Steiner Tree Approximation via Iterative Randomized Rounding

Jarosław Byrka, Fabrizio Grandoni,
Thomas Rothvoß, Laura Sanità

EPFL, Lausanne,
Switzerland

Lugano, 20.07.10
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.
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 $\leq 2 \cdot \text{OPT}$. 
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 ∈ {1, 2} [Bern & Plassmann ’89]
- no < $\frac{96}{95}$-apx unless NP = 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 \textbf{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-directed edges

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

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

$$z_e \geq 0 \quad \forall e \in E.$$
Bi-directed cut relaxation

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

\[
\min \sum_{e \in E} c(e)z_e \quad \text{(BCR)}
\]

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

\[
z_e \geq 0 \quad \forall e \in E.
\]
Bi-directed cut relaxation

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

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

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

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

**Theorem (Edmonds ’67)**

$$R = V \Rightarrow \text{BCR integral}$$

- Integrality gap $\leq 4/3$ for quasi-bipartite graphs [Chakraborty, 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

- $C$ = set of directed components

- $\text{sink}(C)$

- directed component $C$
Directed component cut relaxation

\[\min \sum_{C \in \mathcal{C}} c(C) \cdot x_C \quad \text{(DCR)}\]

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

\[x_C \geq 0 \quad \forall C \in \mathcal{C}\]
Directed component cut relaxation

\[ \text{min} \sum_{C \in \mathcal{C}} c(C) \cdot x_C \quad \text{(DCR)} \]

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

\[ x_C \geq 0 \quad \forall C \in \mathcal{C} \]

\[ x_{C_1} = \frac{1}{2} \]

\[ x_{C_2} = \frac{1}{2} \]

\[ x_{C_3} = \frac{1}{2} \]
Directed component cut relaxation

\[
\min \sum_{C \in \mathcal{C}} c(C') \cdot x_C \quad \text{(DCR)}
\]

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

\[
x_C \geq 0 \quad \forall C \in \mathcal{C}
\]

Properties:

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

Lemma

For any $\epsilon > 0$, a solution $x$ of cost $\leq (1 + \epsilon)OPT_f$ 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 $2^{[1/\varepsilon]} = O(1)$
  [Borchers & Du ’97]: Increases cost by $\leq 1 + \varepsilon$
  → # 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.

- Use only components of size $2^{[1/\varepsilon]} = O(1)$
  
  [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.

- Use only components of size $2^{[1/\varepsilon]} = O(1)$ [Borchers & Du ’97]: Increases cost by $\leq 1 + \varepsilon$ → # variables polynomial
- Compact flow formulation → # 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:

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

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:

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

and contract it.

4. IF all terminals connected THEN output sampled components

$\begin{align*}
\vphantom{\frac{1}{2}}
 x_{C_1} &= \frac{1}{2} \\
 x_{C_2} &= \frac{1}{2} \\
 x_{C_3} &= \frac{1}{2}
\end{align*}$
An iterative randomized rounding algo

(1) FOR $t = 1, \ldots, \infty$ DO
    (2) Compute opt. LP solution $x$
    (3) Sample a component:

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

    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:

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

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:

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

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:

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

and contract it.

(4) IF all terminals connected THEN output sampled components

[Diagram]

[Diagram]
An iterative randomized rounding algo

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

(2) Compute opt. LP solution $x$

(3) Sample a component:

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

and contract it.

(4) \textbf{IF} all terminals connected \textbf{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:

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

and contract it.

(4) IF all terminals connected THEN output sampled components

- W.l.o.g. $M := 1^T x$ 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
  \[
  \sum_{t \geq 1} E[c(\text{comp. sampled in it. } t)] \leq \sum_{t \geq 1} \frac{1}{M} \cdot E[OPT \text{ in iteration } t]
  \]

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

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

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

\[ t = \#\text{iterations} \]
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:

$$Br_S(C) = \arg\max\{c(B) \mid B \subseteq S, \ S \setminus B \cup C \text{ is connected}\}$$
Bridges

- Let $S$ be Steiner tree, $C$ a component

**Bridges:**

$$Br_S(C) = \arg\max\{c(B) \mid B \subseteq S, \ S \setminus B \cup C \text{ is connected}\}$$
The saving function

Definition

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

$$c(Br_S(C)) = w(\text{saving tree})$$
A saving lemma

Lemma

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

$$c(\text{Br}_S(C)) = w(\text{saving tree})$$

- Consider forest $S \setminus \text{Br}_S(C)$
A saving lemma

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

$$c(\text{Br}_S(C)) = w(\text{saving tree})$$

- Consider forest $S \setminus \text{Br}_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\iff b_i$ 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

\[ c(\text{Br}_S(C)) = w(\text{saving tree}) \]

- Consider forest $S \setminus \text{Br}_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\iff b_i$ 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

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

- Consider forest $S \setminus Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree $\iff b_i$ 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

\[ c(Br_S(C)) = w(\text{saving tree}) \]

- Consider forest $S \setminus Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\iff b_i$ 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

$$c(\text{Br}_S(C)) = w(\text{saving tree})$$

- Consider forest $S \setminus \text{Br}_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\iff b_i$ 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

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

- Consider forest $S \setminus Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree $\iff b_i$ 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

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

- Consider forest $S \setminus Br_S(C)$
- Take edge $e_i = (u, v)$ into saving tree
  $\Leftrightarrow b_i$ connects trees of $u$ and $v$
- Then $w(e_i) = c(b_i)$.  \hfill $\square$
The Bridge Lemma (1)

Lemma (Bridge Lemma)

For $T$ terminal spanning tree, $x$ LP solution:

$$\sum_{C \in \mathcal{C}} x_C \cdot c(Br_T(C)) \geq c(T)$$
The Bridge Lemma (1)

Lemma (Bridge Lemma)

For $T$ terminal spanning tree, $x$ LP solution:

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

- 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:

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

- For any $C$, $\exists$ saving tree:
  $$c(Br_T(C)) = w($$ 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 a terminal spanning tree, \(x\) LP solution:

\[ \sum_{C \in \mathcal{C}} x_C \cdot c(Br_T(C)) \geq c(T) \]

- For any \(C\), \(\exists\) saving tree:
  \[ c(Br_T(C)) = w(\text{saving tree of } C) \]
- Transfer capacity from component to its saving tree
  \[ \rightarrow \text{capacity reservation } y : E \rightarrow \mathbb{Q}_+ \]
The Bridge Lemma (1)

Lemma (Bridge Lemma)

For $T$ terminal spanning tree, $x$ LP solution:

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

- 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:

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

- For any $C$, $\exists$ saving tree:
  $$c(Br_T(C)) = w(\text{saving tree of } C)$$
- Transfer capacity from component to its saving tree
  → capacity reservation $y : E \rightarrow \mathbb{Q}_+$
The Bridge Lemma (1)

**Lemma (Bridge Lemma)**

For $T$ terminal spanning tree, $x$ LP solution:

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

- 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 $y : E \rightarrow \mathbb{Q}_+$

$$\sum_{C \in \mathcal{C}} x_C \cdot c(Br_T(C)) = w(y)$$
The Bridge Lemma (2)

\[ \sum_{C \in \mathcal{C}} x_C \cdot c(Br_T(C)) = w(y) \]

![Diagram of a tree structure with labeled edges.]
The Bridge Lemma (2)

\[ \sum_{C \in \mathcal{C}} x_C \cdot c(Br_T(C)) = w(y) \geq w(F) \]
The Bridge Lemma (2)

\[ \sum_{C \in \mathcal{C}} x_C \cdot c(Br_T(C)) = w(y) \geq w(F) \geq c(T) \]
A 1st bound on OPT

**Lemma**

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

Lemma

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

- 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

\[ E[c(\text{new MST})] \leq c(\text{old MST}) - E[c(\text{Br}_{\text{old MST}}(C))] \]
A 1st bound on OPT

Lemma

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

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

\[ E[c(\text{new MST})] \leq c(\text{old MST}) - E[c(Br_{\text{old MST}}(C))] \]
\[ = c(\text{old MST}) - \frac{1}{M} \sum_{C \in \mathcal{C}} x_C \cdot c(Br_{\text{old MST}}(C)) \]
A 1st bound on OPT

Lemma

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

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

\[
E[c(\text{new MST})] \leq c(\text{old MST}) - E[c(Br_{\text{old MST}}(C))] \\
= c(\text{old MST}) - \frac{1}{M} \sum_{C \in C} x_C \cdot c(Br_{\text{old MST}}(C)) \\
\geq c(\text{old MST})
\]
A 1st bound on OPT

Lemma

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

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

\[
E[c(\text{new MST})] \leq c(\text{old MST}) - E[c(Br_{\text{old MST}}(C))] \\
= c(\text{old MST}) - \frac{1}{M} \sum_{C \in \mathcal{C}} x_C \cdot c(Br_{\text{old MST}}(C)) \\
\geq c(\text{old MST}) \\
\leq \left(1 - \frac{1}{M}\right) \cdot c(\text{old MST})
\]

\[ \square \]
A 2nd bound on $OPT$

**Theorem**

*In any iteration*

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

- Let $S$ be opt. Steiner tree
A 2nd bound on $OPT$

**Theorem**

*In any iteration*

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

- 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

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

- 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

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

- 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

$$E[\text{save on } S] \geq E[\text{save on } T] \geq \frac{1}{M} \cdot \min_{T} c(T) \geq \frac{1}{2M} \cdot c(S) \geq \frac{1}{2} c(S)$$
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**

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

\[E[OPT \text{ after } t \text{ it}] \leq (1 - \frac{1}{M})^t \cdot 2 \cdot OPT \leq 2e^{-t/M} \cdot OPT\]

\[E[OPT \text{ after } t \text{ it}] \leq (1 - \frac{1}{2M})^t \cdot OPT \leq e^{-t/(2M)} \cdot OPT\]

- **Cost of sampled components:**

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

**Theorem**

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

\[ E[OPT \text{ after } t \text{ it}] \leq (1 - \frac{1}{M})^t \cdot 2 \cdot OPT \leq 2e^{-t/M} \cdot OPT \]

\[ E[OPT \text{ after } t \text{ it}] \leq (1 - \frac{1}{2M})^t \cdot OPT \leq e^{-t/(2M)} \cdot OPT \]

- Cost of sampled components:

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

\[
\lim_{M \to \infty} OPT \cdot \int_0^{\infty} \min\{2e^{-x}, e^{-x/2}\} \, dx
\]
The approximation guarantee

**Theorem**

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

![Graph showing the approximation guarantee](image_url)

- Cost of sampled components:

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

\[
\xrightarrow{M \to \infty} OPT \cdot \int_0^{\infty} \min\{2e^{-x}, e^{-x/2}\} \, dx = 1.5 \cdot OPT
\]

\[ \square \]
Open problems

<table>
<thead>
<tr>
<th>Open Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.01 \leq \text{Steiner tree approximability} \leq 1.39$</td>
</tr>
</tbody>
</table>
Open problems

Open Problem

1.01 ≤ Steiner tree approximability ≤ 1.39

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

Thanks for your attention