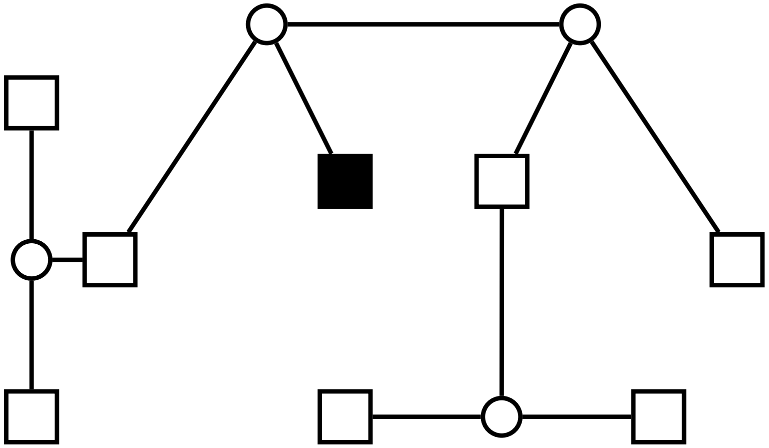
Directing Components

- Direct the edges of an optimal Steiner tree towards a root terminal $r \in R$. This way we obtain **directed components**



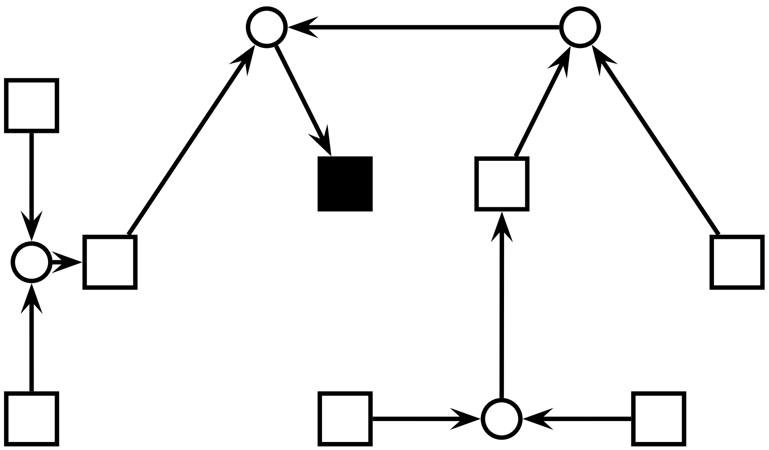
Directing Components

- Direct the edges of an optimal Steiner tree towards a root terminal $r \in R$. This way we obtain **directed components**



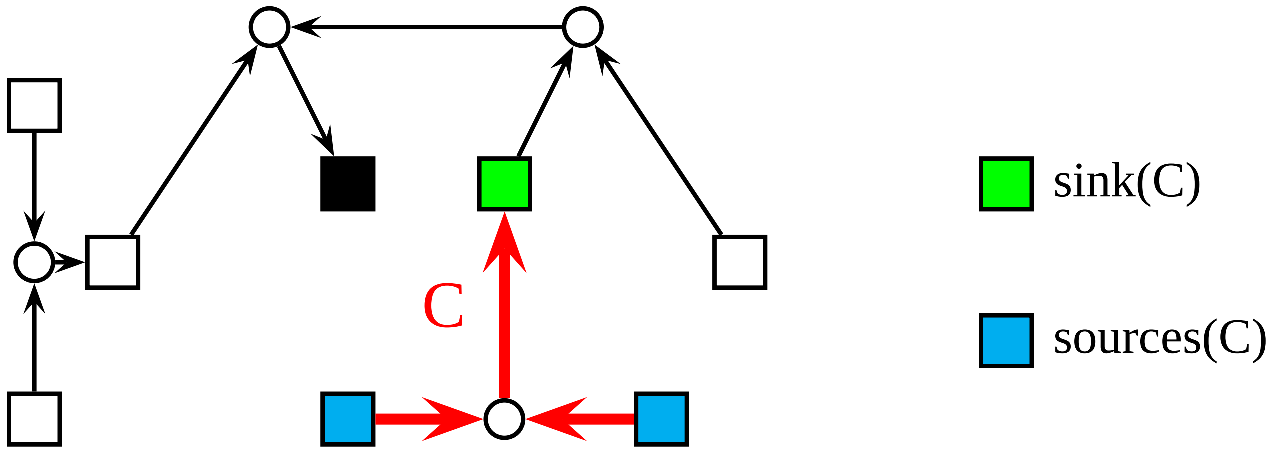
Directing Components

- Direct the edges of an optimal Steiner tree towards a root terminal $r \in R$. This way we obtain **directed components**



Directing Components

- Direct the edges of an optimal Steiner tree towards a root terminal $r \in R$. This way we obtain **directed components**



Directed-component Cut Relaxation

$$\min \sum_{C \in \mathcal{C}} c(C) x_C \qquad \text{(DCR)}$$

$$\sum_{C \in \delta^+_{\mathcal{C}}(U)} x_C \geq 1 \qquad \forall U \subseteq R - r, U \neq \emptyset$$

$$x_C \geq 0 \qquad \forall C \in \mathcal{C}$$

- $\mathcal{C}$ is the set of candidate directed components
- $\delta^+_{\mathcal{C}}(U) = \{C \in \mathcal{C} : \, sources(C) \cap U \neq \emptyset$ and $sink(C) \notin U\}$

Directed-component Cut Relaxation

$$\min \sum_{C \in \mathcal{C}} c(C) x_C \qquad \text{(DCR)}$$

$$\sum_{C \in \delta^+_{\mathcal{C}}(U)} x_C \geq 1 \qquad \forall U \subseteq R - r, U \neq \emptyset$$

$$x_C \geq 0 \qquad \forall C \in \mathcal{C}$$

- $\mathcal{C}$ is the set of candidate directed components
- $\delta^+_{\mathcal{C}}(U) = \{C \in \mathcal{C} : \; sources(C) \cap U \neq \emptyset$ and $sink(C) \notin U\}$

Lem A $(1 + \varepsilon)$ approximation of the optimal fractional solution opt^f to DCR can be computed in polynomial time

Lem The cost of a minimum terminal spanning tree is $\leq 2\, opt^f$

Lem DCR is strictly stronger than BCR

Iterative
Randomized Rounding

Iterative Randomized Rounding

- Solve an LP-relaxation for the problem
- Sample one variable with probability proportional to its fractional value, and round it
- Iterate the process on the residual problem

Iterative Randomized Rounding

- Solve an LP-relaxation for the problem
- Sample one variable with probability proportional to its fractional value, and round it
- Iterate the process on the residual problem

Rem In **randomized rounding** variables are rounded randomly and (typically) simultaneously

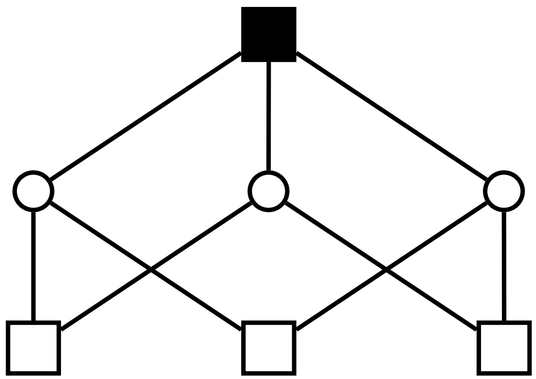
Rem In **iterative rounding** variables are rounded deterministically and (typically) one at a time

Algorithm IRR

- For $t = 1, 2, \ldots$
  - ◇ Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - ◇ Sample a component $C = C^t$ with probability $p_C^t := x_C^t / \sum_{D \in \mathcal{C}} x_D^t$
  - ◇ Contract C^t and update DCR consequently
  - ◇ If there is only one terminal, output the sampled components

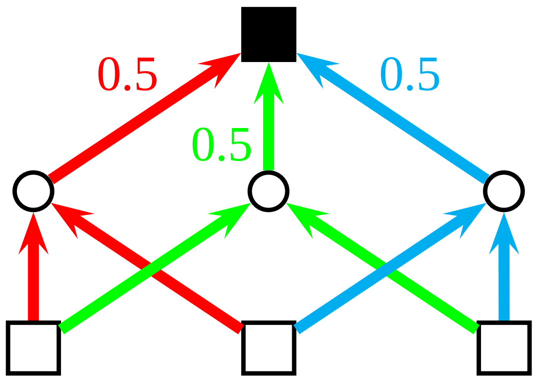
Algorithm `IRR`

- For $t = 1, 2, \ldots$
  - ◇ Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - ◇ Sample a component $C = C^t$ with probability $p_C^t := x_C^t / \sum_{D \in \mathcal{C}} x_D^t$
  - ◇ Contract C^t and update DCR consequently
  - ◇ If there is only one terminal, output the sampled components



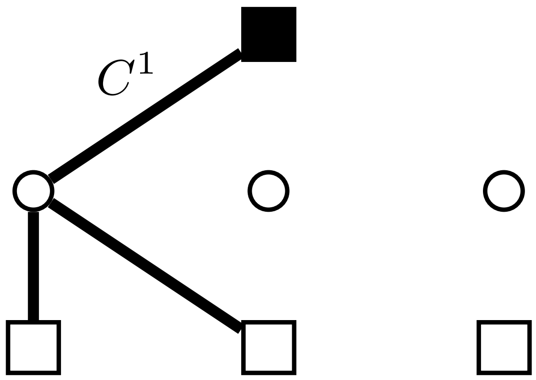
Algorithm IRR

- For $t = 1, 2, \ldots$
  - ◇ Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - ◇ Sample a component $C = C^t$ with probability $p_C^t := x_C^t / \sum_{D \in \mathcal{C}} x_D^t$
  - ◇ Contract C^t and update DCR consequently
  - ◇ If there is only one terminal, output the sampled components



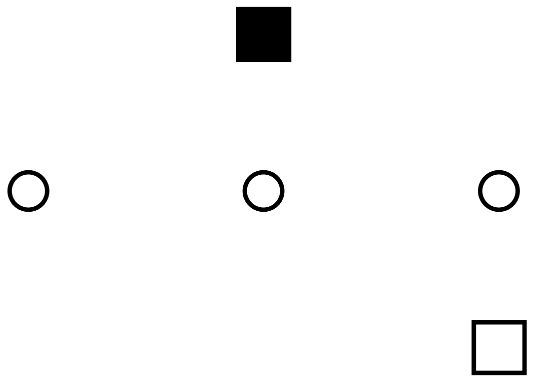
Algorithm IRR

- For $t = 1, 2, \ldots$
  - Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - Sample a component $C = C^t$ with probability $p^t_C := x^t_C / \sum_{D \in \mathcal{C}} x^t_D$
  - Contract C^t and update DCR consequently
  - If there is only one terminal, output the sampled components



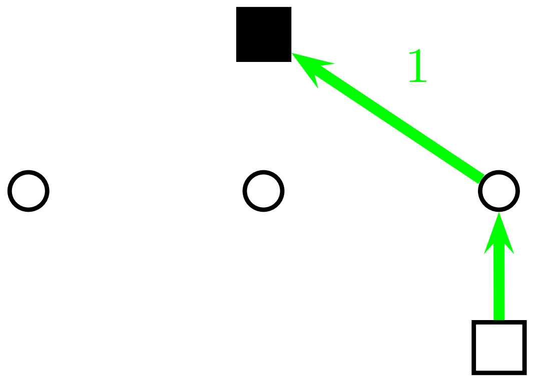
Algorithm `IRR`

- For $t = 1, 2, \ldots$
  - Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - Sample a component $C = C^t$ with probability $p_C^t := x_C^t / \sum_{D \in \mathcal{C}} x_D^t$
  - Contract C^t and update DCR consequently
  - If there is only one terminal, output the sampled components



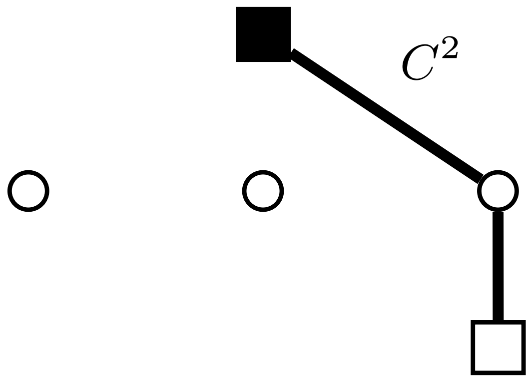
Algorithm IRR

- For $t = 1, 2, \ldots$
  - ◇ Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - ◇ Sample a component $C = C^t$ with probability $p^t_C := x^t_C / \sum_{D \in \mathcal{C}} x^t_D$
  - ◇ Contract C^t and update DCR consequently
  - ◇ If there is only one terminal, output the sampled components



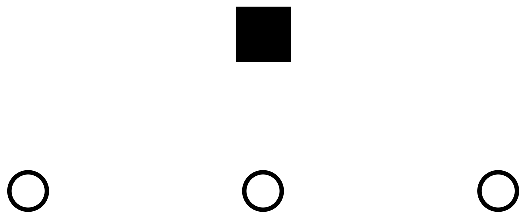
Algorithm IRR

- For $t = 1, 2, \ldots$
  - Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - Sample a component $C = C^t$ with probability $p_C^t := x_C^t / \sum_{D \in \mathcal{C}} x_D^t$
  - Contract C^t and update DCR consequently
  - If there is only one terminal, output the sampled components



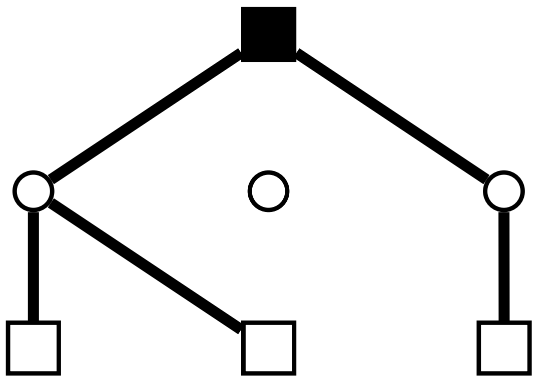
Algorithm IRR

- For $t = 1, 2, \ldots$
  - Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - Sample a component $C = C^t$ with probability $p_C^t := x_C^t / \sum_{D \in \mathcal{C}} x_D^t$
  - Contract C^t and update DCR consequently
  - If there is only one terminal, output the sampled components



Algorithm IRR

- For $t = 1, 2, \ldots$
  - Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - Sample a component $C = C^t$ with probability $p_C^t := x_C^t / \sum_{D \in \mathcal{C}} x_D^t$
  - Contract C^t and update DCR consequently
  - If there is only one terminal, output the sampled components



Algorithm IRR

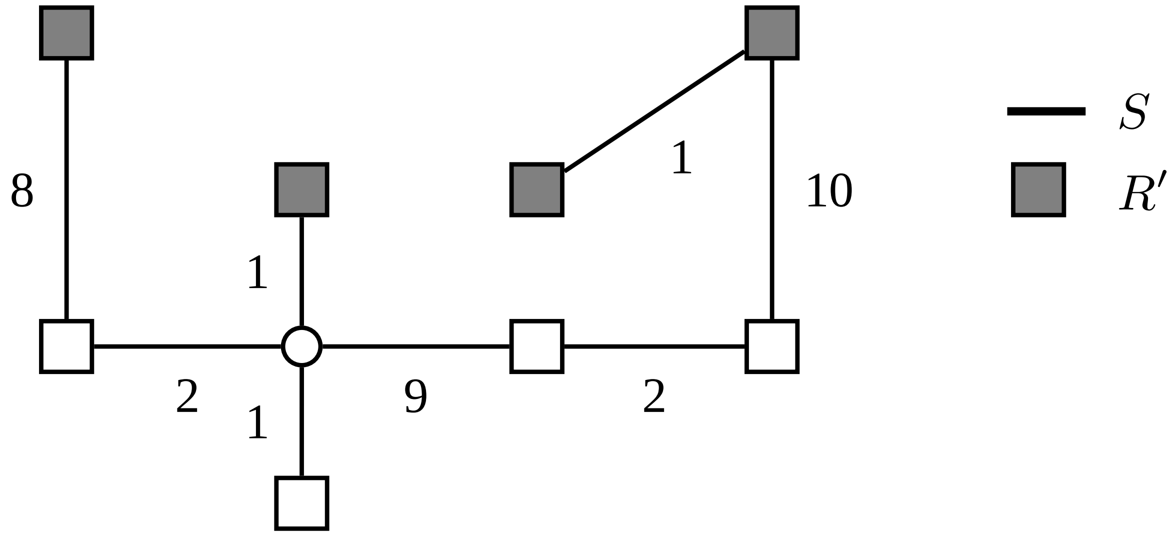
- For $t = 1, 2, \ldots$
  - ◇ Compute a $(1+\varepsilon)$-apx solution x^t for DCR
  - ◇ Sample a component $C = C^t$ with probability $p^t_C := x^t_C / \sum_{D \in \mathcal{C}} x^t_D$
  - ◇ Contract C^t and update DCR consequently
  - ◇ If there is only one terminal, output the sampled components

Rem By adding a dummy component in the root, we can assume w.l.o.g. that $M := \sum_{D \in \mathcal{C}} x^t_D$ is fixed for all t

Bridge Lemma

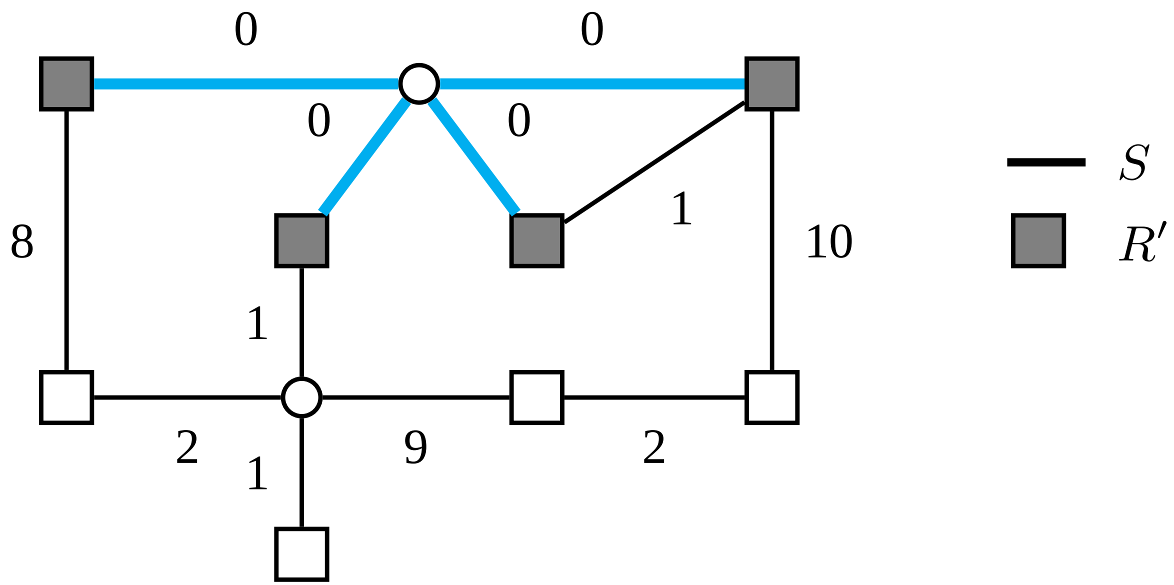
Bridges

Def Given a Steiner tree S and $R' \subseteq R$, the **bridges** $br_{S,c}(R')$ of S w.r.t. R' (and edge costs c) are the edges of S which do not belong to the minimum spanning tree of $V(S)$ after the contraction of R'



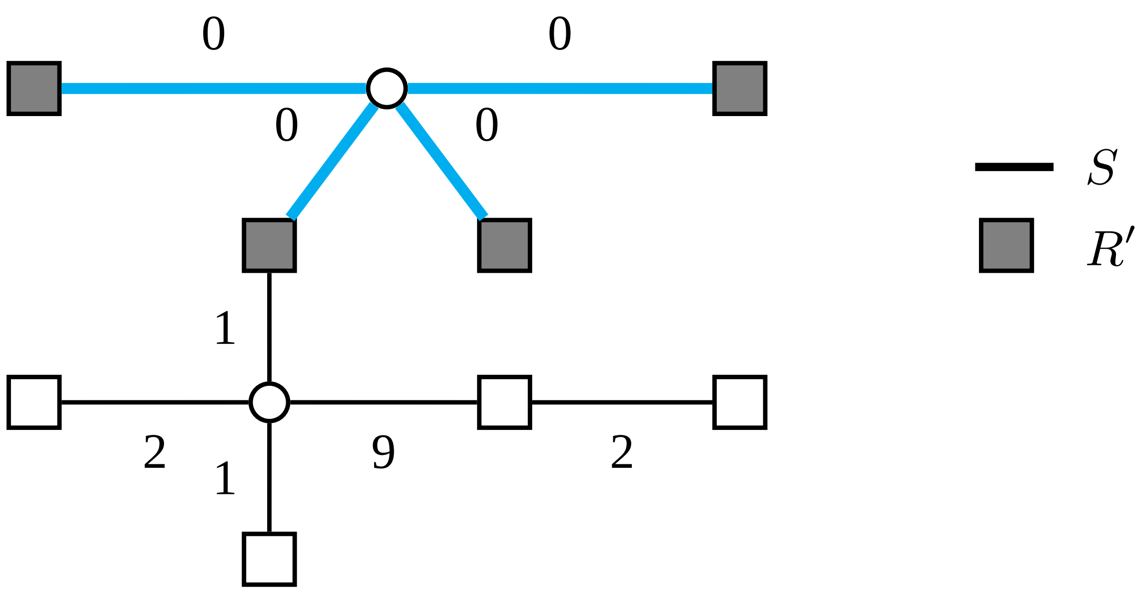
Bridges

Def Given a Steiner tree S and $R' \subseteq R$, the **bridges** $br_{S,c}(R')$ of S w.r.t. R' (and edge costs c) are the edges of S which do not belong to the minimum spanning tree of $V(S)$ after the contraction of R'



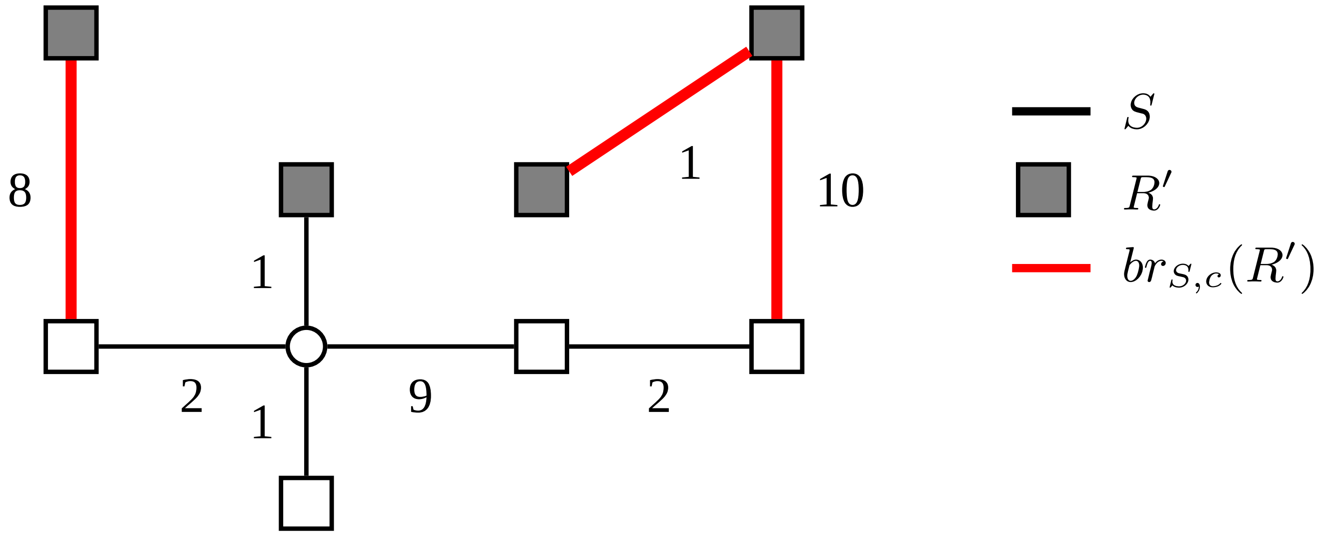
Bridges

Def Given a Steiner tree S and $R' \subseteq R$, the **bridges** $br_{S,c}(R')$ of S w.r.t. R' (and edge costs c) are the edges of S which do not belong to the minimum spanning tree of $V(S)$ after the contraction of R'



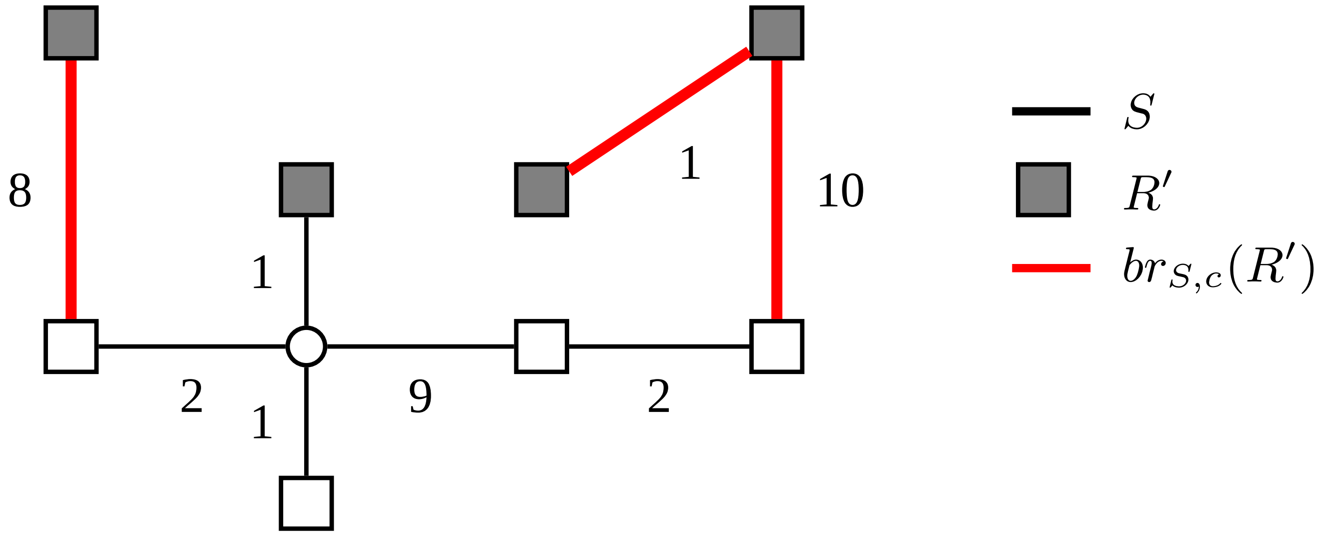
Bridges

Def Given a Steiner tree S and $R' \subseteq R$, the **bridges** $br_{S,c}(R')$ of S w.r.t. R' (and edge costs c) are the edges of S which do not belong to the minimum spanning tree of $V(S)$ after the contraction of R'



Bridges

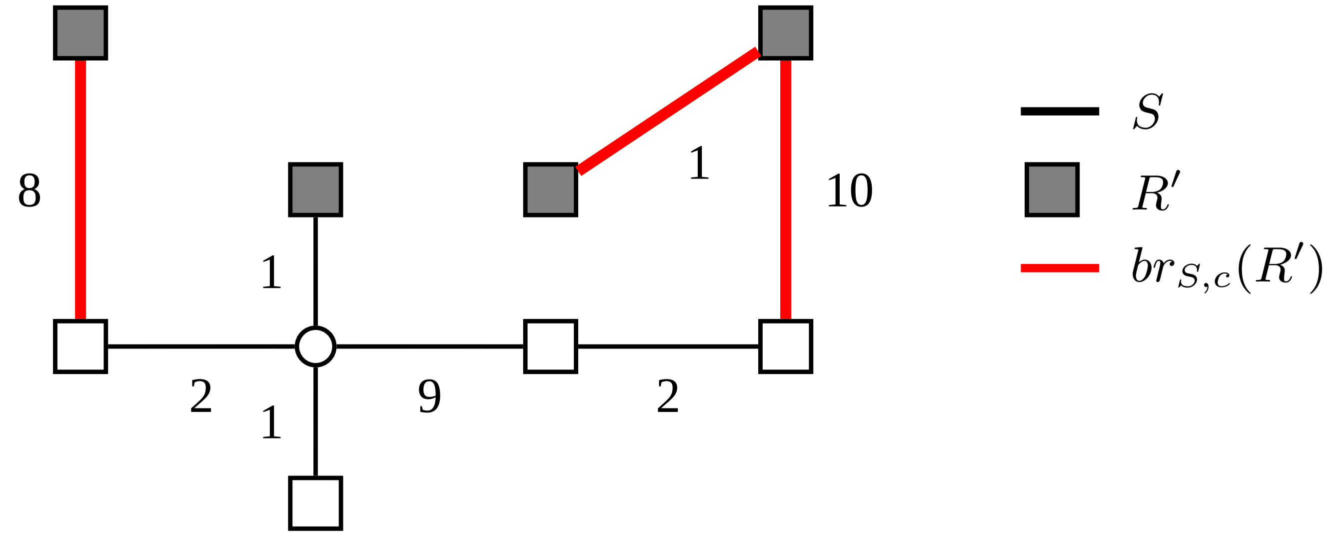
Def Given a Steiner tree S and $R' \subseteq R$, the **bridges** $br_{S,c}(R')$ of S w.r.t. R' (and edge costs c) are the edges of S which do not belong to the minimum spanning tree of $V(S)$ after the contraction of R'



Rem The most expensive edge on a path between two gray nodes is a bridge

Bridges

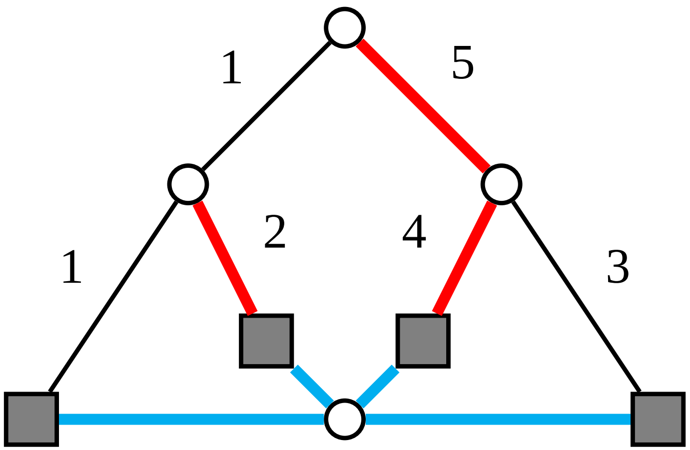
Def Given a Steiner tree S and $R' \subseteq R$, the **bridges** $br_{S,c}(R')$ of S w.r.t. R' (and edge costs c) are the edges of S which do not belong to the minimum spanning tree of $V(S)$ after the contraction of R'



Rem Let $br_S(R')$=$br_{S,c}(R')$, $br_S(R')$:=$c(br_S(R'))$ and $br_S(C)$:=$br_S(R \cap C)$.

Bridges

Lem For any Steiner tree S on R, $br_S(R) \geq \frac{1}{2}c(S)$



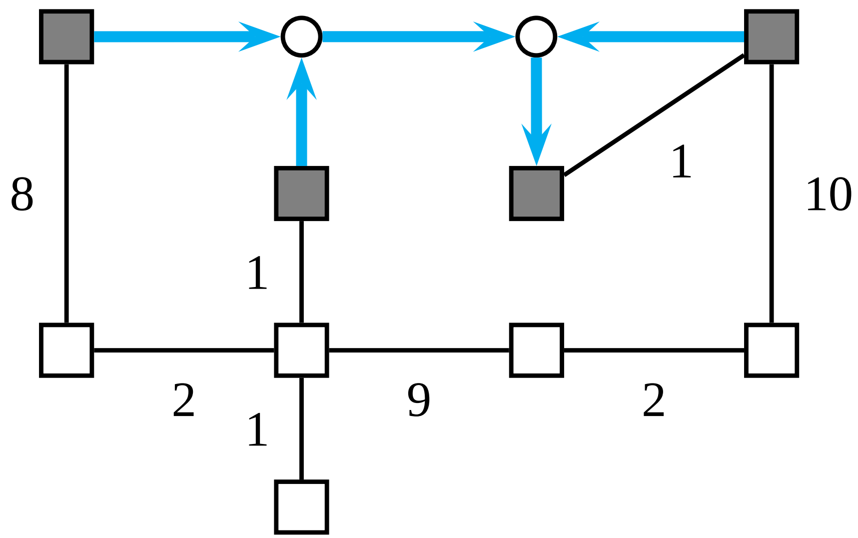
The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

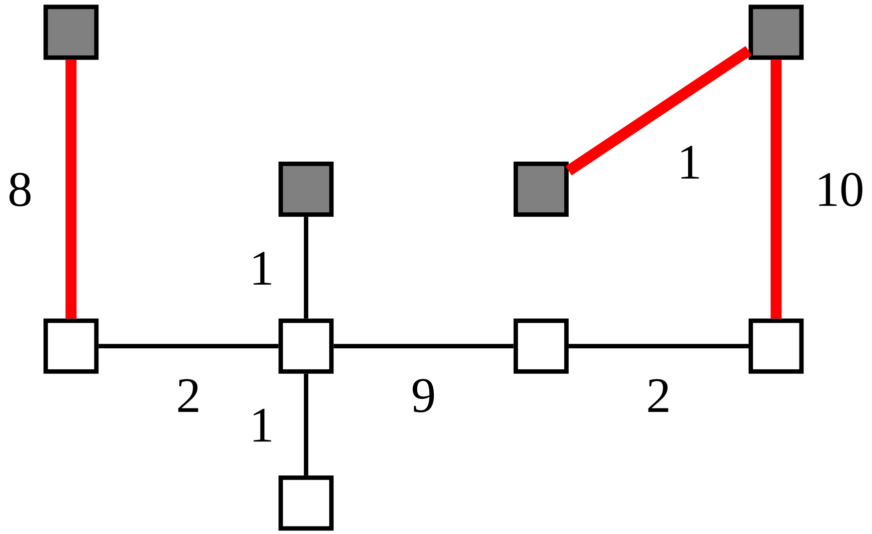
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

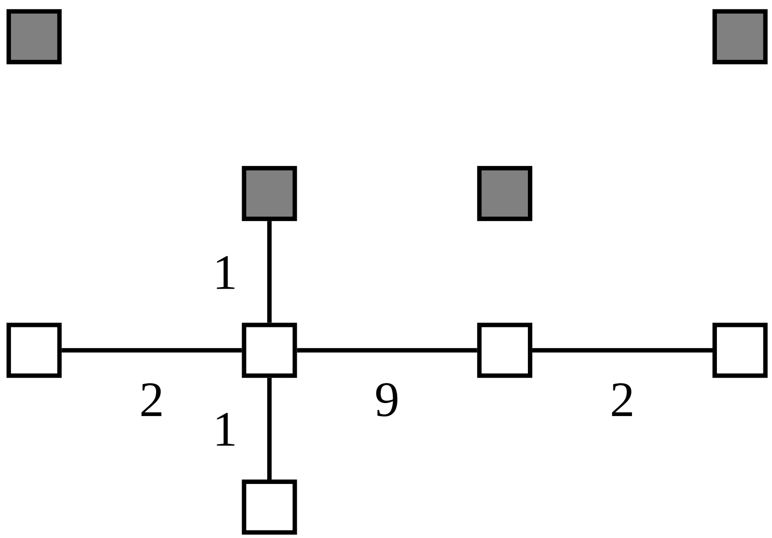
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

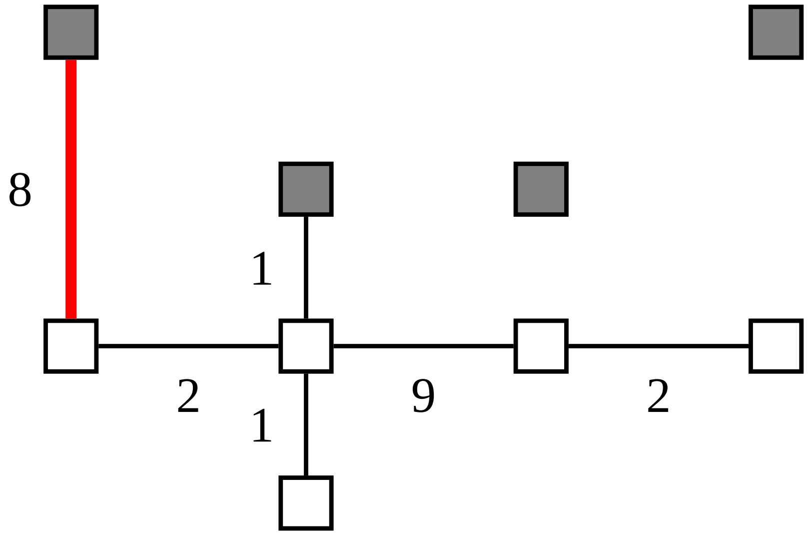
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

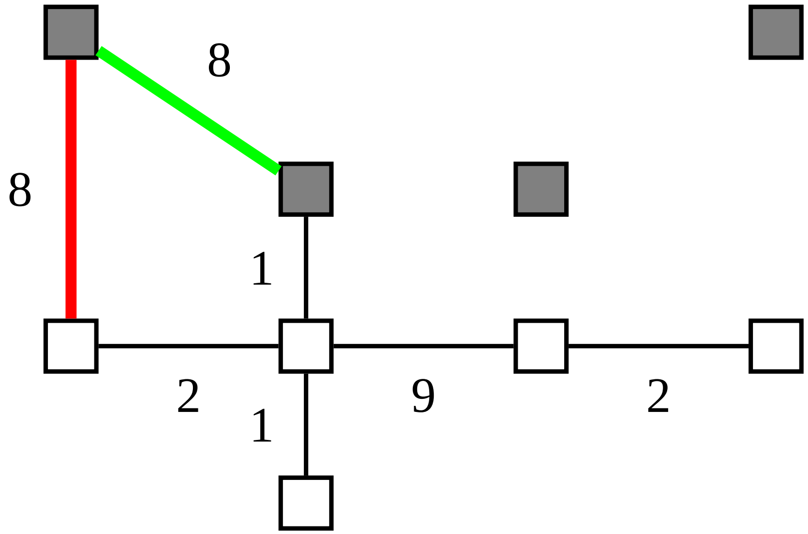
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

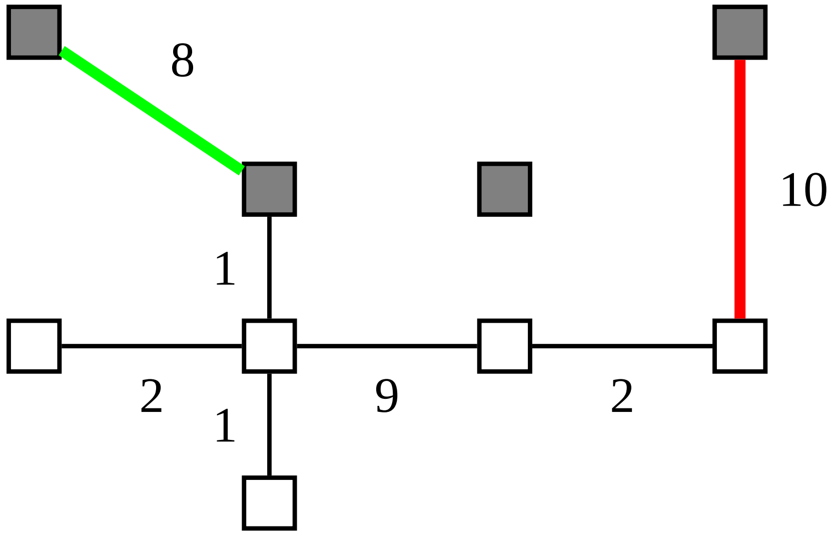
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

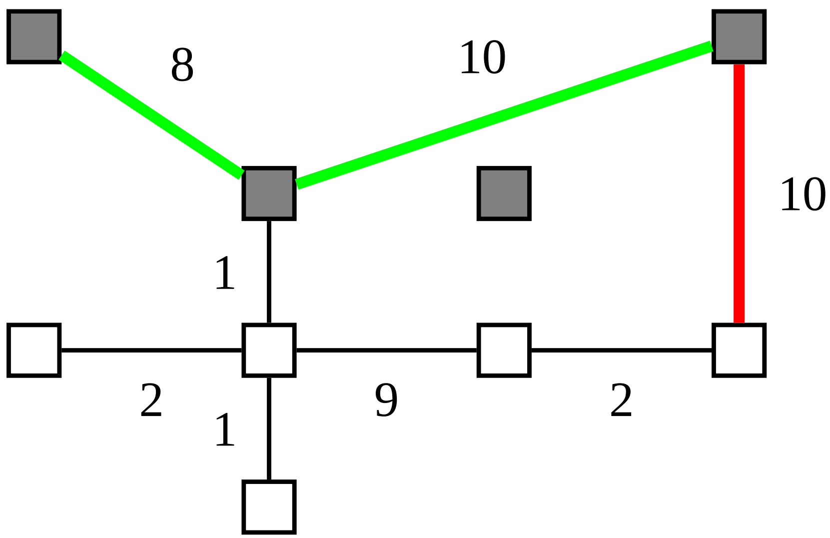
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

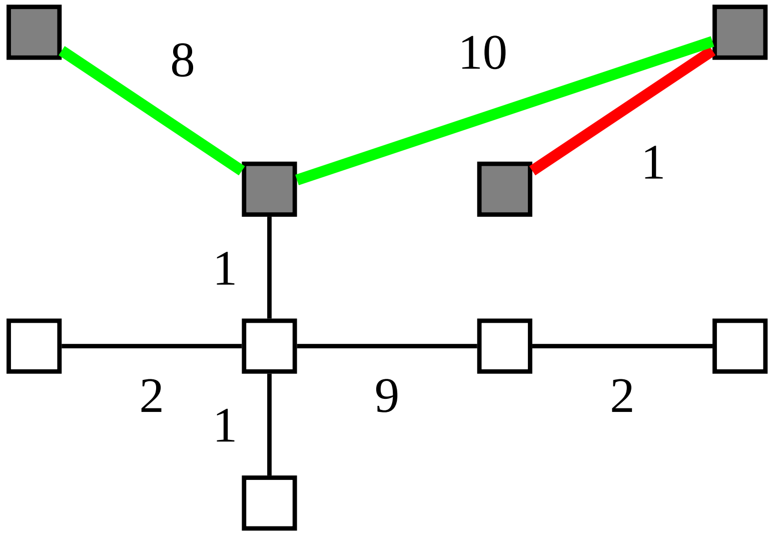
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

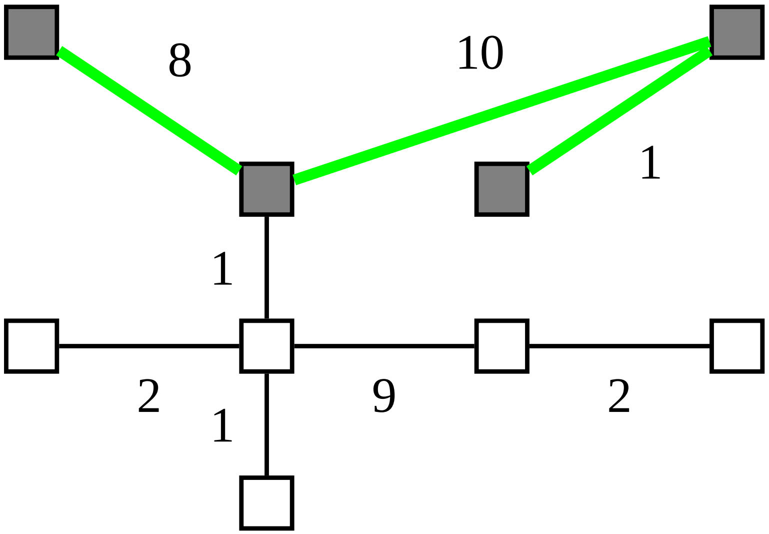
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

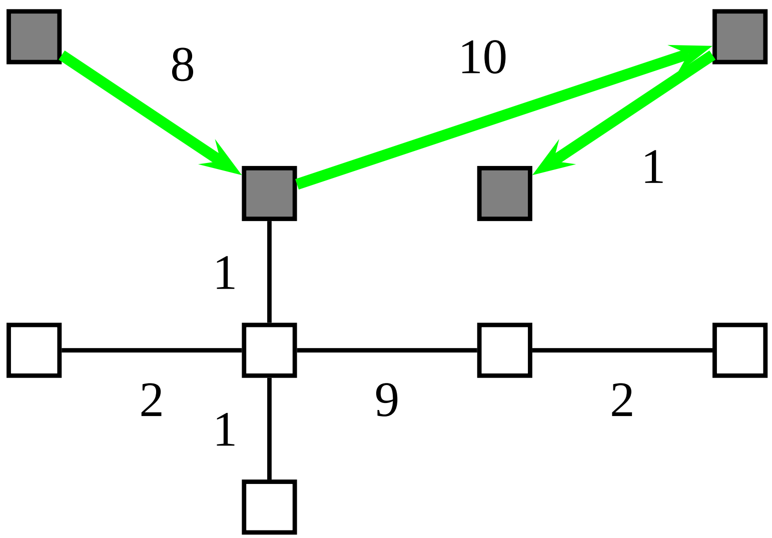
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

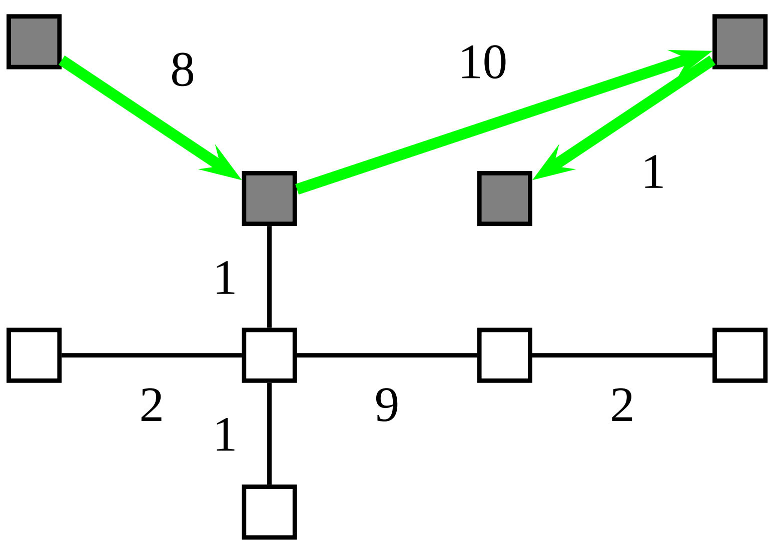
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

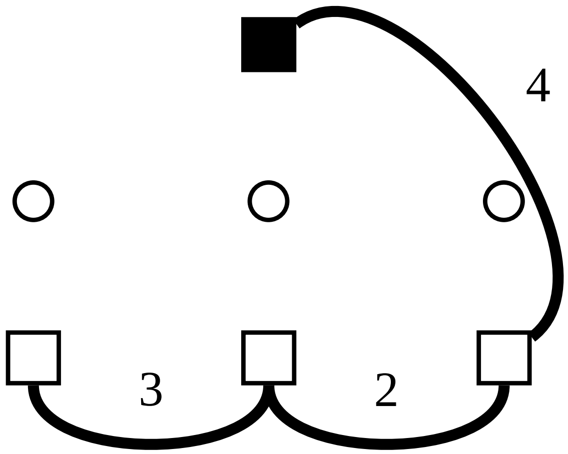
- For every $C \in \mathcal{C}$, with capacity x_C, construct a directed terminal spanning tree Y_C on $R \cap C$, with capacity x_C and edge weights w, as follows



Rem Y_C supports the same flow to the root as C w.r.t. terminals

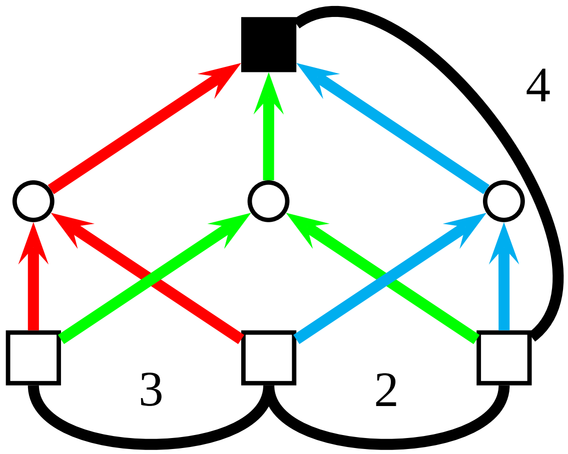
The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$



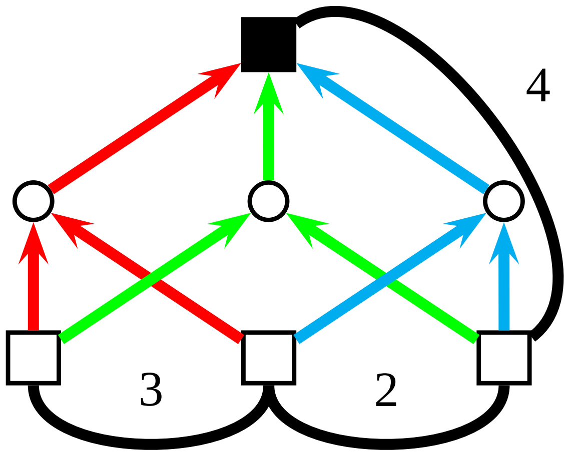
The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$



The Bridge Lemma

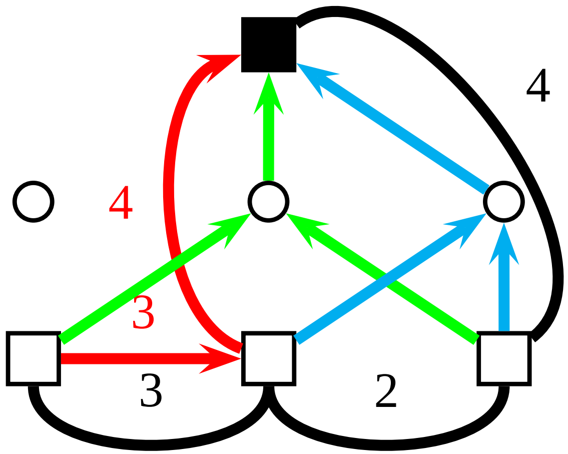
Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$



- Replace each component C with the corresponding Y_C (cumulating capacities)

The Bridge Lemma

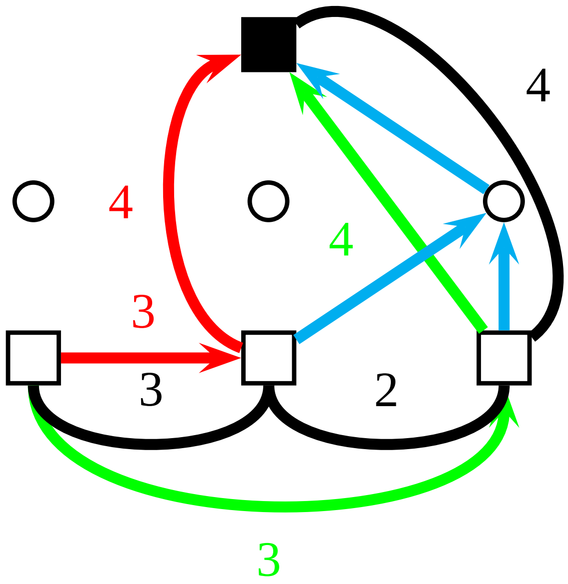
Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$



- Replace each component C with the corresponding Y_C (cumulating capacities)

The Bridge Lemma

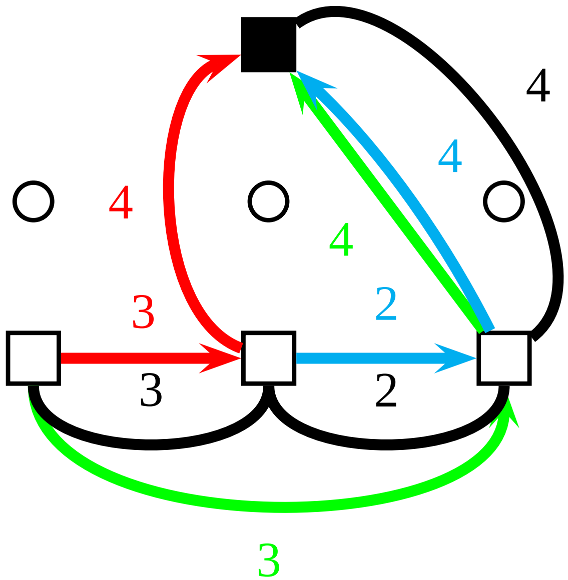
Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$



- Replace each component C with the corresponding Y_C (cumulating capacities)

The Bridge Lemma

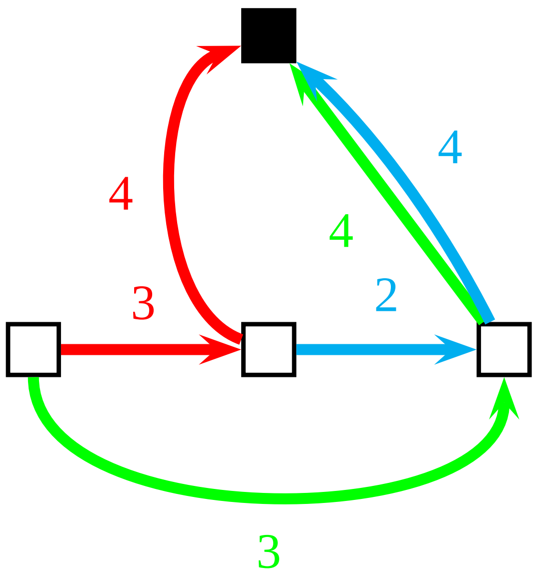
Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$



- Replace each component C with the corresponding Y_C (cumulating capacities)

The Bridge Lemma

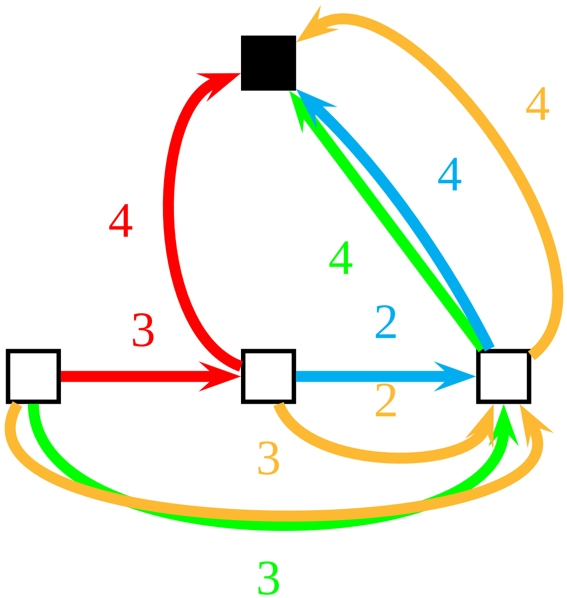
Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$



- We obtain a feasible fractional directed terminal spanning tree on a directed graph with $V = R$ and edge costs w

The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C\in\mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

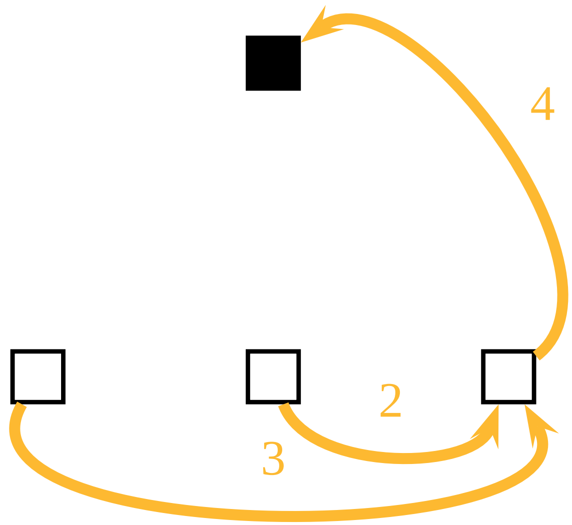


- We obtain a feasible fractional directed terminal spanning tree on a directed graph with $V = R$ and edge costs w

$\Rightarrow$ By Edmod's thr there is a cheaper (w.r.t. w) integral directed terminal spanning tree F

The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

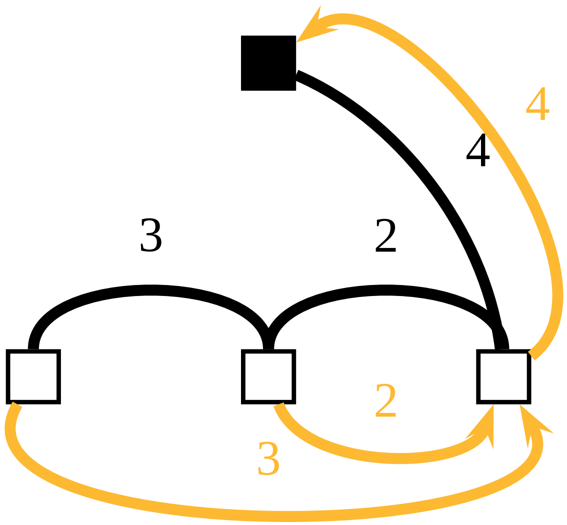


- We obtain a feasible fractional directed terminal spanning tree on a directed graph with $V = R$ and edge costs w

$\Rightarrow$ By Edmod's thr there is a cheaper (w.r.t. w) integral directed terminal spanning tree F

The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$



- The new terminal spanning tree F is more expensive than the original terminal spanning tree T by the cycle-rule

The Bridge Lemma

Lem (Bridge Lemma) For *any* terminal spanning tree T and *any* feasible fractional solution x to DCR, $\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) \geq c(T)$

- Summarizing

$$\sum_{C \in \mathcal{C}} x_C \cdot br_T(C) = \underbrace{\sum_{C \in \mathcal{C}} x_C \cdot w(Y_C)}_{\text{w-cost of fractional terminal spanning tree}} \geq \underbrace{w(F)}_{\text{w-cost of integral terminal spanning tree}} \geq c(T)$$

Approximation Factor

A First Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1 + \ln 2 + \varepsilon)\, opt^f$

A First Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1 + \ln 2 + \varepsilon)\, opt^f$

Cor The integrality gap of DCR is at most $1 + \ln 2 < 1.7$

A First Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1 + \ln 2 + \varepsilon)\, opt^f$

$$E[apx] = \sum_{t \geq 1} E[c(C^t)] \leq \sum_{t \geq 1} E[\sum_C \frac{x_C^t}{M} c(C)] \leq \frac{1+\varepsilon}{M} \sum_{t \geq 1} E[opt^{f,t}]$$

$$\leq \frac{1+\varepsilon}{M} \sum_{t=1}^{M \ln 2} opt^f + \frac{1+\varepsilon}{M} \sum_{t > M \ln 2} E[c(T^t)]$$

- T^t is a minimum terminal spanning tree at step t

A First Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1 + \ln 2 + \varepsilon)\, opt^f$

$$E[apx] = \sum_{t\geq 1} E[c(C^t)] \leq \sum_{t\geq 1} E[\sum_C \frac{x_C^t}{M} c(C)] \leq \frac{1+\varepsilon}{M} \sum_{t\geq 1} E[opt^{f,t}]$$

$$\leq \frac{1+\varepsilon}{M} \sum_{t=1}^{M \ln 2} opt^f + \frac{1+\varepsilon}{M} \sum_{t > M \ln 2} E[c(T^t)]$$

Lem For any t, $E[c(T^{t+1})] \leq (1 - \frac{1}{M}) c(T^t)$

$$E[c(T^{t+1})] \leq c(T^t) - E[br_{T^t}(C^t)] = c(T^t) - \sum_C \frac{x_C^t}{M} br_{T^t}(C)$$

$$\stackrel{\text{Bridge Lem}}{\leq} c(T^t) - \frac{1}{M} c(T^t)$$

A First Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1 + \ln 2 + \varepsilon)\, opt^f$

$$E[apx] = \sum_{t \geq 1} E[c(C^t)] \leq \sum_{t \geq 1} E[\sum_{C} \frac{x_C^t}{M} c(C)] \leq \frac{1+\varepsilon}{M} \sum_{t \geq 1} E[opt^{f,t}]$$

$$\leq \frac{1+\varepsilon}{M} \sum_{t=1}^{M \ln 2} opt^f + \frac{1+\varepsilon}{M} \sum_{t > M \ln 2} E[c(T^t)]$$

Lem For any t, $E[c(T^{t+1})] \leq (1 - \frac{1}{M}) c(T^t)$

$$E[c(T^{t+1})] \leq c(T^t) - E[br_{T^t}(C^t)] = c(T^t) - \sum_{C} \frac{x_C^t}{M} br_{T^t}(C)$$

$$\overset{\text{Bridge Lem}}{\leq} c(T^t) - \frac{1}{M} c(T^t)$$

Cor $E[c(T^t)] \leq (1 - \frac{1}{M})^{t-1} c(T^1) \leq (1 - \frac{1}{M})^{t-1} 2\, opt^f$

A First Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1 + \ln 2 + \varepsilon)\, opt^f$

$$E[apx] = \sum_{t \geq 1} E[c(C^t)] \leq \sum_{t \geq 1} E[\sum_C \frac{x_C^t}{M} c(C)] \leq \frac{1+\varepsilon}{M} \sum_{t \geq 1} E[opt^{f,t}]$$

$$\leq \frac{1+\varepsilon}{M} \sum_{t=1}^{M \ln 2} opt^f + \frac{1+\varepsilon}{M} \sum_{t > M \ln 2} E[c(T^t)]$$

$$\leq opt^f (1+\varepsilon) \ln 2 + 2\, opt^f (1+\varepsilon) \sum_{t > M \ln 2} \frac{1}{M} \left(1 - \frac{1}{M}\right)^{t-1}$$

$$\leq (1+\varepsilon)(\ln 2 + 2e^{-\ln 2}) \cdot opt^f$$

A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Rem This bound might not hold w.r.t. opt^f

A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

$$E[apx] = \sum_{t \geq 1} E[c(C^t)] \leq \sum_{t \geq 1} E[\sum_C \frac{x_C^t}{M} c(C)] \leq \frac{1+\varepsilon}{M} \sum_{t \geq 1} E[opt^{f,t}]$$

$$\leq \frac{1+\varepsilon}{M} \sum_{t=1}^{\mathbf{M \ln 4}} \mathbf{E[c(S^t)]} + \frac{1+\varepsilon}{M} \sum_{t > \mathbf{M \ln 4}} E[c(T^t)]$$

- S^t is a minimum Steiner tree at step t

A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M}) c(S^t)$

A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M}) c(S^t)$

- Construct a terminal spanning tree (Y^t, w) w.r.t. S^t and all its terminals $R^t = R \cap S^t$ as in the proof of the bridge lemma.

A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M})c(S^t)$

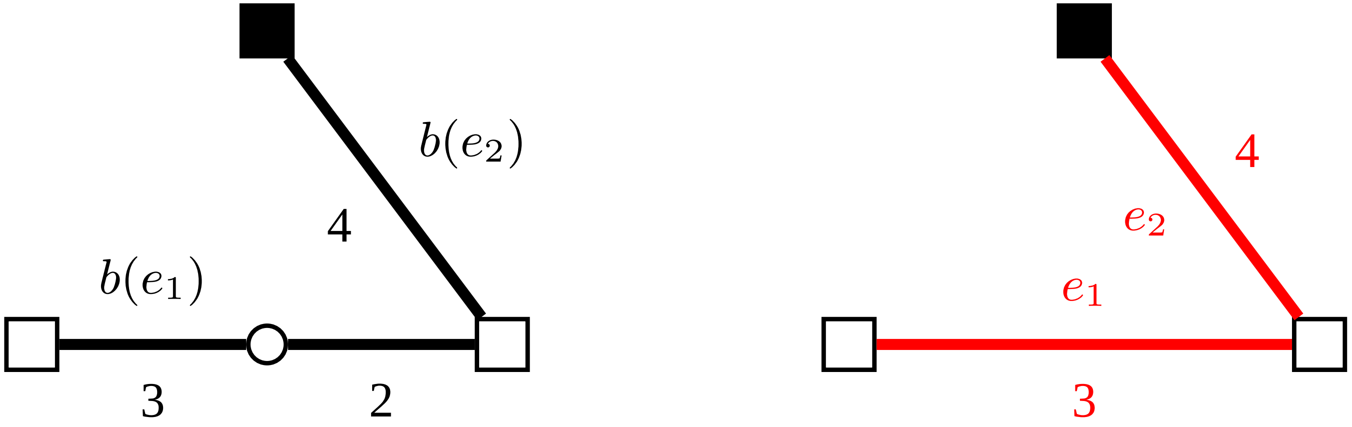
- Construct a terminal spanning tree (Y^t, w) w.r.t. S^t and all its terminals $R^t = R \cap S^t$ as in the proof of the bridge lemma.
- Let $b(e) \in S^t$ be the bridge associated to $e \in Y^t$.

A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M}) c(S^t)$

- Construct a terminal spanning tree (Y^t, w) w.r.t. S^t and all its terminals $R^t = R \cap S^t$ as in the proof of the bridge lemma.
- Let $b(e) \in S^t$ be the bridge associated to $e \in Y^t$.

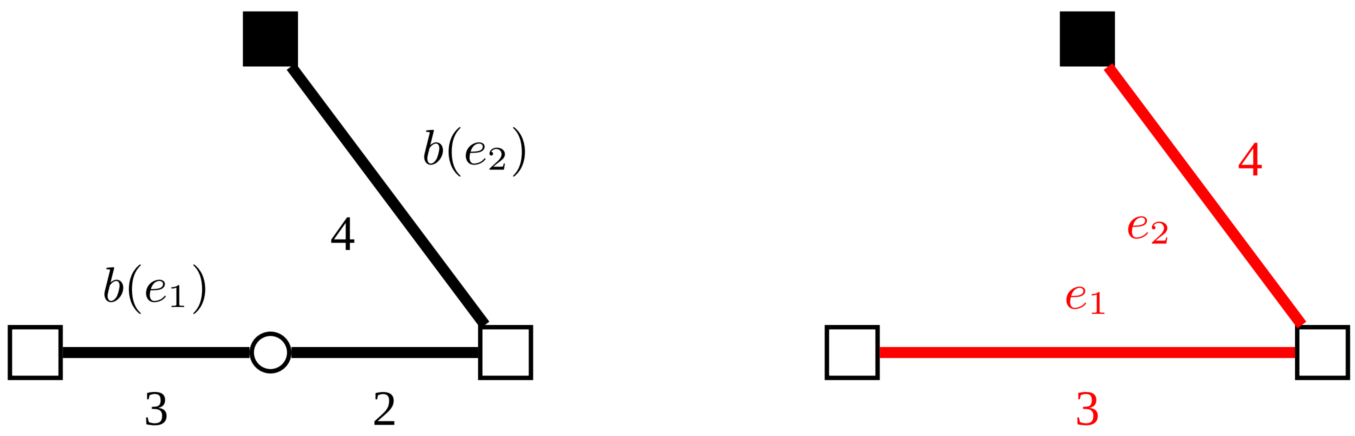


A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M})c(S^t)$

- $S' := S^t - \{b(e) \in S^t \mid e \in br_{Y^t,\mathbf{w}}(C^t)\}$ is a feasible Steiner tree at step $t+1$

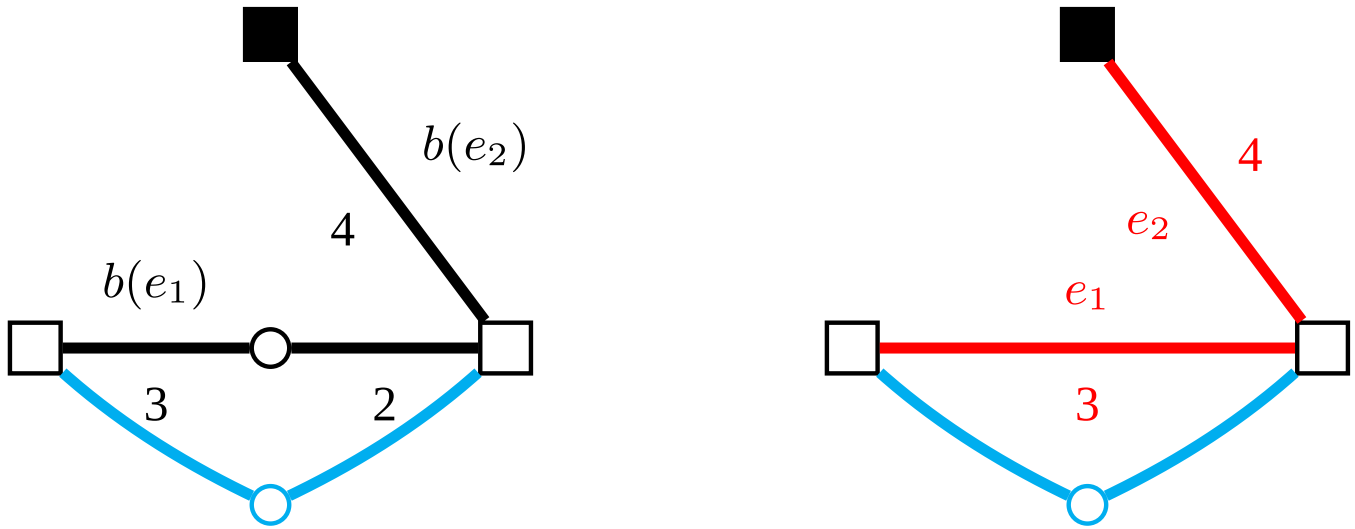


A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M})c(S^t)$

- $S' := S^t - \{b(e) \in S^t \mid e \in br_{Y^t,\mathbf{w}}(C^t)\}$ is a feasible Steiner tree at step $t+1$

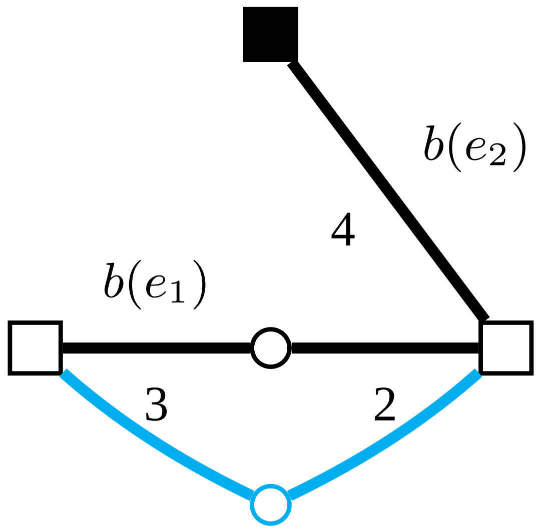


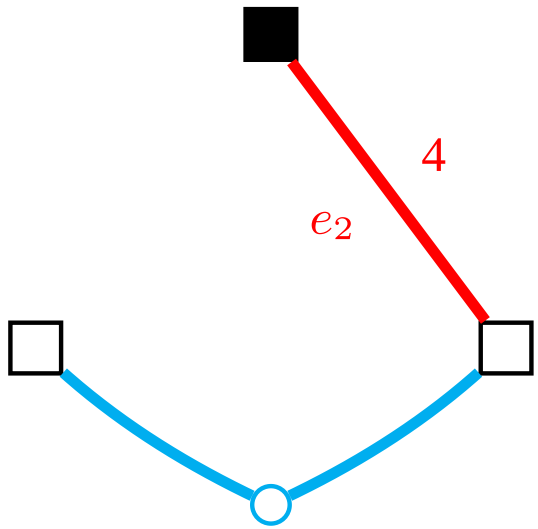
A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M}) c(S^t)$

- $S' := S^t - \{b(e) \in S^t \mid e \in br_{Y^t,\mathbf{w}}(C^t)\}$ is a feasible Steiner tree at step $t+1$



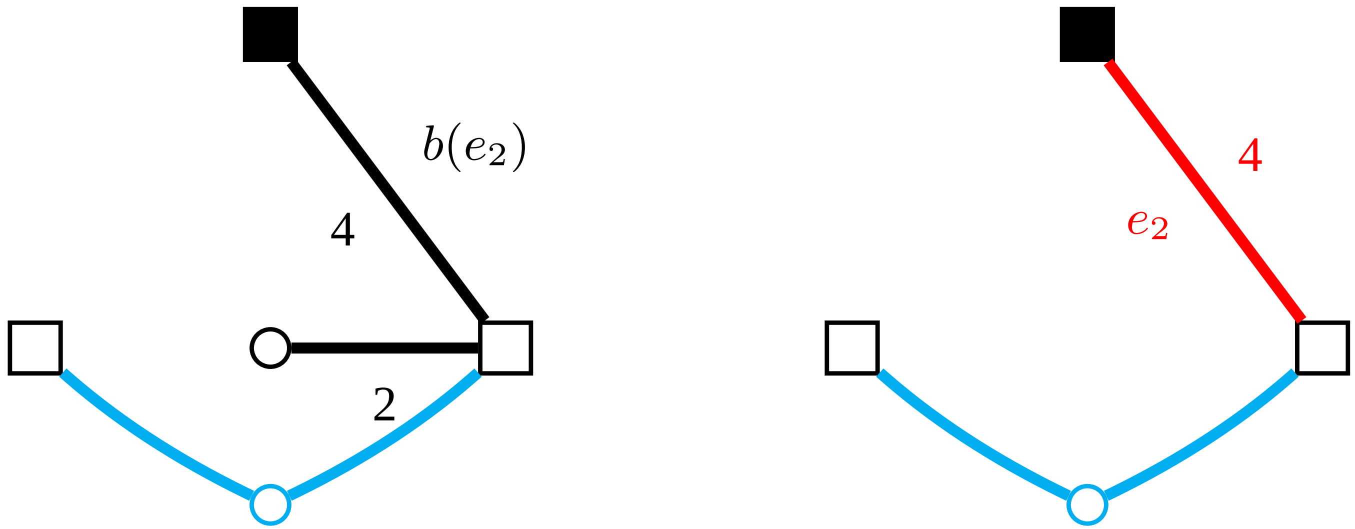


A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M})c(S^t)$

- S':=S^t-$\{b(e) \in S^t \mid e \in br_{Y^t,\mathbf{w}}(C^t)\}$ is a feasible Steiner tree at step $t+1$



A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M}) c(S^t)$

$$
\begin{aligned}
E[c(S^{t+1})] \leq E[c(S')] &= c(S^t) - E[c(\{b(e) \in S^t \mid e \in br_{Y^t,\mathbf{w}}(C^t)\})] \\
&= c(S^t) - E[br_{Y^t,\mathbf{w}}(C^t)] \\
&= c(S^t) - \frac{1}{M} \sum_C x_C^t br_{Y^t,\mathbf{w}}(C) \\
&\overset{\text{Bridge Lem}}{\leq} c(S^t) - \frac{1}{M} w(Y^t) \\
&= c(S^t) - \frac{1}{M} br_{S^t,\mathbf{c}}(R^t) \\
&\leq c(S^t) - \frac{1}{M} \frac{c(S^t)}{2}
\end{aligned}
$$

A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

Lem For any t, $E[c(S^{t+1})] \leq (1 - \frac{1}{\mathbf{2}M})c(S^t)$

$$\begin{aligned} E[c(S^{t+1})] \leq E[c(S')] &= c(S^t) - E[c(\{b(e) \in S^t \mid e \in br_{Y^t,\mathbf{w}}(C^t)\})] \\ &= c(S^t) - E[br_{Y^t,\mathbf{w}}(C^t)] \\ &= c(S^t) - \frac{1}{M}\sum_C x_C^t br_{Y^t,\mathbf{w}}(C) \\ &\overset{\text{Bridge Lem}}{\leq} c(S^t) - \frac{1}{M} w(Y^t) \\ &= c(S^t) - \frac{1}{M} br_{S^t,\mathbf{c}}(R^t) \\ &\leq c(S^t) - \frac{1}{M}\frac{c(S^t)}{2} \end{aligned}$$

Cor $E[c(S^t)] \leq (1 - \frac{1}{\mathbf{2}M})^{t-1} c(S^1) = (1 - \frac{1}{\mathbf{2}M})^{t-1} opt$

A Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (1.5 + \varepsilon)\, opt$

$$E[apx] = \sum_{t \geq 1} E[c(C^t)] \leq \sum_{t \geq 1} E[\sum_j \frac{x_j^t}{M} c(C_j^t)] \leq \frac{1+\varepsilon}{M} \sum_{t \geq 1} E[opt^{f,t}]$$

$$\leq \frac{1+\varepsilon}{M} \sum_{t=1}^{M \ln 4} E[c(S^t)] + \frac{1+\varepsilon}{M} \sum_{t > M \ln 4} E[c(T^t)]$$

$$\leq (\frac{1+\varepsilon}{M} opt) \cdot (\sum_{t=1}^{M \ln 4} (1 - \frac{1}{2M})^{t-1} + \sum_{t > M \ln 4} 2(1 - \frac{1}{M})^{t-1})$$

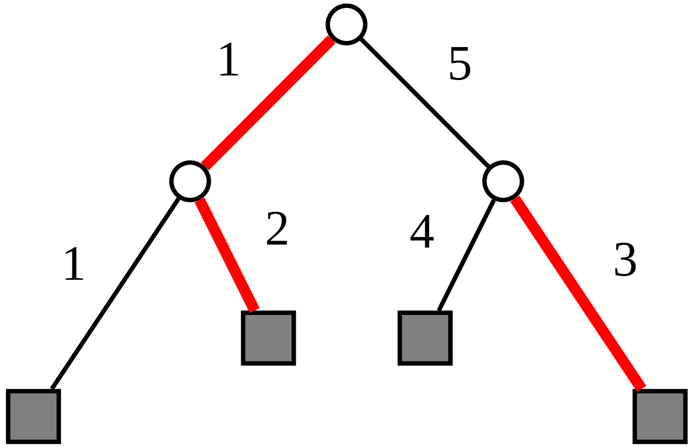
$$\leq (1+\varepsilon)(2 - 2e^{-\ln(4)/2} + 2e^{-\ln(4)}) \cdot opt$$

An Even Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$

An Even Better Bound

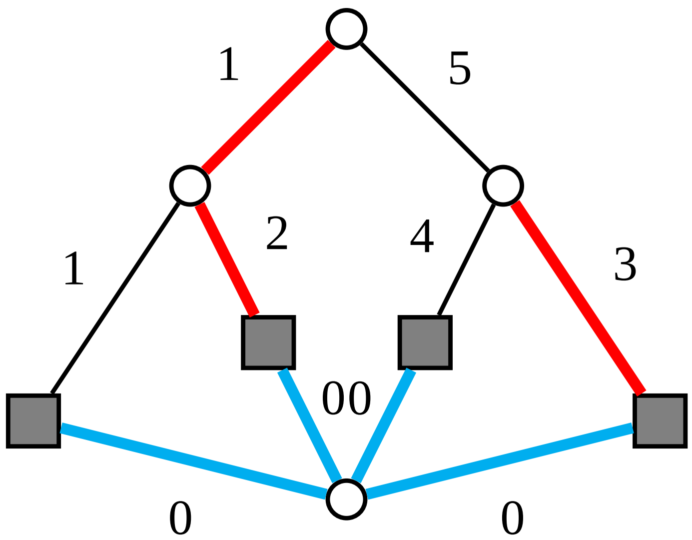
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- We define a random terminal spanning tree W (*witness* tree)

An Even Better Bound

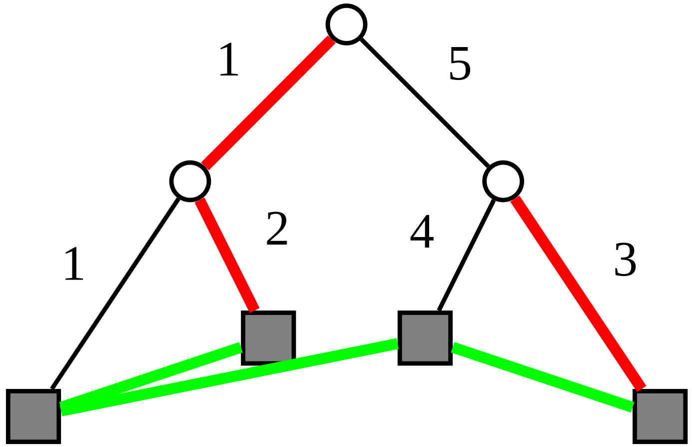
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- We define a random terminal spanning tree W (*witness* tree)

An Even Better Bound

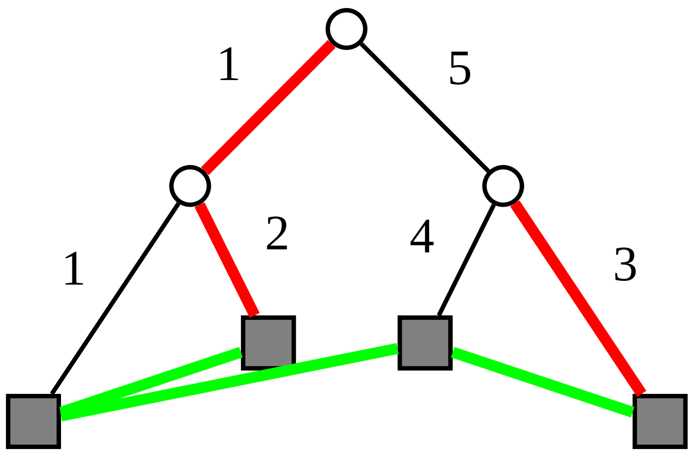
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- We define a random terminal spanning tree W (*witness* tree)

An Even Better Bound

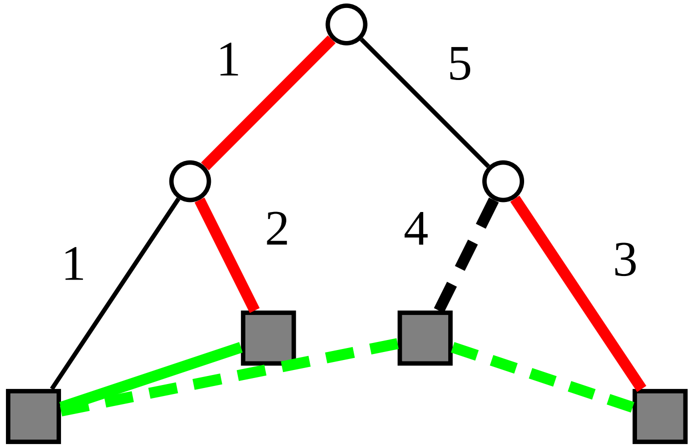
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- We associate to each e in the Steiner tree S the edges $W(e)$ of W such that the corresponding path in S contains e
- Observe that $|W(e)|$ is 1, 2... with probability $\frac{1}{2}$, $\frac{1}{4}$, ...

An Even Better Bound

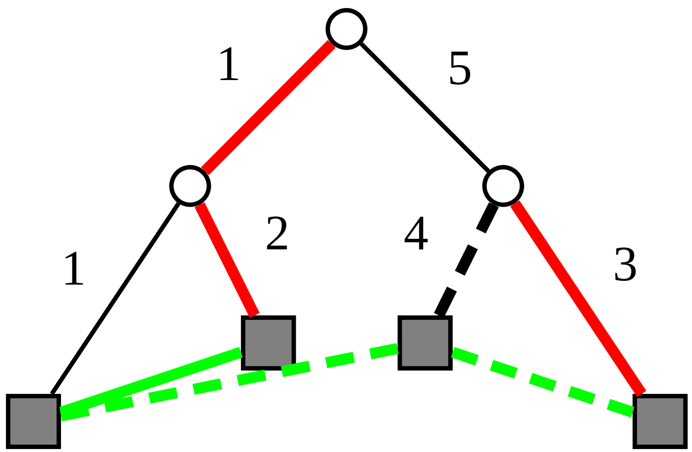
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- We associate to each e in the Steiner tree S the edges $W(e)$ of W such that the corresponding path in S contains e
- Observe that $|W(e)|$ is 1, 2… with probability $\frac{1}{2}$, $\frac{1}{4}$, …

An Even Better Bound

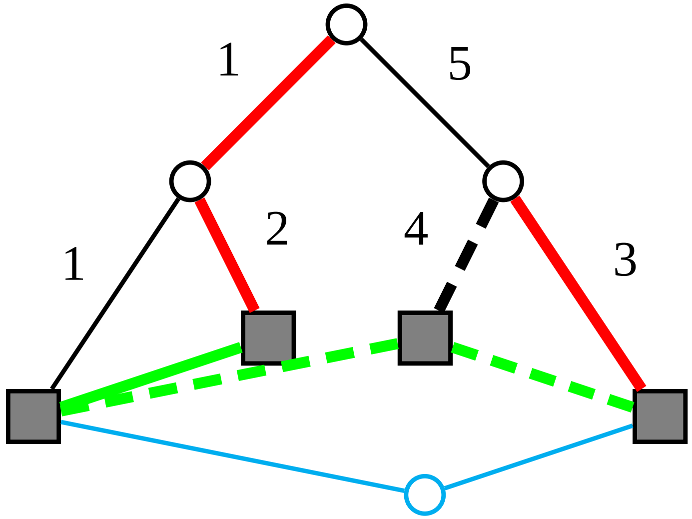
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- For any sampled component C^t, we delete from W a random set of bridges such that each edge of W is deleted with probability $\geq 1/M$ ($\Leftarrow$ Farkas' lemma+Bridge lemma)
- When $W(e)$ is deleted, we delete e from S

An Even Better Bound

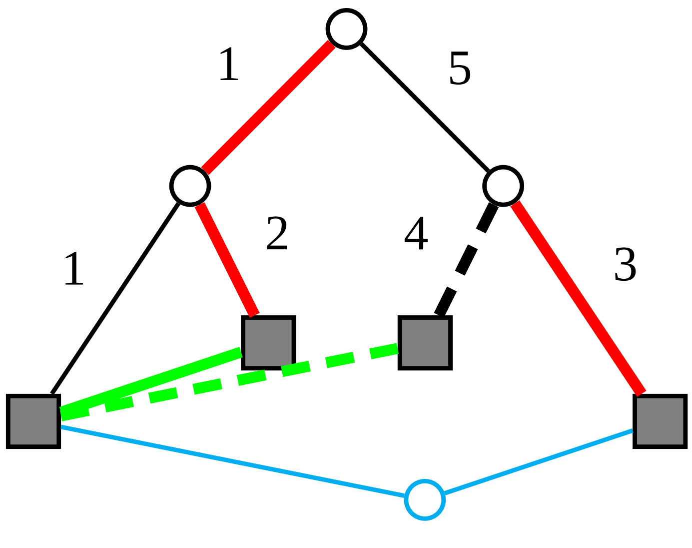
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- For any sampled component C^t, we delete from W a random set of bridges such that each edge of W is deleted with probability $\geq 1/M$ ($\Leftarrow$ Farkas' lemma+Bridge lemma)
- When $W(e)$ is deleted, we delete e from S

An Even Better Bound

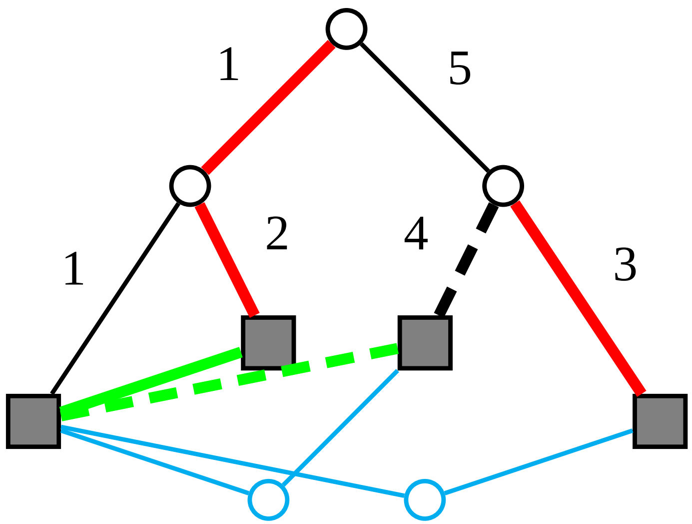
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- For any sampled component C^t, we delete from W a random set of bridges such that each edge of W is deleted with probability $\geq 1/M$ ($\Leftarrow$ Farkas' lemma+Bridge lemma)
- When $W(e)$ is deleted, we delete e from S

An Even Better Bound

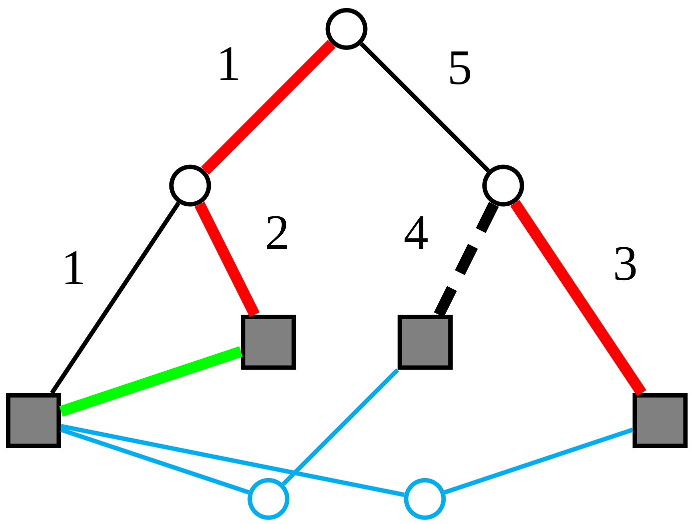
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- For any sampled component C^t, we delete from W a random set of bridges such that each edge of W is deleted with probability $\geq 1/M$ ($\Leftarrow$ Farkas' lemma+Bridge lemma)
- When $W(e)$ is deleted, we delete e from S

An Even Better Bound

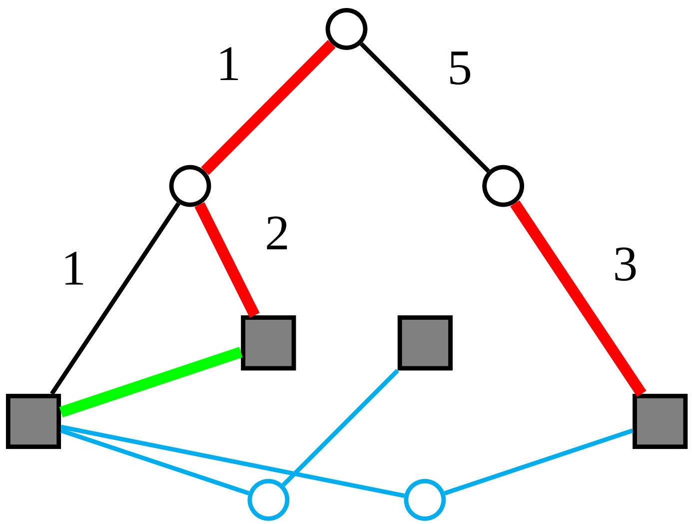
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- For any sampled component C^t, we delete from W a random set of bridges such that each edge of W is deleted with probability $\geq 1/M$ ($\Leftarrow$ Farkas' lemma+Bridge lemma)
- When $W(e)$ is deleted, we delete e from S

An Even Better Bound

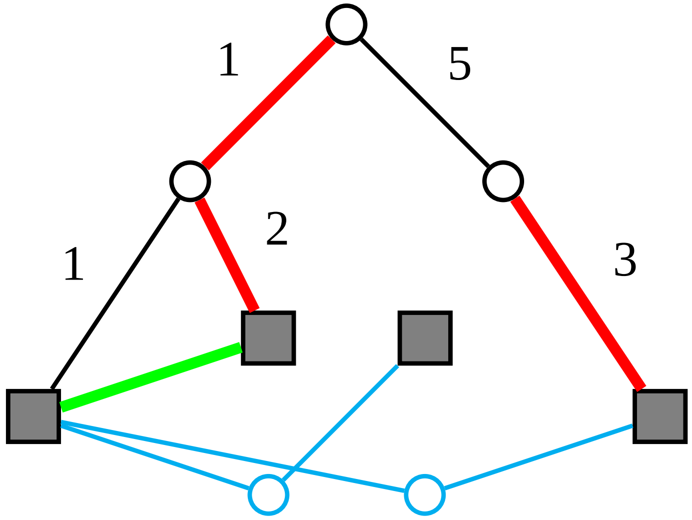
Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- For any sampled component C^t, we delete from W a random set of bridges such that each edge of W is deleted with probability $\geq 1/M$ ($\Leftarrow$ Farkas' lemma+Bridge lemma)
- When $W(e)$ is deleted, we delete e from S

An Even Better Bound

Thr Algorithm `IRR` computes a solution of expected cost $\leq (\ln 4 + \varepsilon)\, opt$



- Each $e \in S$ survives in expectation $M \cdot \ln 4$ rounds

Derandomization

Thr There is a $\ln 4 + \varepsilon$ deterministic approximation algorithm for Steiner tree

Derandomization

Thr There is a $\ln 4 + \varepsilon$ deterministic approximation algorithm for Steiner tree

- We define a phase-based randomized algorithm, with $1/\varepsilon^2$ phases s
- At each phase, we sample a proper number of components (without updating the LP)
- It is sufficient to guarantee that, at each phase:
  - ◇ Each component is sampled with probability $O(\varepsilon) x_C^s$
  - ◇ Each edge of the witness tree W is marked with probability $\Omega(\varepsilon)$
- This can be done by using only $O(\log n)$ random bits per phase

Open Problems

- The best 1.39 (and even 1.5) bound is w.r.t. the optimal integral solution. Does is hold w.r.t. the fractional one?
- Other applications of iterative randomized rounding?
  - Prize-collecting Steiner tree
  - k-MST
  - Single-Sink Rent-or-Buy
  - $\ldots$

THANKS!!!