Steiner Tree Approximation via Iterative Randomized Rounding

Jarosław Byrka, Fabrizio Grandoni,

Thomas Rothvoß, Laura Sanità

EPFL, Lausanne,
Switzerland

Lugano, 20.07.10

DISCRETE OPTIMIZATION

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

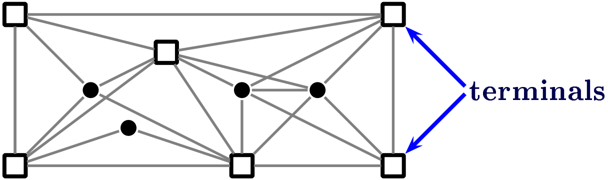
Steiner Tree

Given:

- undirected graph $G = (V, E)$
- cost $c : E \to \mathbb{Q}_+$
- terminals $R \subseteq V$

Find: Min-cost Steiner tree, spanning R.

$$OPT := \min\{c(S) \mid S \text{ spans } R\}$$



W.l.o.g.: c is metric.

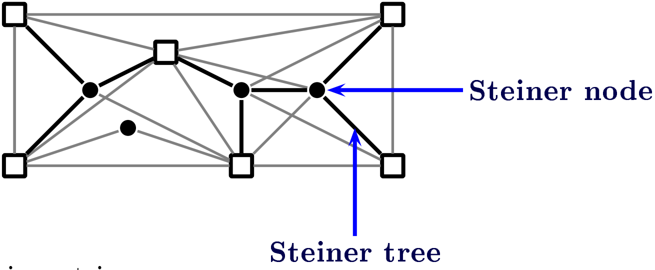
Steiner Tree

Given:

- undirected graph $G = (V, E)$
- cost $c : E \rightarrow \mathbb{Q}_+$
- terminals $R \subseteq V$

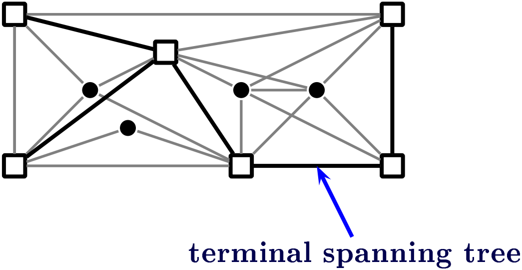
Find: Min-cost Steiner tree, spanning R.

$$OPT := \min\{c(S) \mid S \text{ spans } R\}$$



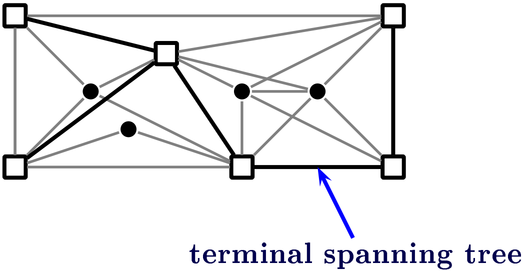
W.l.o.g.: c is metric.

Spanning tree



Min-cost terminal spanning tree (MST):

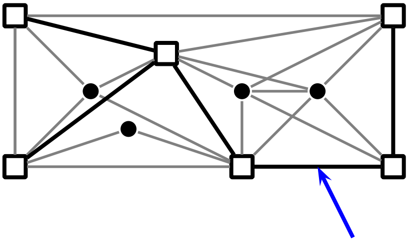
Spanning tree



Min-cost terminal spanning tree (MST):

- Can be computed in poly-time.

Spanning tree



Min-cost terminal spanning tree (MST):

- Can be computed in poly-time.
- Costs $\leq 2 \cdot OPT$.

Known results for Steiner tree:

Approximations:

- 2-apx (*minimum spanning tree heuristic*)
- 1.83-apx [Zelikovsky '93]
- 1.667-apx [Prömel & Steger '97]
- 1.644-apx [Karpinski & Zelikovsky '97]
- 1.598-apx [Hougardy & Prömel '99]
- 1.55-apx [Robins & Zelikovsky '00]

Known results for Steiner tree:

Approximations:

- 2-apx (*minimum spanning tree heuristic*)
- 1.83-apx [Zelikovsky '93]
- 1.667-apx [Prömel & Steger '97]
- 1.644-apx [Karpinski & Zelikovsky '97]
- 1.598-apx [Hougardy & Prömel '99]
- 1.55-apx [Robins & Zelikovsky '00]

Hardness:

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

Our results:

Theorem

There is a polynomial time 1.39-approximation.

- LP-based! *(Directed-Component Cut Relaxation)*
- Algorithmic framework: *Iterative Randomized Rounding*

Our results:

Theorem

There is a polynomial time 1.39-approximation.

- LP-based! *(Directed-Component Cut Relaxation)*
- Algorithmic framework: *Iterative Randomized Rounding*
- Here: Simpler $(1.5 + \varepsilon)$-apx

Our results:

Theorem

There is a polynomial time 1.39-approximation.

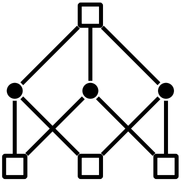
- LP-based! *(Directed-Component Cut Relaxation)*
- Algorithmic framework: *Iterative Randomized Rounding*
- Here: Simpler $(1.5 + \varepsilon)$-apx

Theorem

The Directed-Component Cut Relaxation has an integrality gap of at most 1.55.

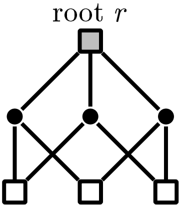
- First < 2 bound for *any* LP-relaxation.

Bi-directed cut relaxation



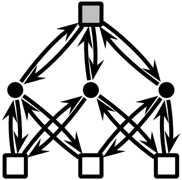
Bi-directed cut relaxation

- Pick a **root** $r \in R$



Bi-directed cut relaxation

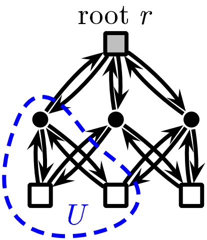
- Pick a **root** $r \in R$
- Bi-direct edges



Bi-directed cut relaxation

- Pick a **root** $r \in R$
- Bi-direct edges

$$
\begin{aligned}
\min \sum_{e \in E} c(e) z_e & \qquad \text{(BCR)} \\
\sum_{e \in \delta^+(U)} z_e \geq 1 & \qquad \forall U \subseteq V \setminus \{r\} : U \cap R \neq \emptyset \\
z_e \geq 0 & \qquad \forall e \in E.
\end{aligned}
$$



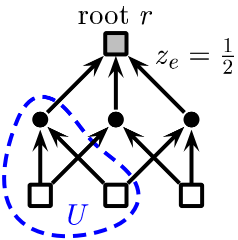
Bi-directed cut relaxation

- Pick a **root** $r \in R$
- Bi-direct edges

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

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

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



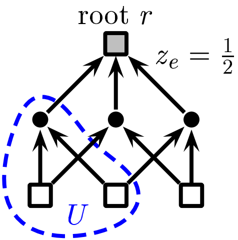
Bi-directed cut relaxation

- Pick a **root** $r \in R$
- Bi-direct edges

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

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

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



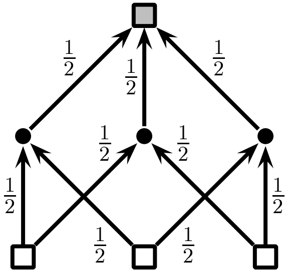
Theorem (Edmonds '67)

$R = V \Rightarrow BCR$ integral

- Integrality gap $\leq 4/3$ for quasi-bipartite graphs [Chakrabarty, Devanur, Vazirani '08]
- Integrality gap $\in [1.16, 2]$

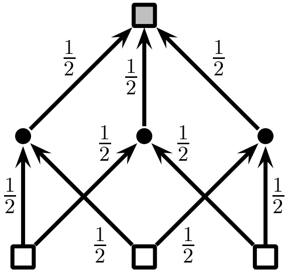
How to exploit BCR?

- Is there always an edge e with $x_e \geq 1/2$?



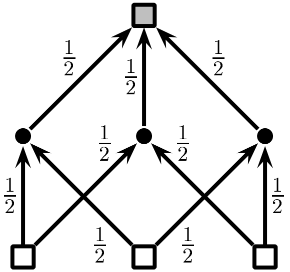
How to exploit BCR?

- Is there always an edge e with $x_e \geq 1/2$? **No!**



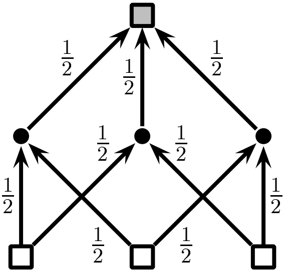
How to exploit BCR?

- Is there always an edge e with $x_e \geq 1/2$? **No!**
- Primal dual algorithm?



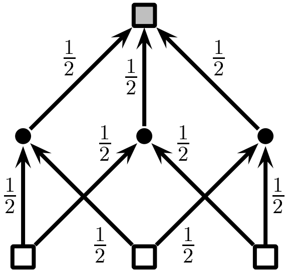
How to exploit BCR?

- Is there always an edge e with $x_e \geq 1/2$? **No!**
- Primal dual algorithm?
  [CDV '08] **"dual solution lacks structure"**



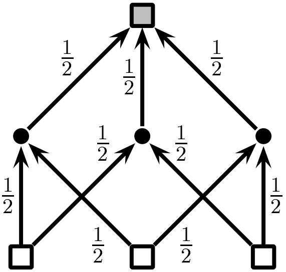
How to exploit BCR?

- Is there always an edge e with $x_e \geq 1/2$? **No!**
- Primal dual algorithm?
  [CDV '08] **"dual solution lacks structure"**
- Randomized rounding of a single edge?



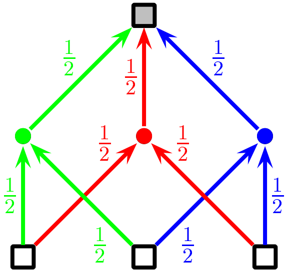
How to exploit BCR?

- Is there always an edge e with $x_e \geq 1/2$? **No!**
- Primal dual algorithm?
  [CDV '08] **"dual solution lacks structure"**
- Randomized rounding of a single edge?
  [Rajagopalan, Vazirani '99] **Doesn't work!**



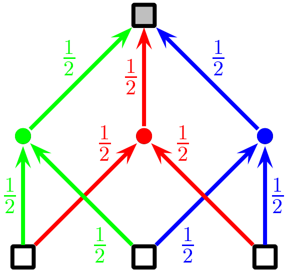
How to exploit BCR?

- Is there always an edge e with $x_e \geq 1/2$? **No!**
- Primal dual algorithm?
  [CDV '08] **"dual solution lacks structure"**
- Randomized rounding of a single edge?
  [Rajagopalan, Vazirani '99] **Doesn't work!**
- Can a solution be decomposed into components?

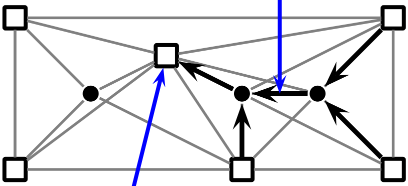


How to exploit BCR?

- Is there always an edge e with $x_e \geq 1/2$? **No!**
- Primal dual algorithm?
  [CDV '08] **"dual solution lacks structure"**
- Randomized rounding of a single edge?
  [Rajagopalan, Vazirani '99] **Doesn't work!**
- Can a solution be decomposed into components? **No!**



Components



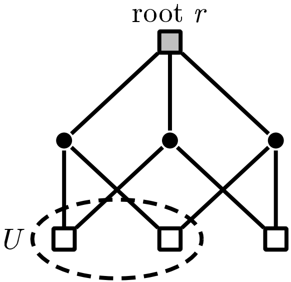
- $\mathbf{C}$ = set of directed components

Directed component cut relaxation

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

$$\sum_{\substack{C \in \mathbf{C} : R(C) \cap U \neq \emptyset, \\ \text{sink}(C) \notin U}} x_C \geq 1 \quad \forall \emptyset \subset U \subseteq R \setminus \{r\}$$

$$x_C \geq 0 \quad \forall C \in \mathbf{C}$$

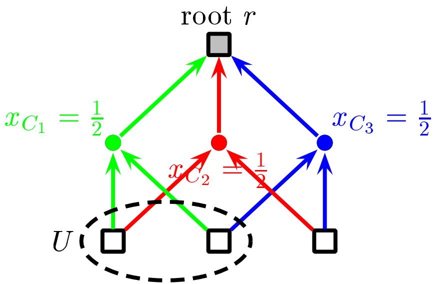


Directed component cut relaxation

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

$$\sum_{\substack{C \in \mathbf{C} : R(C) \cap U \neq \emptyset, \\ \text{sink}(C) \notin U}} x_C \geq 1 \quad \forall \emptyset \subset U \subseteq R \setminus \{r\}$$

$$x_C \geq 0 \quad \forall C \in \mathbf{C}$$



Directed component cut relaxation

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

$$\sum_{\substack{C \in \mathbf{C} : R(C) \cap U \neq \emptyset, \\ \text{sink}(C) \notin U}} x_C \geq 1 \quad \forall \emptyset \subset U \subseteq R \setminus \{r\}$$

$$x_C \geq 0 \quad \forall C \in \mathbf{C}$$

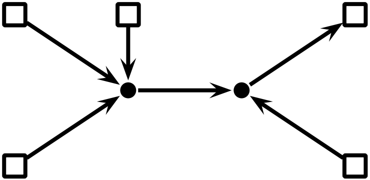
Properties:

- Number of variables: exponential
- Number of constraints: exponential
- Approximable within $1 + \varepsilon$ (we ignore the ε here).

Solvability of the LP

Lemma

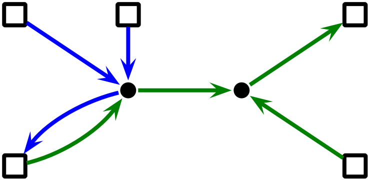
For any $\varepsilon > 0$, a solution x of cost $\leq (1+\varepsilon)OPT_f$ can be computed in polynomial time.



Solvability of the LP

Lemma

For any $\varepsilon > 0$, a solution x of cost $\leq (1+\varepsilon)OPT_f$ can be computed in polynomial time.



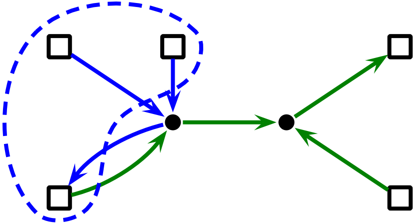
- Use only components of size $2^{\lceil 1/\varepsilon \rceil} = O(1)$
  [Borchers & Du '97]: Increases cost by $\leq 1 + \varepsilon$
  $\rightarrow$ # variables polynomial

Solvability of the LP

Lemma

For any $\varepsilon > 0$, a solution x of cost $\leq (1+\varepsilon)OPT_f$ can be computed in polynomial time.

$\leq 2^{\lceil 1/\varepsilon \rceil}$



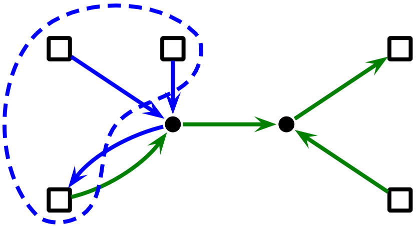
- Use only components of size $2^{\lceil 1/\varepsilon \rceil} = O(1)$
  [Borchers & Du '97]: Increases cost by $\leq 1 + \varepsilon$
  $\rightarrow$ # variables polynomial

Solvability of the LP

Lemma

For any $\varepsilon > 0$, a solution x of cost $\leq (1+\varepsilon)OPT_f$ can be computed in polynomial time.

$\leq 2^{\lceil 1/\varepsilon \rceil}$



- Use only components of size $2^{\lceil 1/\varepsilon \rceil} = O(1)$
  [Borchers & Du '97]: Increases cost by $\leq 1 + \varepsilon$
  $\rightarrow$ # variables polynomial
- Compact flow formulation $\rightarrow$ # constraints polynomial
  (or solve with ellipsoid method). □

An iterative randomized rounding algo

(1) FOR $t = 1, \ldots, \infty$ DO

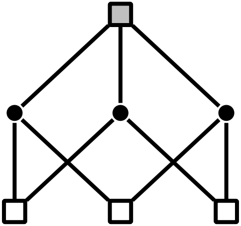
(2) Compute opt. LP solution x

(3) Sample a component:

$$\Pr[\text{sample } C] = \frac{x_C}{\mathbf{1}^T x}$$

and contract it.

(4) IF all terminals connected THEN output sampled components



An iterative randomized rounding algo

(1) FOR $t = 1, \dots, \infty$ DO

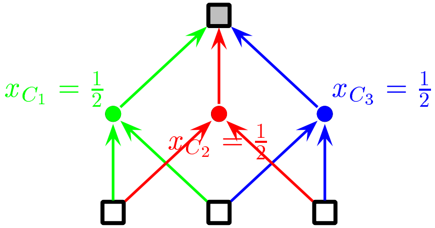
(2) Compute opt. LP solution x

(3) Sample a component:

$$\Pr[\text{sample } C] = \frac{x_C}{\mathbf{1}^T x}$$

and contract it.

(4) IF all terminals connected THEN output sampled components



An iterative randomized rounding algo

(1) FOR $t = 1, \ldots, \infty$ DO

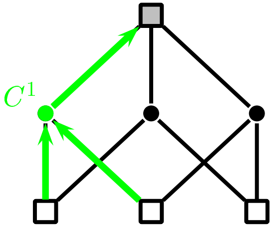
(2) Compute opt. LP solution x

(3) Sample a component:

$$\Pr[\text{sample } C] = \frac{x_C}{\mathbf{1}^T x}$$

and contract it.

(4) IF all terminals connected THEN output sampled components



An iterative randomized rounding algo

(1) FOR $t = 1, \dots, \infty$ DO

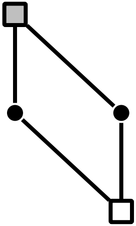
(2) Compute opt. LP solution x

(3) Sample a component:

$$\Pr[\text{sample } C] = \frac{x_C}{\mathbf{1}^T x}$$

and contract it.

(4) IF all terminals connected THEN output sampled components



An iterative randomized rounding algo

(1) FOR $t = 1, \ldots, \infty$ DO

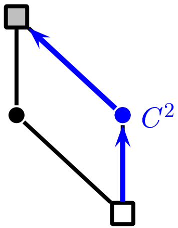
  (2) Compute opt. LP solution x

  (3) Sample a component:

$$\Pr[\text{sample } C] = \frac{x_C}{\mathbf{1}^T x}$$

  and contract it.

  (4) IF all terminals connected THEN output sampled components



An iterative randomized rounding algo

(1) FOR $t = 1, \ldots, \infty$ DO

(2) Compute opt. LP solution x

(3) Sample a component:

$$\Pr[\text{sample } C] = \frac{x_C}{\mathbf{1}^T x}$$

and contract it.

(4) IF all terminals connected THEN output sampled components

An iterative randomized rounding algo

(1) FOR $t = 1, \ldots, \infty$ DO

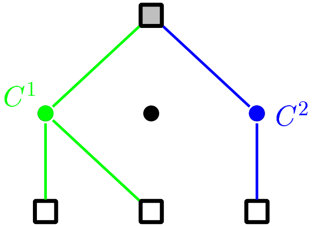
  (2) Compute opt. LP solution x

  (3) Sample a component:

$$\Pr[\text{sample } C] = \frac{x_C}{\mathbf{1}^T x}$$

  and contract it.

  (4) IF all terminals connected THEN output sampled components



An iterative randomized rounding algo

(1) FOR $t = 1, \ldots, \infty$ DO

  (2) Compute opt. LP solution x

  (3) Sample a component:

$$\Pr[\text{sample } C] = \frac{x_C}{\mathbf{1}^T x}$$

  and contract it.

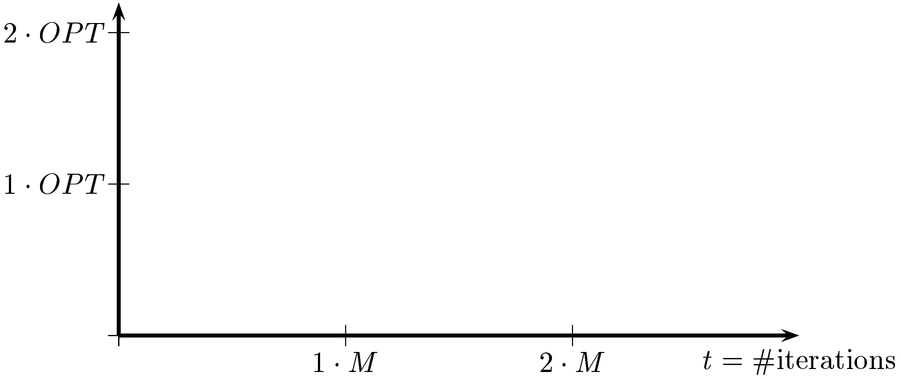
  (4) IF all terminals connected THEN output sampled components

- W.l.o.g. $M := \mathbf{1}^T x$ invariant

Roadmap

- In one iteration t:

  $$E[c(\text{comp. sampled in it. } t)] = \sum_C \frac{x_C}{M} \cdot c(C) \leq \frac{1}{M} \cdot OPT \text{ in it } t$$



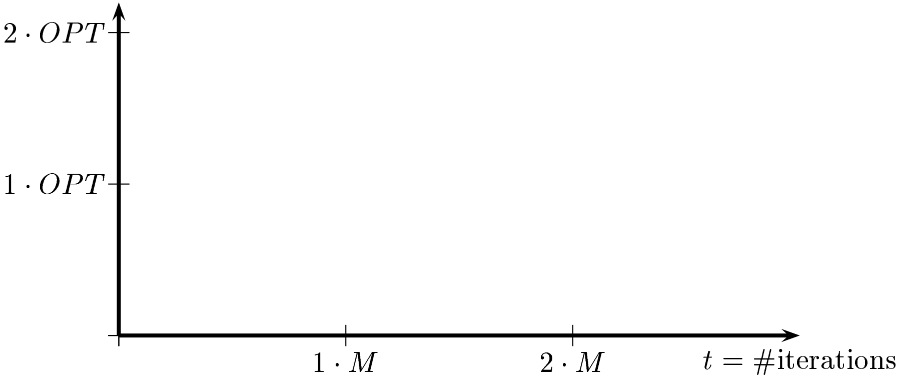
Roadmap

- In one iteration t:

$$E[c(\text{comp. sampled in it. } t)] = \sum_C \frac{x_C}{M} \cdot c(C) \leq \frac{1}{M} \cdot OPT \text{ in it } t$$

- In total

$$\sum_{t \geq 1} E[c(\text{comp. sampled in it. } t)] \leq \sum_{t \geq 1} \frac{1}{M} \cdot E[OPT \text{ in iteration } t]$$



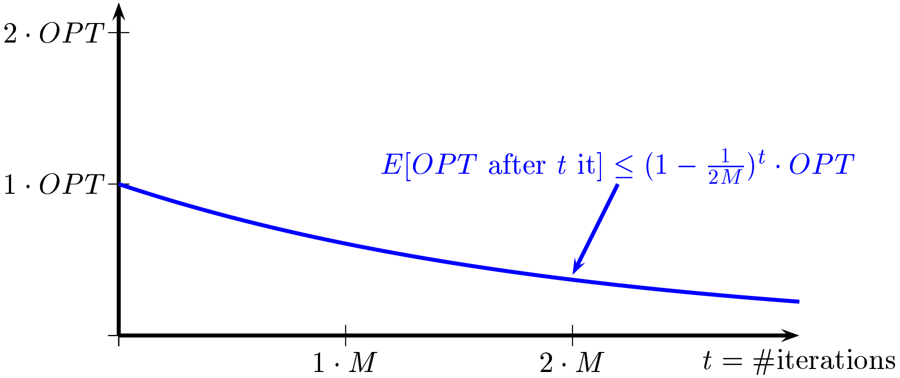
Roadmap

- In one iteration t:

  $$E[c(\text{comp. sampled in it. } t)] = \sum_{C} \frac{x_C}{M} \cdot c(C) \leq \frac{1}{M} \cdot OPT \text{ in it } t$$

- In total

  $$\sum_{t \geq 1} E[c(\text{comp. sampled in it. } t)] \leq \sum_{t \geq 1} \frac{1}{M} \cdot E[OPT \text{ in iteration } t]$$



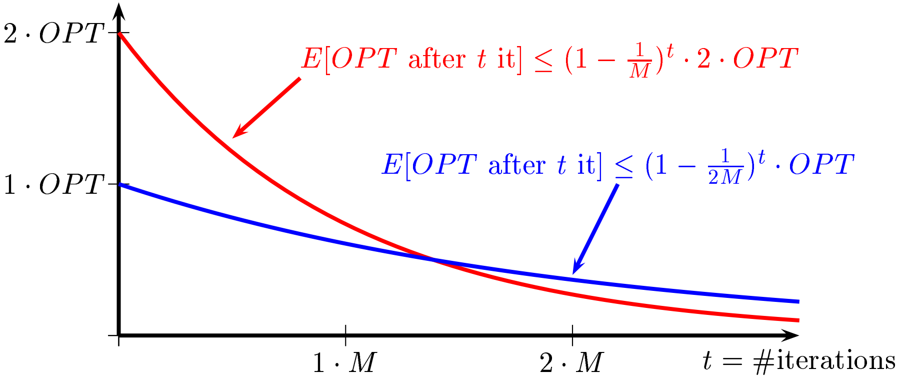
Roadmap

- In one iteration t:

$$E[c(\text{comp. sampled in it. } t)] = \sum_C \frac{x_C}{M} \cdot c(C) \leq \frac{1}{M} \cdot OPT \text{ in it } t$$

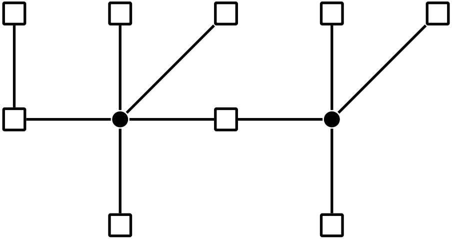
- In total

$$\sum_{t \geq 1} E[c(\text{comp. sampled in it. } t)] \leq \sum_{t \geq 1} \frac{1}{M} \cdot E[OPT \text{ in iteration } t]$$



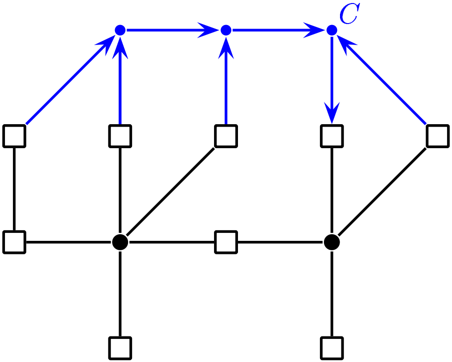
Bridges

- Let S be Steiner tree



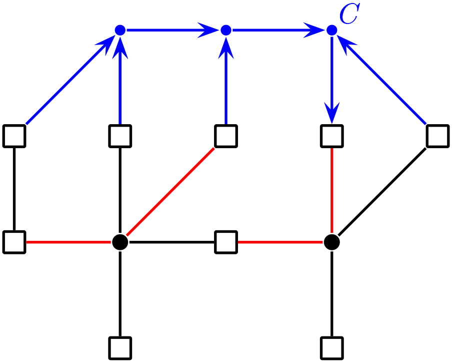
Bridges

- Let S be Steiner tree, C a component



Bridges

- Let S be Steiner tree, C a component

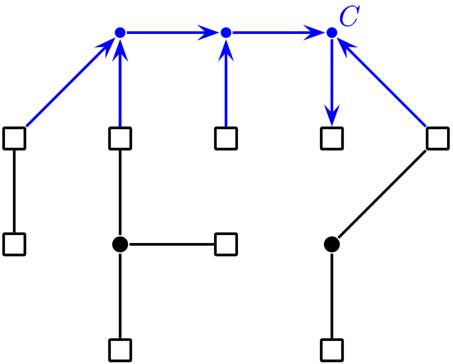


- **Bridges:**

$$Br_S(C) = \text{argmax}\{c(B) \mid B \subseteq S,\ S \backslash B \cup C \text{ is connected}\}$$

Bridges

- Let S be Steiner tree, C a component



- **Bridges:**

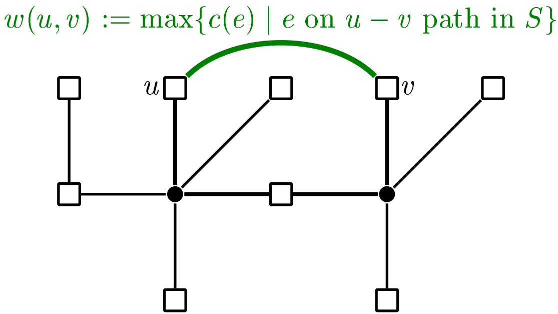
$$Br_S(C) = \text{argmax}\{c(B) \mid B \subseteq S,\ S \backslash B \cup C \text{ is connected}\}$$

The saving function

Definition

For a Steiner tree S, the **saving function** $w : E \to \mathbb{Q}_+$ is defined as

$$w(u, v) := \max\{c(e) \mid e \text{ on } u - v \text{ path in } S\}.$$

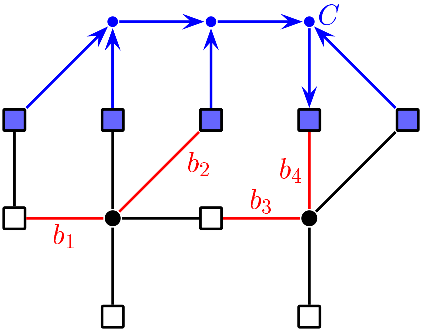


A saving lemma

Lemma

For any Steiner tree S and component C, $\exists$ saving tree spanning the terminals of C with

$$c(Br_S(C)) = w(\textit{saving tree})$$



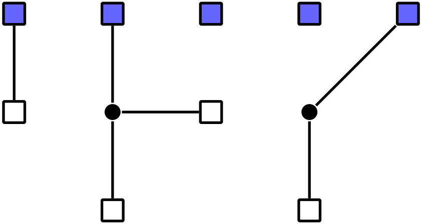
A saving lemma

Lemma

For any Steiner tree S and component C, $\exists$ saving tree spanning the terminals of C with

$$c(Br_S(C)) = w(\textit{saving tree})$$

- Consider forest $S \backslash Br_S(C)$



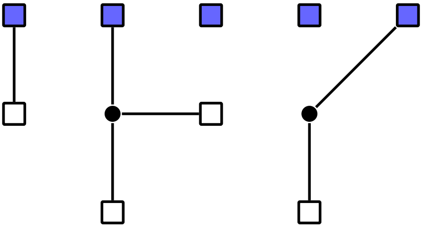
A saving lemma

Lemma

For any Steiner tree S and component C, $\exists$ saving tree spanning the terminals of C with

$$c(Br_S(C)) = w(\textit{saving tree})$$

- Consider forest $S \backslash Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\Leftrightarrow$ b_i connects trees of u and v
- Then $w(e_i) = c(b_i)$.



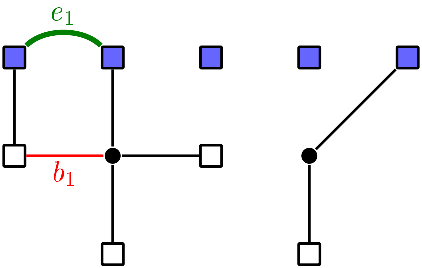
A saving lemma

Lemma

For any Steiner tree S and component C, $\exists$ saving tree spanning the terminals of C with

$$c(Br_S(C)) = w(\textit{saving tree})$$

- Consider forest $S \backslash Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\Leftrightarrow$ b_i connects trees of u and v
- Then $w(e_i) = c(b_i)$.



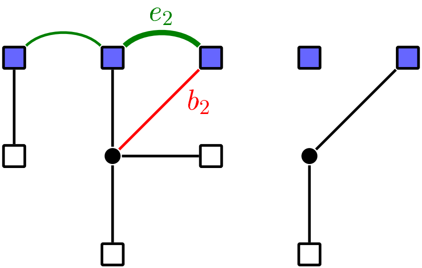
A saving lemma

Lemma

For any Steiner tree S and component C, $\exists$ saving tree spanning the terminals of C with

$$c(Br_S(C)) = w(\textit{saving tree})$$

- Consider forest $S \backslash Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\Leftrightarrow$ b_i connects trees of u and v
- Then $w(e_i) = c(b_i)$.



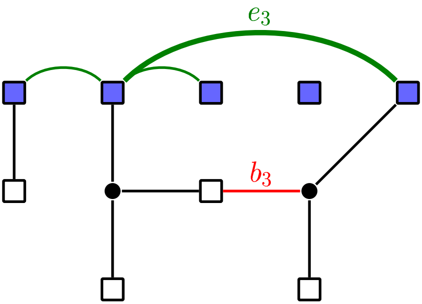
A saving lemma

Lemma

For any Steiner tree S and component C, $\exists$ saving tree spanning the terminals of C with

$$c(Br_S(C)) = w(\textit{saving tree})$$

- Consider forest $S \backslash Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\Leftrightarrow$ b_i connects trees of u and v
- Then $w(e_i) = c(b_i)$.



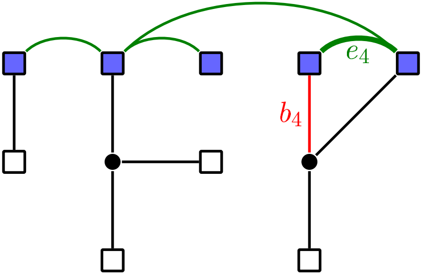
A saving lemma

Lemma

For any Steiner tree S and component C, $\exists$ saving tree spanning the terminals of C with

$$c(Br_S(C)) = w(\textit{saving tree})$$

- Consider forest $S \backslash Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\Leftrightarrow$ b_i connects trees of u and v
- Then $w(e_i) = c(b_i)$.

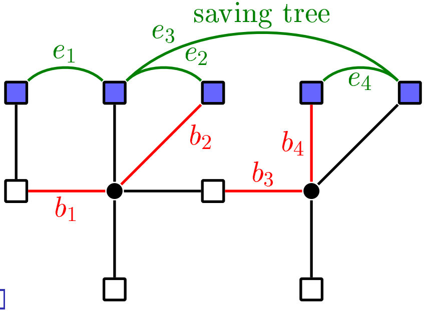


A saving lemma

Lemma

For any Steiner tree S and component C, $\exists$ saving tree spanning the terminals of C with

$$c(Br_S(C)) = w(\textit{saving tree})$$



- Consider forest $S \backslash Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\Leftrightarrow$ b_i connects trees of u and v
- Then $w(e_i) = c(b_i)$. $\square$

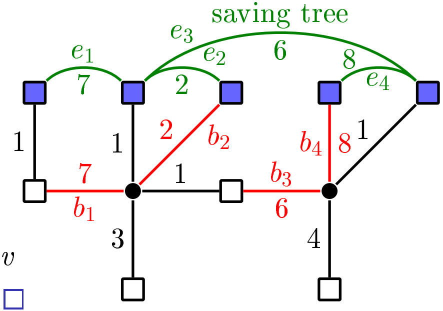
A saving lemma

Lemma

For any Steiner tree S and component C, $\exists$ saving tree spanning the terminals of C with

$$c(Br_S(C)) = w(\textit{saving tree})$$

- Consider forest $S \backslash Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\Leftrightarrow$ b_i connects trees of u and v
- Then $w(e_i) = c(b_i)$. □

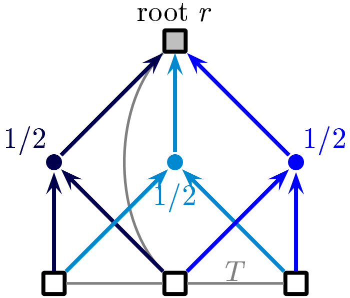


The Bridge Lemma (1)

Lemma (Bridge Lemma)

For T terminal spanning tree, x LP solution:

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) \geq c(T)$$



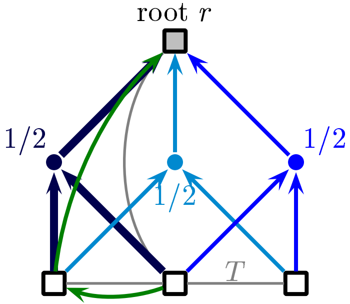
The Bridge Lemma (1)

Lemma (Bridge Lemma)

For T terminal spanning tree, x LP solution:

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) \geq c(T)$$

- For any C, $\exists$ saving tree: $c(Br_T(C)) = w(\text{saving tree of } C)$



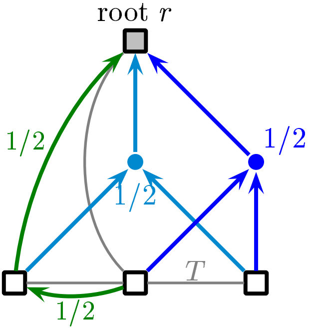
The Bridge Lemma (1)

Lemma (Bridge Lemma)

For T terminal spanning tree, x LP solution:

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) \geq c(T)$$

- For any C, $\exists$ saving tree: $c(Br_T(C)) = w(\text{saving tree of } C)$
- Transfer capacity from component to its saving tree $\rightarrow$ capacity reservation $y : E \rightarrow \mathbb{Q}_+$



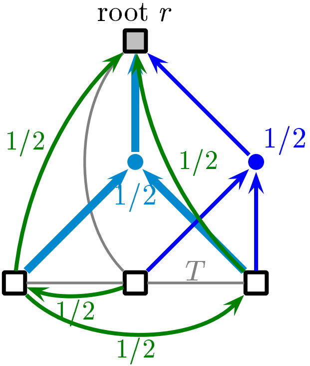
The Bridge Lemma (1)

Lemma (Bridge Lemma)

For T terminal spanning tree, x LP solution:

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) \geq c(T)$$

- For any C, $\exists$ saving tree: $c(Br_T(C)) = w(\text{saving tree of } C)$
- Transfer capacity from component to its saving tree $\rightarrow$ capacity reservation $y : E \rightarrow \mathbb{Q}_+$



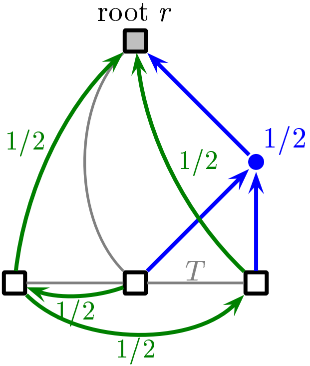
The Bridge Lemma (1)

Lemma (Bridge Lemma)

For T terminal spanning tree, x LP solution:

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) \geq c(T)$$

- For any C, $\exists$ saving tree: $c(Br_T(C)) = w(\text{saving tree of } C)$
- Transfer capacity from component to its saving tree $\rightarrow$ capacity reservation $y : E \rightarrow \mathbb{Q}_+$



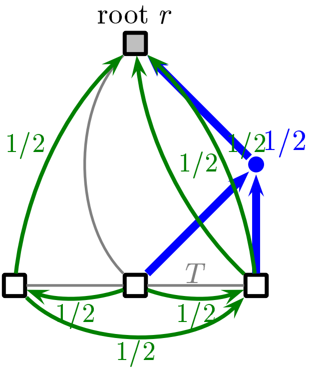
The Bridge Lemma (1)

Lemma (Bridge Lemma)

For T terminal spanning tree, x LP solution:

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) \geq c(T)$$

- For any C, $\exists$ saving tree: $c(Br_T(C)) = w(\text{saving tree of } C)$
- Transfer capacity from component to its saving tree $\rightarrow$ capacity reservation $y : E \rightarrow \mathbb{Q}_+$



The Bridge Lemma (1)

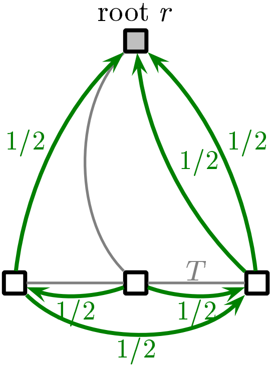
Lemma (Bridge Lemma)

For T terminal spanning tree, x LP solution:

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) \geq c(T)$$

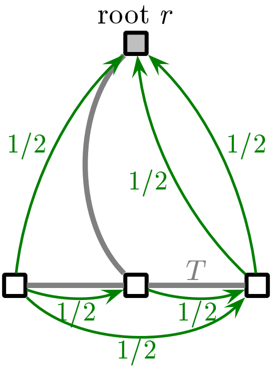
- For any C, $\exists$ saving tree: $c(Br_T(C)) = w(\text{saving tree of } C)$
- Transfer capacity from component to its saving tree $\rightarrow$ capacity reservation $y : E \rightarrow \mathbb{Q}_+$

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) = w(y)$$



The Bridge Lemma (2)

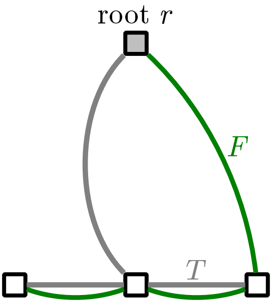
$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) = w(y)$$



The Bridge Lemma (2)

Edmonds Thm

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) = w(y) \geq w(F)$$

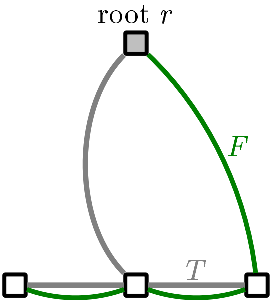


The Bridge Lemma (2)

$$\sum_{C \in \mathbf{C}} x_C \cdot c(Br_T(C)) = w(y) \geq w(F) \geq c(T)$$

Edmonds Thm

Cycle rule



A 1st bound on OPT

Lemma

$E[OPT \text{ after it. } t] \leq \left(1 - \frac{1}{M}\right)^t \cdot 2 \cdot OPT.$

A 1st bound on OPT

Lemma

$E[OPT \text{ after it. } t] \leq \left(1 - \frac{1}{M}\right)^t \cdot 2 \cdot OPT.$

- Initially $c(\text{MST}) \leq 2 \cdot OPT$

A 1st bound on OPT

Lemma

$E[OPT \text{ after it. } t] \leq \left(1 - \frac{1}{M}\right)^t \cdot 2 \cdot OPT.$

- Initially $c(\text{MST}) \leq 2 \cdot OPT$
- In any iteration

$$E[c(\text{new MST})] \quad \leq \quad c(\text{old MST}) - E[c(Br_{\text{old MST}}(C))]$$

A 1st bound on OPT

Lemma

$E[OPT\ \textit{after it. } t] \leq \left(1 - \frac{1}{M}\right)^t \cdot 2 \cdot OPT.$

- Initially $c(\text{MST}) \leq 2 \cdot OPT$
- In any iteration

$$\begin{aligned} E[c(\text{new MST})] &\leq c(\text{old MST}) - E[c(Br_{\text{old MST}}(C))] \\ &= c(\text{old MST}) - \frac{1}{M} \sum_{C \in \mathbf{C}} x_C \cdot c(Br_{\text{old MST}}(C)) \end{aligned}$$

A 1st bound on OPT

Lemma

$E[OPT \textit{ after it. } t] \leq \left(1 - \frac{1}{M}\right)^t \cdot 2 \cdot OPT.$

- Initially $c(\text{MST}) \leq 2 \cdot OPT$
- In any iteration

$$\begin{aligned} E[c(\text{new MST})] &\leq c(\text{old MST}) - E[c(Br_{\text{old MST}}(C))] \\ &= c(\text{old MST}) - \frac{1}{M} \underbrace{\sum_{C \in \mathbf{C}} x_C \cdot c(Br_{\text{old MST}}(C))}_{\geq c(\text{old MST})} \end{aligned}$$

A 1st bound on OPT

Lemma

$E[OPT\ \textit{after it. } t] \leq \left(1 - \frac{1}{M}\right)^t \cdot 2 \cdot OPT.$

- Initially $c(\text{MST}) \leq 2 \cdot OPT$
- In any iteration

$$
\begin{aligned}
E[c(\text{new MST})] &\leq c(\text{old MST}) - E[c(Br_{\text{old MST}}(C))] \\
&= c(\text{old MST}) - \frac{1}{M} \underbrace{\sum_{C \in \mathbf{C}} x_C \cdot c(Br_{\text{old MST}}(C))}_{\geq c(\text{old MST})} \\
&\leq \left(1 - \frac{1}{M}\right) \cdot c(\text{old MST}) \quad \square
\end{aligned}
$$

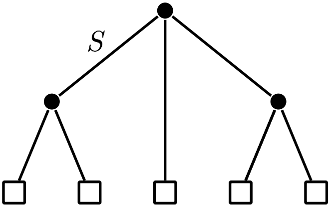
A 2nd bound on OPT

Theorem

In any iteration

$$E[new\ OPT] \leq \left(1 - \frac{1}{2M}\right) \cdot old\ OPT$$

- Let S be opt. Steiner tree



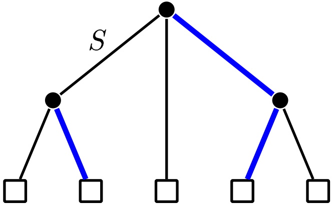
A 2nd bound on OPT

Theorem

In any iteration

$$E[new\ OPT] \leq \left(1 - \frac{1}{2M}\right) \cdot old\ OPT$$

- Let S be opt. Steiner tree
- From each inner node in S: Contract the cheapest edge going to a child



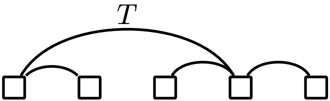
A 2nd bound on OPT

Theorem

In any iteration

$$E[new\ OPT] \leq \left(1 - \frac{1}{2M}\right) \cdot old\ OPT$$

- Let S be opt. Steiner tree
- From each inner node in S: Contract the cheapest edge going to a child
- A terminal spanning tree T remains



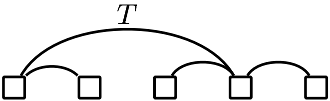
A 2nd bound on OPT

Theorem

In any iteration

$$E[\textit{new } OPT] \leq \left(1 - \frac{1}{2M}\right) \cdot \textit{old } OPT$$

- Let S be opt. Steiner tree
- From each inner node in S: Contract the cheapest edge going to a child
- A terminal spanning tree T remains



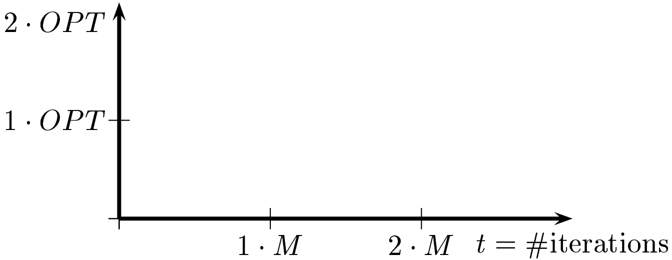
$$E[\text{save on } S] \geq E[\text{save on } T] \overset{\text{Bridge Lem}}{\geq} \frac{1}{M} \cdot \underbrace{c(T)}_{\geq \frac{1}{2} c(S)} \geq \frac{1}{2M} \cdot c(S)$$

□

The approximation guarantee

Theorem

$E[APX] \leq (1.5+\varepsilon) \cdot OPT.$



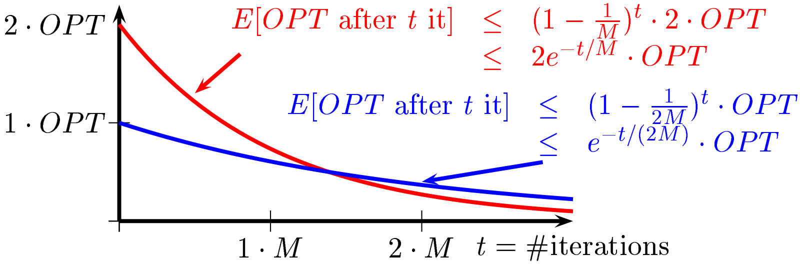
- Cost of sampled components:

$$\sum_{t=1}^{\infty} \frac{1}{M} \cdot E[OPT \text{ in it. } t]$$

The approximation guarantee

Theorem

$E[APX] \leq (1.5+\varepsilon) \cdot OPT.$



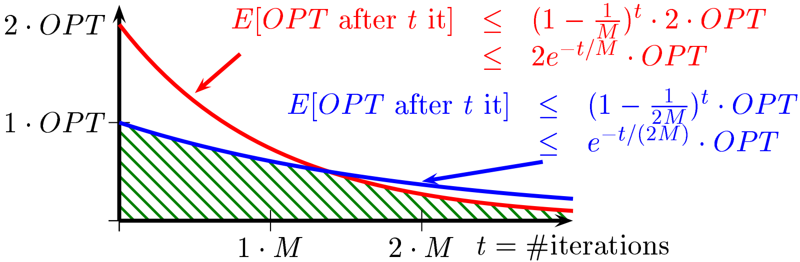
- Cost of sampled components:

$$\sum_{t=1}^{\infty} \frac{1}{M} \cdot E[OPT \text{ in it. } t]$$

The approximation guarantee

Theorem

$E[APX] \leq (1.5+\varepsilon) \cdot OPT.$



- Cost of sampled components:

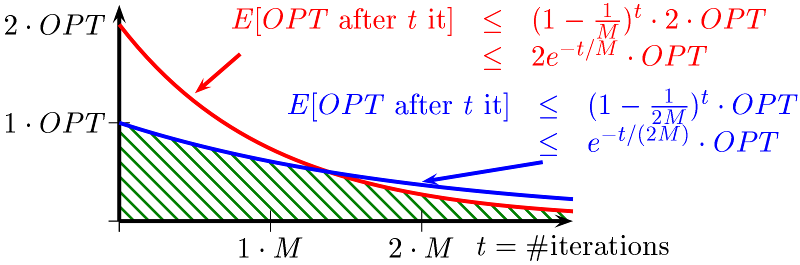
$$\sum_{t=1}^{\infty} \frac{1}{M} \cdot E[OPT \text{ in it. } t]$$

$$\stackrel{M\to\infty}{\to} \quad OPT \cdot \int_0^{\infty} \min\{2e^{-x}, e^{-x/2}\}\, dx$$

The approximation guarantee

Theorem

$E[APX] \le (1.5+\varepsilon) \cdot OPT.$



- Cost of sampled components:

$$\sum_{t=1}^{\infty} \frac{1}{M} \cdot E[OPT \text{ in it. } t]$$

$$\overset{M\to\infty}{\to} \quad OPT \cdot \int_0^{\infty} \min\{2e^{-x}, e^{-x/2}\}\, dx = 1.5 \cdot OPT \qquad \square$$

Open problems

Open Problem

$1.01 \leq$ Steiner tree approximability ≤ 1.39

Open problems

Open Problem

$1.01 \leq$ Steiner tree approximability ≤ 1.39

- Byrka, Grandoni, Rothvoß, Sanità - STOC'10:
  An improved LP-based approximation for Steiner Tree

http://infoscience.epfl.ch/record/148220/files/SteinerTree-STOC2010.pdf

Thanks for your attention