# Cost Sharing and Approximation Algorithms

— Lecture 3 —

Guido Schäfer
CWI Amsterdam / VU University Amsterdam
g.schaefer@cwi.nl

ADFOCS 2010
11th Max Planck Advanced Course on the Foundations of Computer Science
August 2–6, 2010, Saarbrücken, Germany

# Recap: Randomized Framework

**Sample-and-Augment Algorithm for MROB:**

1: Mark each terminal pair with probability $1/M$. Let $D$ be set of marked terminal pairs.

2: Compute an $\alpha$-approximate Steiner forest $F$ for $D$ and buy all edges in $F$.

3: For all terminal pairs $(s,t) \notin D$: rent unit capacity on a shortest $s,t$-path in contracted graph $G|F$.

$G|F =$ graph obtained from $G$ by contracting all edges in $F \subseteq E$

# Recap: Strictness

## Definition

A Steiner forest algorithm *ALG* is $\beta$-strict if there exist **cost shares** $\xi_{st} \geq 0$ for every $(s,t) \in R$ such that:

1. $\sum_{(s,t)\in R} \xi_{st} \leq c(F^*)$ (competitiveness)
2. For every $(s,t) \in R$, $c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}$ ($\beta$-strictness)

**Notation:**

- $F^* =$ optimal Steiner forest for $R$
- $F_{-st} =$ Steiner forest computed by *ALG* for $R_{-st} = R \setminus \{(s,t)\}$
- $G|F_{-st} =$ graph obtained if all components of $F_{-st}$ are contracted

# Randomized Framework

**Theorem**

*Given an $\alpha$-approximate and $\beta$-strict Steiner forest algorithm, Sample-and-Augment is an (expected) $(\alpha + \beta)$-approximation algorithm for MROB.*

[Gupta, Kumar, Pál, Roughgarden, JACM '07]

**Remark:** framework applies to other network design problems

- single-sink rent-or-buy
- multicast rent-or-buy
- virtual private network design
- single-sink buy-at-bulk

# State of Affairs

## Multicommodity Rent-or-Buy

| | |
|---|---|
| [Kumar, Gupta, Roughgarden, FOCS ’02] | O(1) |
| [Gupta, Kumar, Pál, Roughgarden, FOCS ’03] | 12, later 8 |
| [Becchetti, Könemann, Leonardi, Pál, SODA ’05] | 6.82 |
| [Fleischer, Könemann, Leonardi, Schäfer, STOC ’06] | 5 |

**Theorem**

*The primal-dual 2-approximate Steiner forest algorithm AKR of* [Agrawal, Klein, Ravi, SICOMP ’95] *is 3-strict.*

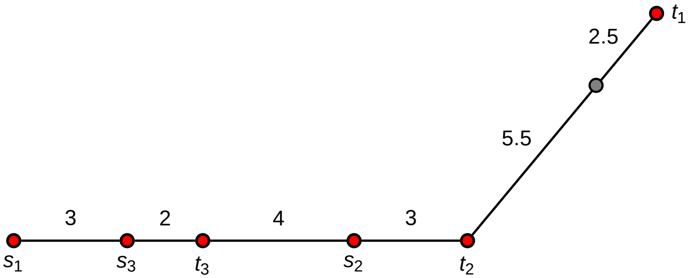
[Fleischer, Könemann, Leonardi, Schäfer, STOC ’06]

**Remark:** $\frac{8}{3}$ is a lower bound on the strictness factor of every 2-approximate Steiner forest algorithm

# Strict Steiner Forest Algorithm

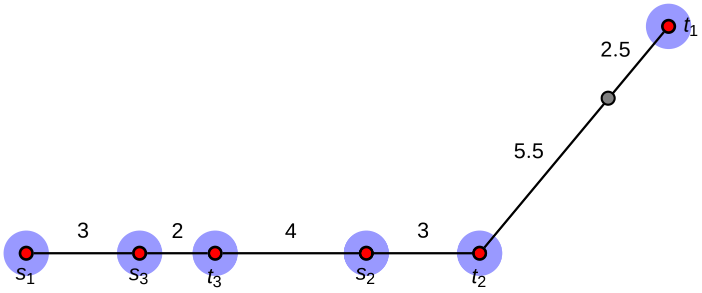
# Recall: Steiner Forest Algorithm AKR

$\tau = 0.0$

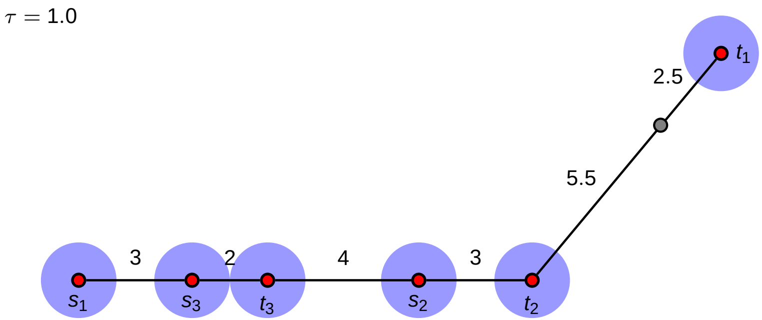


# Recall: Steiner Forest Algorithm AKR

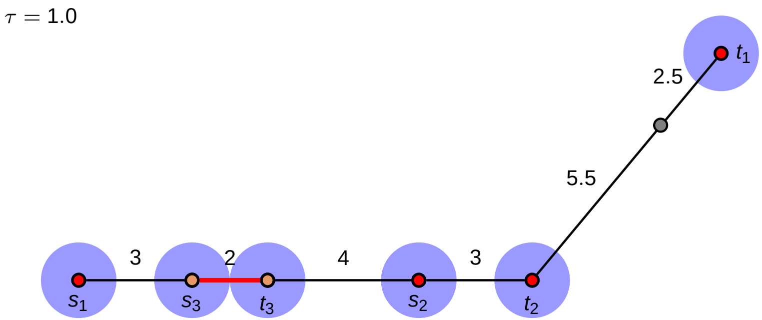
$\tau = 0.3$



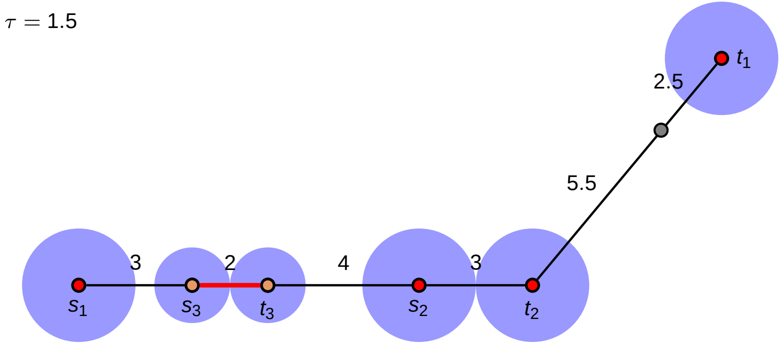
# Recall: Steiner Forest Algorithm AKR



# Recall: Steiner Forest Algorithm AKR

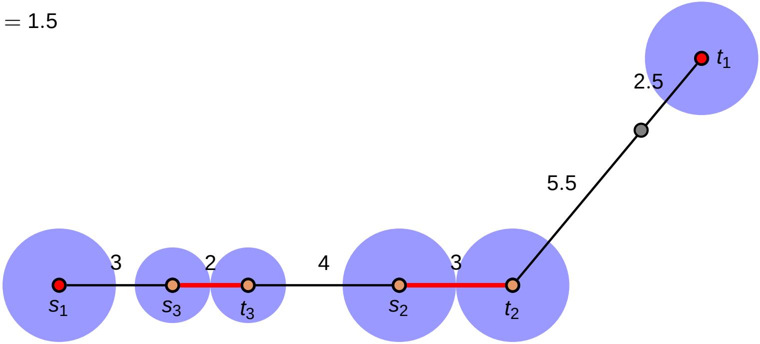


# Recall: Steiner Forest Algorithm AKR

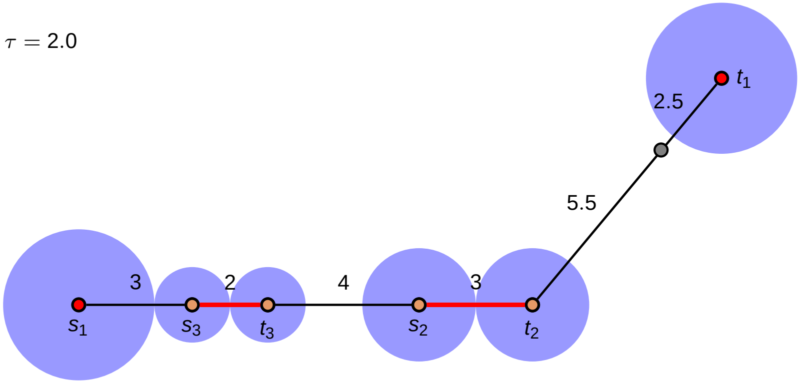


# Recall: Steiner Forest Algorithm AKR

$\tau = 1.5$

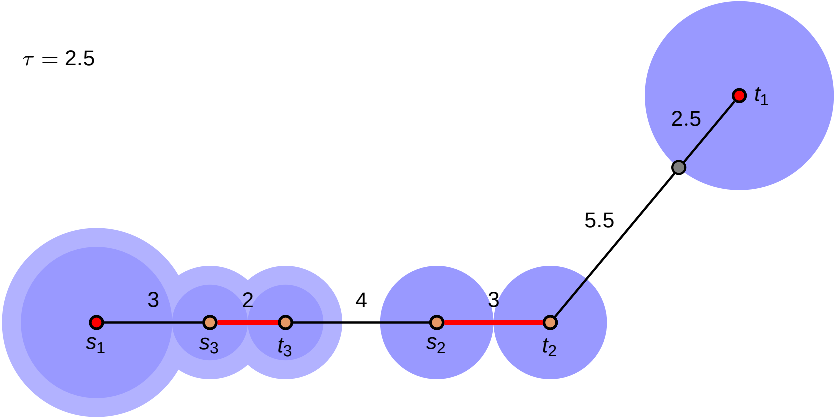


# Recall: Steiner Forest Algorithm AKR



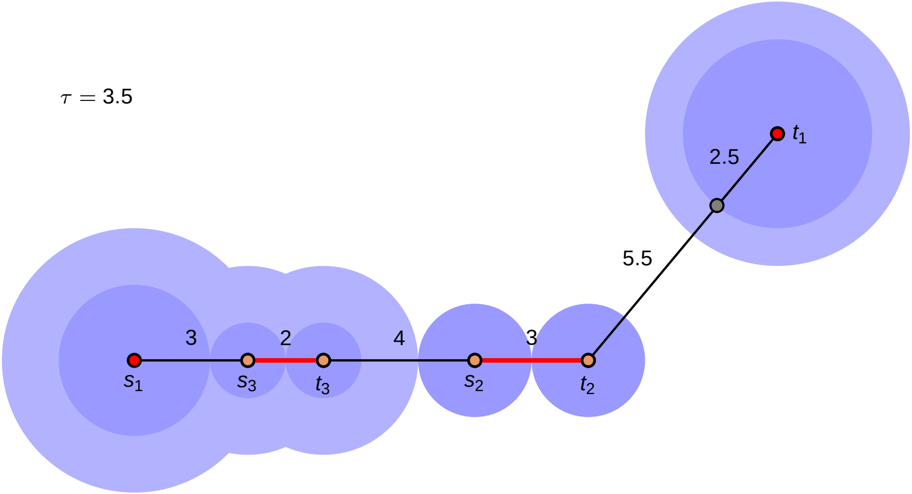
# Recall: Steiner Forest Algorithm AKR

$\tau = 2.5$



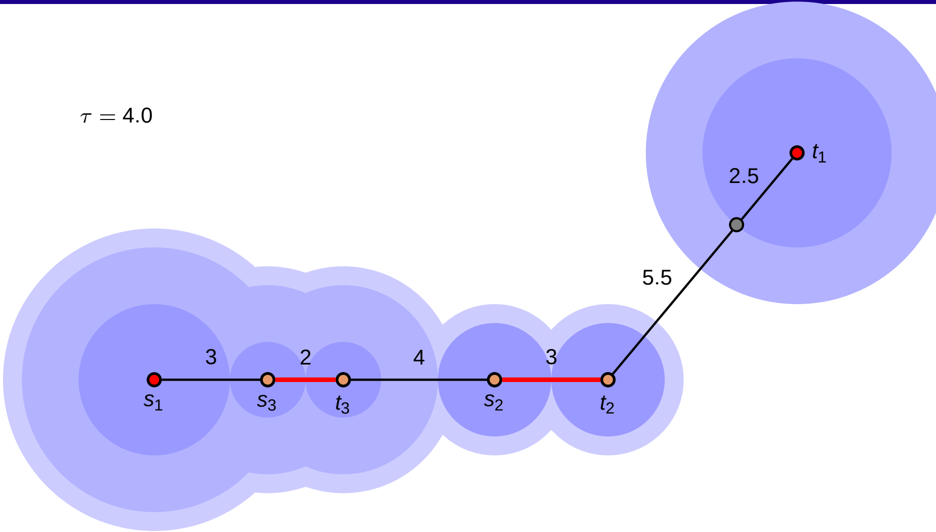
# Recall: Steiner Forest Algorithm AKR

$\tau = 3.5$

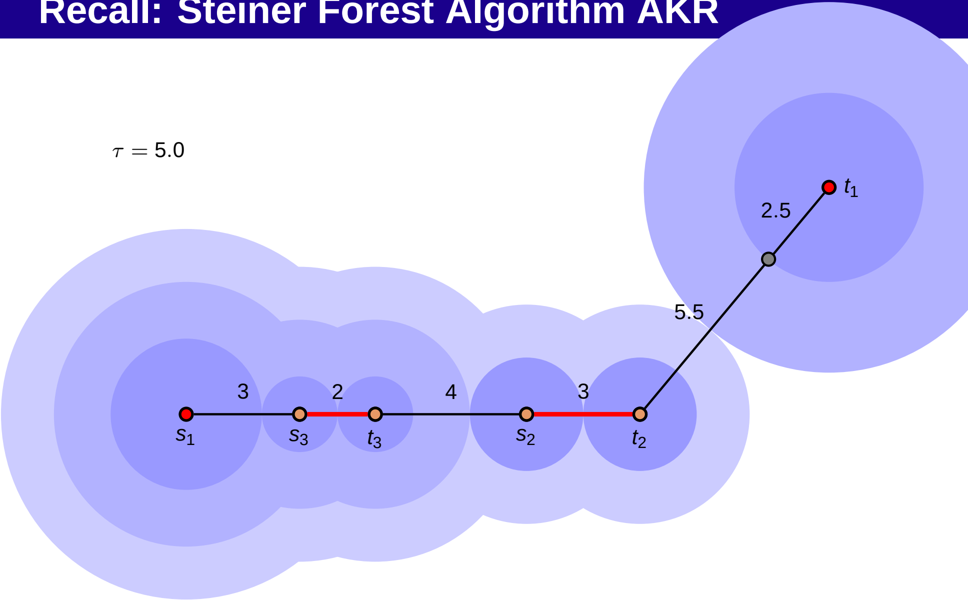


# Recall: Steiner Forest Algorithm AKR

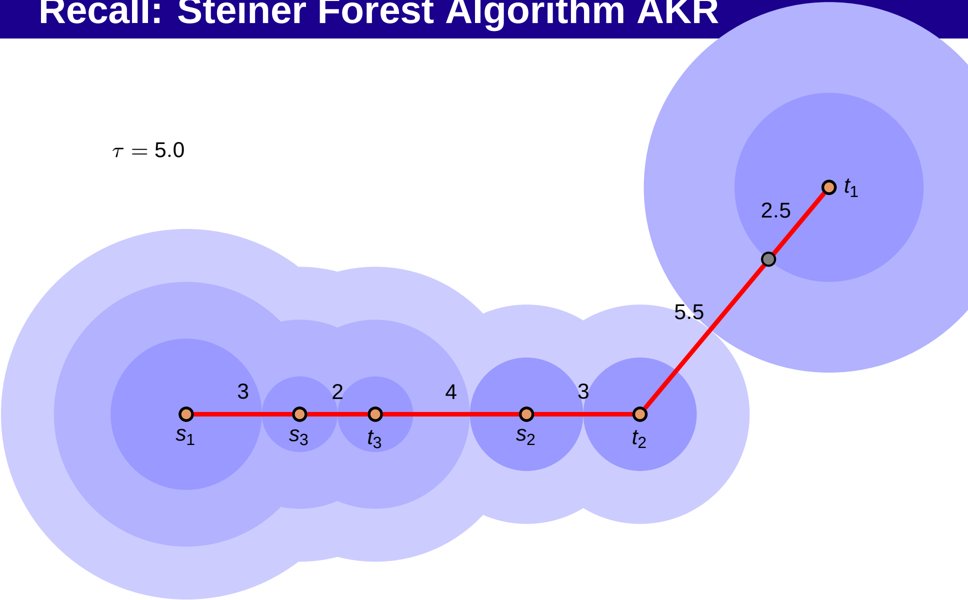
$\tau = 4.0$



# Recall: Steiner Forest Algorithm AKR



# Recall: Steiner Forest Algorithm AKR



# Adding Competitiveness

**Competitiveness:** compute cost shares $\xi_{st}$ for all $(s,t) \in R$ such that

$$\sum_{(s,t)\in R} \xi_{st} \leq OPT$$

**Idea:**

- forest $F$ computed by AKR has cost at most $2OPT$
- whenever a path $P_i$ becomes tight, can distribute half of the cost of the added edges as cost share

$$\xi(e) = \frac{1}{2}c(e)$$

- total distributed cost share is

$$\sum_{e\in F} \xi(e) = \frac{1}{2}c(F) \leq OPT$$

# Adding Strictness

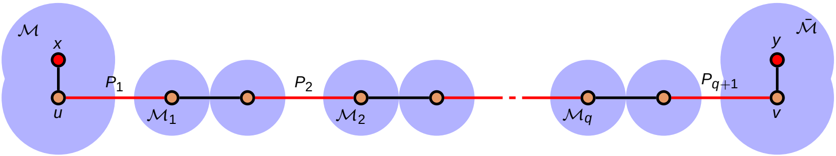
**$\beta$-Strictness:** connecting $s$ and $t$ in the contracted graph $G|F_{-st}$ has cost at most

$$c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}.$$

**Idea:**

- consider the unique $s,t$-path $P_{st}$ in forest $F$ computed by AKR for $R$
- some edges of $P_{st}$ might be missing in $F_{-st}$
- use $\beta \cdot \xi_{st}$ to pay for adding the missing edges

# Crucial Notion: Witnesses

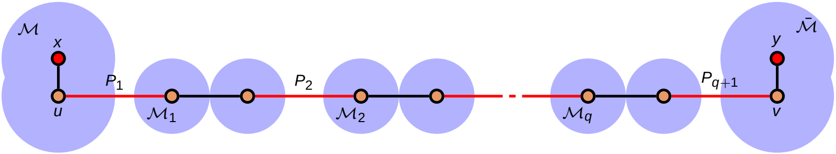



**Event:** moats $\mathcal{M}$ and $\bar{\mathcal{M}}$ make path $P$ tight:

- $P$ passes through inactive moats $\mathcal{M}_1, \ldots, \mathcal{M}_q$
- every edge $e \in P_1 \cup \cdots \cup P_{q+1}$ is added to existing forest
- $x =$ active terminal in $\mathcal{M}$ whose moat intersects $P$ earliest
  $y =$ active terminal in $\bar{\mathcal{M}}$ whose moat intersects $P$ earliest

Call $\mathcal{W}_e = \{x, y\}$ the witnesses of edge $e \in P_1 \cup \cdots \cup P_{q+1}$

# Crucial Notion: Witnesses




**Event:** moats $\mathcal{M}$ and $\bar{\mathcal{M}}$ make path $P$ tight:

- $P$ passes through inactive moats $\mathcal{M}_1, \ldots, \mathcal{M}_q$
- every edge $e \in P_1 \cup \cdots \cup P_{q+1}$ is added to existing forest
- $x =$ active terminal in $\mathcal{M}$ whose moat intersects $P$ earliest
  $y =$ active terminal in $\bar{\mathcal{M}}$ whose moat intersects $P$ earliest

Call $\mathcal{W}_e = \{x, y\}$ the witnesses of edge $e \in P_1 \cup \cdots \cup P_{q+1}$

# Witness Lemma

$\tau_{st}$ = time when $s$ and $t$ become inactive in AKR

**Lemma (Witness Lemma)**

*Let $e$ be an edge that has been added to forest $F$ at time $\tau_e \leq \tau_{st}$. If neither $s$ nor $t$ is witness for $e$ then $e$ is part of $F_{-st}$.*

**Proof (sketch):** Can show by induction over $\tau \leq \tau_{st}$ that for every terminal $v \neq s, t$:

$$\mathcal{M}^{\tau}_{-st}(v) = \mathcal{M}^{\tau}(v) \text{ if } \mathcal{M}^{\tau}(v) \cap \{s,t\} = \emptyset$$
$$\mathcal{M}^{\tau}_{-st}(v) \subseteq \mathcal{M}^{\tau}(v) \text{ otherwise}$$

$\Rightarrow$ every terminal $v \neq s, t$ that is active at time $\tau \leq \tau_{st}$ in AKR($R$) must be active at that time in AKR($R_{-st}$). □

# Witness Lemma

$\tau_{st}$ = time when $s$ and $t$ become inactive in AKR

**Lemma (Witness Lemma)**

*Let $e$ be an edge that has been added to forest $F$ at time $\tau_e \leq \tau_{st}$. If neither $s$ nor $t$ is witness for $e$ then $e$ is part of $F_{-st}$.*

**Proof (sketch):** Can show by induction over $\tau \leq \tau_{st}$ that for every terminal $v \neq s, t$:

$$\mathcal{M}^{\tau}_{-st}(v) = \mathcal{M}^{\tau}(v) \text{ if } \mathcal{M}^{\tau}(v) \cap \{s,t\} = \emptyset$$
$$\mathcal{M}^{\tau}_{-st}(v) \subseteq \mathcal{M}^{\tau}(v) \text{ otherwise}$$

$\Rightarrow$ every terminal $v \neq s, t$ that is active at time $\tau \leq \tau_{st}$ in AKR($R$) must be active at that time in AKR($R_{-st}$). □

# Witness Lemma

$\tau_{st}$ = time when $s$ and $t$ become inactive in AKR

**Lemma (Witness Lemma)**

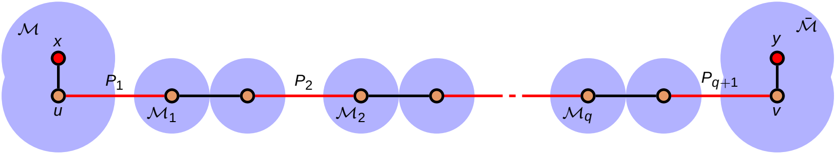
*Let $e$ be an edge that has been added to forest $F$ at time $\tau_e \leq \tau_{st}$. If neither $s$ nor $t$ is witness for $e$ then $e$ is part of $F_{-st}$.*

**Proof (sketch):** Can show by induction over $\tau \leq \tau_{st}$ that for every terminal $v \neq s, t$:

$$\mathcal{M}^{\tau}_{-st}(v) = \mathcal{M}^{\tau}(v) \text{ if } \mathcal{M}^{\tau}(v) \cap \{s,t\} = \emptyset$$
$$\mathcal{M}^{\tau}_{-st}(v) \subseteq \mathcal{M}^{\tau}(v) \text{ otherwise}$$

$\Rightarrow$ every terminal $v \neq s, t$ that is active at time $\tau \leq \tau_{st}$ in AKR($R$) must be active at that time in AKR($R_{-st}$). □

# Symmetric Cost Sharing




$x, y$ = witnesses for the edges $e \in P_1 \cup \ldots \cup P_{q+1}$

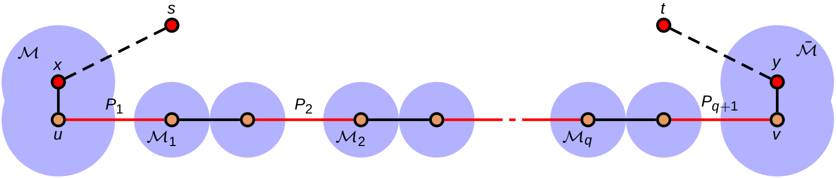
**Cost Share Distribution:** if edge $e$ is witnessed by $v \in \{x, y\}$

$$\xi_v(e) = \frac{1}{2}\xi(e) = \frac{1}{4}c(e)$$

Cost share of terminal pair $(s, t)$:

$$\xi_{st} = \sum_{e \in F} \xi_s(e) + \xi_t(e)$$

# Path Reconstruction

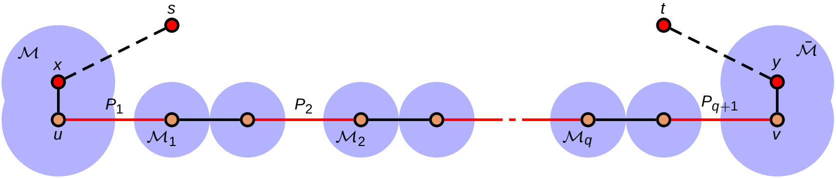



**Goal:** augment forest $F_{-st}$ at cost $c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}$

- consider the unique $s,t$-path $P_{st}$ in $F$
- some edges of $P_{st}$ might be missing in $F_{-st}$
- **Witness Lemma:** If $e \in P_{st} \setminus F_{-st}$ then $\{s,t\} \cap \mathcal{W}_e \neq \emptyset$.
- each witness of $e$ received $\frac{1}{2}\xi(e)$ as cost share
- cost of edge $e$ is $2\xi(e)$

  $\Rightarrow 4\xi_{st}$ sufficient to pay for all missing edges on $P_{st}$

**Thus:** AKR is 2-approximate and 4-strict



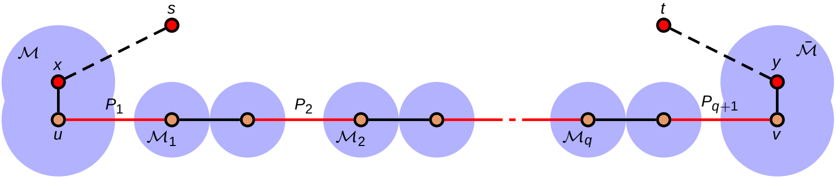

**Goal:** augment forest $F_{-st}$ at cost $c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}$

- consider the unique $s,t$-path $P_{st}$ in $F$
- some edges of $P_{st}$ might be missing in $F_{-st}$
- **Witness Lemma:** If $e \in P_{st} \setminus F_{-st}$ then $\{s,t\} \cap \mathcal{W}_e \neq \emptyset$.
- each witness of $e$ received $\frac{1}{2}\xi(e)$ as cost share
- cost of edge $e$ is $2\xi(e)$

  $\Rightarrow 4\xi_{st}$ sufficient to pay for all missing edges on $P_{st}$

**Thus:** AKR is 2-approximate and 4-strict

# Path Reconstruction



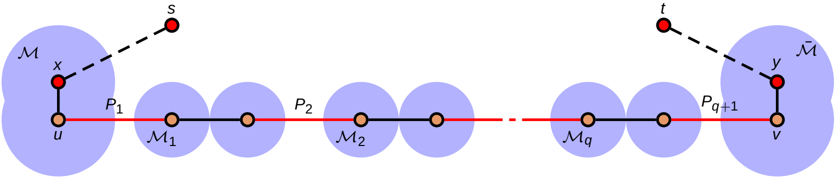

**Goal:** augment forest $F_{-st}$ at cost $c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}$

- consider the unique $s,t$-path $P_{st}$ in $F$
- some edges of $P_{st}$ might be missing in $F_{-st}$
- **Witness Lemma:** If $e \in P_{st} \setminus F_{-st}$ then $\{s,t\} \cap \mathcal{W}_e \neq \emptyset$.
- each witness of $e$ received $\frac{1}{2}\xi(e)$ as cost share
- cost of edge $e$ is $2\xi(e)$

  $\Rightarrow 4\xi_{st}$ sufficient to pay for all missing edges on $P_{st}$

**Thus:** AKR is 2-approximate and 4-strict

# Path Reconstruction



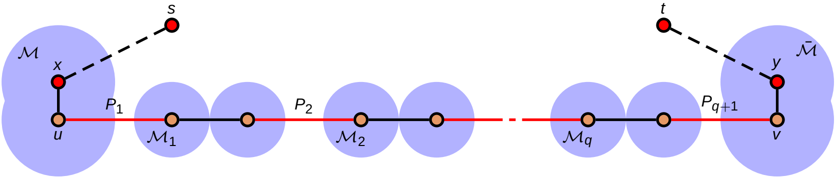

**Goal:** augment forest $F_{-st}$ at cost $c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}$

- consider the unique $s,t$-path $P_{st}$ in $F$
- some edges of $P_{st}$ might be missing in $F_{-st}$
- **Witness Lemma:** If $e \in P_{st} \setminus F_{-st}$ then $\{s,t\} \cap \mathcal{W}_e \neq \emptyset$.
- each witness of $e$ received $\frac{1}{2}\xi(e)$ as cost share
- cost of edge $e$ is $2\xi(e)$

  $\Rightarrow 4\xi_{st}$ sufficient to pay for all missing edges on $P_{st}$

**Thus:** AKR is 2-approximate and 4-strict

# Path Reconstruction



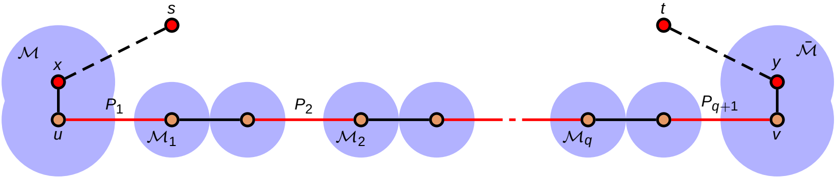

**Goal:** augment forest $F_{-st}$ at cost $c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}$

- consider the unique $s,t$-path $P_{st}$ in $F$
- some edges of $P_{st}$ might be missing in $F_{-st}$
- **Witness Lemma:** If $e \in P_{st} \setminus F_{-st}$ then $\{s,t\} \cap \mathcal{W}_e \neq \emptyset$.
- each witness of $e$ received $\frac{1}{2}\xi(e)$ as cost share
- cost of edge $e$ is $2\xi(e)$

  $\Rightarrow 4\xi_{st}$ sufficient to pay for all missing edges on $P_{st}$

**Thus:** AKR is 2-approximate and 4-strict

# Path Reconstruction



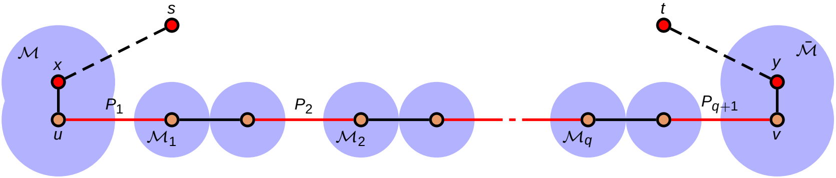

**Goal:** augment forest $F_{-st}$ at cost $c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}$

- consider the unique $s,t$-path $P_{st}$ in $F$
- some edges of $P_{st}$ might be missing in $F_{-st}$
- **Witness Lemma:** If $e \in P_{st} \setminus F_{-st}$ then $\{s,t\} \cap \mathcal{W}_e \neq \emptyset$.
- each witness of $e$ received $\frac{1}{2}\xi(e)$ as cost share
- cost of edge $e$ is $2\xi(e)$

  $\Rightarrow 4\xi_{st}$ sufficient to pay for all missing edges on $P_{st}$

**Thus:** AKR is 2-approximate and 4-strict

# Path Reconstruction



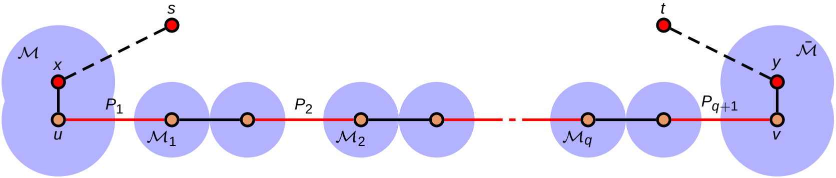

**Goal:** augment forest $F_{-st}$ at cost $c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}$

- consider the unique $s,t$-path $P_{st}$ in $F$
- some edges of $P_{st}$ might be missing in $F_{-st}$
- **Witness Lemma:** If $e \in P_{st} \setminus F_{-st}$ then $\{s,t\} \cap \mathcal{W}_e \neq \emptyset$.
- each witness of $e$ received $\frac{1}{2}\xi(e)$ as cost share
- cost of edge $e$ is $2\xi(e)$

  $\Rightarrow 4\xi_{st}$ sufficient to pay for all missing edges on $P_{st}$

**Thus:** AKR is 2-approximate and 4-strict

# Path Reconstruction



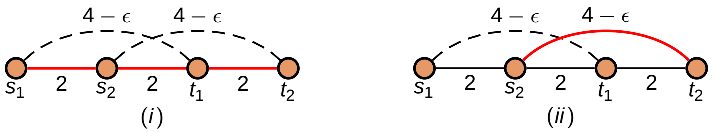

**Goal:** augment forest $F_{-st}$ at cost $c_{G|F_{-st}}(s,t) \leq \beta \cdot \xi_{st}$

- consider the unique $s,t$-path $P_{st}$ in $F$
- some edges of $P_{st}$ might be missing in $F_{-st}$
- **Witness Lemma:** If $e \in P_{st} \setminus F_{-st}$ then $\{s,t\} \cap \mathcal{W}_e \neq \emptyset$.
- each witness of $e$ received $\frac{1}{2}\xi(e)$ as cost share
- cost of edge $e$ is $2\xi(e)$

  $\Rightarrow 4\xi_{st}$ sufficient to pay for all missing edges on $P_{st}$

**Thus:** AKR is 2-approximate and 4-strict

# Tight Example and Insights




Analysis is tight:

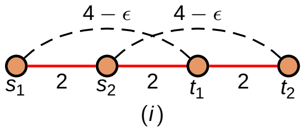
- cost share of $(s_1, t_1)$ for path $\langle s_1, s_2, t_1 \rangle$ is 1
- reconstructing this path in $G|F_{-s_1 t_1}$ costs 4

**But:** we are not using $\xi_{t_1}(t_1, t_2)$!

- total cost share of $(s_1, t_1)$ in our algorithm is $\frac{3}{2}$
- we could have shown $\frac{4}{3/2} = \frac{8}{3}$-strictness!

**Open Problem**: Does the symmetric cost sharing rule lead to $\frac{8}{3}$-strictness?

# Tight Example and Insights




Analysis is tight:

- cost share of $(s_1, t_1)$ for path $\langle s_1, s_2, t_1 \rangle$ is 1
- reconstructing this path in $G|F_{-s_1 t_1}$ costs 4

**But:** we are not using $\xi_{t_1}(t_1, t_2)$!

- total cost share of $(s_1, t_1)$ in our algorithm is $\frac{3}{2}$
- we could have shown $\frac{4}{3/2} = \frac{8}{3}$-strictness!

**Open Problem**: Does the symmetric cost sharing rule lead to $\frac{8}{3}$-strictness?

# Tight Example and Insights


Analysis is tight:

- cost share of $(s_1, t_1)$ for path $\langle s_1, s_2, t_1 \rangle$ is 1
- reconstructing this path in $G|F_{-s_1 t_1}$ costs 4

**But:** we are not using $\xi_{t_1}(t_1, t_2)$!

- total cost share of $(s_1, t_1)$ in our algorithm is $\frac{3}{2}$
- we could have shown $\frac{4}{3/2} = \frac{8}{3}$-strictness!

**Open Problem**: Does the symmetric cost sharing rule lead to $\frac{8}{3}$-strictness?

# Asymmetric Cost Sharing

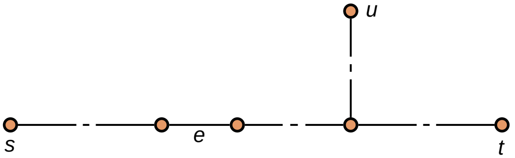



Figure shows path $P_{st}$ in forest $F$ and an edge $e$ with $\mathcal{W}_e = \{s, u\}$ for some $u \neq t$.

Let $\bar{u}$ be the mate of $u$. $\tau_{st}$ ($\tau_{u\bar{u}}$) is the time when $s$ and $t$ ($u$ and $\bar{u}$) meet in AKR.

Cost share $\xi_s(e)$ will now depend on:

- whether $e \in P_{u\bar{u}}$ or not, and
- meeting times $\tau_{st}$ and $\tau_{u\bar{u}}$.

# Asymmetric Cost Sharing

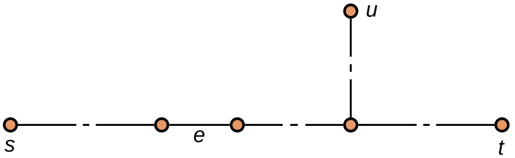



Figure shows path $P_{st}$ in forest $F$ and an edge $e$ with $\mathcal{W}_e = \{s, u\}$ for some $u \neq t$.

Let $\bar{u}$ be the mate of $u$. $\tau_{st}$ ($\tau_{u\bar{u}}$) is the time when $s$ and $t$ ($u$ and $\bar{u}$) meet in AKR.

Cost share $\xi_s(e)$ will now depend on:

- whether $e \in P_{u\bar{u}}$ or not, and
- meeting times $\tau_{st}$ and $\tau_{u\bar{u}}$.

# Asymmetric Cost Sharing

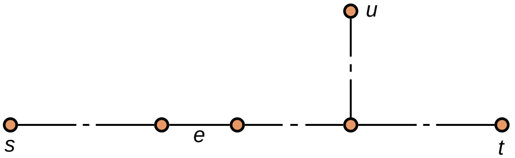



Figure shows path $P_{st}$ in forest $F$ and an edge $e$ with $\mathcal{W}_e = \{s, u\}$ for some $u \neq t$.

Let $\bar{u}$ be the mate of $u$. $\tau_{st}$ ($\tau_{u\bar{u}}$) is the time when $s$ and $t$ ($u$ and $\bar{u}$) meet in AKR.

Cost share $\xi_s(e)$ will now depend on:

- whether $e \in P_{u\bar{u}}$ or not, and
- meeting times $\tau_{st}$ and $\tau_{u\bar{u}}$.

# Asymmetric Cost Sharing

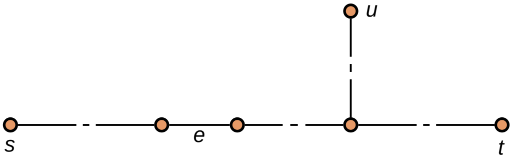


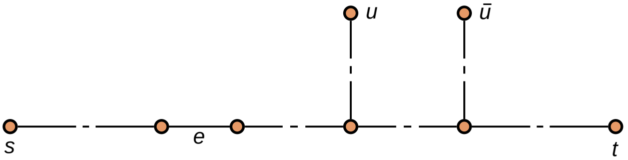

Figure shows path $P_{st}$ in forest $F$ and an edge $e$ with $\mathcal{W}_e = \{s, u\}$ for some $u \neq t$.

Let $\bar{u}$ be the mate of $u$. $\tau_{st}$ ($\tau_{u\bar{u}}$) is the time when $s$ and $t$ ($u$ and $\bar{u}$) meet in AKR.

Cost share $\xi_s(e)$ will now depend on:

- whether $e \in P_{u\bar{u}}$ or not, and
- meeting times $\tau_{st}$ and $\tau_{u\bar{u}}$.

# Edges not needed by $(u, \bar{u})$



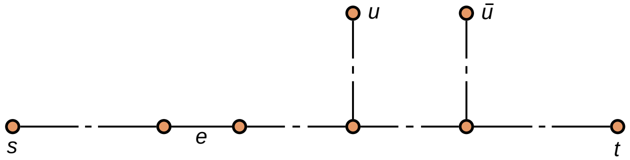

**Cost Share Distribution:** if edge $e$ is on $P_{st} \setminus P_{u\bar{u}}$

$$\xi_s(e) = \frac{2}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{1}{3}\xi(e)$$

**Note:** cost of such an edge $e$ is $2\xi(e)$

$\Rightarrow 3\xi_s(e)$ is sufficient to pay for $e$

# Edges not needed by $(u, \bar{u})$



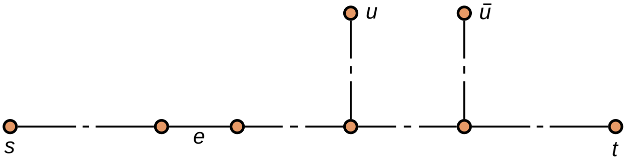

**Cost Share Distribution:** if edge $e$ is on $P_{st} \setminus P_{u\bar{u}}$

$$\xi_s(e) = \frac{2}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{1}{3}\xi(e)$$

**Note:** cost of such an edge $e$ is $2\xi(e)$

$\Rightarrow 3\xi_s(e)$ is sufficient to pay for $e$

# Edges not needed by $(u, \bar{u})$



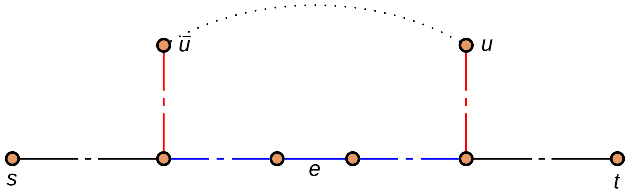

**Cost Share Distribution:** if edge $e$ is on $P_{st} \setminus P_{u\bar{u}}$

$$\xi_s(e) = \frac{2}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{1}{3}\xi(e)$$

**Note:** cost of such an edge $e$ is $2\xi(e)$

$\Rightarrow 3\xi_s(e)$ is sufficient to pay for $e$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$



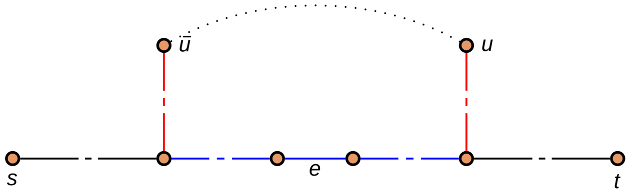

**Observe:** all blue and red edges are built before time $\tau_{st}$

$B, R =$ blue, red edges missing in $F_{-st}$

**Witness Lemma:** For all $e \in B \cup R$: $\{s, t\} \cap \mathcal{W}_e \neq \emptyset$.

**Idea:** use cost share obtained for edges in $B$ and $R$ and fact that $u$ and $\bar{u}$ are connected to connect $s$ and $t$ in $F_{-st}$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$



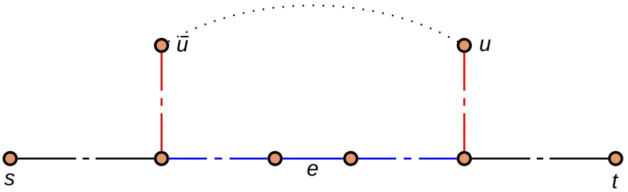

**Observe:** all blue and red edges are built before time $\tau_{st}$

$B, R =$ blue, red edges missing in $F_{-st}$

**Witness Lemma:** For all $e \in B \cup R$: $\{s, t\} \cap \mathcal{W}_e \neq \emptyset$.

**Idea:** use cost share obtained for edges in $B$ and $R$ and fact that $u$ and $\bar{u}$ are connected to connect $s$ and $t$ in $F_{-st}$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$



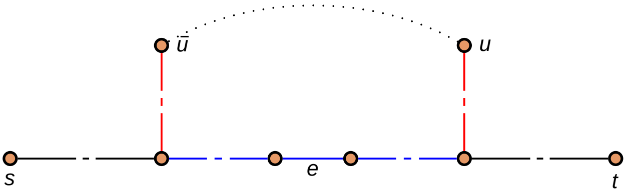

**Observe:** all blue and red edges are built before time $\tau_{st}$

$B, R =$ blue, red edges missing in $F_{-st}$

**Witness Lemma:** For all $e \in B \cup R$: $\{s, t\} \cap \mathcal{W}_e \neq \emptyset$.

**Idea:** use cost share obtained for edges in $B$ and $R$ and fact that $u$ and $\bar{u}$ are connected to connect $s$ and $t$ in $F_{-st}$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$



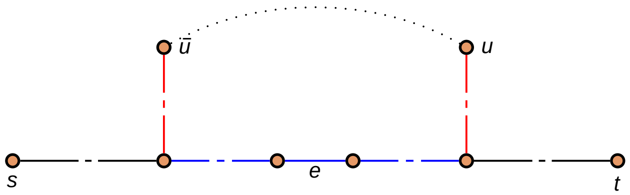

**Observe:** all blue and red edges are built before time $\tau_{st}$

$B, R =$ blue, red edges missing in $F_{-st}$

**Witness Lemma:** For all $e \in B \cup R$: $\{s, t\} \cap \mathcal{W}_e \neq \emptyset$.

**Idea:** use cost share obtained for edges in $B$ and $R$ and fact that $u$ and $\bar{u}$ are connected to connect $s$ and $t$ in $F_{-st}$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \le \tau_{st}$




**Cost Share Distribution:** If $e \in P_{u\bar{u}}$ and $\tau_{u\bar{u}} \le \tau_{st}$

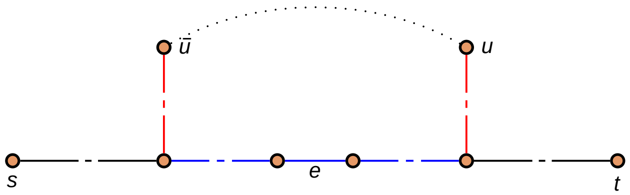
$$\xi_s(e) = \frac{1}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{2}{3}\xi(e)$$

$s, t$ obtain at least $\frac{1}{3}\xi(e)$ for each $e \in B \cup R$

**Assume:** $\sum_{e \in B} \xi(e) \ge \sum_{e \in R} \xi(e)$

Total cost share obtained for $B \cup R$ is at least $\frac{2}{3} \sum_{e \in R} \xi(e)$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$




**Cost Share Distribution:** If $e \in P_{u\bar{u}}$ and $\tau_{u\bar{u}} \leq \tau_{st}$

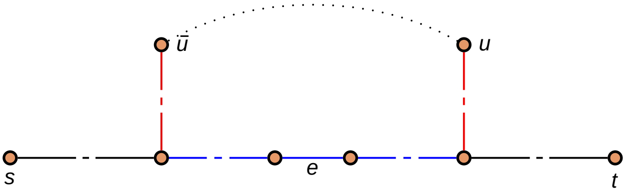
$$\xi_s(e) = \frac{1}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{2}{3}\xi(e)$$

$s, t$ obtain at least $\frac{1}{3}\xi(e)$ for each $e \in B \cup R$

**Assume:** $\sum_{e \in B} \xi(e) \geq \sum_{e \in R} \xi(e)$

Total cost share obtained for $B \cup R$ is at least $\frac{2}{3}\sum_{e \in R} \xi(e)$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$




**Cost Share Distribution:** If $e \in P_{u\bar{u}}$ and $\tau_{u\bar{u}} \leq \tau_{st}$

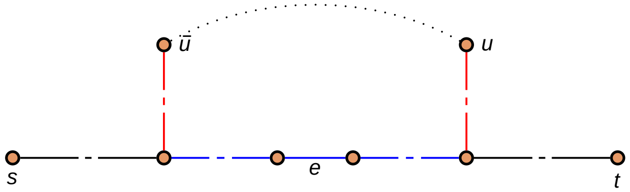
$$\xi_s(e) = \frac{1}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{2}{3}\xi(e)$$

$s, t$ obtain at least $\frac{1}{3}\xi(e)$ for each $e \in B \cup R$

**Assume:** $\sum_{e \in B} \xi(e) \geq \sum_{e \in R} \xi(e)$

Total cost share obtained for $B \cup R$ is at least $\frac{2}{3} \sum_{e \in R} \xi(e)$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$




**Cost Share Distribution:** If $e \in P_{u\bar{u}}$ and $\tau_{u\bar{u}} \leq \tau_{st}$

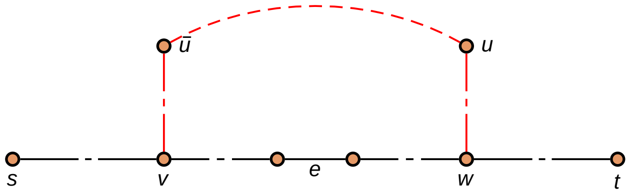
$$\xi_s(e) = \frac{1}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{2}{3}\xi(e)$$

$s, t$ obtain at least $\frac{1}{3}\xi(e)$ for each $e \in B \cup R$

**Assume:** $\sum_{e \in B} \xi(e) \geq \sum_{e \in R} \xi(e)$

Total cost share obtained for $B \cup R$ is at least $\frac{2}{3}\sum_{e \in R} \xi(e)$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$



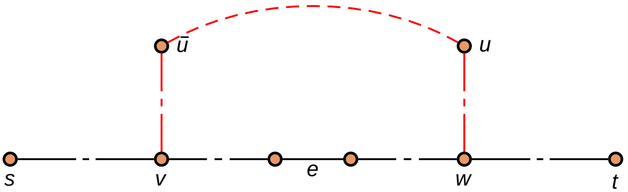

Total cost share obtained for $B \cup R$ is at least $\frac{2}{3} \sum_{e \in R} \xi(e)$

Reconstructing path $\langle v, \bar{u}, u, w \rangle$ costs at most $2 \sum_{e \in R} \xi(e)$

$\Rightarrow 3 \sum_{e \in R} \xi_{st}(e)$ is sufficient to reconstruct this path

Similar argument applies when $\sum_{e \in R} \xi(e) > \sum_{e \in B} \xi(e)$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$



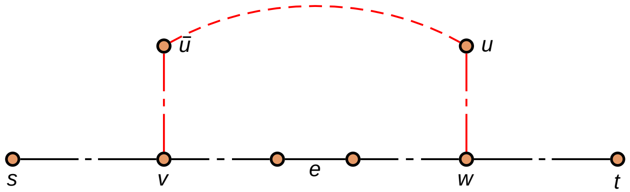

Total cost share obtained for $B \cup R$ is at least $\frac{2}{3} \sum_{e \in R} \xi(e)$

Reconstructing path $\langle v, \bar{u}, u, w \rangle$ costs at most $2 \sum_{e \in R} \xi(e)$

$\Rightarrow 3 \sum_{e \in R} \xi_{st}(e)$ is sufficient to reconstruct this path

Similar argument applies when $\sum_{e \in R} \xi(e) > \sum_{e \in B} \xi(e)$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$



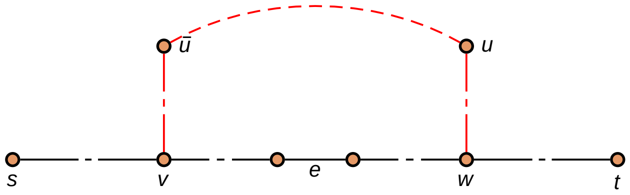

Total cost share obtained for $B \cup R$ is at least $\frac{2}{3} \sum_{e \in R} \xi(e)$

Reconstructing path $\langle v, \bar{u}, u, w \rangle$ costs at most $2 \sum_{e \in R} \xi(e)$

$\Rightarrow 3 \sum_{e \in R} \xi_{st}(e)$ is sufficient to reconstruct this path

Similar argument applies when $\sum_{e \in R} \xi(e) > \sum_{e \in B} \xi(e)$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} \leq \tau_{st}$



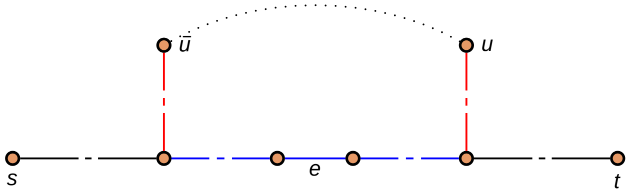

Total cost share obtained for $B \cup R$ is at least $\frac{2}{3} \sum_{e \in R} \xi(e)$

Reconstructing path $\langle v, \bar{u}, u, w \rangle$ costs at most $2 \sum_{e \in R} \xi(e)$

$\Rightarrow 3 \sum_{e \in R} \xi_{st}(e)$ is sufficient to reconstruct this path

Similar argument applies when $\sum_{e \in R} \xi(e) > \sum_{e \in B} \xi(e)$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} > \tau_{st}$




There may be red edges $e \in R$ that are not part of $F_{-st}$ and $\mathcal{W}_e \cap \{s, t\} = \emptyset$!

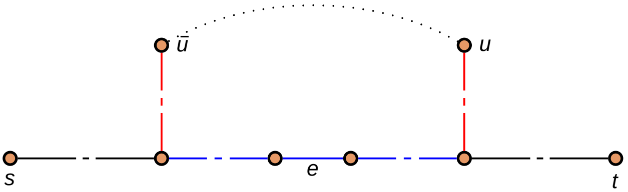
**Cost Share Distribution:** If $e \in P_{u\bar{u}}$ and $\tau_{u\bar{u}} > \tau_{st}$

$$\xi_s(e) = \frac{2}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{1}{3}\xi(e)$$

**Note:** cost of such an edge $e$ is $2\xi(e)$

$\Rightarrow 3\xi_s(e)$ is sufficient to pay for $e$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} > \tau_{st}$




There may be red edges $e \in R$ that are not part of $F_{-st}$ and $\mathcal{W}_e \cap \{s, t\} = \emptyset$!

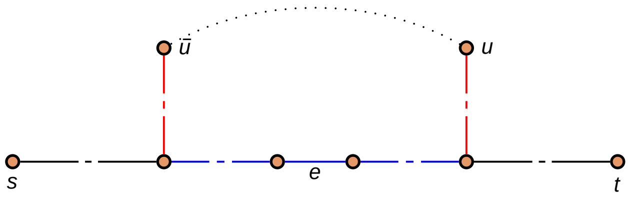
**Cost Share Distribution:** If $e \in P_{u\bar{u}}$ and $\tau_{u\bar{u}} > \tau_{st}$

$$\xi_s(e) = \frac{2}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{1}{3}\xi(e)$$

**Note:** cost of such an edge $e$ is $2\xi(e)$

$\Rightarrow 3\xi_s(e)$ is sufficient to pay for $e$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} > \tau_{st}$




There may be red edges $e \in R$ that are not part of $F_{-st}$ and $\mathcal{W}_e \cap \{s, t\} = \emptyset$!

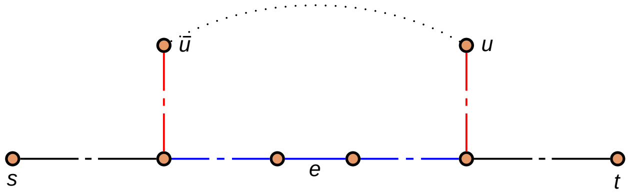
**Cost Share Distribution:** If $e \in P_{u\bar{u}}$ and $\tau_{u\bar{u}} > \tau_{st}$

$$\xi_s(e) = \frac{2}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{1}{3}\xi(e)$$

**Note:** cost of such an edge $e$ is $2\xi(e)$

$\Rightarrow 3\xi_s(e)$ is sufficient to pay for $e$

# Edges needed by $(u, \bar{u})$: $\tau_{u\bar{u}} > \tau_{st}$




There may be red edges $e \in R$ that are not part of $F_{-st}$ and $\mathcal{W}_e \cap \{s, t\} = \emptyset$!

**Cost Share Distribution:** If $e \in P_{u\bar{u}}$ and $\tau_{u\bar{u}} > \tau_{st}$

$$\xi_s(e) = \frac{2}{3}\xi(e) \qquad \text{and} \qquad \xi_u(e) = \frac{1}{3}\xi(e)$$

**Note:** cost of such an edge $e$ is $2\xi(e)$

$\Rightarrow 3\xi_s(e)$ is sufficient to pay for $e$

# Putting All Pieces Together

There may also be edges $e \in P_{st}$ that are witnessed by $s$ and $t$.

$\Rightarrow \xi_{st}(e) = \xi(e)$ and thus $2\xi_{st}(e)$ is sufficient to pay for $e$

**Recall:** for every edge $e \in F$ with $\mathcal{W}_e = \{u, v\}$

$$\xi_u(e) + \xi_v(e) = \xi(e)$$

Altogether

$$\sum_{(s,t)\in R} \xi_{st} \leq \frac{1}{2} c(F) \leq OPT$$

**Thus:** AKR is 2-approximate and 3-strict

# Putting All Pieces Together

There may also be edges $e \in P_{st}$ that are witnessed by $s$ and $t$.
$\Rightarrow \xi_{st}(e) = \xi(e)$ and thus $2\xi_{st}(e)$ is sufficient to pay for $e$

**Recall:** for every edge $e \in F$ with $\mathcal{W}_e = \{u, v\}$

$$\xi_u(e) + \xi_v(e) = \xi(e)$$

Altogether

$$\sum_{(s,t)\in R} \xi_{st} \leq \frac{1}{2} c(F) \leq OPT$$

**Thus:** AKR is 2-approximate and 3-strict

# Putting All Pieces Together

There may also be edges $e \in P_{st}$ that are witnessed by $s$ and $t$.
$\Rightarrow \xi_{st}(e) = \xi(e)$ and thus $2\xi_{st}(e)$ is sufficient to pay for $e$

**Recall:** for every edge $e \in F$ with $\mathcal{W}_e = \{u, v\}$

$$\xi_u(e) + \xi_v(e) = \xi(e)$$

Altogether

$$\sum_{(s,t)\in R} \xi_{st} \leq \frac{1}{2} c(F) \leq OPT$$

**Thus:** AKR is 2-approximate and 3-strict

# Putting All Pieces Together

There may also be edges $e \in P_{st}$ that are witnessed by $s$ and $t$.
$\Rightarrow \xi_{st}(e) = \xi(e)$ and thus $2\xi_{st}(e)$ is sufficient to pay for $e$

**Recall:** for every edge $e \in F$ with $\mathcal{W}_e = \{u, v\}$

$$\xi_u(e) + \xi_v(e) = \xi(e)$$

Altogether

$$\sum_{(s,t) \in R} \xi_{st} \leq \frac{1}{2} c(F) \leq OPT$$

**Thus:** AKR is 2-approximate and 3-strict

# Putting All Pieces Together

There may also be edges $e \in P_{st}$ that are witnessed by $s$ and $t$.
$\Rightarrow \xi_{st}(e) = \xi(e)$ and thus $2\xi_{st}(e)$ is sufficient to pay for $e$

**Recall:** for every edge $e \in F$ with $\mathcal{W}_e = \{u, v\}$

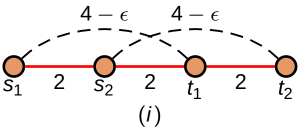
$$\xi_u(e) + \xi_v(e) = \xi(e)$$

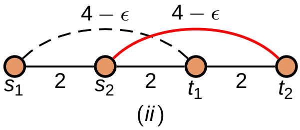
Altogether

$$\sum_{(s,t) \in R} \xi_{st} \leq \frac{1}{2} c(F) \leq OPT$$

**Thus:** AKR is 2-approximate and 3-strict

# Strictness Lower Bound




The forest that AKR computes in (i) has cost 6 and the maximum total cost share that can be distributed is 3.

**Assume:** $\xi_{s_1 t_1} \leq \frac{3}{2}$

Running AKR on terminal pairs $R_{-s_1 t_1}$ yields the forest in (ii) of cost $4 - \epsilon$.

**Thus:** lower bound of $8/3$ for strictness of AKR

# Extensions

**Generalized Steiner Forest:** find a minimum cost forest that connects a given set $R = \{g_1, \ldots, g_k\}$ of terminal groups $g_i \subseteq V$.

AKR also yields 2-approximate and 4-strict algorithm for generalized Steiner forest
$\Rightarrow$ 6-approximation for the multicast rent-or-buy problem

**Open Problem:** Can the asymmetric 3-strict cost sharing scheme be adapted to work for generalized Steiner forest?

# Extensions

**Generalized Steiner Forest:** find a minimum cost forest that connects a given set $R = \{g_1, \ldots, g_k\}$ of terminal groups $g_i \subseteq V$.

AKR also yields 2-approximate and 4-strict algorithm for generalized Steiner forest
$\Rightarrow$ 6-approximation for the multicast rent-or-buy problem

**Open Problem:** Can the asymmetric 3-strict cost sharing scheme be adapted to work for generalized Steiner forest?

# Extensions

**Generalized Steiner Forest:** find a minimum cost forest that connects a given set $R = \{g_1, \ldots, g_k\}$ of terminal groups $g_i \subseteq V$.

AKR also yields 2-approximate and 4-strict algorithm for generalized Steiner forest
$\Rightarrow$ 6-approximation for the multicast rent-or-buy problem

**Open Problem:** Can the asymmetric 3-strict cost sharing scheme be adapted to work for generalized Steiner forest?

# Further Results

**Derandomization:** some Sample-and-Augment algorithms for network design problems can be derandomized

- single-sink rent-or-buy
- virtual private network design
- single-sink buy-at-bulk

[Van Zuylen, Algorithmica ’09]

**Open Problem:** derandomize the Sample-and-Augment algorithm for MROB

# Connections to Stochastic Optimization

# Stochastic Optimization with Recourse

**Stochastic Steiner Tree:**

- network $N = (V, E, c)$, root vertex $r \in V$, terminal set $R \subseteq V$
- probability distribution $\pi : 2^R \to [0, 1]$ (sampling oracle)
  $\pi(S) =$ probability that terminal set $S \subseteq R$ realizes
- inflation factor $\sigma > 1$

**Stage 1:** choose a subset $E_1$ of edges at cost $c(E_1)$

**Stage 2:** actual set $S$ of terminal realizes: augment $E_1$ to a feasible Steiner tree solution for $S \cup \{r\}$ by adding a set of edges $E_S$ at cost $\sigma c(E_S)$

**Objective:** minimize $c(E_1) + \sigma \mathbf{E}[c(E_S)]$

# Boosted Sampling Framework

**Boost-and-Sample:**

1: Sample $\sigma$ times from $\pi$ to obtain terminal sets $D_1, \ldots, D_\sigma$
2: Build $\alpha$-approximate Steiner tree $T$ for $D = \cup_i D_i \cup \{r\}$
3: When actual set $S$ realizes, augment $T$ to feasible solution for $S \cup \{r\}$

**Theorem**

*Given an $\alpha$-approximate and $\beta$-group-strict Steiner tree algorithm, Boost-and-Sample is an $(\alpha + \beta)$-approximation algorithm for stochastic Steiner tree.*

[Gupta, Pál, Ravi, Sinha, STOC '04]

**Remark:** framework applies to stochastic version of optimization problems that are sub-additive, e.g., facility location, vertex cover, Steiner forest, etc.

# Group-Strictness

## Definition

A Steiner tree algorithm is $\beta$-group-strict if there exists a cost share $\xi_t \geq 0$ for every $t \in R$ such that

1. $\sum_{t \in R} \xi_t \leq c(F^*)$
2. for every $S \subseteq R$, $c_{G|F_{-S}}(S) \leq \beta \cdot \sum_{t \in S} \xi_t$

## Group-Strict Algorithms:

| **Problem** | $\alpha$ | $\beta$ | $\alpha + \beta$ |
|---|---|---|---|
| Steiner tree | 1.55 | 2 | 3.55 |
| facility location | 3 | 5.45 | 8.45 |
| vertex cover | 2 | 6 | 8 |

**Question:** How about group-strict Steiner forest algorithm?

(see also [Gupta, Kumar, STOC '09])

# Open Problem

Develop an $\alpha$-approximate and $\beta$-group-strict Steiner forest algorithm with $\alpha, \beta \in O(1)$.

**Reward:** bottle of champagne

# Open Problem

Develop an $\alpha$-approximate and $\beta$-group-strict Steiner forest algorithm with $\alpha, \beta \in O(1)$.

**Reward:** bottle of champagne

# Strict Steiner Forest Algorithm

**Independent Decision Model:** every terminal $t \in R$ is realized independently with probability $\pi_t$

**Theorem**

*Given an $\alpha$-approximate and $\beta$-strict Steiner tree algorithm, Boost-and-Sample is an $(\alpha + \beta)$-approximation algorithm for stochastic Steiner tree in the independent decision model.*

[Gupta, Pál, Ravi, Sinha, STOC '04]

**Further Consequences:**

- 5-approximation algorithm for stochastic Steiner forest in the independent decision model
- 6-approximation algorithm for stochastic Steiner tree without a fixed root in the general model

[Gupta, Pál, ICALP '05]

[Fleischer, Könemann, Leonardi, Schäfer, STOC '06]

# Strict Steiner Forest Algorithm

**Independent Decision Model:** every terminal $t \in R$ is realized independently with probability $\pi_t$

## Theorem

*Given an $\alpha$-approximate and $\beta$-strict Steiner tree algorithm, Boost-and-Sample is an $(\alpha + \beta)$-approximation algorithm for stochastic Steiner tree in the independent decision model.*

[Gupta, Pál, Ravi, Sinha, STOC '04]

**Further Consequences:**

- 5-approximation algorithm for stochastic Steiner forest in the independent decision model
- 6-approximation algorithm for stochastic Steiner tree without a fixed root in the general model

[Gupta, Pál, ICALP '05]
[Fleischer, Könemann, Leonardi, Schäfer, STOC '06]

# Approximation Algorithms for Connected Facility Location

# Connected Facility Location

**Given:**

- graph $G = (V, E)$ with edge cost $c : E \to \mathbb{R}^+$
- set of facilities $\mathcal{F} \subseteq V$ with opening cost $f_i$ for every $i \in \mathcal{F}$
- set of clients $\mathcal{D} \subseteq V$ with demand $d_j$ for every $j \in \mathcal{D}$
  (can assume without loss of generality $d_j = 1$ for every $j$)
- parameter $M \geq 1$

**Goal:**

- determine a subset $F \subseteq \mathcal{F}$ of facilities to be opened
- assign each client $j \in \mathcal{D}$ to some open facility $\sigma(j) \in F$
- build a Steiner tree $T$ on $F$ so as to minimize

$$\sum_{i \in F} f_i + M \sum_{e \in T} c_e + \sum_{j \in \mathcal{D}} d_j \cdot \ell(j, \sigma(j))$$

$\ell(u, v)$ = shortest path distance between nodes $u$ and $v$ in $G$

# Connected Facility Location

## Given:

- graph $G = (V, E)$ with edge cost $c : E \to \mathbb{R}^+$
- set of facilities $\mathcal{F} \subseteq V$ with opening cost $f_i$ for every $i \in \mathcal{F}$
- set of clients $\mathcal{D} \subseteq V$ with demand $d_j$ for every $j \in \mathcal{D}$ (can assume without loss of generality $d_j = 1$ for every $j$)
- parameter $M \geq 1$

## Goal:

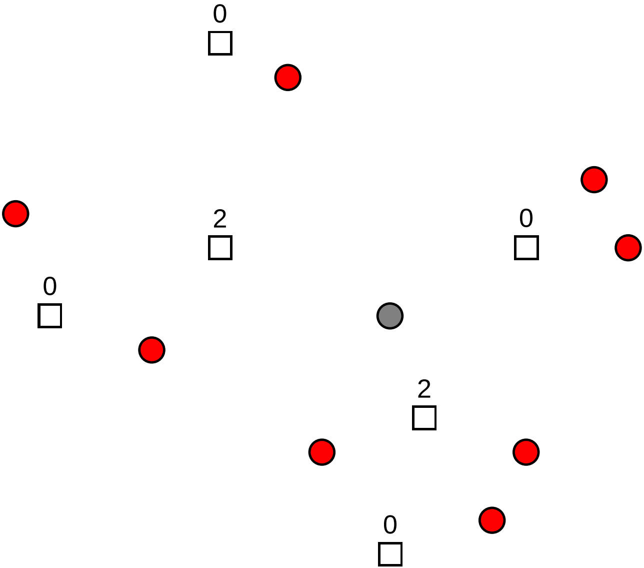
- determine a subset $F \subseteq \mathcal{F}$ of facilities to be opened
- assign each client $j \in \mathcal{D}$ to some open facility $\sigma(j) \in F$
- build a Steiner tree $T$ on $F$ so as to minimize

$$\sum_{i \in F} f_i + M \sum_{e \in T} c_e + \sum_{j \in \mathcal{D}} d_j \cdot \ell(j, \sigma(j))$$

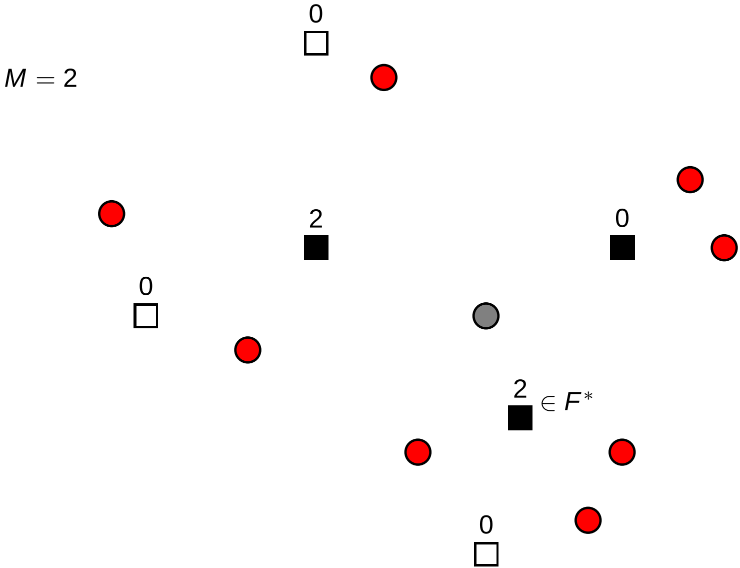
$\ell(u, v)$ = shortest path distance between nodes $u$ and $v$ in $G$

# Example: Connected Facility Location

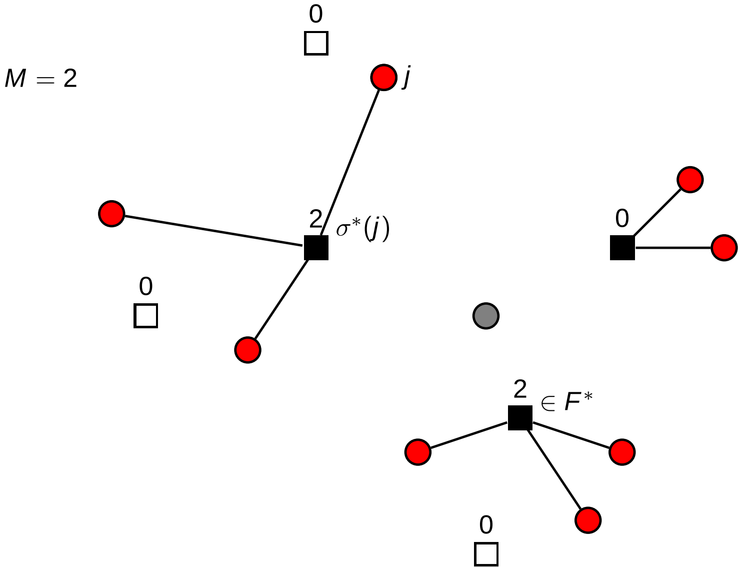
$M = 2$



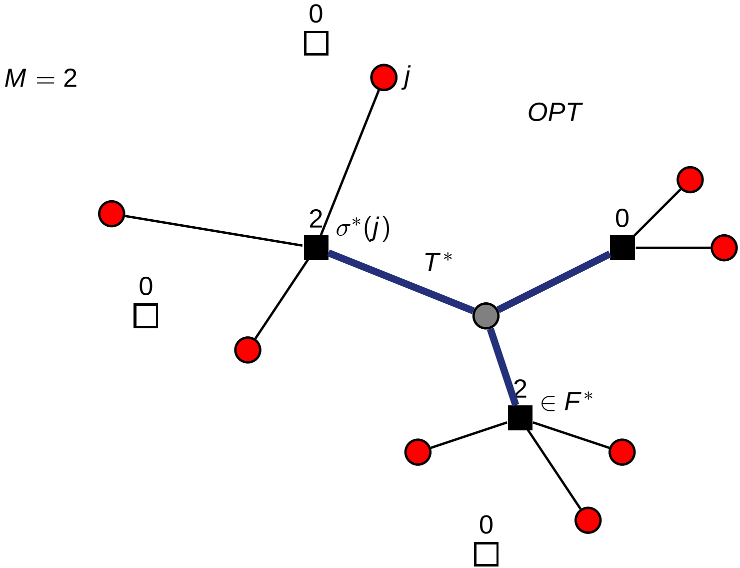
# Example: Connected Facility Location



# Example: Connected Facility Location



# Example: Connected Facility Location



# Variants

**Single-Sink Rent-or-Buy:** special case of CFL where every node is a facility with zero opening cost

**Connected $k$-Facility Location:** can open at most $k$ facilities

**Connected Soft-Capacitated Facility Location:**

- every facility $i$ can serve at most $b_i$ clients
- can open several copies of each facility $i$ (incurring opening cost $f_i$ each time)

**Tour-Connected Facility Location:** connect open facilities by a minimum-cost traveling salesman tour

# Contributions

We give simple and currently best approximation algorithms for all mentioned variants of the connected facility location problem

**Obstacles**: need to incorporate that facilities can only be opened at certain nodes and incur some opening cost

**Naïve Two-Level Approach:**

1. solve the (unconnected) facility location problem
2. build a Steiner tree on top of the opened facilities

$\Rightarrow$ fails because of prohibitively large Steiner tree cost due to outlier facilities

**High-Level Idea:** use random sampling approach to choose a good subset of the facilities opened in the unconnected facility location solution

# Our Algorithm

## Algorithm randCFL

1: Compute a $\rho_{fl}$-approximate solution $(F_U, \sigma_U)$ for the unconnected facility location instance.

2: Mark a client $j^* \in \mathcal{D}$ uniformly at random and mark every other client independently with probability $\alpha/M$. Let $D$ be the set of marked clients.

3: Open facility $i \in F_U$ if there is at least one marked client $j$ with $\sigma_U(j) = i$. Let $F$ be the set of open facilities.

4: Compute a $\rho_{st}$-approximate Steiner tree on $D$. Augment this tree by adding the shortest path between every $j \in D$ and the corresponding open facility $\sigma_U(j) \in F$. Extract a tree $T$ spanning $F$ from the resulting multi-graph.

5: Assign each client $j \in \mathcal{D}$ to a closest open facility $\sigma(j) \in F$.

# Our Algorithm

## Algorithm randCFL

1: Compute a $\rho_{fl}$-approximate solution $(F_U, \sigma_U)$ for the unconnected facility location instance.

2: Mark a client $j^* \in \mathcal{D}$ uniformly at random and mark every other client independently with probability $\alpha/M$. Let $D$ be the set of marked clients.

3: Open facility $i \in F_U$ if there is at least one marked client $j$ with $\sigma_U(j) = i$. Let $F$ be the set of open facilities.

4: Compute a $\rho_{st}$-approximate Steiner tree on $D$. Augment this tree by adding the shortest path between every $j \in D$ and the corresponding open facility $\sigma_U(j) \in F$. Extract a tree $T$ spanning $F$ from the resulting multi-graph.

5: Assign each client $j \in \mathcal{D}$ to a closest open facility $\sigma(j) \in F$.

# Our Algorithm

## Algorithm randCFL

1: Compute a $\rho_{fl}$-approximate solution $(F_U, \sigma_U)$ for the unconnected facility location instance.

2: Mark a client $j^* \in \mathcal{D}$ uniformly at random and mark every other client independently with probability $\alpha/M$. Let $D$ be the set of marked clients.

3: Open facility $i \in F_U$ if there is at least one marked client $j$ with $\sigma_U(j) = i$. Let $F$ be the set of open facilities.

4: Compute a $\rho_{st}$-approximate Steiner tree on $D$. Augment this tree by adding the shortest path between every $j \in D$ and the corresponding open facility $\sigma_U(j) \in F$. Extract a tree $T$ spanning $F$ from the resulting multi-graph.

5: Assign each client $j \in \mathcal{D}$ to a closest open facility $\sigma(j) \in F$.

# Our Algorithm

## Algorithm randCFL

1: Compute a $\rho_{fl}$-approximate solution $(F_U, \sigma_U)$ for the unconnected facility location instance.

2: Mark a client $j^* \in \mathcal{D}$ uniformly at random and mark every other client independently with probability $\alpha/M$. Let $D$ be the set of marked clients.

3: Open facility $i \in F_U$ if there is at least one marked client $j$ with $\sigma_U(j) = i$. Let $F$ be the set of open facilities.

4: Compute a $\rho_{st}$-approximate Steiner tree on $D$. Augment this tree by adding the shortest path between every $j \in D$ and the corresponding open facility $\sigma_U(j) \in F$. Extract a tree $T$ spanning $F$ from the resulting multi-graph.

5: Assign each client $j \in \mathcal{D}$ to a closest open facility $\sigma(j) \in F$.

# Our Algorithm

## Algorithm randCFL

1: Compute a $\rho_{fl}$-approximate solution $(F_U, \sigma_U)$ for the unconnected facility location instance.

2: Mark a client $j^* \in \mathcal{D}$ uniformly at random and mark every other client independently with probability $\alpha/M$. Let $D$ be the set of marked clients.

3: Open facility $i \in F_U$ if there is at least one marked client $j$ with $\sigma_U(j) = i$. Let $F$ be the set of open facilities.

4: Compute a $\rho_{st}$-approximate Steiner tree on $D$. Augment this tree by adding the shortest path between every $j \in D$ and the corresponding open facility $\sigma_U(j) \in F$. Extract a tree $T$ spanning $F$ from the resulting multi-graph.

5: Assign each client $j \in \mathcal{D}$ to a closest open facility $\sigma(j) \in F$.

# Our Algorithm

## Algorithm randCFL

1: Compute a $\rho_{fl}$-approximate solution $(F_U, \sigma_U)$ for the unconnected facility location instance.

2: Mark a client $j^* \in \mathcal{D}$ uniformly at random and mark every other client independently with probability $\alpha/M$. Let $D$ be the set of marked clients.

3: Open facility $i \in F_U$ if there is at least one marked client $j$ with $\sigma_U(j) = i$. Let $F$ be the set of open facilities.

4: Compute a $\rho_{st}$-approximate Steiner tree on $D$. Augment this tree by adding the shortest path between every $j \in D$ and the corresponding open facility $\sigma_U(j) \in F$. Extract a tree $T$ spanning $F$ from the resulting multi-graph.

5: Assign each client $j \in \mathcal{D}$ to a closest open facility $\sigma(j) \in F$.

# Our Algorithm

## Algorithm randCFL

1: Compute a $\rho_{fl}$-approximate solution $(F_U, \sigma_U)$ for the unconnected facility location instance.

2: Mark a client $j^* \in \mathcal{D}$ uniformly at random and mark every other client independently with probability $\alpha/M$. Let $D$ be the set of marked clients.

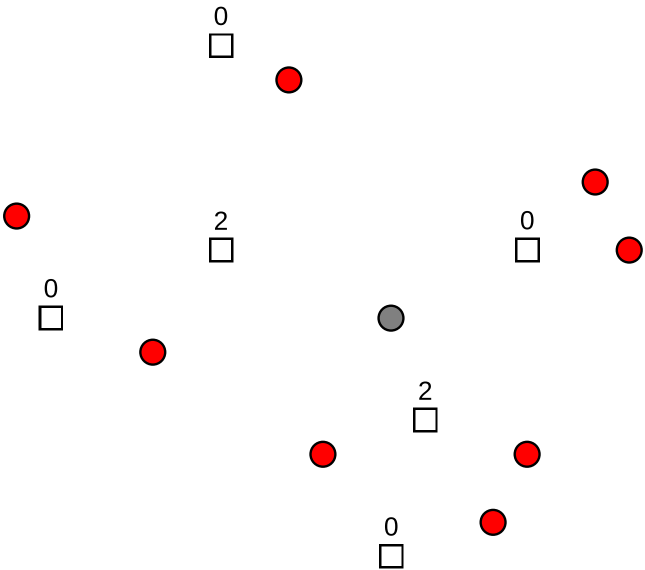
3: Open facility $i \in F_U$ if there is at least one marked client $j$ with $\sigma_U(j) = i$. Let $F$ be the set of open facilities.

4: Compute a $\rho_{st}$-approximate Steiner tree on $D$. Augment this tree by adding the shortest path between every $j \in D$ and the corresponding open facility $\sigma_U(j) \in F$. Extract a tree $T$ spanning $F$ from the resulting multi-graph.

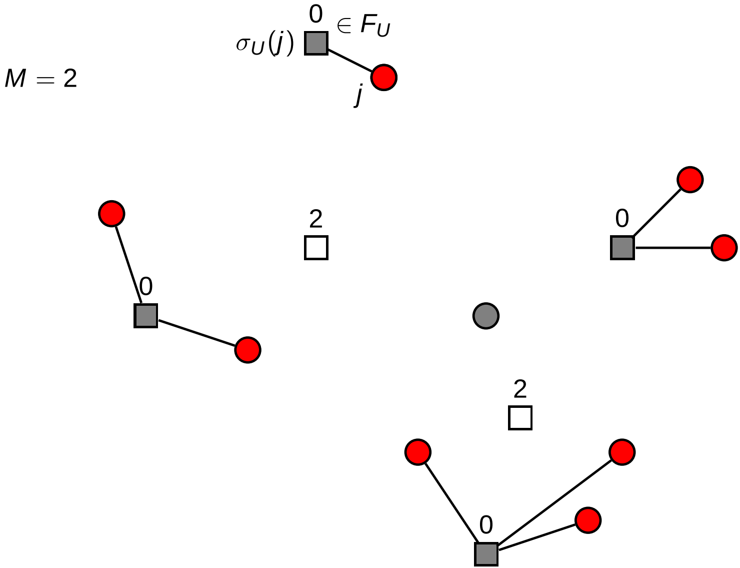
5: Assign each client $j \in \mathcal{D}$ to a closest open facility $\sigma(j) \in F$.

# Illustration: randCFL

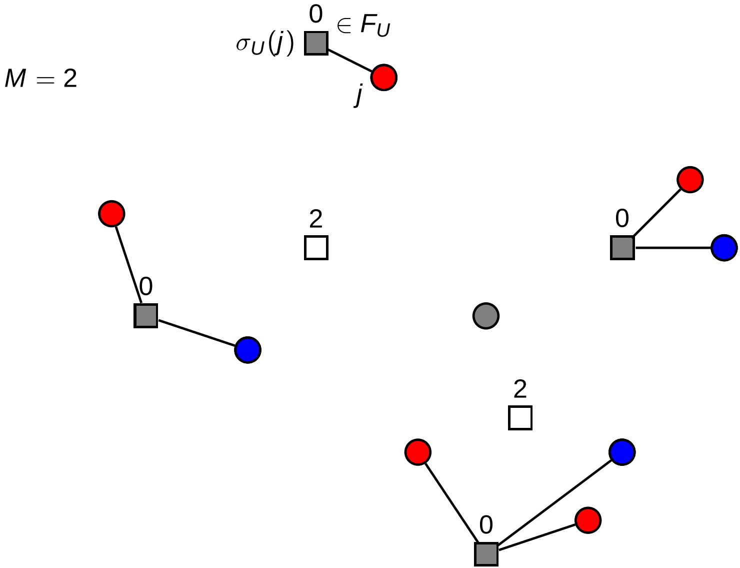
$M = 2$



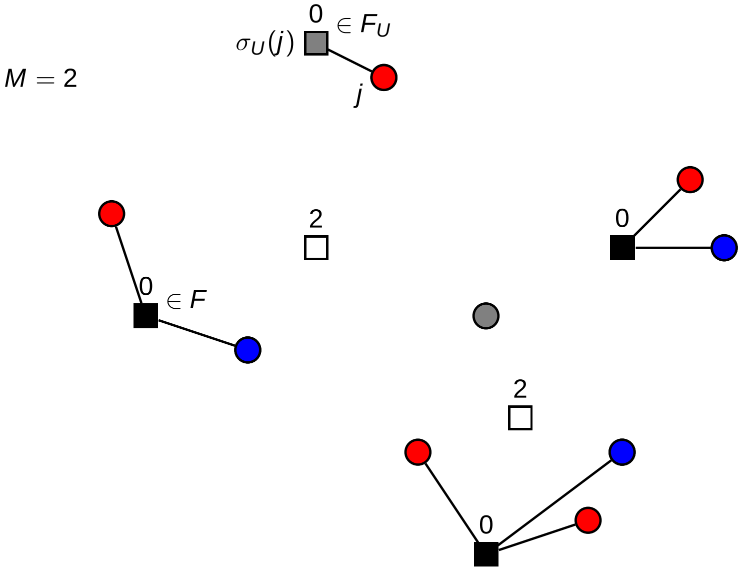
# Illustration: randCFL



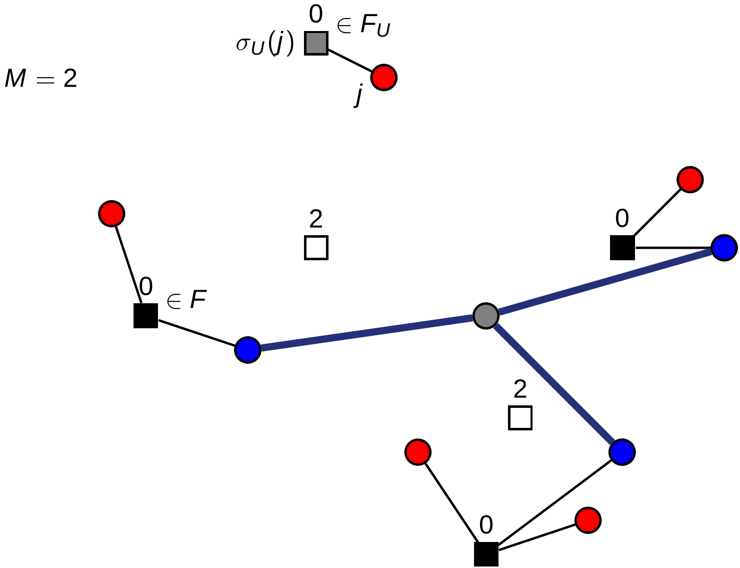
# Illustration: randCFL



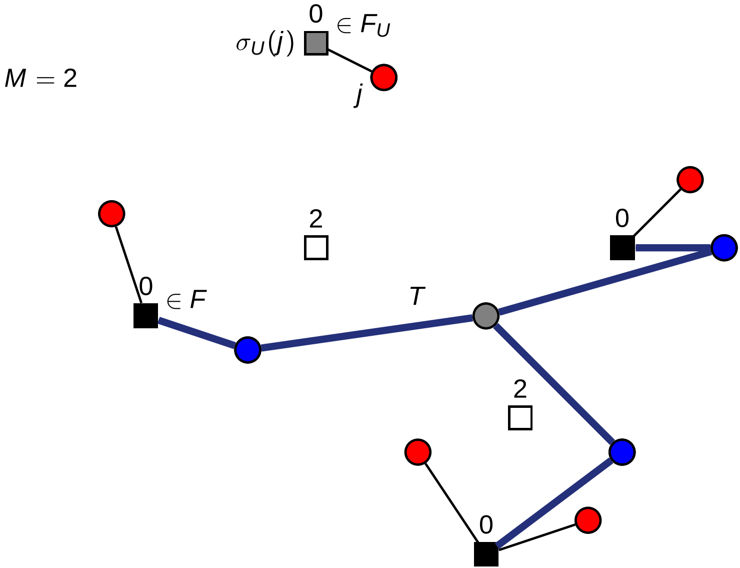
# Illustration: randCFL



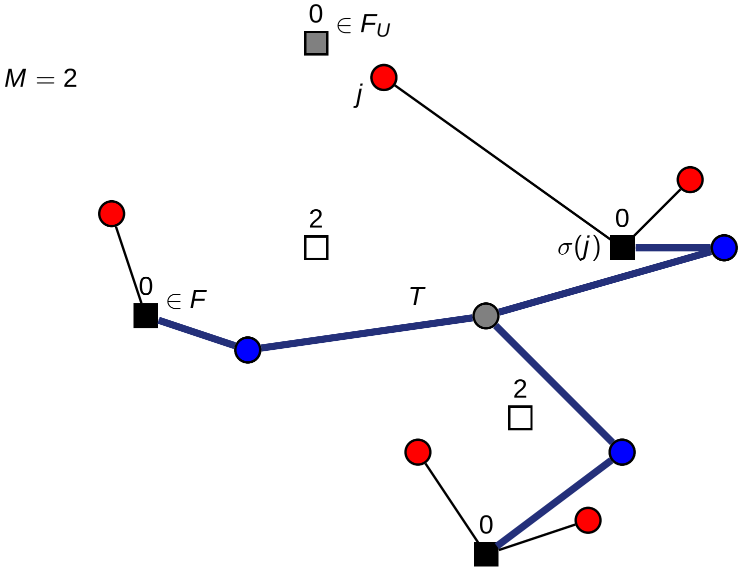
# Illustration: randCFL



# Illustration: randCFL



# Illustration: randCFL



# Main Results

## Theorem

*Algorithm randCFL is an (expected)*

- *4.55-approximation algorithm for connected facility location*
- *3.05-approximation algorithm for single-sink rent-or-buy.*

[Eisenbrand, Grandoni, Rothvoß, Schäfer, JCSS '10]

**Remarks:**

- stated approximation guarantees are with respect to (previously) best approximation guarantees
  - $\rho_{fl} = 1.52$ for facility location [Mahdian, Ye, Zhang, APPROX '03]
  - $\rho_{st} = 1.55$ for Steiner tree [Robins, Zelikovsky, SODA '00]
- obtain slightly superior results by using recent improvements
  - $\rho_{fl} = 1.5$ for facility location [Byrka, Aardal, SICOMP '10+]
  - $\rho_{st} \approx 1.39$ for Steiner tree [Byrka, Grandoni, Rothvoß, Sanita, STOC '10]

# Main Results

## Theorem

*Algorithm randCFL is an (expected)*

- *4.55-approximation algorithm for connected facility location*
- *3.05-approximation algorithm for single-sink rent-or-buy.*

[Eisenbrand, Grandoni, Rothvoß, Schäfer, JCSS '10]

**Remarks:**

- stated approximation guarantees are with respect to (previously) best approximation guarantees
  - $\rho_{fl} = 1.52$ for facility location [Mahdian, Ye, Zhang, APPROX '03]
  - $\rho_{st} = 1.55$ for Steiner tree [Robins, Zelikovsky, SODA '00]
- obtain slightly superior results by using recent improvements
  - $\rho_{fl} = 1.5$ for facility location [Byrka, Aardal, SICOMP '10+]
  - $\rho_{st} \approx 1.39$ for Steiner tree [Byrka, Grandoni, Rothvoß, Sanita, STOC '10]

# Analysis

Let $(F^*, T^*, \sigma^*)$ be an optimal solution for the CFL instance of cost

$$\underbrace{\sum_{i \in F^*} f_i}_{\text{opening cost } O^*} + \underbrace{M \sum_{e \in T^*} c_e}_{\text{Steiner cost } S^*} + \underbrace{\sum_{j \in \mathcal{D}} \ell(j, \sigma^*(j))}_{\text{connection cost } C^*}$$

## Theorem

*If $|\mathcal{D}|/M = O(1)$ then there is a polynomial-time approximation scheme for CFL.*

**Assumption:** $M/|\mathcal{D}| \leq \epsilon$ for sufficiently small $\epsilon > 0$

# Steiner Cost

**Lemma**

*The Steiner cost is at most* $\rho_{st}(S^* + (\alpha + \epsilon)\, C^*) + (\alpha + \epsilon) C_U$.

**Proof:**

Augment the Steiner tree $T^*$ to a feasible Steiner tree on $D$ by adding the shortest path from each client in $D$ to $T^*$:

$$\sum_{e \in T^*} c(e) + \sum_{j \in \mathcal{D}} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) \ell(j, F^*) = \frac{1}{M} S^* + \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C^*$$

Thus the expected cost of the $\rho_{st}$-approximate Steiner tree over $D$ computed in Step 4 is at most

$$\frac{\rho_{st}}{M} S^* + \rho_{st} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C^*$$

# Steiner Cost

**Lemma**

*The Steiner cost is at most* $\rho_{st}(S^* + (\alpha + \epsilon)\, C^*) + (\alpha + \epsilon) C_U$.

**Proof:**

Augment the Steiner tree $T^*$ to a feasible Steiner tree on $D$ by adding the shortest path from each client in $D$ to $T^*$:

$$\sum_{e \in T^*} c(e) + \sum_{j \in \mathcal{D}} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) \ell(j, F^*) = \frac{1}{M} S^* + \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C^*$$

Thus the expected cost of the $\rho_{st}$-approximate Steiner tree over $D$ computed in Step 4 is at most

$$\frac{\rho_{st}}{M} S^* + \rho_{st} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C^*$$

# Steiner Cost

## Lemma

*The Steiner cost is at most* $\rho_{st}(S^* + (\alpha + \epsilon)\, C^*) + (\alpha + \epsilon)C_U$.

**Proof:**

Augment the Steiner tree $T^*$ to a feasible Steiner tree on $D$ by adding the shortest path from each client in $D$ to $T^*$:

$$\sum_{e \in T^*} c(e) + \sum_{j \in \mathcal{D}} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) \ell(j, F^*) = \frac{1}{M} S^* + \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C^*$$

Thus the expected cost of the $\rho_{st}$-approximate Steiner tree over $D$ computed in Step 4 is at most

$$\frac{\rho_{st}}{M} S^* + \rho_{st} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C^*$$

# Steiner Cost

## Lemma

*The Steiner cost is at most* $\rho_{st}(S^* + (\alpha + \epsilon)\, C^*) + (\alpha + \epsilon) C_U$.

**Proof:**

The expected cost of adding the shortest paths from each client $j \in D$ to the corresponding open facility $\sigma_U(j) \in F_U$ is at most

$$\sum_{j \in \mathcal{D}} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) \ell(j, F_U) = \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C_U$$

Altogether we obtain

$$\mathbf{E}[S] \leq M \left( \frac{\rho_{st}}{M} S^* + \rho_{st} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C^* + \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C_U \right)$$
$$\leq \rho_{st}(S^* + (\alpha + \epsilon)\, C^*) + (\alpha + \epsilon) C_U$$

□

# Steiner Cost

## Lemma

*The Steiner cost is at most* $\rho_{st}(S^* + (\alpha + \epsilon)\, C^*) + (\alpha + \epsilon)C_U$.

**Proof:**

The expected cost of adding the shortest paths from each client $j \in D$ to the corresponding open facility $\sigma_U(j) \in F_U$ is at most

$$\sum_{j \in \mathcal{D}} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) \ell(j, F_U) = \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C_U$$

Altogether we obtain

$$\begin{aligned} \mathbf{E}[S] &\leq M \left( \frac{\rho_{st}}{M} S^* + \rho_{st} \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C^* + \left( \frac{\alpha}{M} + \frac{1}{|\mathcal{D}|} \right) C_U \right) \\ &\leq \rho_{st}(S^* + (\alpha + \epsilon)\, C^*) + (\alpha + \epsilon)C_U \end{aligned}$$

□

# Connection Cost

**Lemma**

*The opening cost is at most $O_U$.*

**Proof:**
Set of opened facilities $F$ is a subset of $F_U$, whose total cost is $O_U$. □

**Lemma**

*The connection cost is at most $2C^* + C_U + S^*/\alpha$.*

**Key Ingredient:** novel core detouring scheme to bound the expected connection cost

# Connection Cost

**Lemma**

*The opening cost is at most $O_U$.*

**Proof:**
Set of opened facilities $F$ is a subset of $F_U$, whose total cost is $O_U$. □

**Lemma**

*The connection cost is at most $2C^* + C_U + S^*/\alpha$.*

**Key Ingredient:** novel core detouring scheme to bound the expected connection cost

# Connection Cost

## Lemma

*The connection cost is at most* $2C^* + C_U + S^*/\alpha$.

**Proof (sketch):**

**Note:** Each client is connected to its closest open facility in $F$.

It suffices to bound the expected cost of an alternative routing scheme that connects every client to some open facility in $F$.

**Idea:** Use the optimal tree $T^*$ as a core through which all clients are routed to some facility in $F$.

# Connection Cost

**Lemma**

*The connection cost is at most $2C^* + C_U + S^*/\alpha$.*

**Proof (sketch):**

**Note:** Each client is connected to its closest open facility in $F$.

It suffices to bound the expected cost of an alternative routing scheme that connects every client to some open facility in $F$.

**Idea:** Use the optimal tree $T^*$ as a core through which all clients are routed to some facility in $F$.

# Connection Cost

**Lemma**

*The connection cost is at most $2C^* + C_U + S^*/\alpha$.*

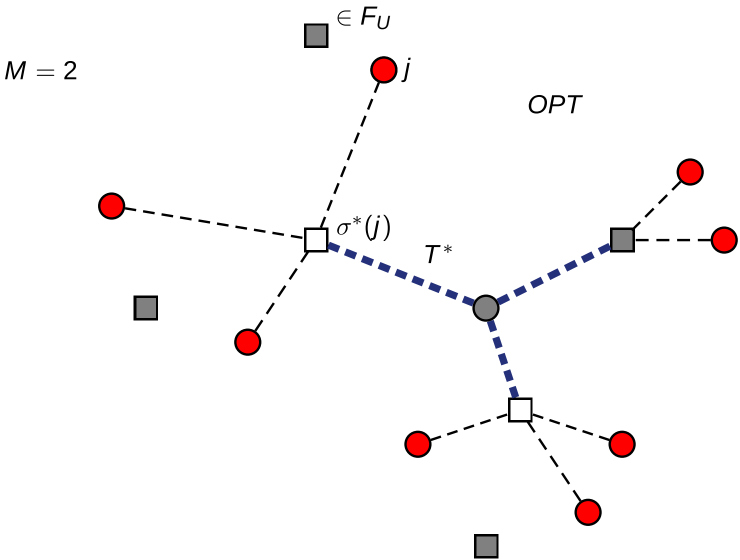
**Proof (sketch):**

**Note:** Each client is connected to its closest open facility in $F$.

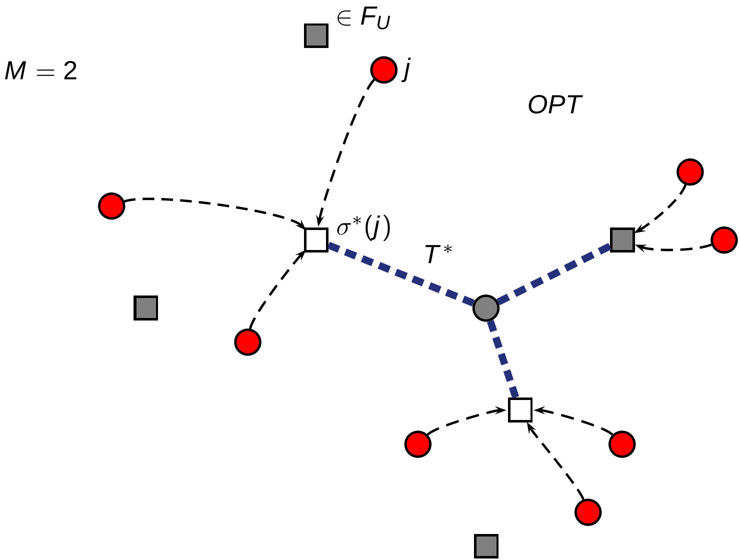
It suffices to bound the expected cost of an alternative routing scheme that connects every client to some open facility in $F$.

**Idea:** Use the optimal tree $T^*$ as a core through which all clients are routed to some facility in $F$.

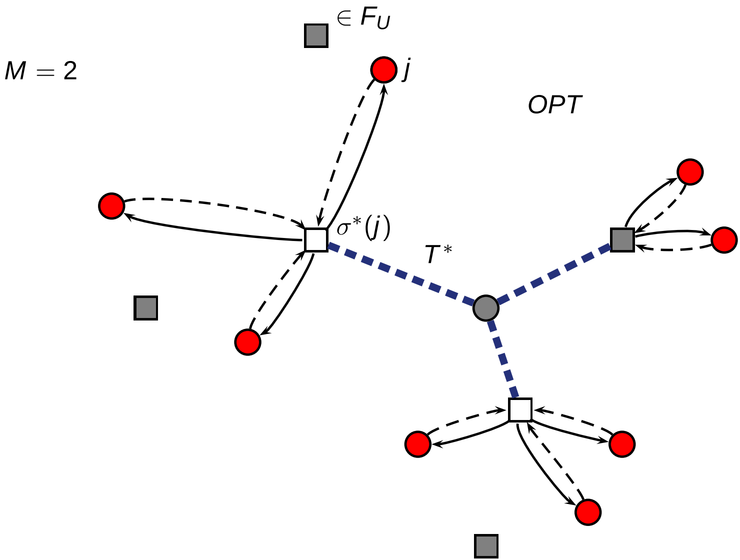
# Core Detouring Scheme



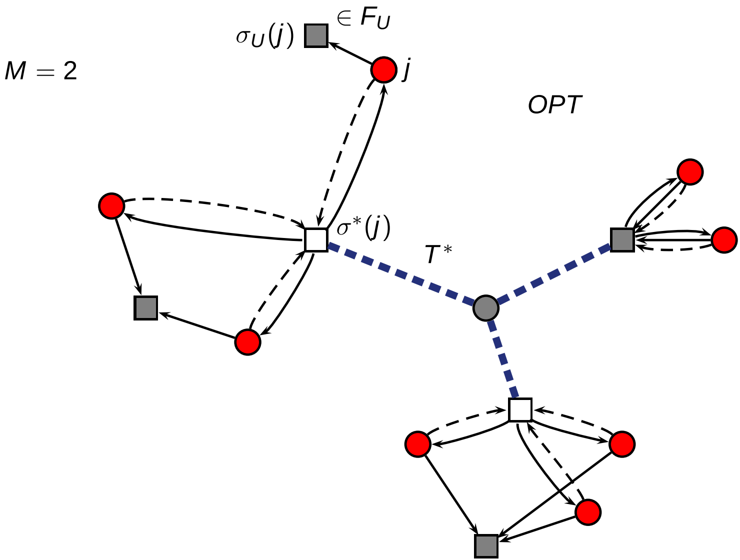
# Core Detouring Scheme



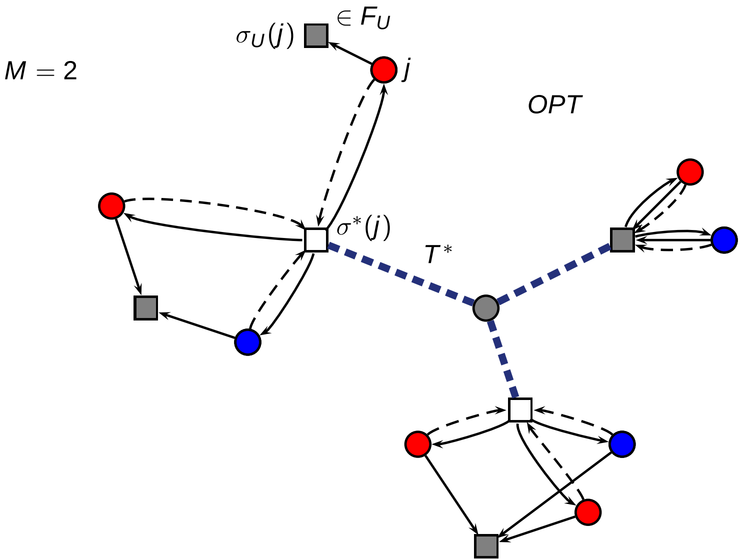
# Core Detouring Scheme



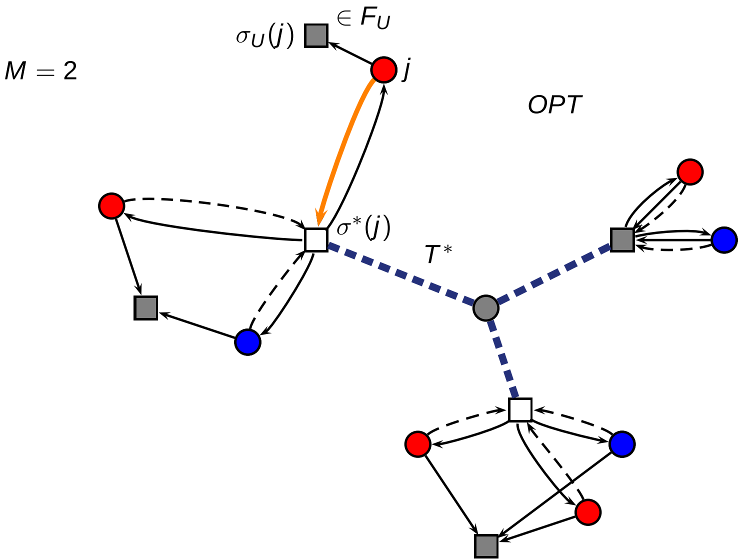
# Core Detouring Scheme



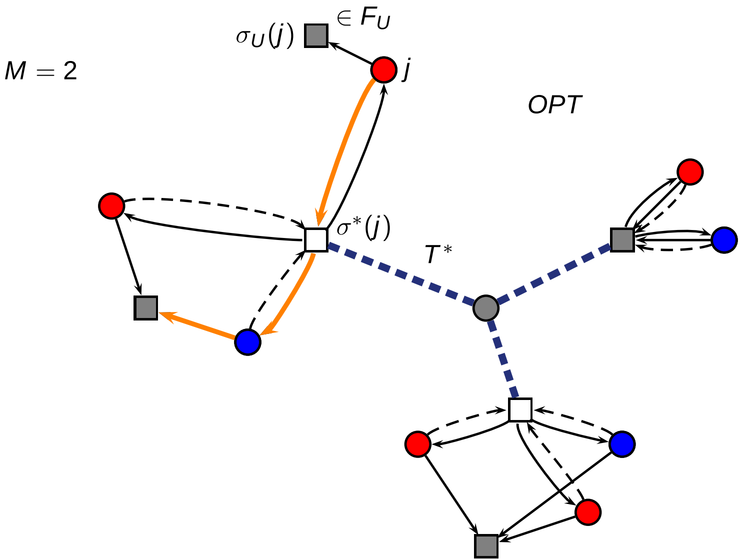
# Core Detouring Scheme



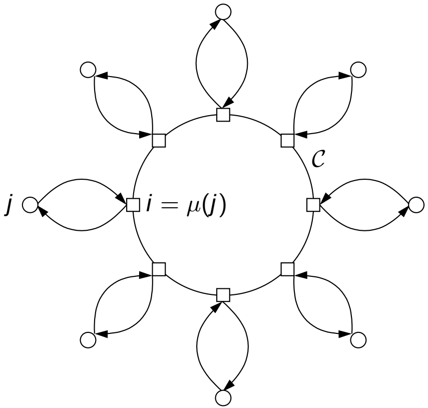
# Core Detouring Scheme



# Core Detouring Scheme



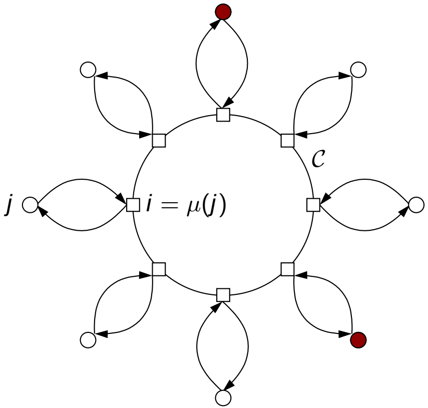
# Cycle-Core Connection Game




**Instance:**

- core nodes connected by undirected cycle $\mathcal{C}$
- each client node $j$ assigned to exactly one core node $\mu(j)$
- $\mathcal{H}_{in}$ and $\mathcal{H}_{out}$ are the edges directed into and out of the core nodes, respectively
- every edge $e$ has a weight $w_e \geq 0$

# Cycle-Core Connection Game

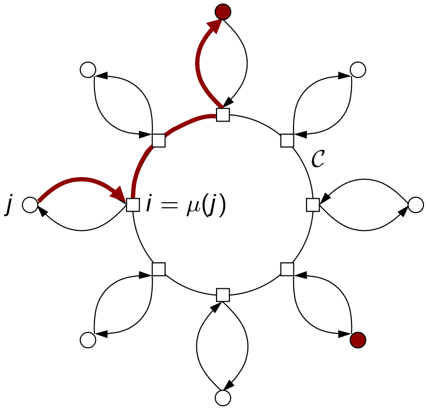



**Random Experiment:**

- mark one client node uniformly at random and every other node independently with probability $p$
- every client node sends one unit of flow to the closest marked client node

**Question:** What is the total cost of the resulting flow?

# Cycle-Core Connection Game

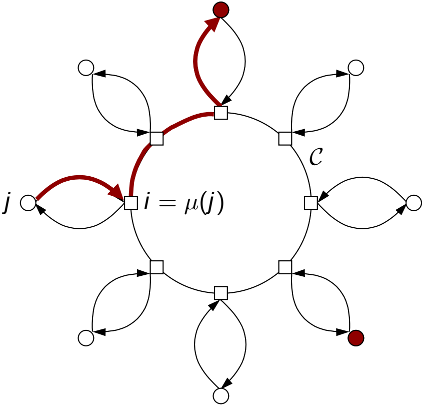



## Random Experiment:

- mark one client node uniformly at random and every other node independently with probability $p$
- every client node sends one unit of flow to the closest marked client node

**Question:** What is the total cost of the resulting flow?

# Cycle-Core Connection Game




**Random Experiment:**

- mark one client node uniformly at random and every other node independently with probability $p$
- every client node sends one unit of flow to the closest marked client node

**Question:** What is the total cost of the resulting flow?

# Cycle-Core Connection Game

**Theorem**

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Consider alternative routing scheme: Each client sends one unit of flow to closest marked client with respect to unit edge weights. Let $f_e$ be the flow on edge $e$. Bound total cost $Y$ with respect to $w$ of $f$. Clearly, $\mathbf{E}[X] \leq \mathbf{E}[Y]$.

$$\mathbf{E}[Y] = \sum_{e \in \mathcal{H}_{in} \cup \mathcal{H}_{out}} \mathbf{E}[f_e] w_e + \sum_{e \in \mathcal{C}} \mathbf{E}[f_e] w_e$$

Consider an edge $e \in \mathcal{H}_{in}$. Then $f_e \leq 1$ (deterministically).
Consider an edge $e \in \mathcal{H}_{out}$. Then $\mathbf{E}[f_e] \leq 1$ (by symmetry).

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Consider alternative routing scheme: Each client sends one unit of flow to closest marked client with respect to unit edge weights. Let $f_e$ be the flow on edge $e$. Bound total cost $Y$ with respect to $w$ of $f$. Clearly, $\mathbf{E}[X] \leq \mathbf{E}[Y]$.

$$\mathbf{E}[Y] = \sum_{e \in \mathcal{H}_{in} \cup \mathcal{H}_{out}} \mathbf{E}[f_e] w_e + \sum_{e \in \mathcal{C}} \mathbf{E}[f_e] w_e$$

Consider an edge $e \in \mathcal{H}_{in}$. Then $f_e \leq 1$ (deterministically).
Consider an edge $e \in \mathcal{H}_{out}$. Then $\mathbf{E}[f_e] \leq 1$ (by symmetry).

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Consider alternative routing scheme: Each client sends one unit of flow to closest marked client with respect to unit edge weights. Let $f_e$ be the flow on edge $e$. Bound total cost $Y$ with respect to $w$ of $f$. Clearly, $\mathbf{E}[X] \le \mathbf{E}[Y]$.

$$\mathbf{E}[Y] = \sum_{e \in \mathcal{H}_{in} \cup \mathcal{H}_{out}} \mathbf{E}[f_e] w_e + \sum_{e \in \mathcal{C}} \mathbf{E}[f_e] w_e$$

Consider an edge $e \in \mathcal{H}_{in}$. Then $f_e \le 1$ (deterministically).
Consider an edge $e \in \mathcal{H}_{out}$. Then $\mathbf{E}[f_e] \le 1$ (by symmetry).

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Consider alternative routing scheme: Each client sends one unit of flow to closest marked client with respect to unit edge weights. Let $f_e$ be the flow on edge $e$. Bound total cost $Y$ with respect to $w$ of $f$. Clearly, $\mathbf{E}[X] \leq \mathbf{E}[Y]$.

$$\mathbf{E}[Y] = \sum_{e \in \mathcal{H}_{in} \cup \mathcal{H}_{out}} \mathbf{E}[f_e] w_e + \sum_{e \in \mathcal{C}} \mathbf{E}[f_e] w_e$$

Consider an edge $e \in \mathcal{H}_{in}$. Then $f_e \leq 1$ (deterministically).
Consider an edge $e \in \mathcal{H}_{out}$. Then $\mathbf{E}[f_e] \leq 1$ (by symmetry).

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Consider alternative routing scheme: Each client sends one unit of flow to closest marked client with respect to unit edge weights. Let $f_e$ be the flow on edge $e$. Bound total cost $Y$ with respect to $w$ of $f$. Clearly, $\mathbf{E}[X] \leq \mathbf{E}[Y]$.

$$\mathbf{E}[Y] = \sum_{e \in \mathcal{H}_{in} \cup \mathcal{H}_{out}} \mathbf{E}[f_e] w_e + \sum_{e \in \mathcal{C}} \mathbf{E}[f_e] w_e$$

Consider an edge $e \in \mathcal{H}_{in}$. Then $f_e \leq 1$ (deterministically).
Consider an edge $e \in \mathcal{H}_{out}$. Then $\mathbf{E}[f_e] \leq 1$ (by symmetry).

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Consider alternative routing scheme: Each client sends one unit of flow to closest marked client with respect to unit edge weights. Let $f_e$ be the flow on edge $e$. Bound total cost $Y$ with respect to $w$ of $f$. Clearly, $\mathbf{E}[X] \leq \mathbf{E}[Y]$.

$$\mathbf{E}[Y] = \sum_{e \in \mathcal{H}_{in} \cup \mathcal{H}_{out}} \mathbf{E}[f_e] w_e + \sum_{e \in \mathcal{C}} \mathbf{E}[f_e] w_e$$

Consider an edge $e \in \mathcal{H}_{in}$. Then $f_e \leq 1$ (deterministically).
Consider an edge $e \in \mathcal{H}_{out}$. Then $\mathbf{E}[f_e] \leq 1$ (by symmetry).

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Consider alternative routing scheme: Each client sends one unit of flow to closest marked client with respect to unit edge weights. Let $f_e$ be the flow on edge $e$. Bound total cost $Y$ with respect to $w$ of $f$. Clearly, $\mathbf{E}[X] \leq \mathbf{E}[Y]$.

$$\mathbf{E}[Y] = \sum_{e \in \mathcal{H}_{in} \cup \mathcal{H}_{out}} \mathbf{E}[f_e] w_e + \sum_{e \in \mathcal{C}} \mathbf{E}[f_e] w_e$$

Consider an edge $e \in \mathcal{H}_{in}$. Then $f_e \leq 1$ (deterministically).

Consider an edge $e \in \mathcal{H}_{out}$. Then $\mathbf{E}[f_e] \leq 1$ (by symmetry).

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Consider alternative routing scheme: Each client sends one unit of flow to closest marked client with respect to unit edge weights. Let $f_e$ be the flow on edge $e$. Bound total cost $Y$ with respect to $w$ of $f$. Clearly, $\mathbf{E}[X] \leq \mathbf{E}[Y]$.

$$\mathbf{E}[Y] = \sum_{e \in \mathcal{H}_{in} \cup \mathcal{H}_{out}} \mathbf{E}[f_e] w_e + \sum_{e \in \mathcal{C}} \mathbf{E}[f_e] w_e$$

Consider an edge $e \in \mathcal{H}_{in}$. Then $f_e \leq 1$ (deterministically).
Consider an edge $e \in \mathcal{H}_{out}$. Then $\mathbf{E}[f_e] \leq 1$ (by symmetry).

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Let $X_j$ be the number of edges in $\mathcal{C}$ used by flow of client $j$.

$$\sum_{e \in \mathcal{C}} f_e = \sum_{j \in \mathcal{D}} X_j \quad \text{and thus by symmetry} \quad \mathbf{E}[f_e] = \mathbf{E}[X_j].$$

Now $X_j > k$ iff $j$ and the first $k$ neighbors to the left and right of $j$ are not marked:

$$Pr(X_j > k) < (1-p)^{2k+1}$$

$$\Rightarrow \mathbf{E}[f_e] = \mathbf{E}[X_j] = \sum_{k \geq 0} Pr(X_j > k) \leq \frac{1-p}{1-(1-p)^2} \leq \frac{1}{2p}. \qquad \square$$

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Let $X_j$ be the number of edges in $\mathcal{C}$ used by flow of client $j$.

$$\sum_{e \in \mathcal{C}} f_e = \sum_{j \in \mathcal{D}} X_j \quad \text{and thus by symmetry} \quad \mathbf{E}[f_e] = \mathbf{E}[X_j].$$

Now $X_j > k$ iff $j$ and the first $k$ neighbors to the left and right of $j$ are not marked:

$$Pr(X_j > k) < (1-p)^{2k+1}$$

$$\Rightarrow \quad \mathbf{E}[f_e] = \mathbf{E}[X_j] = \sum_{k \geq 0} Pr(X_j > k) \leq \frac{1-p}{1-(1-p)^2} \leq \frac{1}{2p}. \qquad \square$$

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Let $X_j$ be the number of edges in $\mathcal{C}$ used by flow of client $j$.

$$\sum_{e \in \mathcal{C}} f_e = \sum_{j \in \mathcal{D}} X_j \quad \text{and thus by symmetry} \quad \mathbf{E}[f_e] = \mathbf{E}[X_j].$$

Now $X_j > k$ iff $j$ and the first $k$ neighbors to the left and right of $j$ are not marked:

$$Pr(X_j > k) < (1-p)^{2k+1}$$

$$\Rightarrow \quad \mathbf{E}[f_e] = \mathbf{E}[X_j] = \sum_{k \geq 0} Pr(X_j > k) \leq \frac{1-p}{1-(1-p)^2} \leq \frac{1}{2p}. \qquad \square$$

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Let $X_j$ be the number of edges in $\mathcal{C}$ used by flow of client $j$.

$$\sum_{e \in \mathcal{C}} f_e = \sum_{j \in \mathcal{D}} X_j \quad \text{and thus by symmetry} \quad \mathbf{E}[f_e] = \mathbf{E}[X_j].$$

Now $X_j > k$ iff $j$ and the first $k$ neighbors to the left and right of $j$ are not marked:

$$Pr(X_j > k) < (1-p)^{2k+1}$$

$$\Rightarrow \mathbf{E}[f_e] = \mathbf{E}[X_j] = \sum_{k \geq 0} Pr(X_j > k) \leq \frac{1-p}{1-(1-p)^2} \leq \frac{1}{2p}. \qquad \square$$

# Cycle-Core Connection Game

## Theorem

*The expected cost $X$ of the flow in the cycle-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{C})/(2p)$.*

**Proof:**

Let $X_j$ be the number of edges in $\mathcal{C}$ used by flow of client $j$.

$$\sum_{e \in \mathcal{C}} f_e = \sum_{j \in \mathcal{D}} X_j \quad \text{and thus by symmetry} \quad \mathbf{E}[f_e] = \mathbf{E}[X_j].$$

Now $X_j > k$ iff $j$ and the first $k$ neighbors to the left and right of $j$ are not marked:

$$Pr(X_j > k) < (1-p)^{2k+1}$$

$$\Rightarrow \mathbf{E}[f_e] = \mathbf{E}[X_j] = \sum_{k \geq 0} Pr(X_j > k) \leq \frac{1-p}{1-(1-p)^2} \leq \frac{1}{2p}. \qquad \square$$

# Tree-Core Connection Game

Suppose core is given by a Steiner tree $\mathcal{T}$ on the core nodes. Every client node is assigned to exactly one core node but a core node can have multiple client nodes assigned to it.

**Theorem**

*The expected cost $X$ of the flow in the tree-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{T})/p$.*

**Proof (sketch):**
Obtain cycle-core connection game by using the standard argument to transform the Steiner tree $\mathcal{T}$ into a cycle of cost at most $2w(\mathcal{T})$ (edge doubling and shortcutting Eulerian tour). $\square$

# Tree-Core Connection Game

Suppose core is given by a Steiner tree $\mathcal{T}$ on the core nodes. Every client node is assigned to exactly one core node but a core node can have multiple client nodes assigned to it.

**Theorem**

*The expected cost $X$ of the flow in the tree-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{T})/p$.*

**Proof (sketch):**
Obtain cycle-core connection game by using the standard argument to transform the Steiner tree $\mathcal{T}$ into a cycle of cost at most $2w(\mathcal{T})$ (edge doubling and shortcutting Eulerian tour). □

# Tree-Core Connection Game

Suppose core is given by a Steiner tree $\mathcal{T}$ on the core nodes. Every client node is assigned to exactly one core node but a core node can have multiple client nodes assigned to it.

**Theorem**

*The expected cost $X$ of the flow in the tree-core connection game is at most $w(\mathcal{H}_{in} \cup \mathcal{H}_{out}) + w(\mathcal{T})/p$.*

**Proof (sketch):**
Obtain cycle-core connection game by using the standard argument to transform the Steiner tree $\mathcal{T}$ into a cycle of cost at most $2w(\mathcal{T})$ (edge doubling and shortcutting Eulerian tour). □

# Putting All Pieces Together

Expected total cost of the constructed solution is at most

$$\rho_{st}(S^* + (\alpha+\epsilon)C^*) + (\alpha+\epsilon)C_U + O_U + 2C^* + C_U + \frac{S^*}{\alpha}$$

$$\overset{(*)}{\leq} \rho_{st}(S^* + (\alpha+\epsilon)C^*) + (1+\alpha+\epsilon)\rho_{fl}(C^* + O^*) + 2C^* + \frac{S^*}{\alpha}$$

$$\leq \left(\rho_{st}(\alpha+\epsilon) + 2 + (1+\alpha+\epsilon)\rho_{fl}\right)(C^* + O^*) + \left(\rho_{st} + \frac{1}{\alpha}\right)S^*$$

**Note:** $C_U + O_U \overset{(*)}{\leq} \rho_{fl}(C^* + O^*)$

Choosing $\epsilon$ sufficiently small and balancing the coefficients of $C^* + O^*$ and $S^*$, the claimed approximation ratio follows with $\alpha = 0.334$.

# Putting All Pieces Together

Expected total cost of the constructed solution is at most

$$\rho_{st}(S^* + (\alpha+\epsilon)C^*) + (\alpha+\epsilon)C_U + O_U + 2C^* + C_U + \frac{S^*}{\alpha}$$

$$\stackrel{(*)}{\leq} \rho_{st}(S^* + (\alpha+\epsilon)C^*) + (1+\alpha+\epsilon)\rho_{fl}(C^* + O^*) + 2C^* + \frac{S^*}{\alpha}$$

$$\leq (\rho_{st}(\alpha+\epsilon) + 2 + (1+\alpha+\epsilon)\rho_{fl})(C^* + O^*) + \left(\rho_{st} + \frac{1}{\alpha}\right) S^*$$

**Note:** $C_U + O_U \stackrel{(*)}{\leq} \rho_{fl}(C^* + O^*)$

Choosing $\epsilon$ sufficiently small and balancing the coefficients of $C^* + O^*$ and $S^*$, the claimed approximation ratio follows with $\alpha = 0.334$.

# Refinements and Derandomization

**Refinements:**

- can improve approximation guarantees by using
  - bifactor approximation algorithm for facility location
  - flow cancelling in the tree-core detouring scheme
- techniques extend to other connected facility location variants

**Derandomization:** can derandomize most of our algorithms

(see also [Van Zuylen, Algorithmica '09])

# Overview of Results

| Problem | Our results | Previous best | |
|---|---|---|---|
| CFL | $4.00^\star$<br>4.23 | 8.55 | [Swamy, Kumar, Algorithmica '04] |
| SROB | $2.92^\star$<br>3.28 | $3.55^\star$<br>4 | [Gupta, Kumar, Roughgarden, STOC '03]<br>[Van Zuylen, Williamson, manuscript] |
| $k$-CFL | $6.85^\star$<br>6.98 | $15.55^\star$ | [Swamy and Kumar, Algorithmica '04] |
| tour-CFL | $4.12^\star$ | $5.83^\star$ | [Ravi, Salman, ESA '99] (special case only) |
| soft-CFL | $6.27^\star$ | | |

$^\star$ = randomized

# Conclusions and Open Problems

# Conclusions

Random sampling is a powerful tool to obtain simple and good approximation algorithms for network design problems.

Cost share viewpoint turned out to be helpful in the analysis of Sample-and-Augment algorithms.

Strict cost shares also play a crucial role in the Boost-and-Sample framework for two-stage stochastic optimization with recourse.

Random sampling approach is versatile enough to attack more complex network design problems such as connected facility location.

# Open Problems

**Open Problem:** Is there a Steiner forest algorithm that admits $O(1)$-group-strict cost shares? (see also [Gupta, Kumar, STOC '09])

**Open Problem:** Can one derandomize the Sample-and-Augment algorithm for MROB?

(see also [Van Zuylen, Algorithmica '09])

**Open Problem:** Is there an analog to the core detouring scheme for problems that have multiple cores (e.g., MROB, single-sink buy-at-bulk, virtual private network design)?

(see also [Grandoni, Rothvoß, ICALP '10])

**Open Problem:** Does the core detouring scheme lead to improved approximation results in the context of two-stage stochastic optimization with recourse?