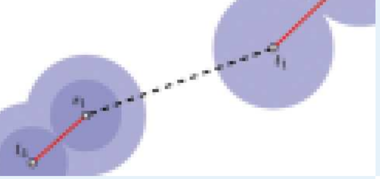
# Cost-Sharing mechanisms for Network Design

Stefano Leonardi

Sapienza Universitá di Roma

ADFOCS 2008, August 18 - 22, MPI Saarbrücken



# Talk Outline



| | |
|---|---|
| Part I | Introduction to cost-sharing mechanisms |
| Part II | The Facility location problem |
| Part III | The Steiner forest problem |
| Part IV | Novel Linear Programming Relaxation for Steiner forest |
| Part V | Lower bounds for cross-monotonic cost-sharing methods |
| Part VI | Summary and conclusions |

# Cost-Sharing Mechanisms

# Cost-Sharing Mechanisms



The ingredients:

- A service provider.
- A set $U$ of potential users (agents, customers).
- Each user $j \in U$ has a (private) utility $u_j$ (the price $j$ is willing to pay to receive the service).
- A cost-function $c$: $c(Q)$ is the cost for servicing a set $Q \subseteq U$. $c(Q)$ is usually given by the solution to an optimization problem.

# Cost-Sharing Mechanisms



## The ingredients:

- A service provider.
- A set $U$ of potential users (agents, customers).
- Each user $j \in U$ has a (private) utility $u_j$ (the price $j$ is willing to pay to receive the service).
- A cost-function $c$: $c(Q)$ is the cost for servicing a set $Q \subseteq U$. $c(Q)$ is usually given by the solution to an optimization problem.

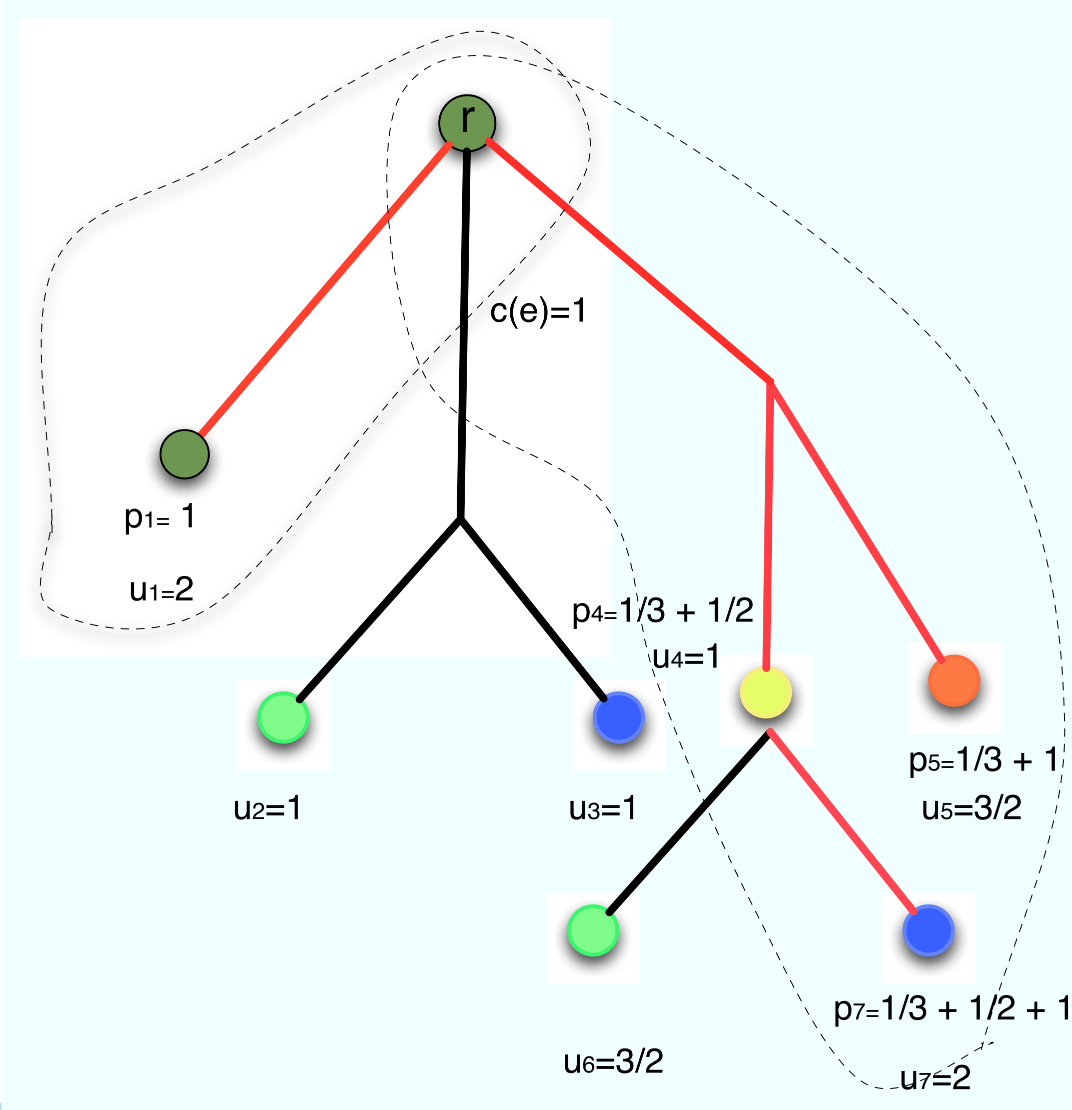
## Cost-Sharing Mechanism:

- Receive bids $b_j$ from all users $j \in U$.
- Select recipients $Q \subseteq U$ using bids.
- Distribute service cost $c(Q)$ among users in $Q$: Determine payment $p_j$ for each $j \in Q$.

# Example: Multicast Transmission

*Shapley cost shares*

- Select a subset $Q$ and a tree $T$ spanning $Q$
- Share the cost of every edge of $T$ evenly between the players served by the edge
- All players in $Q$ should bid more than the individual cost-share



# Cost-Sharing Mechanisms



- Benefit of user $j$ is $u_j - p_j$ if $j \in Q$, and $0$ otherwise.

# Cost-Sharing Mechanisms



- Benefit of user $j$ is $u_j - p_j$ if $j \in Q$, and $0$ otherwise.
- Users may lie about their utilities to increase benefit.

# Cost-Sharing Mechanisms



- Benefit of user $j$ is $u_j - p_j$ if $j \in Q$, and $0$ otherwise.
- Users may lie about their utilities to increase benefit.

Objectives:

- **Strategyproofness:** Dominant strategy for each user is to bid true utility.
- **Group-Strategyproofness:** Same holds even if users collaborate. No side payments between users.
- **Cost Recovery or Budget Balance:** $\sum_{j \in Q} p_j \geq c(Q)$.
- **Competitiveness:** $\sum_{j \in Q} p_j \leq \mathtt{opt}_Q$.

# Cost-Sharing Mechanisms



- Finding such cost-shares and a cost-function is hard if underlying problem is hard.

# Cost-Sharing Mechanisms



- Finding such cost-shares and a cost-function is hard if underlying problem is hard.
- Finding such cost-shares may be impossible if we want to ensure strategyproofness (later in this talk)

# Cost-Sharing Mechanisms



- Finding such cost-shares and a cost-function is hard if underlying problem is hard.

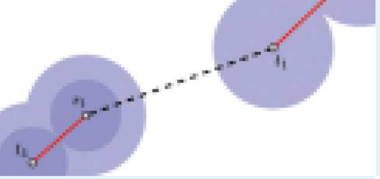
- Finding such cost-shares may be impossible if we want to ensure strategyproofness (later in this talk)

- Relax budget balance condition:
  $\beta$**-budget balance**: $\frac{1}{\beta}c(Q) \leq \sum_{j \in Q} p_j \leq \texttt{opt}_Q, \quad \beta \geq 1$

# Primal-Dual Cost-Sharing Algorithms



- Primal-dual approximation algorithms construct a feasible dual together with an integral solution to the problem.



# Primal-Dual Cost-Sharing Algorithms



- Primal-dual approximation algorithms construct a feasible dual together with an integral solution to the problem.
- Approximation guarantee obtained by relating the cost of the integral solution to a feasible dual.

# Primal-Dual Cost-Sharing Algorithms



- Primal-dual approximation algorithms construct a feasible dual together with an integral solution to the problem.
- Approximation guarantee obtained by relating the cost of the integral solution to a feasible dual.
- Dual variables often have a natural interpretation as costs to be distributed between players.

# Primal-Dual Cost-Sharing Algorithms



- Primal-dual approximation algorithms construct a feasible dual together with an integral solution to the problem.
- Approximation guarantee obtained by relating the cost of the integral solution to a feasible dual.
- Dual variables often have a natural interpretation as costs to be distributed between players.
- Weak duality implies competitiveness.

# Primal-Dual Cost-Sharing Algorithms



- Primal-dual approximation algorithms construct a feasible dual together with an integral solution to the problem.
- Approximation guarantee obtained by relating the cost of the integral solution to a feasible dual.
- Dual variables often have a natural interpretation as costs to be distributed between players.
- Weak duality implies competitiveness.
- Approximation ratio $\beta$ implies $\beta$-budget balance.

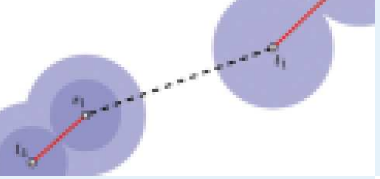
# Metric Facility location



**Input:**
- undirected graph $G = (V, E)$
- non-negative edge costs $c : E \rightarrow \mathbb{R}^+$
- set of **facilities** $F \subseteq V$
- facility $i$ has facility opening cost $f_i$
- set of **demand points** $D \subseteq V$
- $c_{ij}$: cost of connecting demand point $j$ to facility $i$. Connection cost satisfy triangle inequality

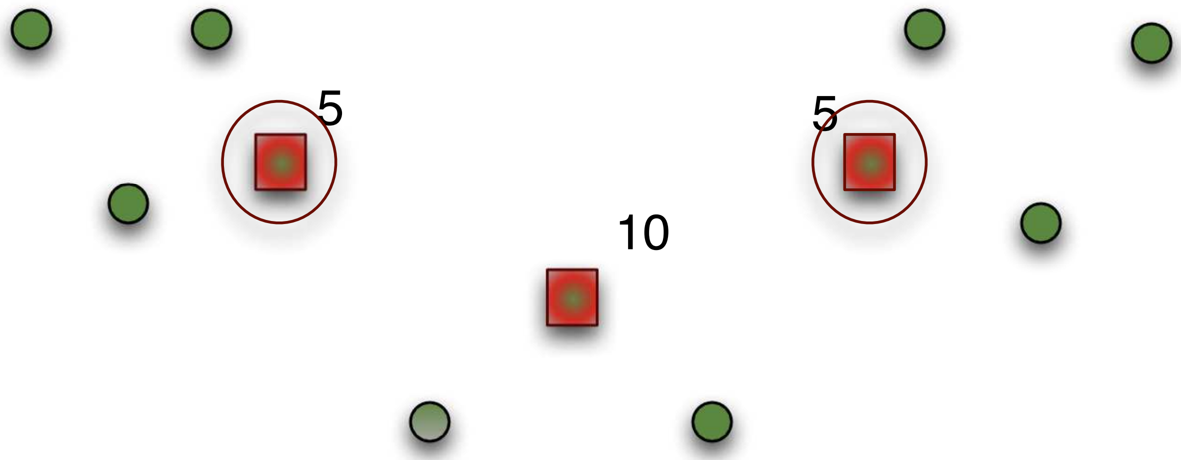
**Goal:** Compute
- set $F' \subseteq F$ of opened facilities; and
- function $\phi : \mathcal{D} \rightarrow \mathcal{F}'$ assigning demand points to opened facilities that minimizes

$$\sum_{i \in F'} f_i + \sum_{j \in \mathcal{D}} c_{\phi(j)j}$$



# Example





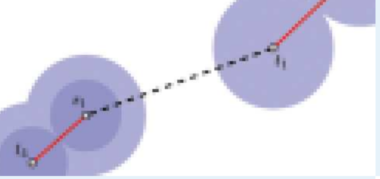

Two facilities of cost 5 are openend

# LP formulation



$$\begin{aligned}
\text{min} \quad & \sum_{i\in F, j\in D} c_{ij}x_{ij} + \sum_{i\in F} f_i y_i \\
\text{s.t.} \quad & \sum_{i\in F} x_{ij} \geq 1 && j \in D \\
& y_i - x_{ij} \geq 0 && i \in F, j \in D \\
& x_{ij} \in \{0,1\} && i \in F, j \in D \\
& y_i \in \{0,1\} && i \in F
\end{aligned}$$

- $y_i = 1$ if facility $i$ is opened;
- $x_{ij} = 1$ if demand $j$ connected to facility $i$.



# LP relaxation:



$$\text{min} \quad \sum_{i \in F, j \in D} c_{ij} x_{ij} + \sum_{i \in \mathcal{F}} f_i y_i$$

$$\begin{array}{lrcll} \text{s.t.} & \sum_{i \in F} x_{ij} & \geq & 1 & j \in D \\ & y_i - x_{ij} & \geq & 0 & i \in F, j \in D \\ & x_{ij} & \geq & 0 & i \in F, j \in D \\ & y_i & \geq & 0 & i \in F \end{array}$$

$$\mathrm{DualProgram} : \text{max} \quad \sum_{j \in D} \alpha_j$$

$$\begin{array}{lrcll} \text{s.t.} & \alpha_j - \beta_{ij} & \leq & c_{ij} & i \in F, j \in D \\ & \sum_{j \in D} \beta_{ij} & \leq & f_i & i \in F \\ & \alpha_j & \geq & 0 & j \in D \\ & \beta_{ij} & \geq & 0 & i \in F, j \in D \end{array}$$

# A 3-approximation algorithm



At time 0, set all $\alpha_j = 0$ and $\beta_{ij} = 0$ and declare all demands unconnected.

While there is an unconnected demand:

- Raise uniformly all $\alpha_j$'s of unconnected demands
- If $\alpha_j = c_{ij}$, declare demand $j$ tight with facility $i$
- For a tight constraint $ij$, raise both $\alpha_j$ and $\beta_{ij}$
- If $\sum_j \beta_{ij} = f_i$ at time $t_i$, declare:
  - Facility $i$ *temporarily opened* at time $t_i$;
  - All unconnected demands $j$ that are tight with $i$ *connected*;

[Jain and Vazirani, 1999][Mettu and Plaxton, 2000]

# A 3-approximation algorithm



Opening facilities:

Demand points contribute to more permanently opened facilities. Not enough money for all of them.

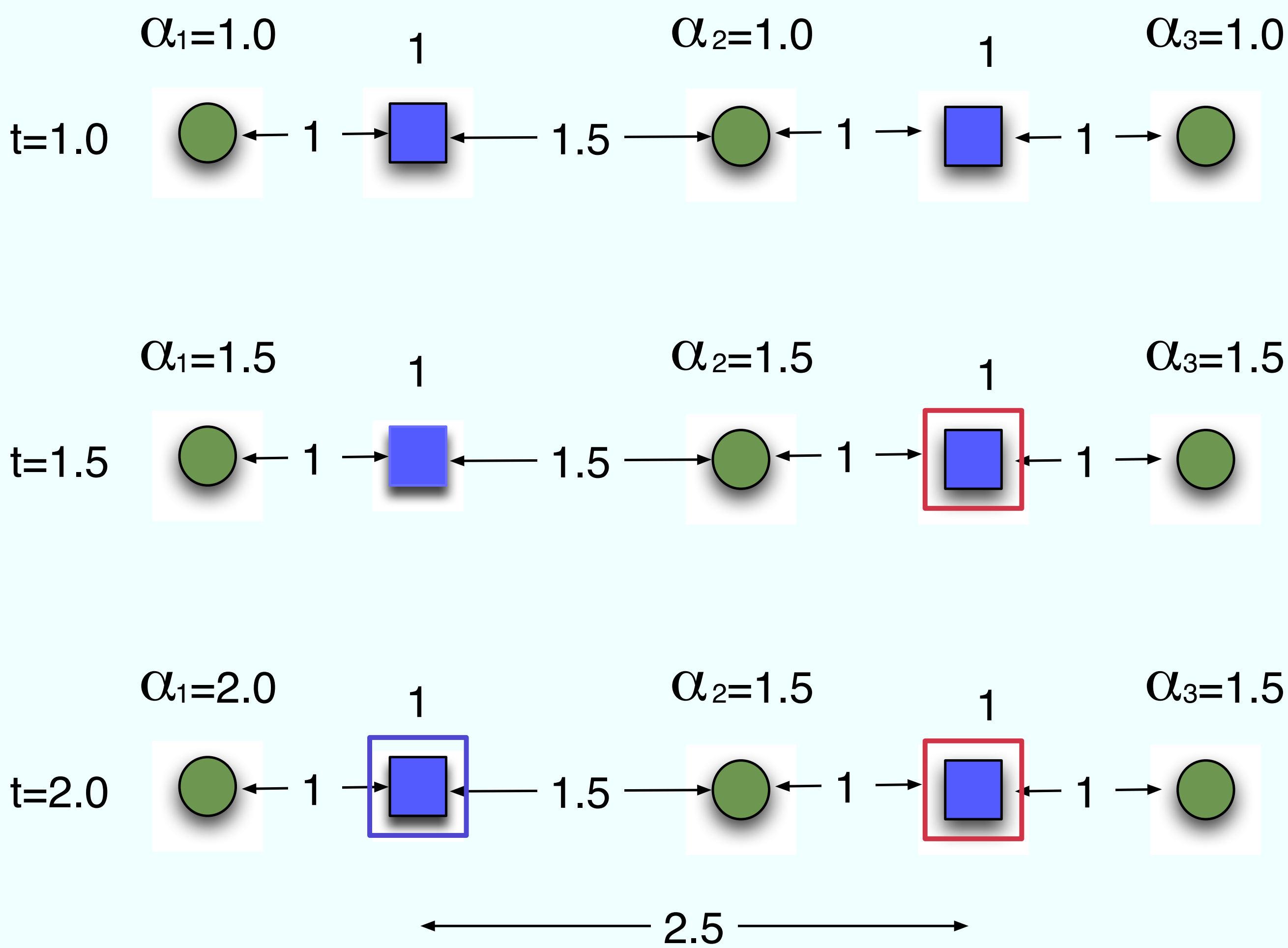
- Facility $i$ *temporarily opened* at time $t_i$;
- Declare facility $i$ *permanently opened* if there is no permanently opened facility within distance $2t_i$.

Open all permanently opened facilities.

Connect each demand to the nearest opened facility.

# Example of execution of the algorithm





# Proof of 3 approximation.



## Demands connected to opened facilities

- $\alpha_j = c_{ij} + \beta_{ij}$ for demands connected to opened facility $i$.
- $\alpha_j$ pays for connection cost $c_{ij}$ and contribute with $\beta_{ij}$ to $f_i$.
- Since other opened facilities are at distance $> t_i$, $\alpha_j$ does not pay for opening any other facility.

## Demands connected to temporarily opened facilities

- Demand $j$ connected to temporarily opened facility $i$. There exists an opened facility $i'$ with $c_{ii'} \leq 2t_i$.
- Since $c_{ji} \leq \alpha_j$ and $t_i \leq \alpha_j$, $c_{ji'} \leq c_{ji} + c_{ii'} \leq 3\alpha_j$

# Strategyproof mechanism for facility location



- If some city's cost share goes beyond its bid, then discard the city from all further considerations.
- If for some closed facility $i$, the total offer it gets is equal to the opening cost, then the facility $i$ is opened, and every city $j$ that has a non-zero offer to $i$ is connected to $i$.
- If some unconnected city $j$'s cost share is equal to its connection cost to an already opened facility $i$, then connect city $j$ to facility $i$.

[Devanur, Mihail, Vazirani, 2003]

# Proof of strategyproofness

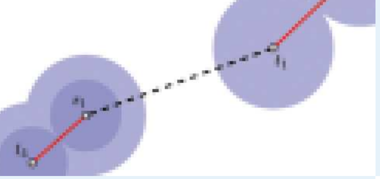


Truthfulness follows from bid independence:

- Lowering the bid might result in early discard: payoff=0
- Raising the bid might result in paying more than the bid: payoff<0

Primal dual algorithms that monotonically increase dual variables often result in truthful cost-sharing mechanism.

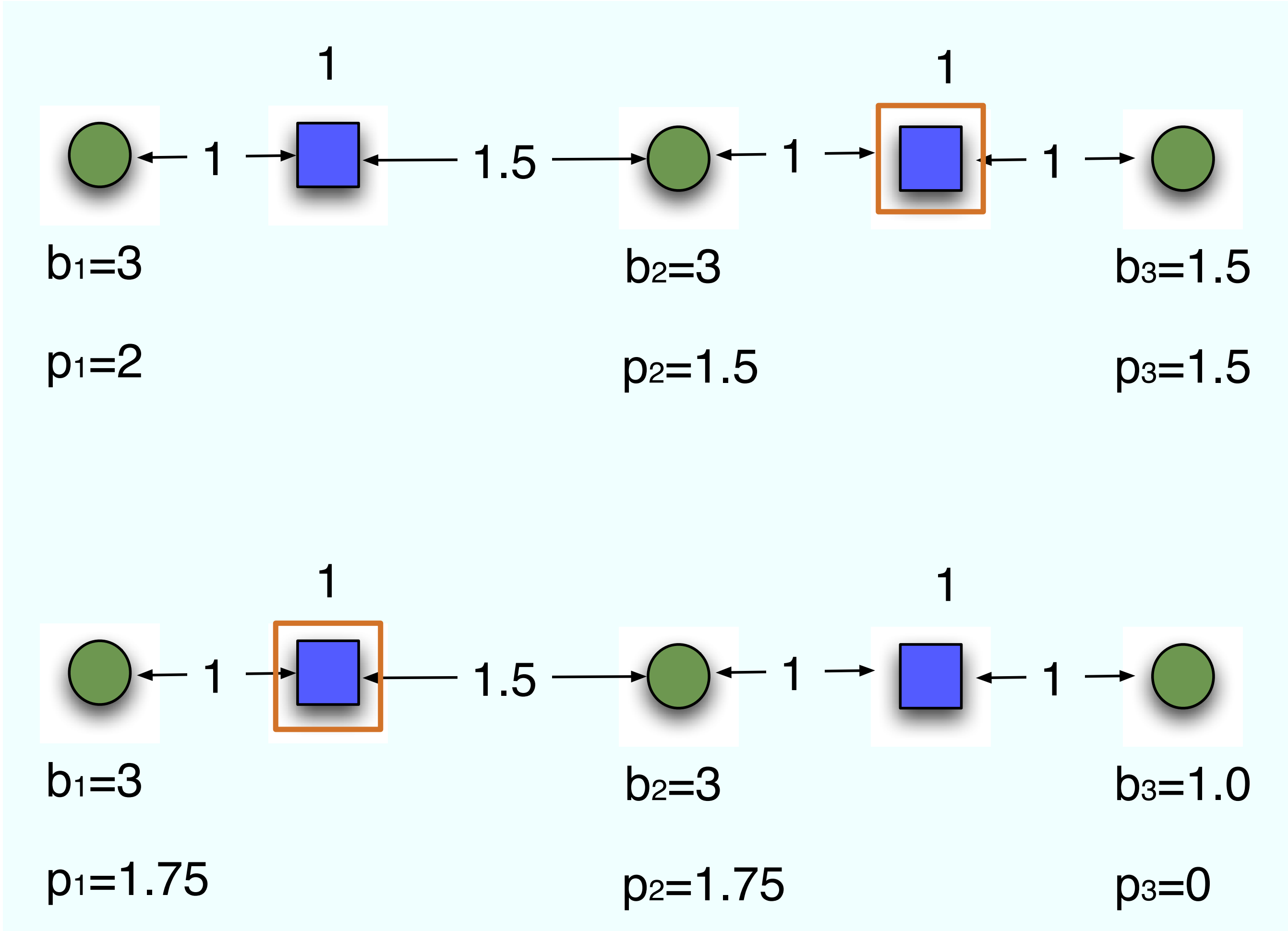
Excercise: Derive a truthful mechanism for set cover.



# The mechanism is not group-strategyproof



## Example:



# A different set of cost shares is needed

- Needs a more equitable notion of cost-sharing
- Intuitively, The cost share of all other players should increase if one player leaves the game
- This would prevent coalitions to manipulate the game by pushing some of the members out of the game
- Observe that the only players of the coalitions that will misreport utilities are those with 0 payoff!
- We do not allow side payments, i.e., transfer utility between members of the coalition

# Cross-Monotonicity



## Cost-Sharing Method:

- Given: Set $Q \subseteq U$ of users.
- Compute: Cost-shares $\xi_Q(j)$ for each $j \in Q$ such that competitiveness and $\beta$-budget balance hold.

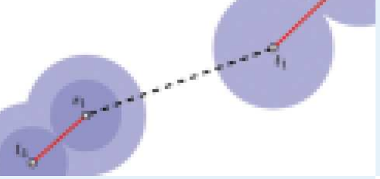
# Cross-Monotonicity



## Cost-Sharing Method:

- Given: Set $Q \subseteq U$ of users.
- Compute: Cost-shares $\xi_Q(j)$ for each $j \in Q$ such that competitiveness and $\beta$-budget balance hold.

$\xi$ is cross-monotonic if each individual cost-share does not increase as additional players join the game:



# Cross-Monotonicity



## Cost-Sharing Method:

- Given: Set $Q \subseteq U$ of users.
- Compute: Cost-shares $\xi_Q(j)$ for each $j \in Q$ such that competitiveness and $\beta$-budget balance hold.

$\xi$ is cross-monotonic if each individual cost-share does not increase as additional players join the game:

$$\forall Q' \subseteq Q,\ \forall j \in Q' : \quad \xi_{Q'}(j) \geq \xi_Q(j).$$

# Cross-Monotonicity



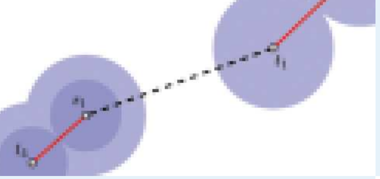
## Cost-Sharing Method:

- Given: Set $Q \subseteq U$ of users.
- Compute: Cost-shares $\xi_Q(j)$ for each $j \in Q$ such that competitiveness and $\beta$-budget balance hold.

$\xi$ is cross-monotonic if each individual cost-share does not increase as additional players join the game:

$$\forall Q' \subseteq Q,\ \forall j \in Q' : \quad \xi_{Q'}(j) \geq \xi_Q(j).$$

**Theorem [Moulin, Shenker ’97]:** The Moulin–Shenker Mechanism is group-strategyproof, and satisfies cost recovery and competitiveness.



# Moulin–Shenker Mechanism



Moulin–Shenker mechanism: Use cross-monotonic cost-sharing method to obtain group-strategyproof mechanisms.

# Moulin–Shenker Mechanism



Moulin–Shenker mechanism: Use cross-monotonic cost-sharing method to obtain group-strategyproof mechanisms.

Moulin–Shenker Mechanism:

1. Initialize: $Q \leftarrow U$.
2. If for each user $j \in Q$: $\xi_Q(j) \leq b_j$ then stop.
3. Otherwise, remove from $Q$ all users with $\xi_Q(j) > b_j$ and repeat.

# Moulin–Shenker Mechanism



Designing a cost-sharing mechanism that is group-strategyproof, satisfies competitiveness and (approximate) budget balance.

$\Downarrow$ reduces to

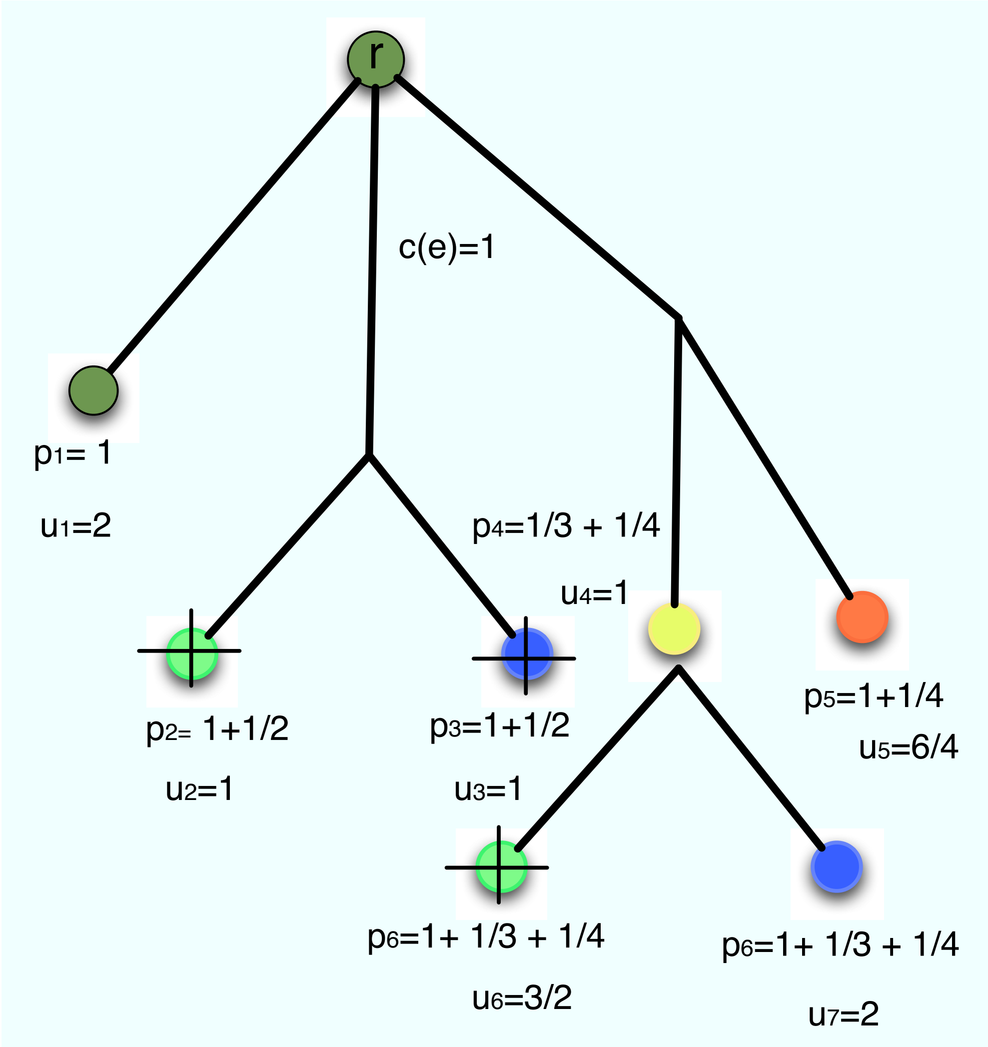
Designing a cross-monotonic cost-sharing method $\xi$ that satisfies competitiveness and (approximate) budget balance.

# Example: Multicast Transmission



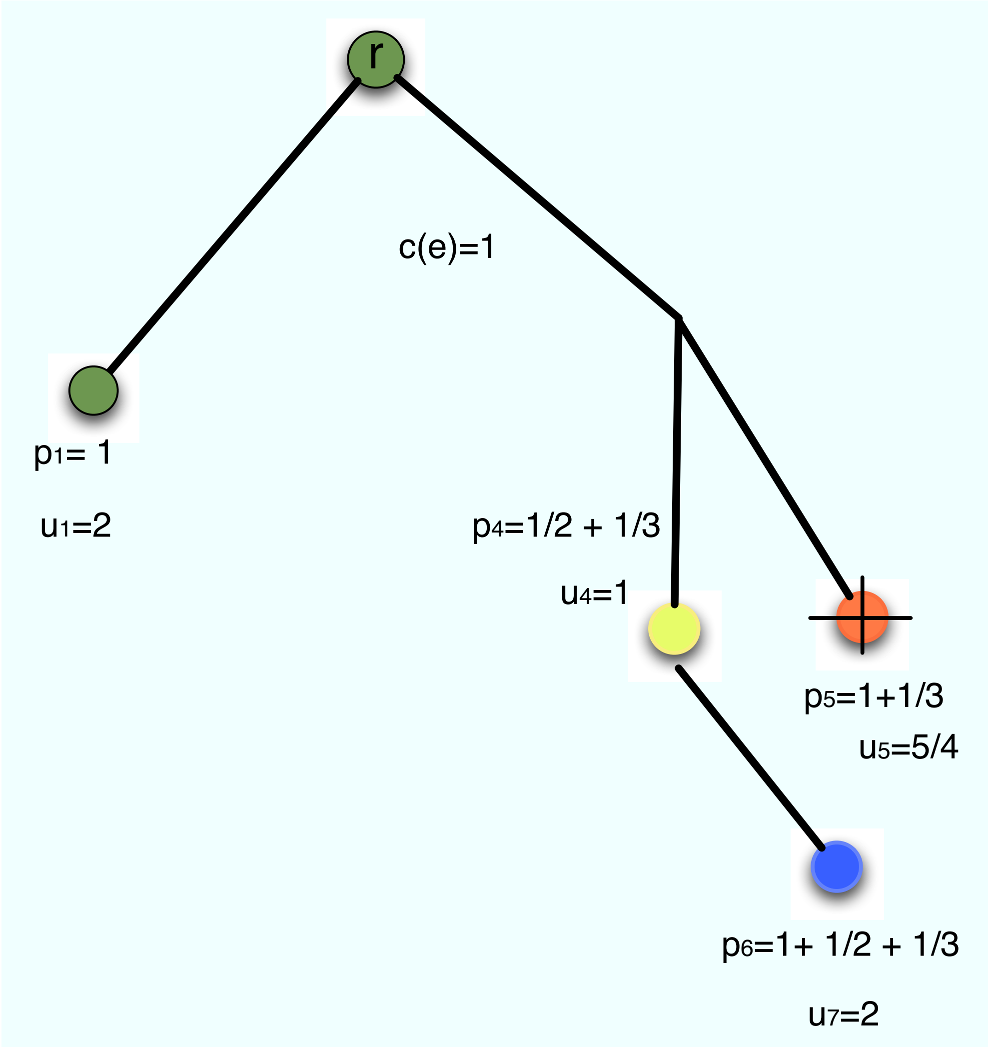
*Moulin Mechanism for Shapley Cost Shares*

- Shapley is a cross-monotonic cost sharing method for Multicast transmission - Submodular function optimization
- Shapley is budget-balance, i.e. recovers the whole cost



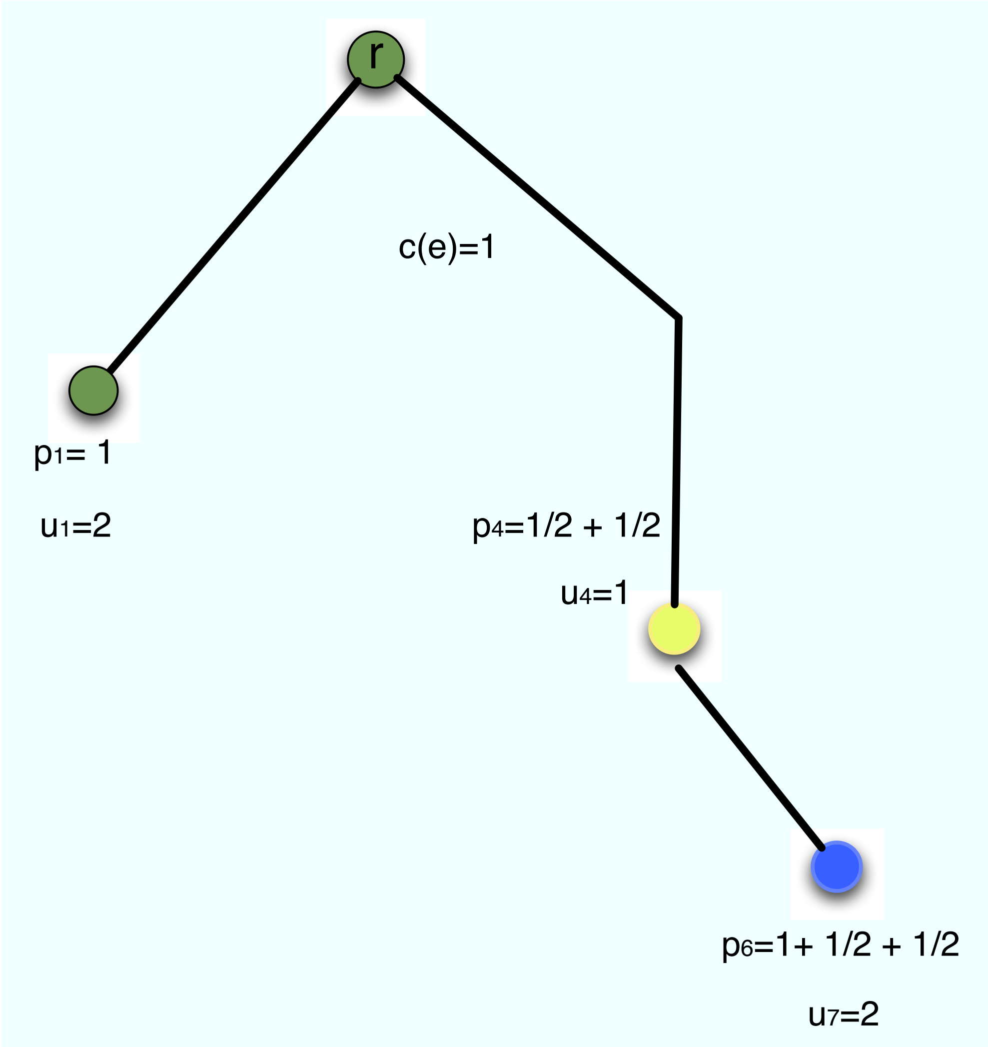
*Moulin Mechanism for Shapley Cost Shares*

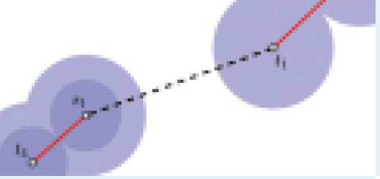
- Shapley is a cross-monotonic cost sharing method for Multicast transmission - Submodular function optimization
- Shapley id budget-balance, i.e. recovers the whole cost



*Moulin Mechanism for Shapley Cost Shares*

- Shapley is a cross-monotonic cost sharing method for Multicast transmission - Submodular function optimization
- Shapley id budget-balance, i.e. recovers the whole cost





# Known Results - Upper Bounds



| **Authors** | **Problem** | $\beta$ |
|---|---|---|
| [Moulin, Shenker '01] | submodular cost | $1$ |
| [Jain, Vazirani '01] | MST | $1$ |
| | Steiner tree and TSP | $2$ |
| [Devanur, Mihail, Vazirani '03] | set cover | $\log n$ |
| (strategyproof only) | facility location | $1.61$ |
| [Pal, Tardos '03] | facility location | $3$ |
| | SRoB | $15$ |
| [Leonardi, Schäfer '03], [Gupta et al. '03] | SRoB | $4$ |
| [Leonardi, Schäfer '03] | CFL | $30$ |
| [Könemann, Leonardi, Schäfer '05] | Steiner forest | $2$ |
| [Gupta, Könemann, Leonardi, Ravi, Schäfer '07] | Prize Collecting Steiner Forest | $3$ |
| [Goyal, Gupta, Leonardi, Ravi '07] | 2-Stage Stochastic Steiner Tree | $O(1)$ |

# Known Results - Lower Bounds



| Authors | Problem | $\beta$ |
|---|---|---|
| **Lower bounds** | | |
| [Immorlica, Mahdian, Mirrokni '05] | edge cover | $2$ |
| | facility location | $3$ |
| | vertex cover | $n^{1/3}$ |
| | set cover | $n$ |
| [Könemann, Leonardi, Schäfer, van Zwam '05] | Steiner tree | $2$ |

# Facility location

# The Pál and Tardos mechanism



- In traditional Primal Dual algorithms, if a new city is added, the cost share of nearby cities is decreased, while farther cities can be negatively affected
- A ghost process uniformly raises every dual variable $\alpha_j$ even after user $j$ is connected, to contribute to open other facilities
- The cost share of user $j$ is still the earliest time of connection of user $j$
- How can we limit the number and the cost of opened facilities?
- How can we recover at least a costant fraction of the opening cost?

[Pal and Tardos, 2003]

# Cost-shares



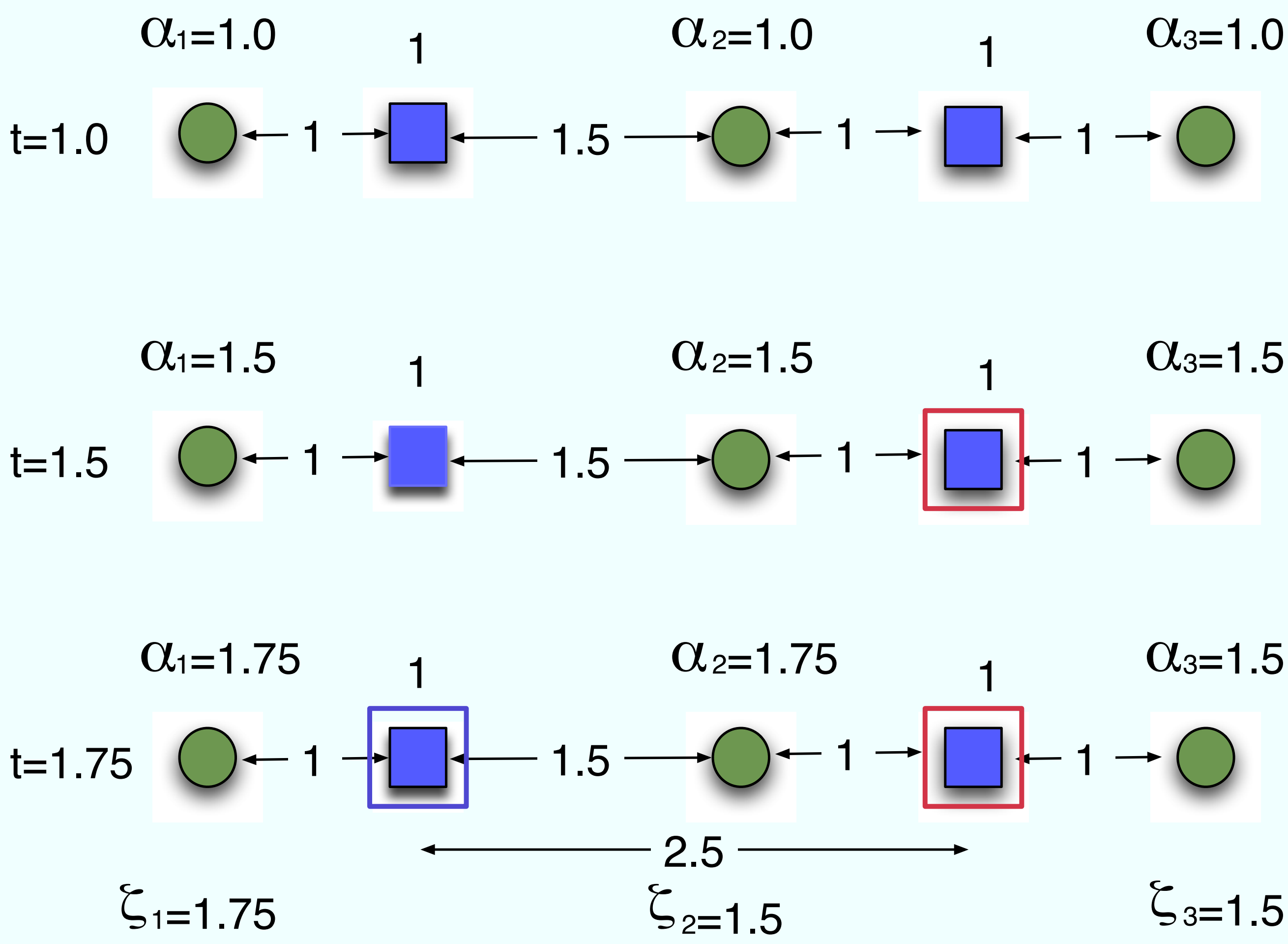
- $t(i)$: when facility $i$ becomes full
- $S_i$ :users contributing to making facility $i$ full, all within distance $t(i)$ from $i$
- Raise cost share $\alpha_j$ till a facility that is touched becomes full or $j$ touches a full facility:

$$\xi_j = \min\{\min_{i:j \in S_i} t(i), \min_{i:j \notin S_i} c_{ij}\}$$

- Cost shares are cross-monotonic since by adding more users, every facility becomes full earlier
- Attention! Not all full facilities are opened

# Example of execution of the algorithm





# Opening facilities



- Open a full facility $i$ if there is no open facility $i'$ at distance $c_{ii'} \leq 2t(i)$
- Assign every city to the closest open facility $i$

**Lemma:** For every two open facilities $i, i'$, $S_i \cap S_{i'} = \emptyset$.

Proof: Assume $i$ to open after $i'$. If there is a point in $S_i \cup S_{i'}$ then $c_{ii'} \leq 2t(i)$.

To prove:

- If $j \in S_i$, $\xi_j$ pays at least for $t(i)/3$
- If $j \notin S_i$ $\xi_j$ pays 1/3 of the connection cost to the closest open facility

# Cost recovery I

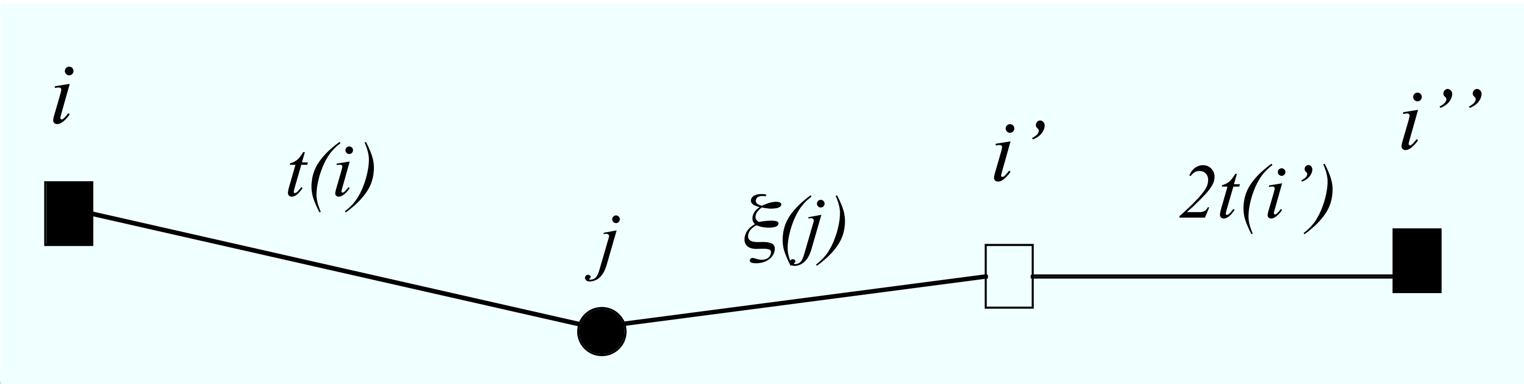


**Lemma:** For every $j \in S_i$, $\xi_j \geq t(i)/3$.

Proof:

- If $\xi_j$ determined by $i$, then $\xi_j = t(i)$ (i.e. 1st full facility touched).
- If determined by facility $i'$ and $i'$ is open we get a contradiction since $c_{ii'} \leq 2t(i)$.
- Otherwise, assume $\xi_j < t(i)/3$ and $i'$ not open. We have a facility $i''$ such that $c_{i'i''} \leq 2t(i') \leq 2\xi_j$. A contradiction since

$$c_{ii''} \leq c_{ij} + c_{ji'} + c_{i'i''} \leq t(i) + \xi_j + 2\xi_j \leq 2t(i)$$



# Cost recovery II

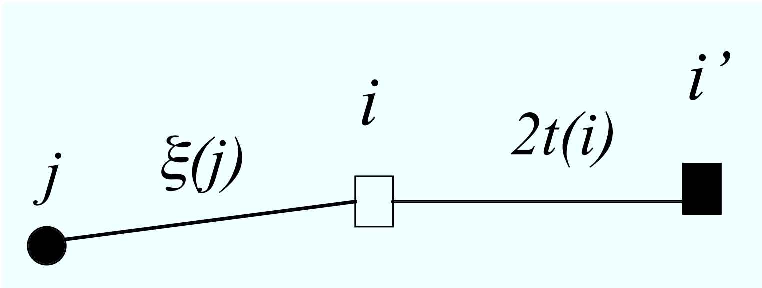
**Lemma:** Assume for every open facility $i$, $j \notin S_i$. If $j$ has been allocated to open facility $i$ then $\xi_j \geq c_{ji}/3$.

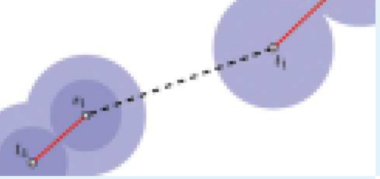
Proof:

Assume $i$ is the first facility that $j$ touches.

- If $i$ is open then $\xi_j = c_{ji}$.
- If $i$ not open , there exists $i'$ such that $c_{ii'} \leq 2t(i)$ to which $j$ is allocated. It follows:

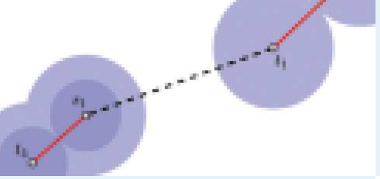
$$c_{ji'} \leq c_{ji} + c_{ii'} \leq \xi_j + 2t(i) \leq 3\xi_j$$







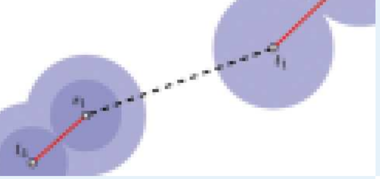
# Steiner Forests



# Steiner forests



- **S**teiner forests



# Steiner forests



- **S**teiner forests

  Input:
  - undirected graph $G = (V, E)$;
  - non-negative edge costs $c : E \to \mathbb{R}^+$;
  - terminal-pairs $R = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$.

# Steiner forests



- **S**teiner forests

  Input:
  - undirected graph $G = (V, E)$;
  - non-negative edge costs $c : E \rightarrow \mathbb{R}^+$;
  - terminal-pairs $R = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$.

  Goal:

  Compute min-cost forest $F$ in $G$ such that $s$ and $t$ are in same tree for all $(s, t) \in R$.

# Steiner forests



- **S**teiner forests

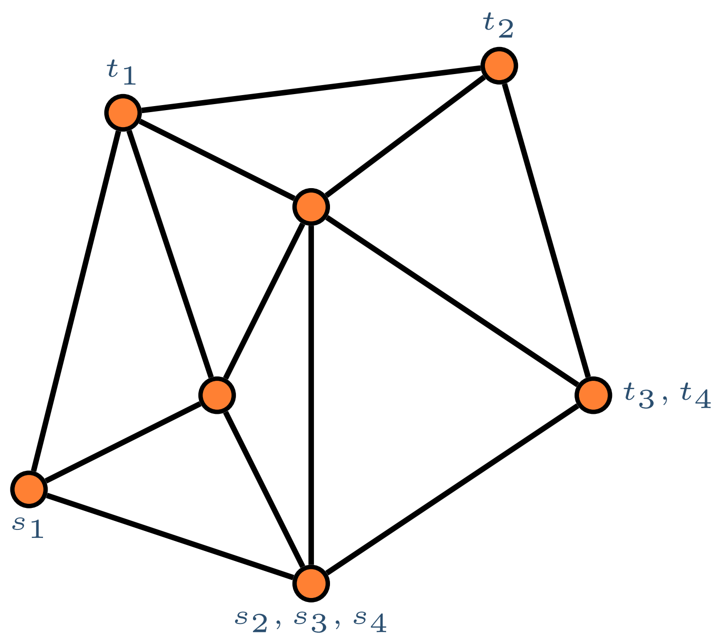
  Input:
  - undirected graph $G = (V, E)$;
  - non-negative edge costs $c : E \to \mathbb{R}^+$;
  - terminal-pairs $R = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$.

  Goal:
  Compute min-cost forest $F$ in $G$ such that $s$ and $t$ are in same tree for all $(s, t) \in R$.

- Special case: Steiner trees.
  Compute a min-cost tree spanning a teminal-set $R \subseteq V$.

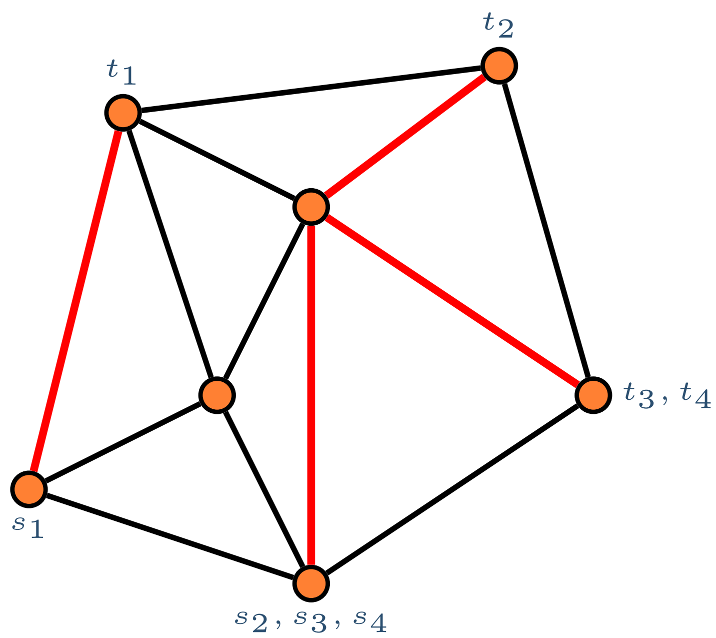
# Steiner forests: Example

- Example with four terminal pairs: $R = \{(s_i, t_i)\}_{1 \le i \le 4}$
- All edges have unit cost.

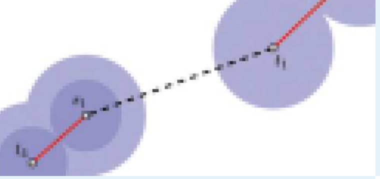


# Steiner forests: Example

- Example with four terminal pairs: $R = \{(s_i, t_i)\}_{1 \leq i \leq 4}$
- All edges have unit cost.




Total cost is 4!



# Previous Work and cross-monotonic result



- [Agrawal, Klein, Ravi '95] (see also [Goemans, Williamson '95]):
  Primal-dual $2$-approximation for Steiner forests.

# Previous Work and cross-monotonic result



- [Agrawal, Klein, Ravi '95] (see also [Goemans, Williamson '95]):
  Primal-dual $2$-approximation for Steiner forests.
- [Jain, Vazirani '01]:
  Group-strategyproof cost-sharing mechanism for Steiner trees that satisfies competitiveness and $2$-budget balance.

# Previous Work and cross-monotonic result

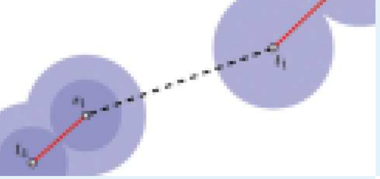


- [Agrawal, Klein, Ravi '95] (see also [Goemans, Williamson '95]):
  Primal-dual $2$-approximation for Steiner forests.
- [Jain, Vazirani '01]:
  Group-strategyproof cost-sharing mechanism for Steiner trees that satisfies competitiveness and $2$-budget balance.
- **[Könemann, L., Schäfer, 2005]:**
  Group-strategyproof cost-sharing mechanism for Steiner forests that satisfies competitiveness and $2$-budget balance.

# Previous Work and cross-monotonic result



- [Agrawal, Klein, Ravi '95] (see also [Goemans, Williamson '95]):
  Primal-dual $2$-approximation for Steiner forests.
- [Jain, Vazirani '01]:
  Group-strategyproof cost-sharing mechanism for Steiner trees that satisfies competitiveness and $2$-budget balance.
- **[Könemann, L., Schäfer, 2005]:**
  Group-strategyproof cost-sharing mechanism for Steiner forests that satisfies competitiveness and $2$-budget balance.



# Steiner Forests: Primal-dual algorithm



- We sketch primal-dual algorithm SF due to [Agrawal, Klein, Ravi ’95] (see also [Goemans, Williamson ’95]).

# Steiner Forests: Primal-dual algorithm



- We sketch primal-dual algorithm SF due to [Agrawal, Klein, Ravi '95] (see also [Goemans, Williamson '95]).
- Algorithm SF computes
  - feasible Steiner forest $F$, and
  - feasible dual solution $y$

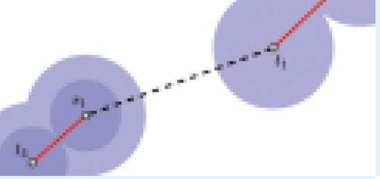
  at the same time.

  Key trick: Use dual $y$ and weak duality to bound cost of $F$.

# Primal LP: Steiner Cuts



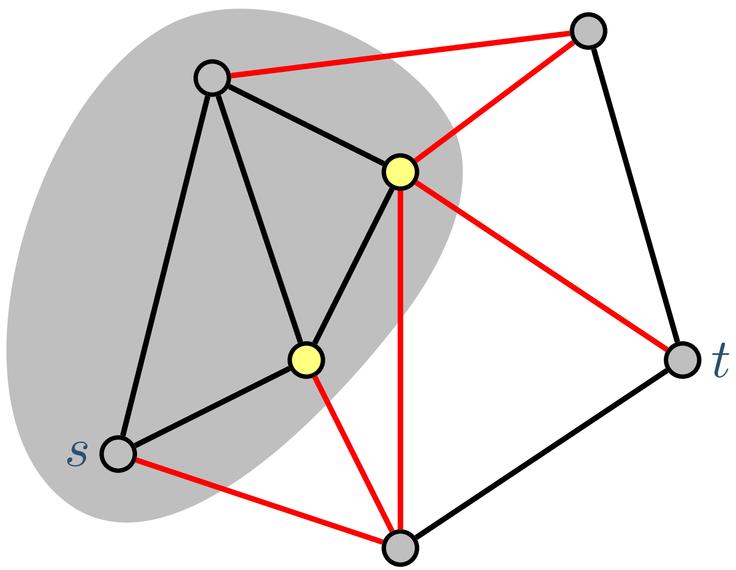
- Primal has variables $x_e$ for all $e \in E$.
  $x_e = 1$ if $e$ is in Steiner forest, $0$ otherwise



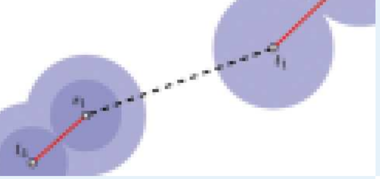
# Primal LP: Steiner Cuts



- Primal has variables $x_e$ for all $e \in E$.
  $x_e = 1$ if $e$ is in Steiner forest, $0$ otherwise
- Steiner cut: Subset of nodes that separates at least one terminal pair $(s, t) \in R$.




Any feasible Steiner forest must contain at least one of the red edges!



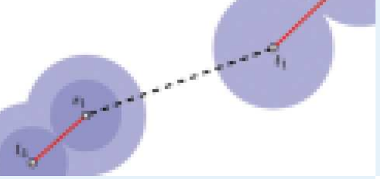
# Primal LP: Steiner Cuts



Primal LP has one constraint for each Steiner cut.

$$
\begin{aligned}
\text{min} \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & \sum_{e \in \delta(U)} x_e \geq 1 \quad \forall \text{ Steiner cut } U \\
& x_e \geq 0 \quad \forall e \in E
\end{aligned}
$$

$\delta(U)$: Edges with exactly one endpoint in $U$.



# Steiner trees: Dual LP



Dual LP has a variable $y_U$ for all Steiner cuts $U$.

$$\max \sum_{U} y_U$$

$$\text{s.t.} \quad \sum_{U:\ e \in \delta(U)} y_U \leq c_e \quad \forall e \in E$$

$$y_U \geq 0 \quad \forall U$$

$\delta(U)$: Edges with exactly one endpoint in $U$.

# Dual LP: Pictorial View



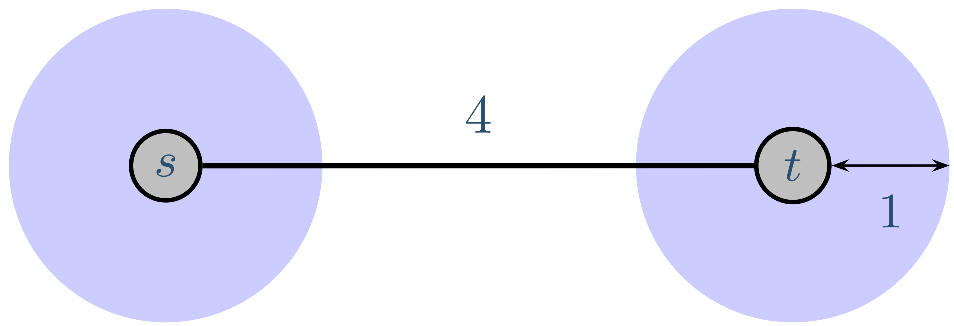
- Can visualize $y_U$ as disks around $U$ with radius $y_U$.
  Example: Terminal pair $(s,t) \in R$, edge $(s,t)$ with cost $4$


$y_s = y_t = 0$

# Dual LP: Pictorial View



- Can visualize $y_U$ as disks around $U$ with radius $y_U$.
  Example: Terminal pair $(s,t) \in R$, edge $(s,t)$ with cost $4$

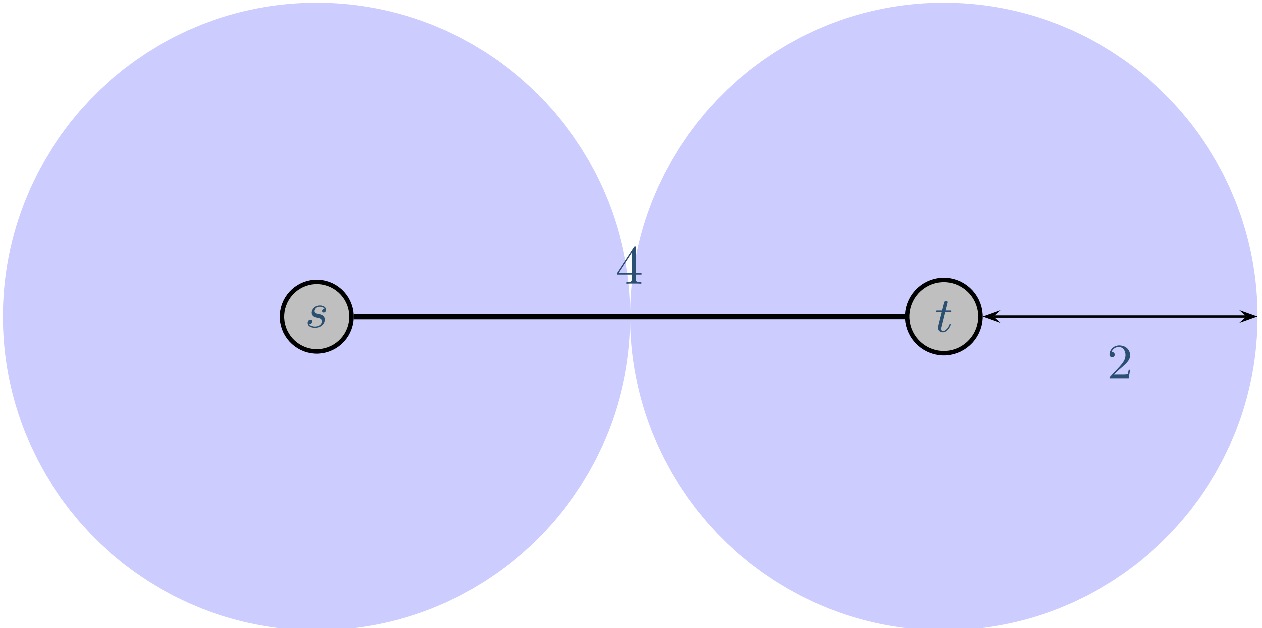



$y_s = y_t = 1$

# Dual LP: Pictorial View

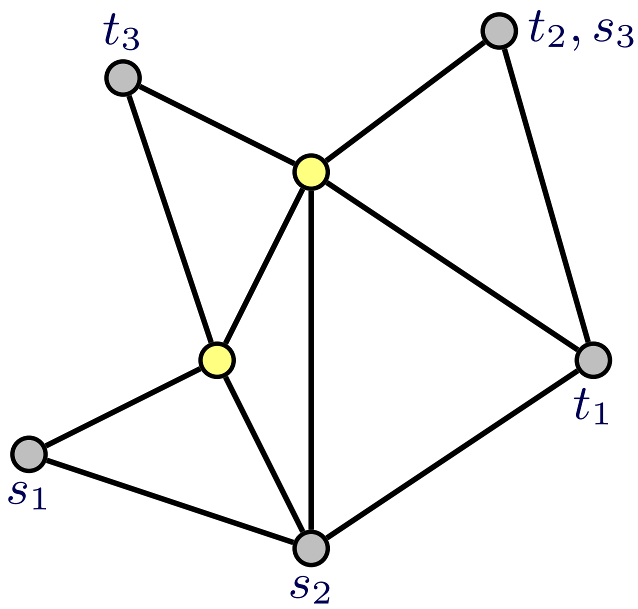


- Can visualize $y_U$ as disks around $U$ with radius $y_U$.
  Example: Terminal pair $(s, t) \in R$, edge $(s, t)$ with cost $4$




$y_s = y_t = 2$ Have: $y_s + y_t = 4 = c_{st}$. Edge $(s, t)$ is tight.

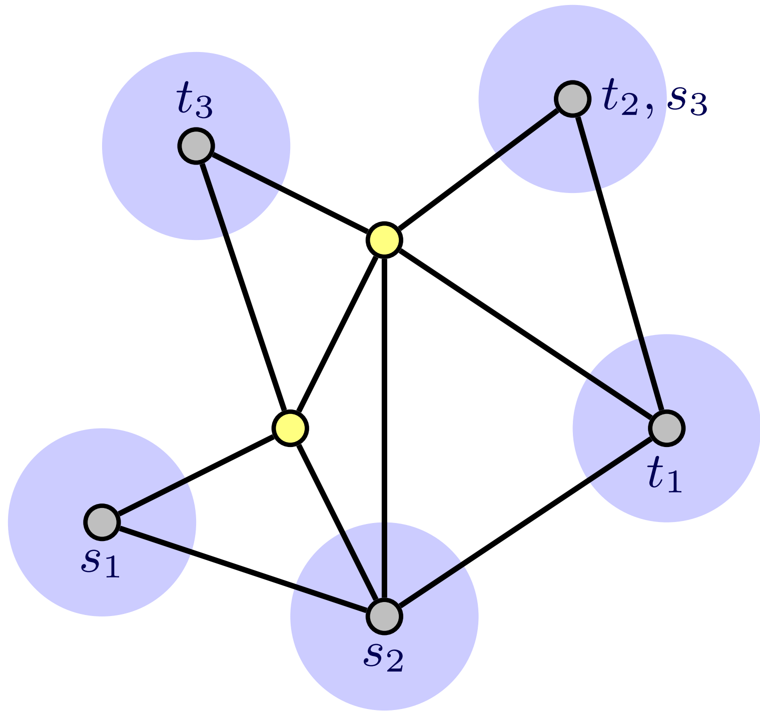
Algorithm grows duals of connected components.



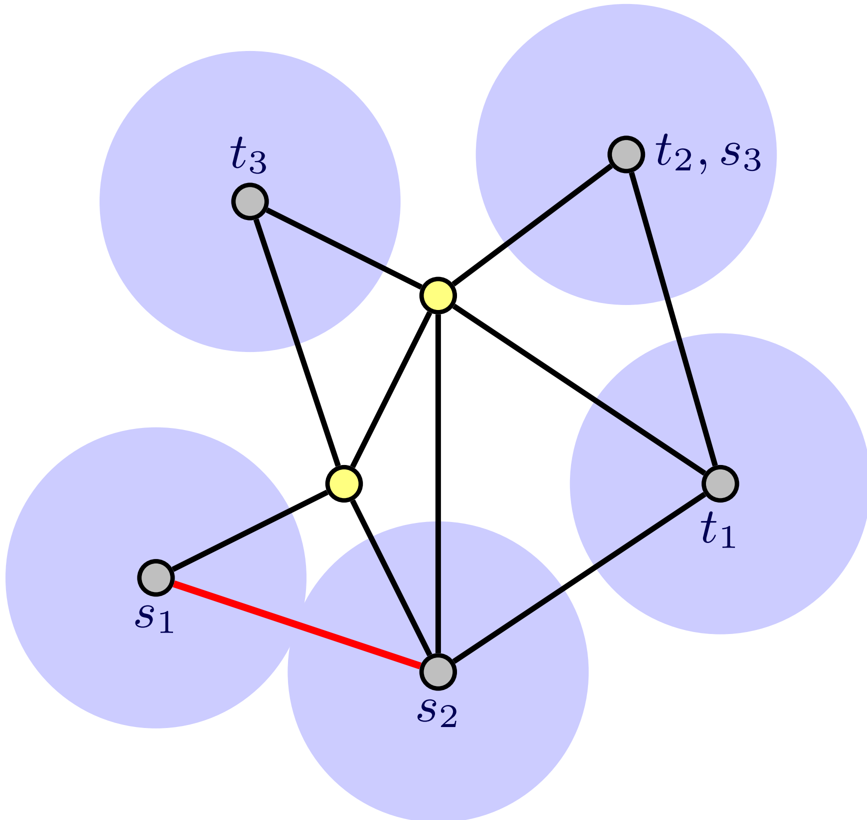
# Algorithm SF: Example

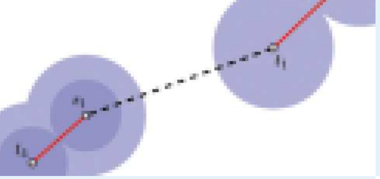


Algorithm grows duals of connected components.



Algorithm grows duals of connected components.

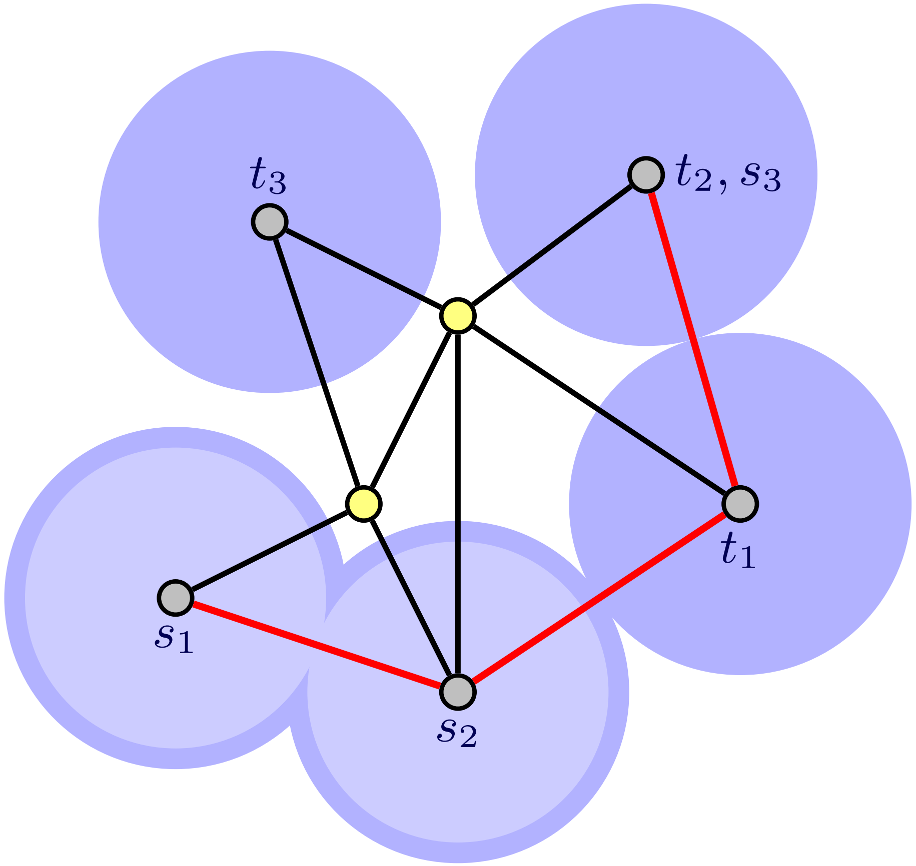




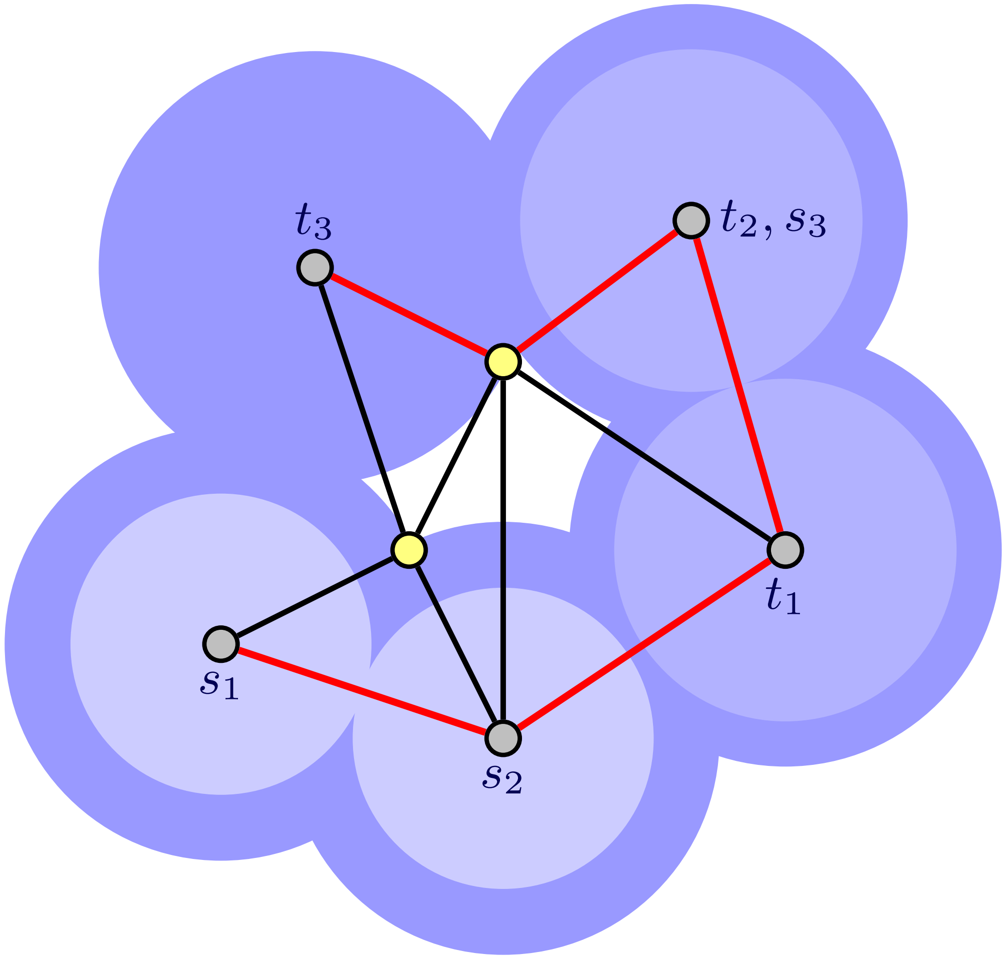
# Algorithm SF: Example



Algorithm grows duals of connected components.



Algorithm grows duals of connected components.

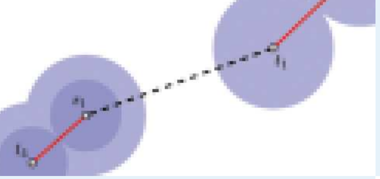


# PD-Algorithm: Properties



**Theorem [Agrawal, Klein, Ravi ’95]:** Algorithm computes forest $F$ and dual $y$ such that

$$c(F) \leq (2 - 1/k) \cdot \sum_U y_U \leq (2 - 1/k) \cdot \texttt{opt}_R.$$



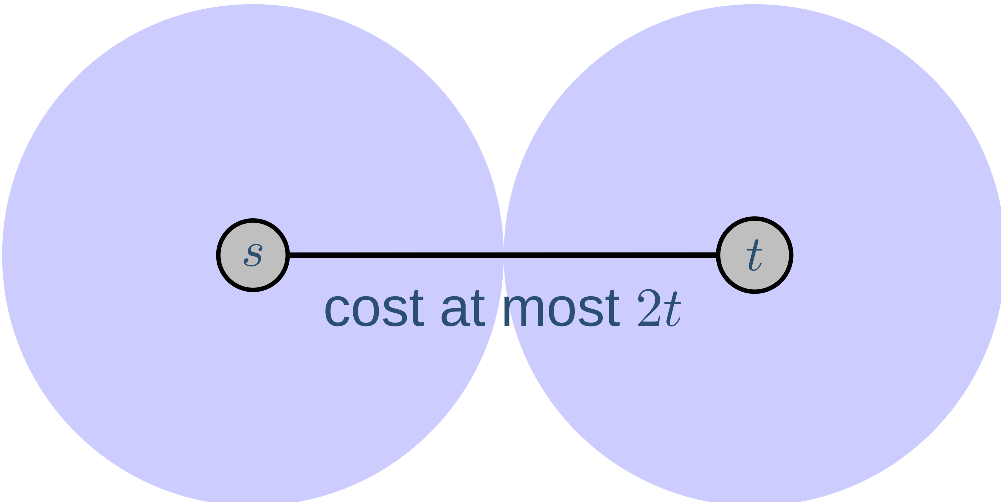
# PD-Algorithm: Properties



**Theorem [Agrawal, Klein, Ravi '95]:** Algorithm computes forest $F$ and dual $y$ such that

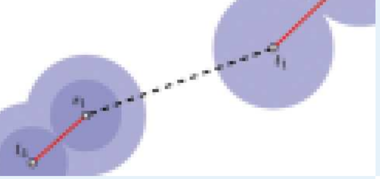
$$c(F) \leq (2 - 1/k) \cdot \sum_U y_U \leq (2 - 1/k) \cdot \mathtt{opt}_R.$$

Main trick: Edge $(s, t)$ becomes tight at time $t$.



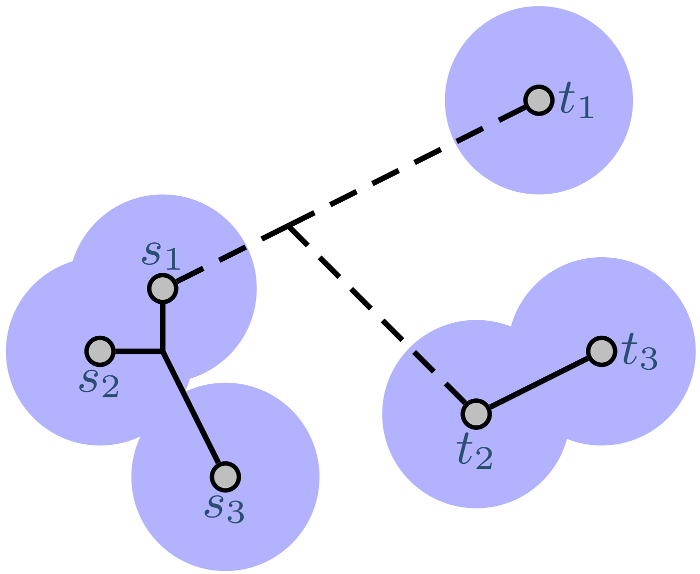

Use twice the dual around $s$ and $t$ to pay for cost of path.

# Steiner Forest Cost-Sharing Mechanism



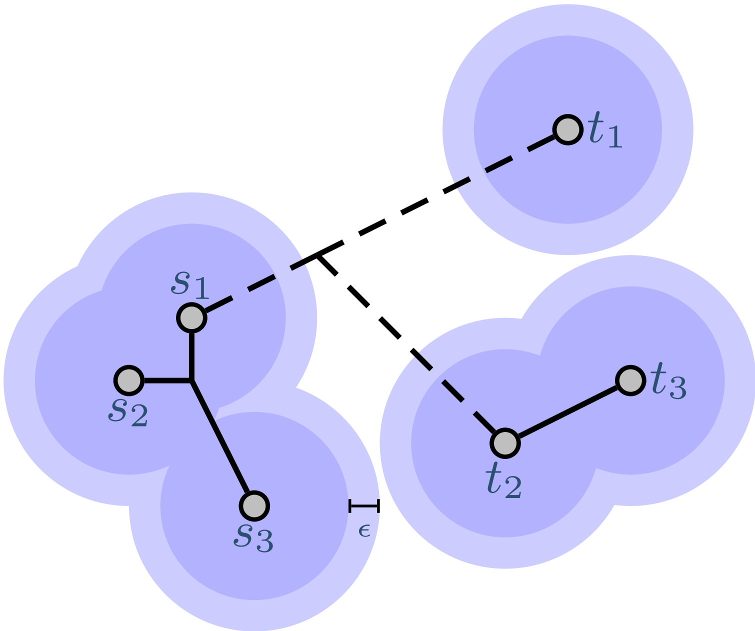
# Try 1: SF and Shapley Value






- Say: terminal pair $(s, t)$ is **active** at time $t$ if $s$ and $t$ are not in same moat.
  Example: All terminals are active.

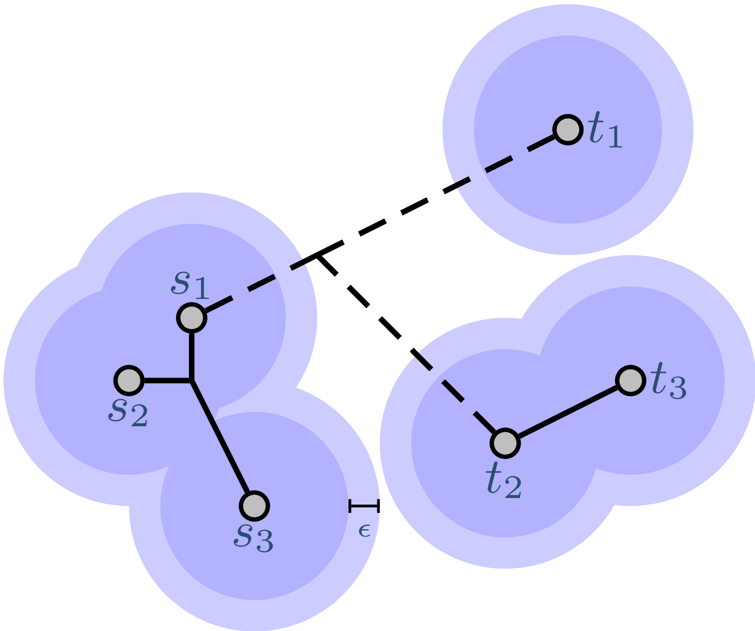
# Try 1: SF and Shapley Value




- Say: terminal pair $(s, t)$ is active at time $t$ if $s$ and $t$ are not in same moat. Example: All terminals are active.
- Grow active moats by $\epsilon$.

# Try 1: SF and Shapley Value

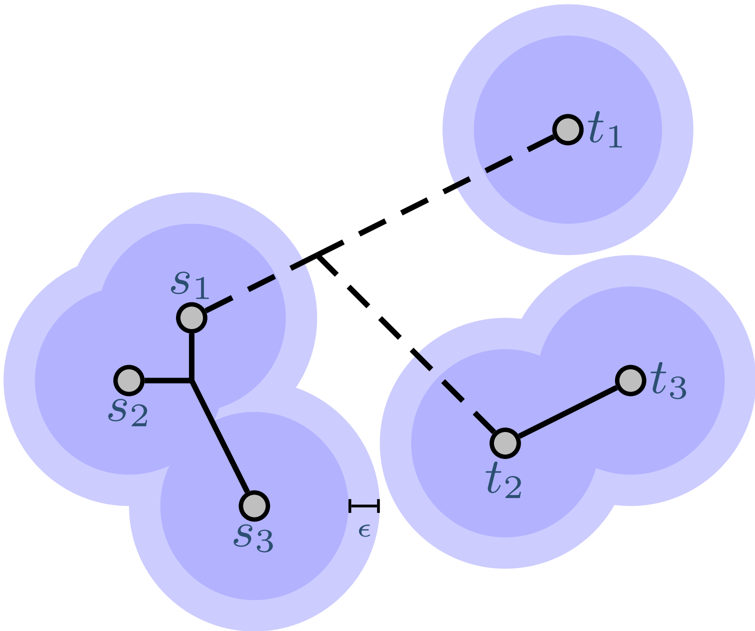





- Say: terminal pair $(s, t)$ is active at time $t$ if $s$ and $t$ are not in same moat.
  Example: All terminals are active.
- Grow active moats by $\epsilon$.
- Growth of moats is shared among active terminals.

# Try 1: SF and Shapley Value





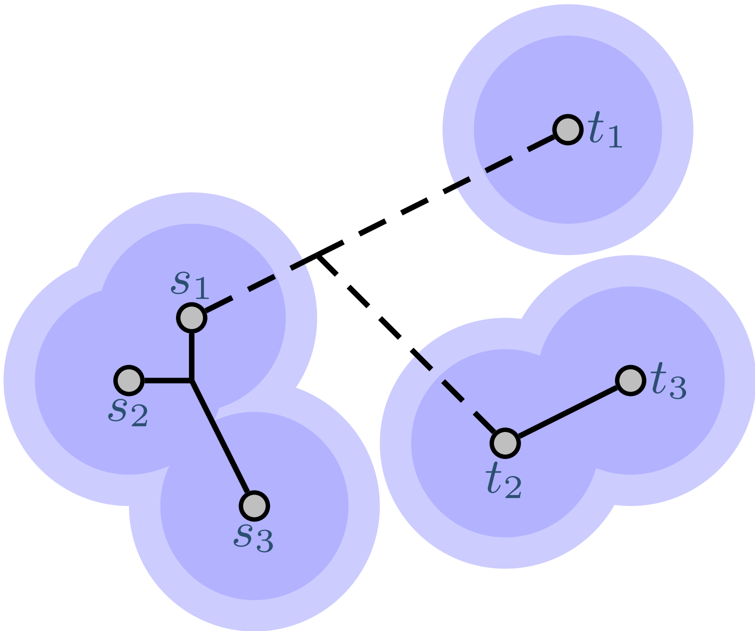

- Say: terminal pair $(s, t)$ is active at time $t$ if $s$ and $t$ are not in same moat. Example: All terminals are active.
- Grow active moats by $\epsilon$.
- Growth of moats is shared among active terminals.
- Cost-share increase for ...

$$s_1 : \epsilon/3$$

$$t_2 : \epsilon/2$$

$$t_1 : \epsilon$$

# Try 1: SF and Shapley Value




- $U^t(r)$ : moat of terminal $r$ at time $t$.

# Try 1: SF and Shapley Value


- Talk Outline

Cost-Sharing Mechanisms

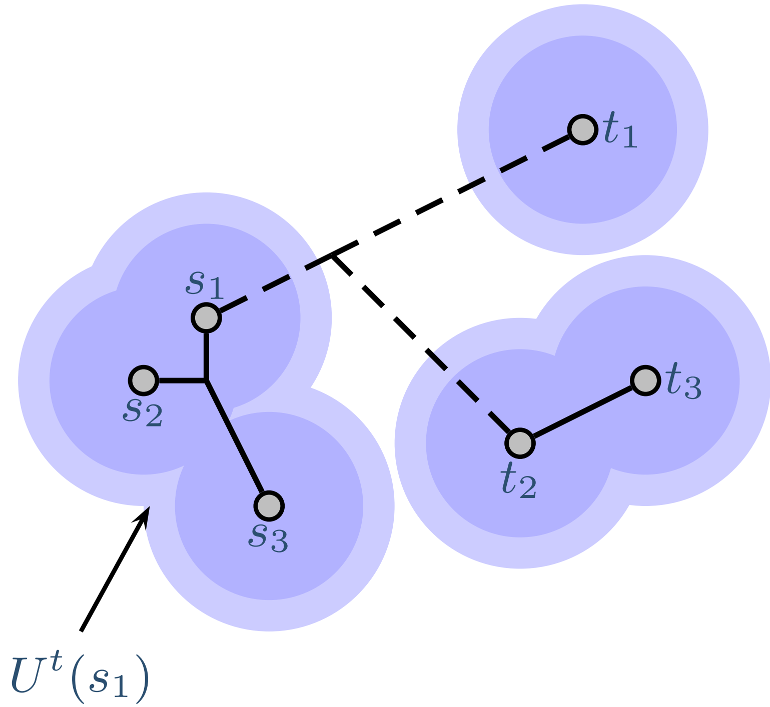
Facility location

Steiner Forests

Steiner Forest CS-Mechanism
- Try 1: SF and Shapley Value
- Try 2: Independent Activity Time
- Proving Cross-Monotonicity
- Proving Cost Recovery and Competitiveness
- Bounding $\sum_r \xi_R(r)$

Lifted-Cut Dual Relaxation

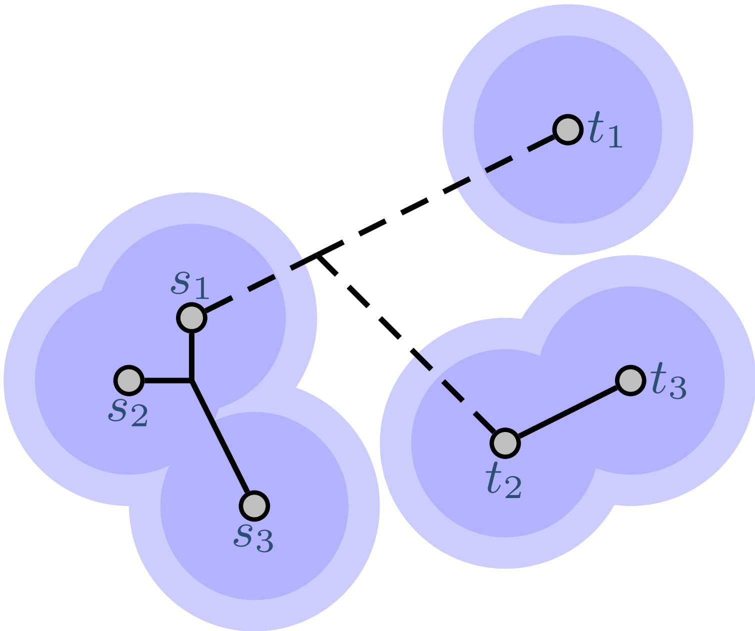
Lower Bounds


- $U^t(r)$ : moat of terminal $r$ at time $t$.

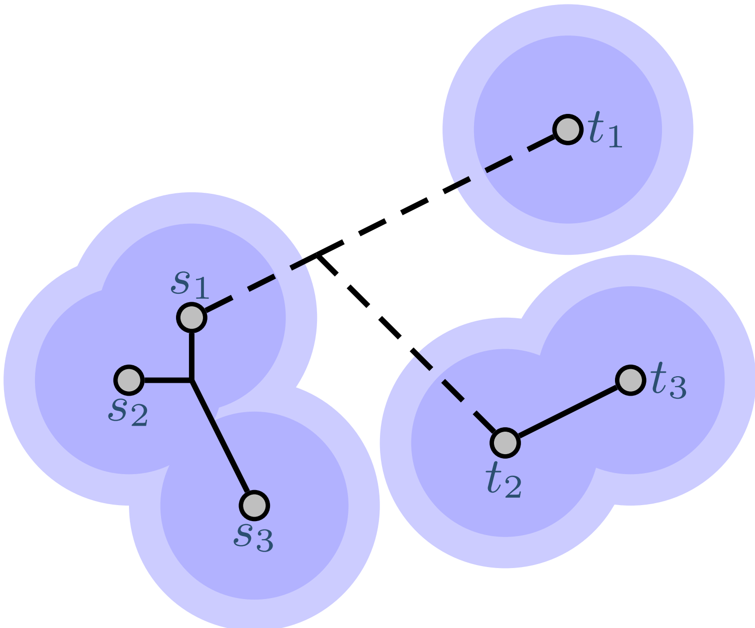
# Try 1: SF and Shapley Value





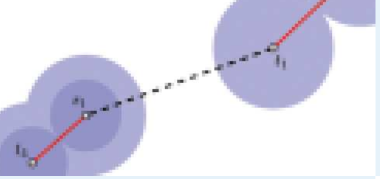

- $U^t(r)$ : moat of terminal $r$ at time $t$.
- $a^t(r)$ : number of active terminals in $U^t(r)$; e.g., $a^t(s_1) = 3$.

# Try 1: SF and Shapley Value



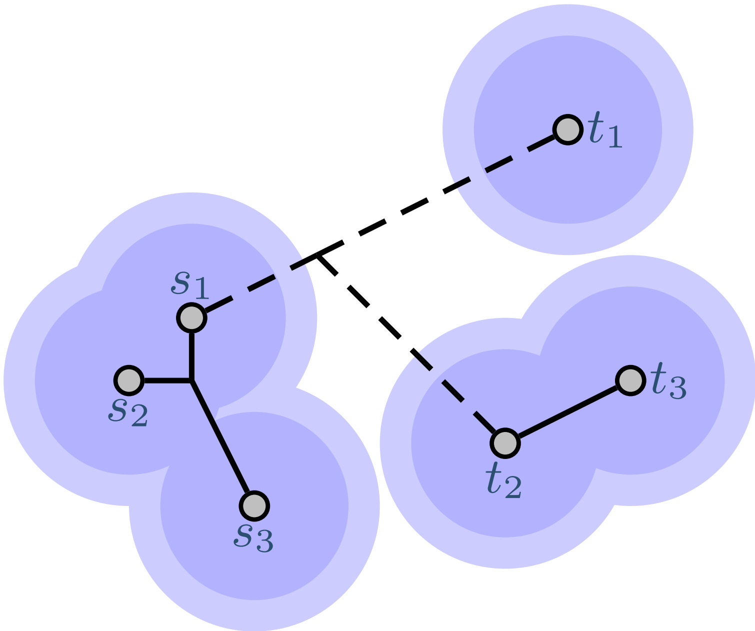

- $U^t(r)$ : moat of terminal $r$ at time $t$.
- $a^t(r)$ : number of active terminals in $U^t(r)$; e.g., $a^t(s_1) = 3$.
- Suppose terminal $r \in R$ becomes inactive at time $T$. Cost-share:

$$\xi_Q(r) = \int_0^T \frac{1}{a^t(r)}\, dt$$



# Try 1: SF and Shapley Value






- $U^t(r)$ : moat of terminal $r$ at time $t$.
- $a^t(r)$ : number of active terminals in $U^t(r)$; e.g., $a^t(s_1) = 3$.
- Suppose terminal $r \in R$ becomes inactive at time $T$. Cost-share:

$$\xi_Q(r) = \int_0^T \frac{1}{a^t(r)}\, dt$$

- For terminal-pair $(s, t) \in R$:

$$\xi_Q(s,t) = \xi_Q(s) + \xi_Q(t)$$

# Try 1: SF and Shapley Value


- Talk Outline

Cost-Sharing Mechanisms

Facility location

Steiner Forests

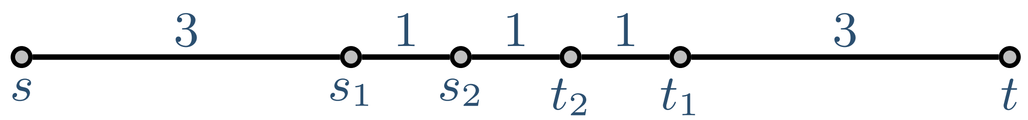
Steiner Forest CS-Mechanism
- Try 1: SF and Shapley Value
- Try 2: Independent Activity Time
- Proving Cross-Monotonicity
- Proving Cost Recovery and Competitiveness
- Bounding $\sum_r \xi_R(r)$

Lifted-Cut Dual Relaxation

Lower Bounds


Q: Is $\xi$ cross-monotonic? A: No!

Simple example: $R = \{(s,t),(s_1,t_1),(s_2,t_2)\}$, $R_0 = R \setminus \{(s_2,t_2)\}$

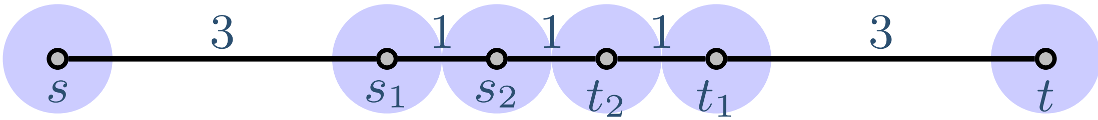


# Try 1: SF and Shapley Value

Q: Is $\xi$ cross-monotonic? A: No!

Simple example: $R = \{(s,t),(s_1,t_1),(s_2,t_2)\}$, $R_0 = R \setminus \{(s_2,t_2)\}$

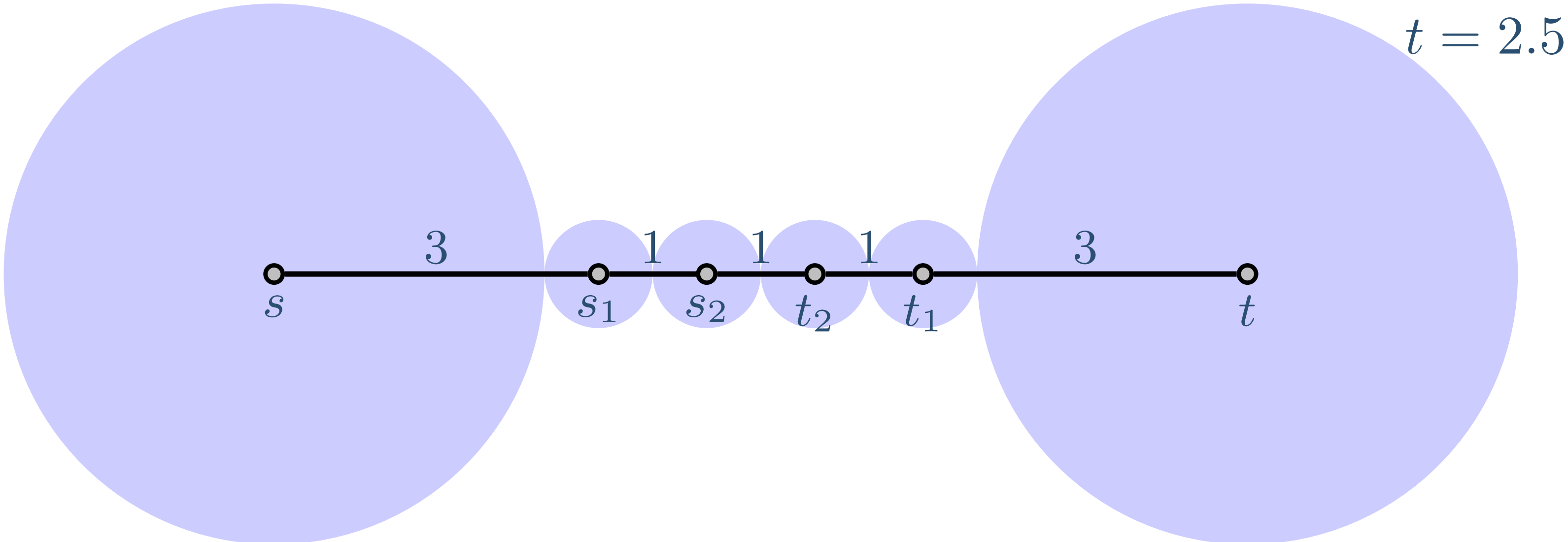
$t = 0.5$



# Try 1: SF and Shapley Value

Q: Is $\xi$ cross-monotonic? A: No!

Simple example: $R = \{(s,t),(s_1,t_1),(s_2,t_2)\}$, $R_0 = R \setminus \{(s_2,t_2)\}$



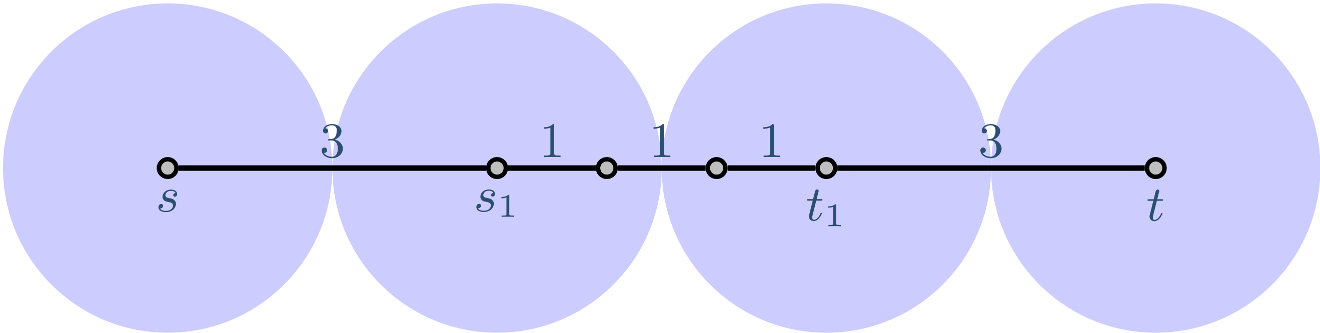

- $\xi_R(s,t) = 5$

# Try 1: SF and Shapley Value

Q: Is $\xi$ cross-monotonic? A: No!

Simple example: $R = \{(s,t),(s_1,t_1),(s_2,t_2)\}$, $R_0 = R \setminus \{(s_2,t_2)\}$

$t = 1.5$



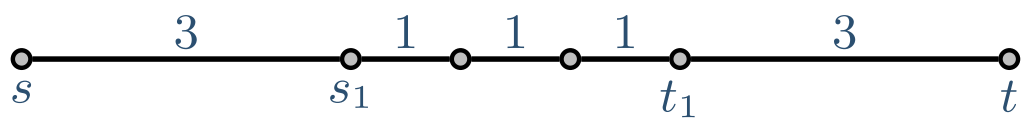

- $\xi_R(s,t) = 5$
- $\xi_{R_0}(s,t) = 3$

# Try 1: SF and Shapley Value



Q: Is $\xi$ cross-monotonic? A: No!

Simple example: $R = \{(s,t),(s_1,t_1),(s_2,t_2)\}$, $R_0 = R \setminus \{(s_2,t_2)\}$




- $\xi_R(s,t) = 5$
- $\xi_{R_0}(s,t) = 3$
- Activity time of $(s,t)$ depends on $(s_2,t_2)$!

# Try 2: Independent Activity Time

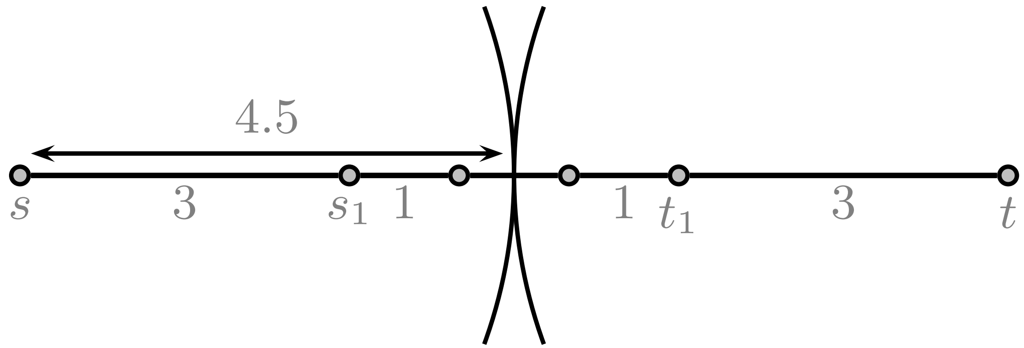


- Previous try: Activity-times of terminal pairs inter-dependent.

# Try 2: Independent Activity Time



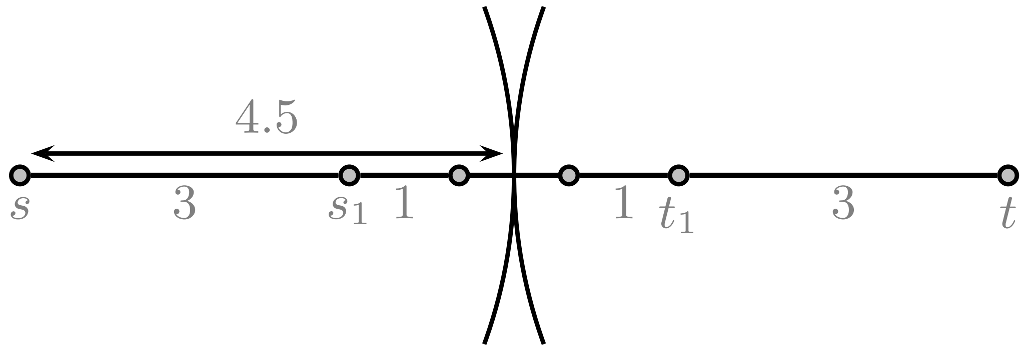
- Previous try: Activity-times of terminal pairs inter-dependent.
  How long would they need to connect if no other terminal was in the game?



# Try 2: Independent Activity Time



- Previous try: Activity-times of terminal pairs inter-dependent. How long would they need to connect if no other terminal was in the game?




- Death time of terminal-pair $(s,t) \in R$:

$$\mathrm{d}(s,t) = \frac{c(s,t)}{2},$$

where $c(s,t)$ is cost of minimum-cost $s,t$-path.

# Try 2: Independent Activity Time



- Extend to terminal nodes: $\mathrm{d}(r) = \mathrm{d}(s,t)$ for $r \in \{s,t\}$.

# Try 2: Independent Activity Time



- Extend to terminal nodes: $\mathrm{d}(r) = \mathrm{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathrm{d}(r)$.

# Try 2: Independent Activity Time



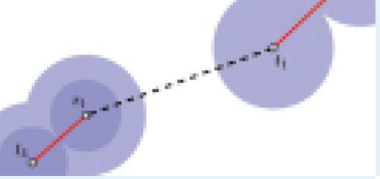
- Extend to terminal nodes: $\mathrm{d}(r) = \mathrm{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathrm{d}(r)$.
- SF grows moats as long as they contain active terminals.

# Try 2: Independent Activity Time



- Extend to terminal nodes: $\mathsf{d}(r) = \mathsf{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathsf{d}(r)$.
- SF grows moats as long as they contain active terminals.
- Cost-share of terminal $r$:

$$\xi_R(r) = \int_0^{\mathsf{d}(r)} \frac{1}{a^t(r)}\, dt.$$

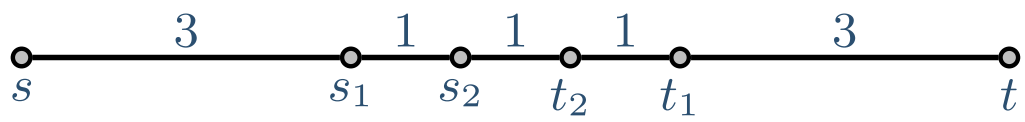


# Try 2: Independent Activity Time



- Extend to terminal nodes: $\mathsf{d}(r) = \mathsf{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathsf{d}(r)$.
- SF grows moats as long as they contain active terminals.
- Cost-share of terminal $r$:

$$\xi_R(r) = \int_0^{\mathsf{d}(r)} \frac{1}{a^t(r)}\, dt.$$



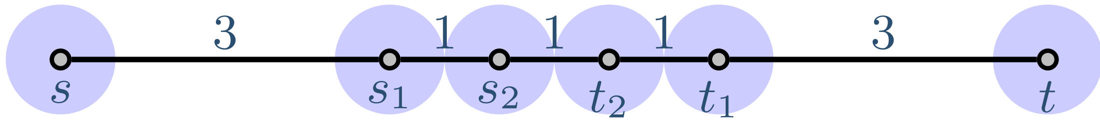
# Try 2: Independent Activity Time



- Extend to terminal nodes: $\mathsf{d}(r) = \mathsf{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathsf{d}(r)$.
- SF grows moats as long as they contain active terminals.
- Cost-share of terminal $r$:

$$\xi_R(r) = \int_0^{\mathsf{d}(r)} \frac{1}{a^t(r)}\, dt.$$

$t = 0.5$



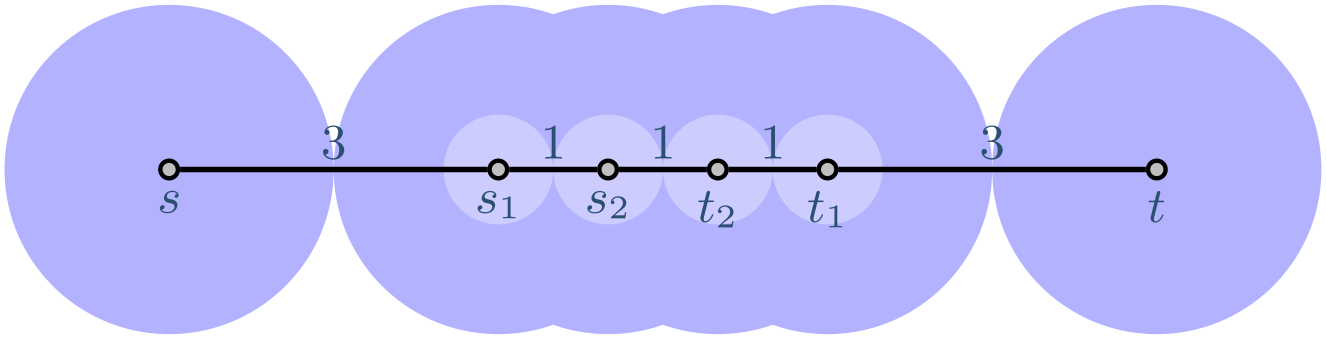
# Try 2: Independent Activity Time



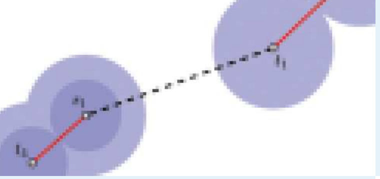
- Extend to terminal nodes: $\mathsf{d}(r) = \mathsf{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathsf{d}(r)$.
- SF grows moats as long as they contain active terminals.
- Cost-share of terminal $r$:

$$\xi_R(r) = \int_0^{\mathsf{d}(r)} \frac{1}{a^t(r)}\, dt.$$

$t = 1.5$




- $\xi_R(s_1, t_1) = 2$

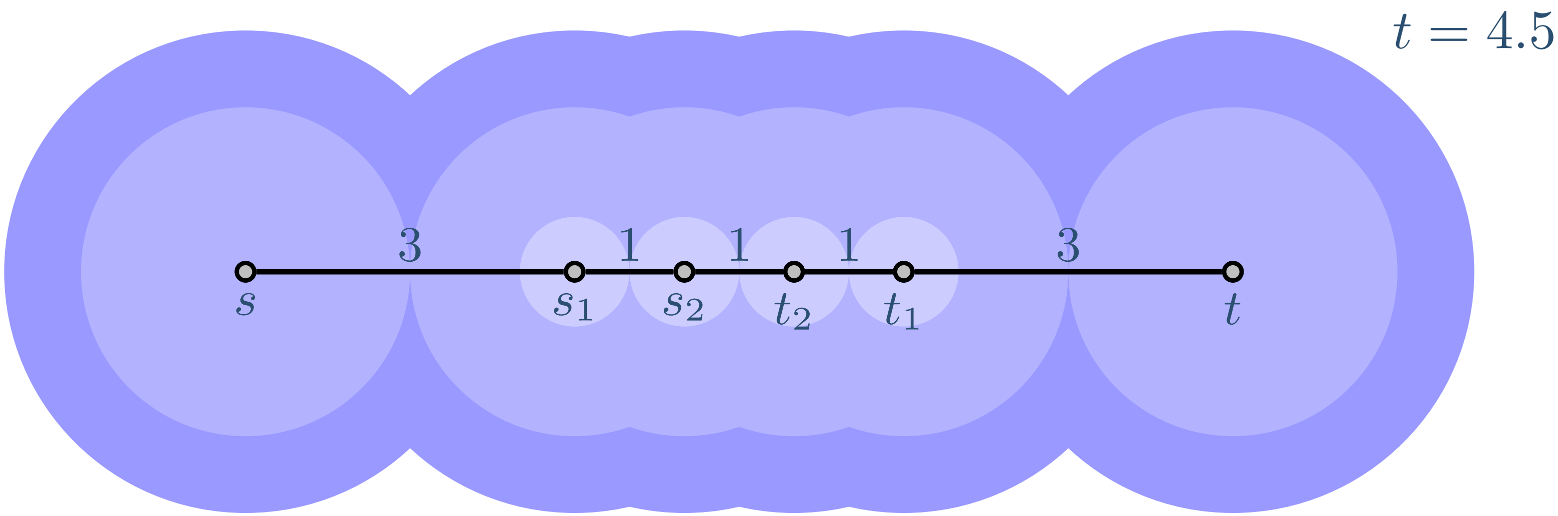


# Try 2: Independent Activity Time

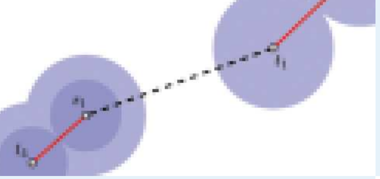


- Extend to terminal nodes: $\mathsf{d}(r) = \mathsf{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathsf{d}(r)$.
- SF grows moats as long as they contain active terminals.
- Cost-share of terminal $r$:

$$\xi_R(r) = \int_0^{\mathsf{d}(r)} \frac{1}{a^t(r)}\, dt.$$




- $\xi_R(s_1,t_1) = 2$, $\xi_R(s,t) = 6$.

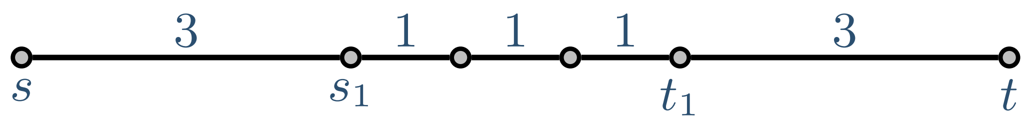


# Try 2: Independent Activity Time



- Extend to terminal nodes: $\mathsf{d}(r) = \mathsf{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathsf{d}(r)$.
- SF grows moats as long as they contain active terminals.
- Cost-share of terminal $r$:

$$\xi_R(r) = \int_0^{\mathsf{d}(r)} \frac{1}{a^t(r)}\, dt.$$



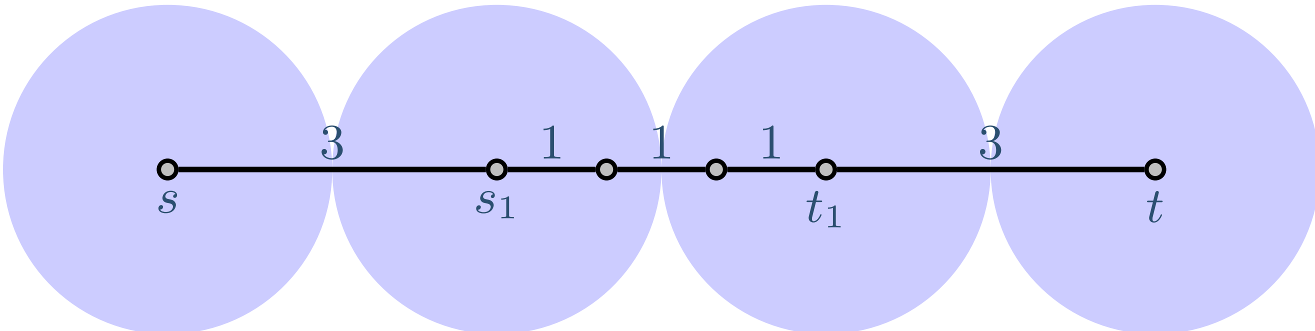

- $\xi_R(s_1,t_1) = 2$, $\xi_R(s,t) = 6$.

# Try 2: Independent Activity Time

- Extend to terminal nodes: $\mathsf{d}(r) = \mathsf{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathsf{d}(r)$.
- SF grows moats as long as they contain active terminals.
- Cost-share of terminal $r$:

$$\xi_R(r) = \int_0^{\mathsf{d}(r)} \frac{1}{a^t(r)}\, dt.$$

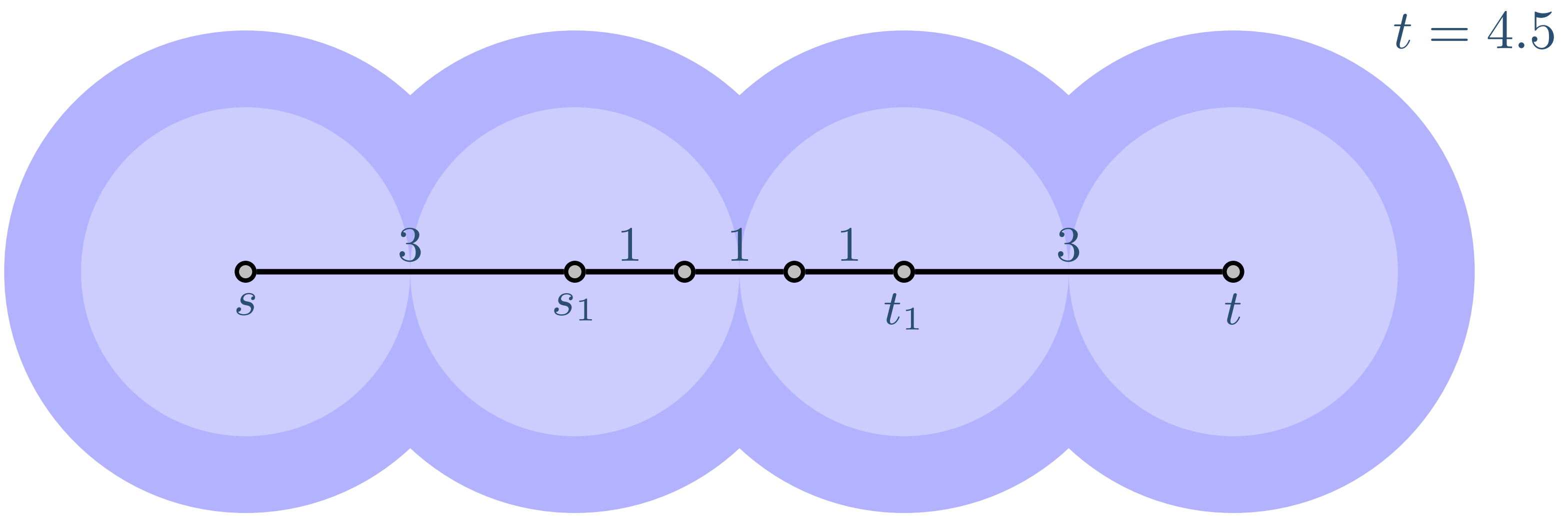
$t = 1.5$




- $\xi_R(s_1,t_1) = 2$, $\xi_R(s,t) = 6$.
- $\xi_{R_0}(s_1,t_1) = 3$

- Extend to terminal nodes: $\mathsf{d}(r) = \mathsf{d}(s,t)$ for $r \in \{s,t\}$.
- Terminal $r$ is active until time $\mathsf{d}(r)$.
- SF grows moats as long as they contain active terminals.
- Cost-share of terminal $r$:

$$\xi_R(r) = \int_0^{\mathsf{d}(r)} \frac{1}{a^t(r)}\, dt.$$




- $\xi_R(s_1,t_1) = 2$, $\xi_R(s,t) = 6$.
- $\xi_{R_0}(s_1,t_1) = 3$, $\xi_{R_0}(s,t) = 6$.

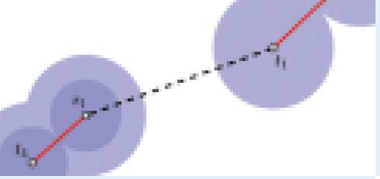
# Proving Cross-Monotonicity



**Lemma:** $\xi$ is cross-monotonic.

Proof:

- $R_0 = R \setminus \{(s,t)\}$.



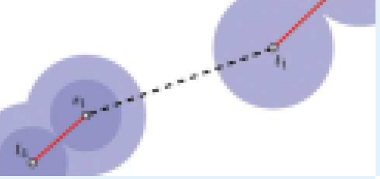
# Proving Cross-Monotonicity



**Lemma:** $\xi$ is cross-monotonic.

Proof:

- $R_0 = R \setminus \{(s,t)\}$.
- $U_0^t(r)$: Moat of $r$ at time $t$ in SF$(R_0)$.



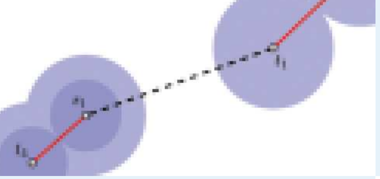
# Proving Cross-Monotonicity



**Lemma:** $\xi$ is cross-monotonic.

Proof:

- $R_0 = R \setminus \{(s,t)\}$.
- $U_0^t(r)$: Moat of $r$ at time $t$ in SF$(R_0)$.
- $a_0^t(r)$: Number of active terminals in $U_0^t(r)$.



# Proving Cross-Monotonicity



**Lemma:** $\xi$ is cross-monotonic.

Proof:

- $R_0 = R \setminus \{(s,t)\}$.
- $U_0^t(r)$: Moat of $r$ at time $t$ in SF$(R_0)$.
- $a_0^t(r)$: Number of active terminals in $U_0^t(r)$.
- Death-times of terminal-pairs are instance independent! Therefore: For each $r \in R_0$:

$$U_0^t(r) \text{ active } \Longrightarrow U^t(r) \text{ active and } U_0^t(r) \subseteq U^t(r).$$

# Proving Cross-Monotonicity



**Lemma:** $\xi$ is cross-monotonic.

Proof:

- $R_0 = R \setminus \{(s,t)\}$.
- $U_0^t(r)$: Moat of $r$ at time $t$ in SF$(R_0)$.
- $a_0^t(r)$: Number of active terminals in $U_0^t(r)$.
- Death-times of terminal-pairs are instance independent! Therefore: For each $r \in R_0$:

$$U_0^t(r) \text{ active } \implies U^t(r) \text{ active and } U_0^t(r) \subseteq U^t(r).$$

- Implies: $a_0^t(r) \leq a^t(r)$ for all $t \geq 0$ and $r \in R_0$.

# Proving Cross-Monotonicity



**Lemma:** $\xi$ is cross-monotonic.

Proof:

- $R_0 = R \setminus \{(s,t)\}$.
- $U_0^t(r)$: Moat of $r$ at time $t$ in SF$(R_0)$.
- $a_0^t(r)$: Number of active terminals in $U_0^t(r)$.
- Death-times of terminal-pairs are instance independent! Therefore: For each $r \in R_0$:

$$U_0^t(r) \text{ active } \implies U^t(r) \text{ active and } U_0^t(r) \subseteq U^t(r).$$

- Implies: $a_0^t(r) \leq a^t(r)$ for all $t \geq 0$ and $r \in R_0$.
- We obtain: For each $r \in R_0$:

$$\xi_R(r) = \int_0^{\mathsf{d}(r)} \frac{1}{a^t(r)}\, dt \leq \int_0^{\mathsf{d}(r)} \frac{1}{a_0^t(r)}\, dt = \xi_{R_0}(r).$$

# Proving Cost Recovery and Competitiveness



**Lemma:** $\xi$ satisfies cost recovery and $2$-approximate competitiveness.

Proof:

- Let $F$ and $y$ be forest and corresponding dual computed by SF.

# Proving Cost Recovery and Competitiveness



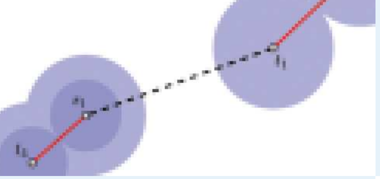
**Lemma:** $\xi$ satisfies cost recovery and $2$-approximate competitiveness.

Proof:

- Let $F$ and $y$ be forest and corresponding dual computed by SF.
- SF-Theorem implies

$$c(F) \leq 2 \cdot \sum_{U \subseteq V} y_U = 2 \cdot \sum_{r \in R} \xi_R(r).$$

$y$ is not dual feasible! Some active moats do not correspond to Steiner cuts.



# Proving Cost Recovery and Competitiveness



**Lemma:** $\xi$ satisfies cost recovery and $2$-approximate competitiveness.

Proof:

- Let $F$ and $y$ be forest and corresponding dual computed by SF.
- SF-Theorem implies

$$c(F) \leq 2 \cdot \sum_{U \subseteq V} y_U = 2 \cdot \sum_{r \in R} \xi_R(r).$$

  $y$ is not dual feasible! Some active moats do not correspond to Steiner cuts.
- Can show: $\sum_{r \in R} \xi_R(r) \leq \mathtt{opt}_R$.

# Proving Cost Recovery and Competitiveness

**Lemma:** $\xi$ satisfies cost recovery and $2$-approximate competitiveness.

Proof:

- Let $F$ and $y$ be forest and corresponding dual computed by SF.
- SF-Theorem implies

$$c(F) \leq 2 \cdot \sum_{U \subseteq V} y_U = 2 \cdot \sum_{r \in R} \xi_R(r).$$

  $y$ is not dual feasible! Some active moats do not correspond to Steiner cuts.

- Can show: $\sum_{r \in R} \xi_R(r) \leq \texttt{opt}_R$.
- This implies:

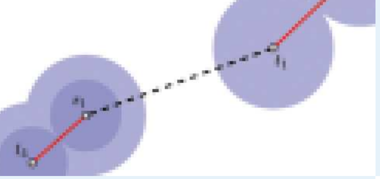
$$c(F) \leq 2 \cdot \sum_{r \in R} \xi_R(r) \leq 2 \cdot \texttt{opt}_R.$$

# Bounding $\sum_r \xi_R(r)$



- Assume that $R = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ and

$$\mathsf{d}(s_1, t_1) \leq \ldots \leq \mathsf{d}(s_k, t_k).$$



# Bounding $\sum_r \xi_R(r)$



- Assume that $R = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ and

$$\mathrm{d}(s_1, t_1) \leq \ldots \leq \mathrm{d}(s_k, t_k).$$

- Define total order: For $u \in \{s_i, t_i\}, v \in \{s_j, t_j\}$:

$$u \prec v \quad \text{iff} \quad \begin{cases} i < j & \text{or} \\ i = j & \text{and } \; u = s_j. \end{cases}$$

# Bounding $\sum_r \xi_R(r)$



- Assume that $R = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ and

$$\mathsf{d}(s_1, t_1) \leq \ldots \leq \mathsf{d}(s_k, t_k).$$

- Define total order: For $u \in \{s_i, t_i\}, v \in \{s_j, t_j\}$:

$$u \prec v \quad \text{iff} \quad \begin{cases} i < j & \text{or} \\ i = j & \text{and} \quad u = s_j. \end{cases}$$

- $v \in R$ is **responsible** at time $t$ if $u \prec v$ for all $u \in U^t(v)$.
  Write: $r^t(v) = 1$ iff $v$ is responsible at time $t$.

# Bounding $\sum_r \xi_R(r)$



- Assume that $R = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ and

$$\mathsf{d}(s_1, t_1) \leq \ldots \leq \mathsf{d}(s_k, t_k).$$

- Define total order: For $u \in \{s_i, t_i\}, v \in \{s_j, t_j\}$:

$$u \prec v \quad \text{iff} \quad \begin{cases} i < j & \text{or} \\ i = j & \text{and} \;\; u = s_j. \end{cases}$$

- $v \in R$ is **responsible** at time $t$ if $u \prec v$ for all $u \in U^t(v)$.
  Write: $r^t(v) = 1$ iff $v$ is responsible at time $t$.
- Total responsibility time of $v \in R$:

$$r(v) = \int_0^{\mathsf{d}(v)} r^t(v)\, dt$$

# Bounding $\sum_r \xi_R(r)$



- Assume that $R = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ and

$$\mathsf{d}(s_1, t_1) \leq \ldots \leq \mathsf{d}(s_k, t_k).$$

- Define total order: For $u \in \{s_i, t_i\}, v \in \{s_j, t_j\}$:

$$u \prec v \quad \text{iff} \quad \begin{cases} i < j & \text{or} \\ i = j & \text{and } \ u = s_j. \end{cases}$$

- $v \in R$ is **responsible** at time $t$ if $u \prec v$ for all $u \in U^t(v)$. Write: $r^t(v) = 1$ iff $v$ is responsible at time $t$.
- Total responsibility time of $v \in R$:

$$r(v) = \int_0^{\mathsf{d}(v)} r^t(v)\, dt$$

- Intuition: No sharing of dual growth. **The responsible terminal gets everything!**

# Bounding $\sum_r \xi_R(r)$



- Exactly one responsible vertex per growing moat in SF. Hence:

$$\sum_{v \in R} \xi_R(v) = \sum_{v \in R} r(v).$$

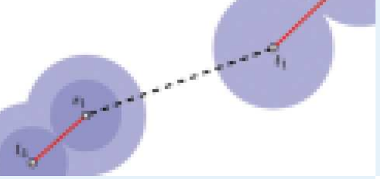
# Bounding $\sum_r \xi_R(r)$



- Exactly one responsible vertex per growing moat in SF. Hence:

$$\sum_{v \in R} \xi_R(v) = \sum_{v \in R} r(v).$$

- Let $F^*$ be a minimum-cost Steiner forest spanning $R$.



# Bounding $\sum_r \xi_R(r)$



- Exactly one responsible vertex per growing moat in SF. Hence:

$$\sum_{v \in R} \xi_R(v) = \sum_{v \in R} r(v).$$

- Let $F^*$ be a minimum-cost Steiner forest spanning $R$.
- Consider tree $T$ in $F^*$ and assume that $T$ spans terminals $\{v_1, \ldots v_p\} \subseteq R$ with

$$r(v_1) \leq \ldots \leq r(v_p).$$

# Bounding $\sum_r \xi_R(r)$



- Exactly one responsible vertex per growing moat in SF. Hence:

$$\sum_{v \in R} \xi_R(v) = \sum_{v \in R} r(v).$$

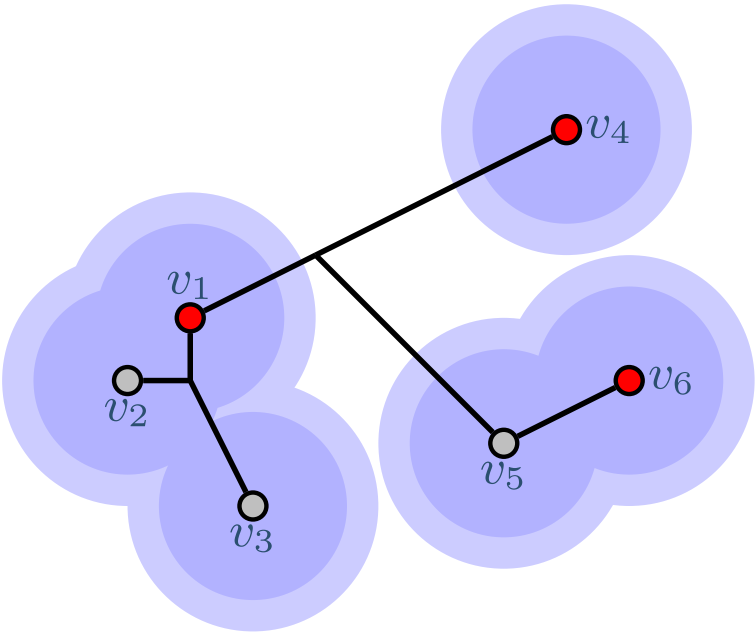
- Let $F^*$ be a minimum-cost Steiner forest spanning $R$.
- Consider tree $T$ in $F^*$ and assume that $T$ spans terminals $\{v_1, \ldots v_p\} \subseteq R$ with

$$r(v_1) \leq \ldots \leq r(v_p).$$

- Must have

$$\{U^t(v_i), \ldots, U^t(v_p)\}$$

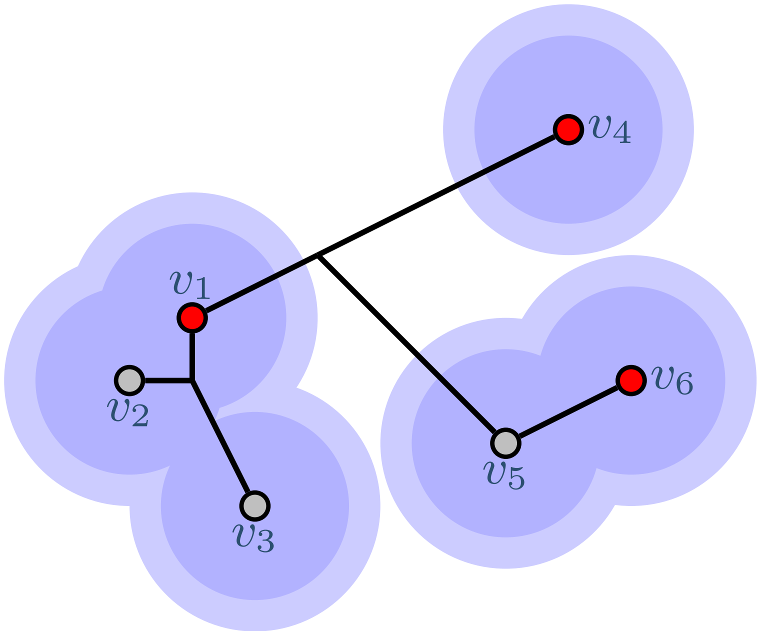
pairwise disjoint for $t \in [r(v_{i-1}), r(v_i))$.




- Example: A tree $T$ of $F^*$ connecting $6$ terminals

# Bounding $\sum_r \xi_R(r)$

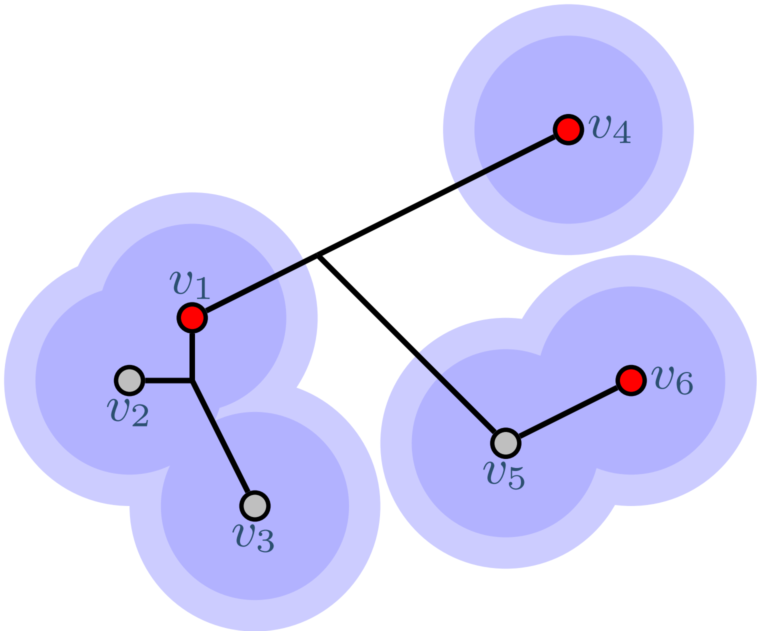





- Example: A tree $T$ of $F^*$ connecting $6$ terminals
- Red terminals are responsible.

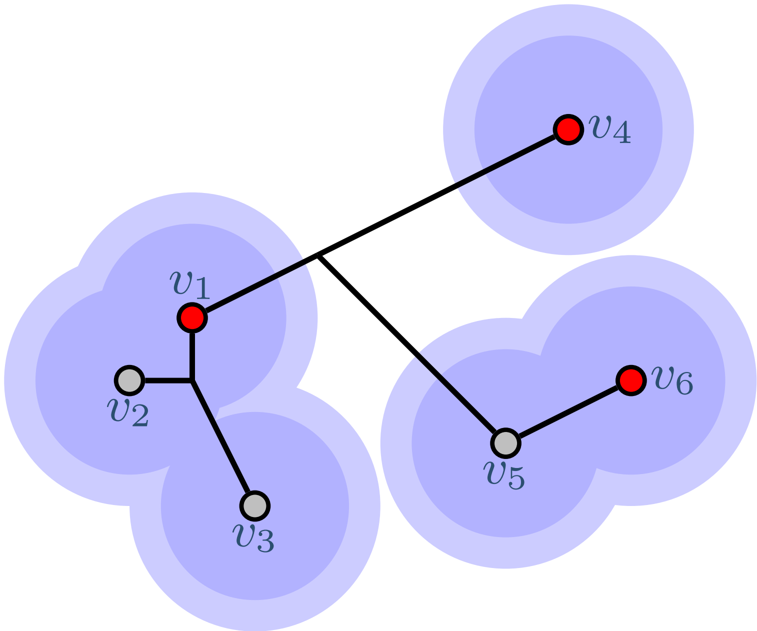
# Bounding $\sum_r \xi_R(r)$






- Example: A tree $T$ of $F^*$ connecting $6$ terminals
- Red terminals are responsible.
- Each vertex $v \in \{v_1, \ldots, v_p\}$ loads distinct part of $T$ of cost $r(v)$!

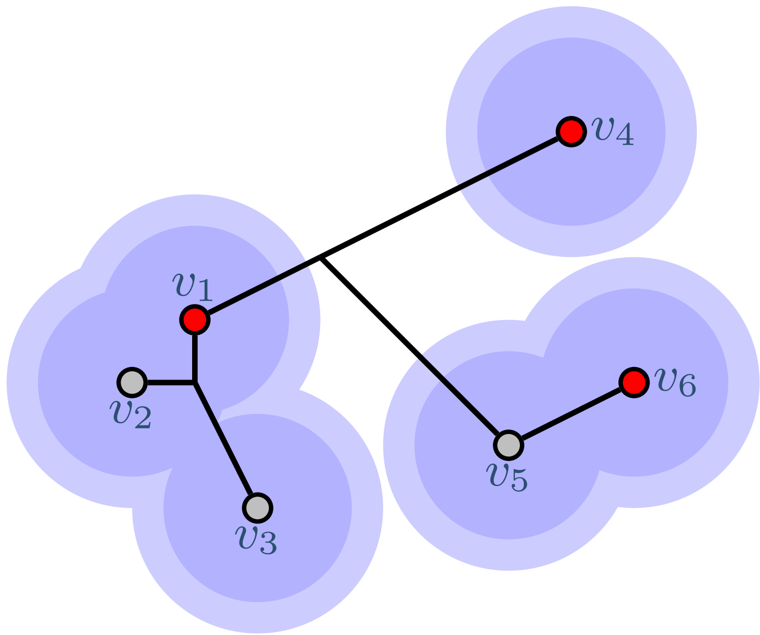
# Bounding $\sum_r \xi_R(r)$




- Example: A tree $T$ of $F^*$ connecting $6$ terminals
- Red terminals are responsible.
- Each vertex $v \in \{v_1, \ldots, v_p\}$ loads distinct part of $T$ of cost $r(v)$!
- Careful: Argument applies if there are at least two responsible terminals at time $t$.

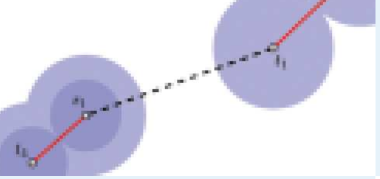
# Bounding $\sum_r \xi_R(r)$





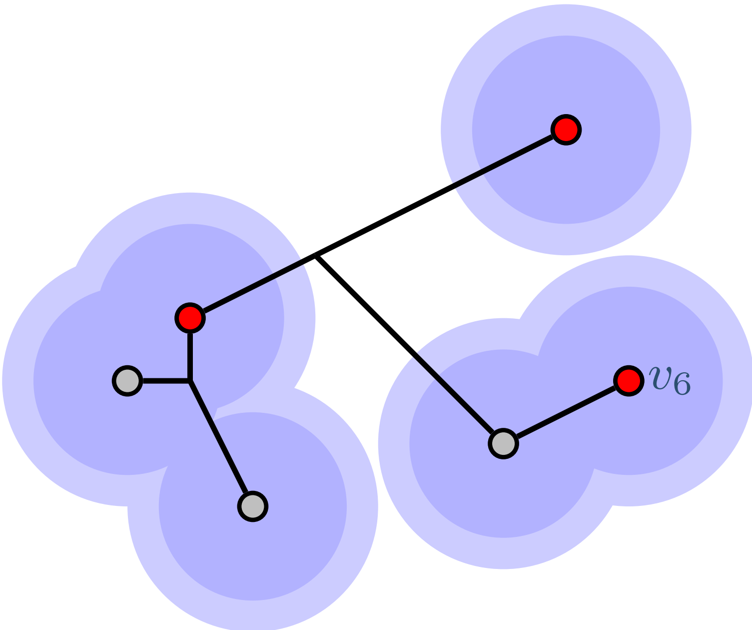

- Example: A tree $T$ of $F^*$ connecting $6$ terminals
- Red terminals are responsible.
- Each vertex $v \in \{v_1, \ldots, v_p\}$ loads distinct part of $T$ of cost $r(v)$!
- Careful: Argument applies if there are at least two responsible terminals at time $t$.
- Let $v_p$ be vertex with highest responsibility time. We get:

$$\sum_{i=1}^{p-1} r(v_i) \leq c(T).$$



# Bounding $\sum_r \xi_R(r)$



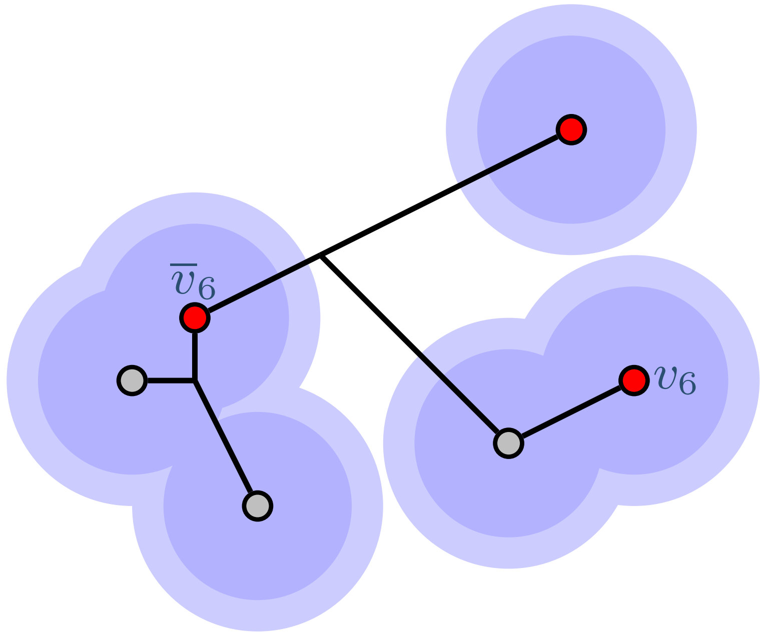



- Let $v_p$ be vertex with highest responsibility time. We get:

$$\sum_{i=1}^{p-1} r(v_i) \leq c(T).$$

# Bounding $\sum_r \xi_R(r)$





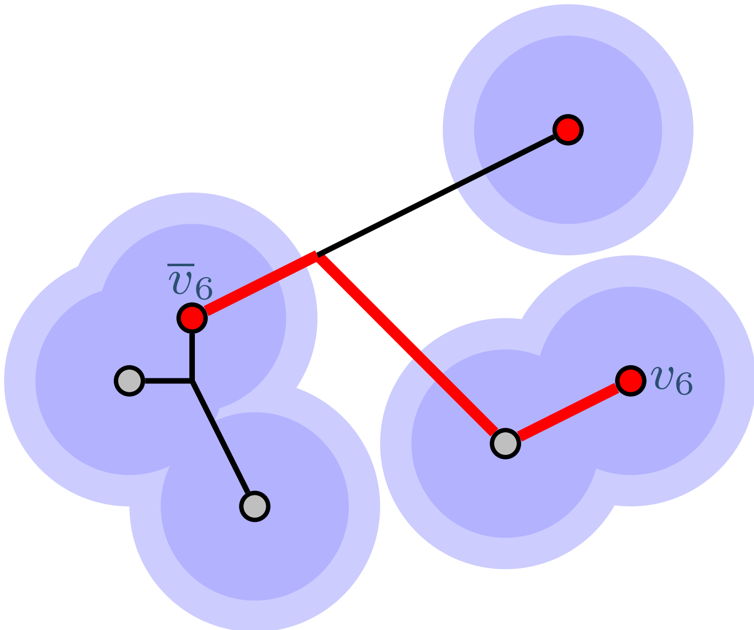

- Let $v_p$ be vertex with highest responsibility time. We get:

$$\sum_{i=1}^{p-1} r(v_i) \leq c(T).$$

- $v_p$'s mate is in $T$ as well!

# Bounding $\sum_r \xi_R(r)$





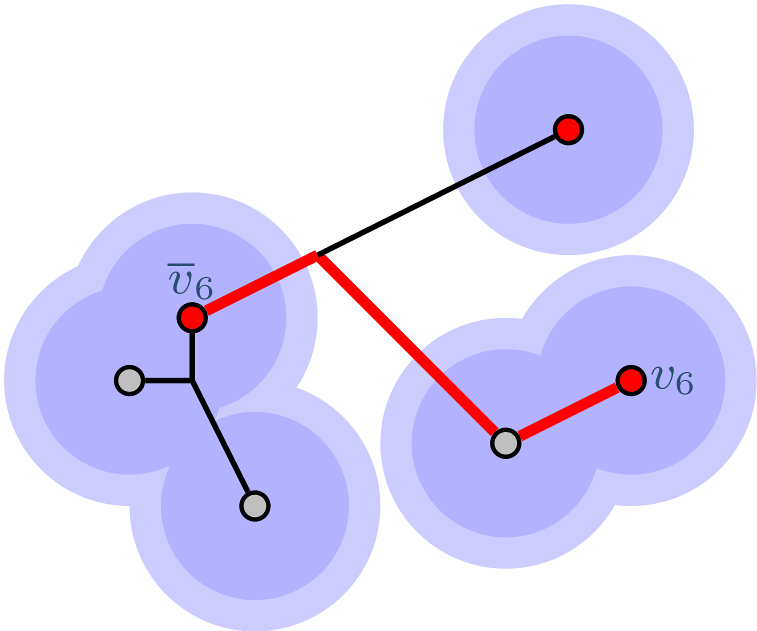

- Let $v_p$ be vertex with highest responsibility time. We get:

$$\sum_{i=1}^{p-1} r(v_i) \leq c(T).$$

- $v_p$'s mate is in $T$ as well!
- $r(v_p) \leq \mathsf{d}(v_p) \leq \frac{1}{2}c(T)$.

# Bounding $\sum_r \xi_R(r)$





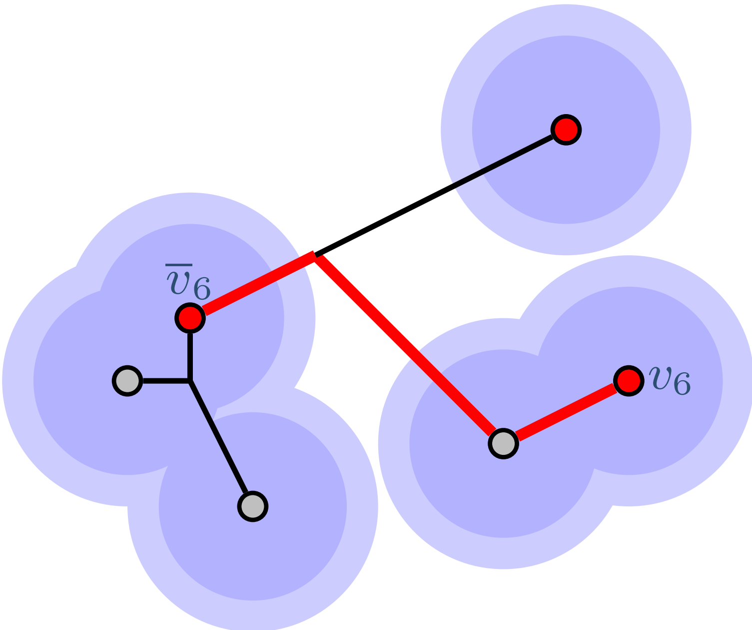

- Let $v_p$ be vertex with highest responsibility time. We get:

$$\sum_{i=1}^{p-1} r(v_i) \leq c(T).$$

- $v_p$'s mate is in $T$ as well!
- $r(v_p) \leq \mathsf{d}(v_p) \leq \frac{1}{2}c(T)$.
- Hence: $\sum_{i=1}^{p} r(v_i) \leq \frac{3}{2}c(T)$.

# Bounding $\sum_r \xi_R(r)$






- Let $v_p$ be vertex with highest responsibility time. We get:

$$\sum_{i=1}^{p-1} r(v_i) \leq c(T).$$

- $v_p$'s mate is in $T$ as well!
- $r(v_p) \leq \mathsf{d}(v_p) \leq \frac{1}{2}c(T)$.
- Hence: $\sum_{i=1}^{p} r(v_i) \leq \frac{3}{2}c(T)$.
- Summing over all trees $T \in F^*$:

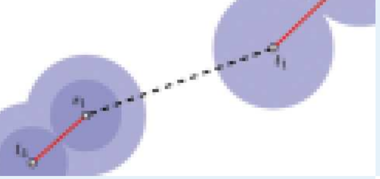
$$\sum_{v \in R} r(v) \leq \frac{3}{2} \cdot c(F^*).$$

# Lifted-Cut Dual Relaxation

# Approximating Steiner Forests



- Suppose our modified Steiner forest algorithm produces forest $F$ and (infeasible) dual $y$.



# Approximating Steiner Forests



- Suppose our modified Steiner forest algorithm produces forest $F$ and (infeasible) dual $y$.
- Can still show

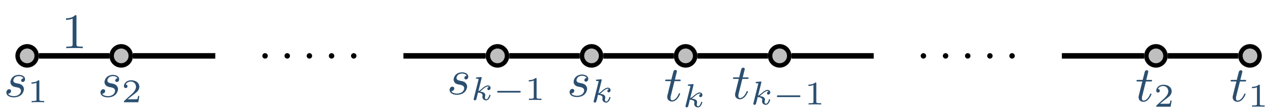
$$c(F) \leq (2 - 1/k) \sum_{U \subseteq V} y_U \leq (2 - 1/k) \cdot \texttt{opt}_R.$$

# Approximating Steiner Forests

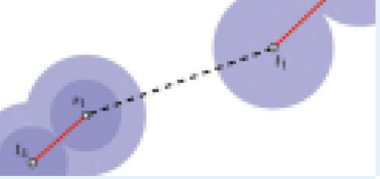
- Suppose our modified Steiner forest algorithm produces forest $F$ and (infeasible) dual $y$.
- Can still show

$$c(F) \leq (2 - 1/k) \sum_{U \subseteq V} y_U \leq (2 - 1/k) \cdot \texttt{opt}_R.$$

- Our dual is often much better than the SF-dual!




| | |
|---|---|
| opt | $2k-1$ |
| Standard SF-dual | $k$ |
| Our dual | $2k-1$ |

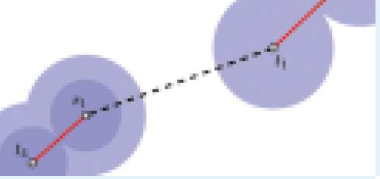


# Lifted-Cut Dual for Steiner Forests



- Fix an order $\prec$ on the terminal pairs:

$$\mathsf{d}(s_1, t_1) \leq \mathsf{d}(s_2, t_2) \leq \cdots \leq \mathsf{d}(s_k, t_k)$$



# Lifted-Cut Dual for Steiner Forests



- Fix an order $\prec$ on the terminal pairs:

$$\mathsf{d}(s_1, t_1) \leq \mathsf{d}(s_2, t_2) \leq \cdots \leq \mathsf{d}(s_k, t_k)$$

- Associate each cut $U \subseteq V$ with a terminal (pair).

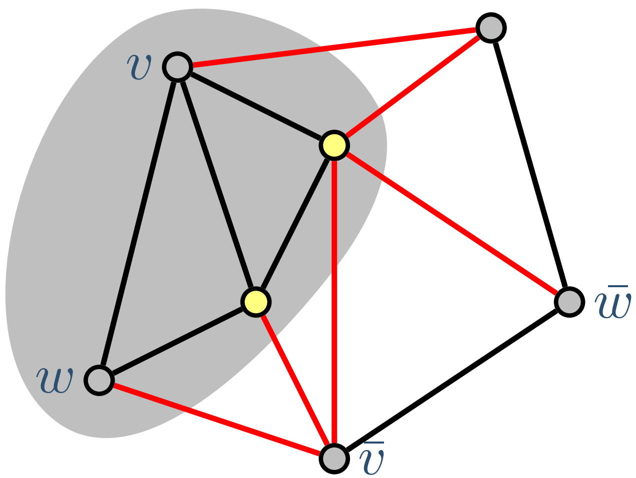
# Lifted-Cut Dual for Steiner Forests

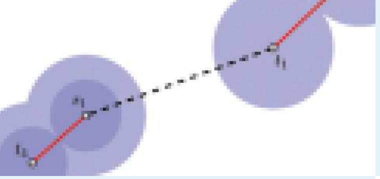


- Fix an order $\prec$ on the terminal pairs:

$$\mathsf{d}(s_1, t_1) \leq \mathsf{d}(s_2, t_2) \leq \cdots \leq \mathsf{d}(s_k, t_k)$$

- Associate each cut $U \subseteq V$ with a terminal (pair).
- Example: $(v, \bar{v}) \prec (w, \bar{w})$.





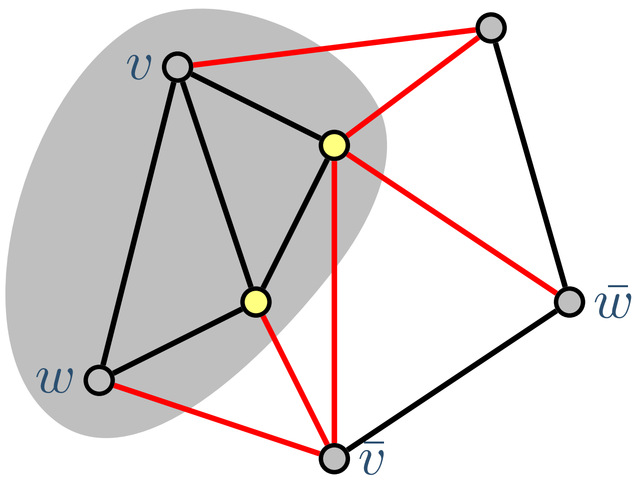
# Lifted-Cut Dual for Steiner Forests



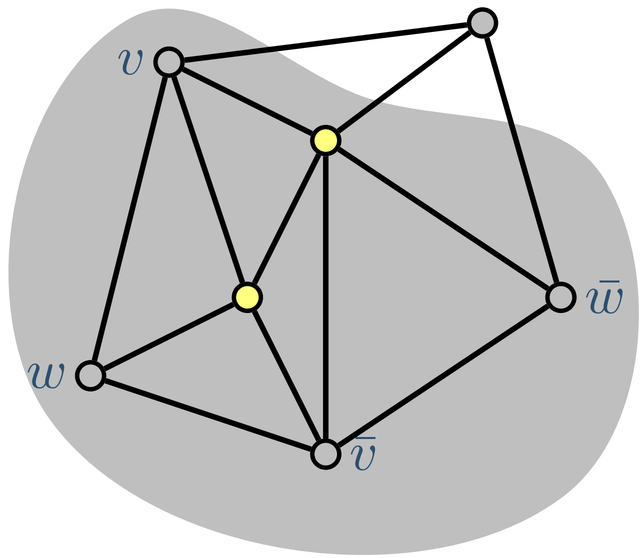
- Fix an order $\prec$ on the terminal pairs:

$$\mathsf{d}(s_1, t_1) \leq \mathsf{d}(s_2, t_2) \leq \cdots \leq \mathsf{d}(s_k, t_k)$$

- Associate each cut $U \subseteq V$ with a terminal (pair).
- Example: $(v, \bar{v}) \prec (w, \bar{w})$.




$S \in \mathcal{U}_w$




$S \in \mathcal{U}_{w,\bar{w}}$

# Lifted-Cut Dual for Steiner Forests



$$\max \quad \sum_{U \subseteq V} y_U$$

$$\text{s.t.} \qquad \begin{aligned} \sum_{U \subseteq V : e \in \delta(U)} y_U &\leq c_e && \forall e \in E \\ \sum_{U \in \mathcal{U}_w} y_U + \sum_{U \in \mathcal{U}_{w,\overline{w}}} y_U &\leq \mathsf{d}(w) && \forall w \in R \\ y &\geq 0 \end{aligned}$$

# Lifted-Cut Primal for Steiner Forests



$$
\begin{array}{llcll}
\min & \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w & & & \\
\text{s.t.} & \sum_{e \in \delta(U)} x_e + x_w & \geq & 1 & \forall U \in \mathcal{U}_w, \ \forall w \in R \\
 & \sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} & \geq & 1 & \forall U \in \mathcal{U}_{w,\bar{w}}, \ \forall (w, \bar{w}) \in R \\
 & x & \geq & 0 &
\end{array}
$$

# Optimal Integral Solution is a Steiner Forest



$$
\begin{aligned}
\text{min} \quad & \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w \\
\text{s.t.} \quad & \sum_{e \in \delta(U)} x_e + x_w \geq 1 \quad \forall U \in \mathcal{U}_w, \ \forall w \in R \\
& \textstyle\sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall U \in \mathcal{U}_{w,\bar{w}}, \ \forall (w, \bar{w}) \in R \\
& x \geq 0
\end{aligned}
$$

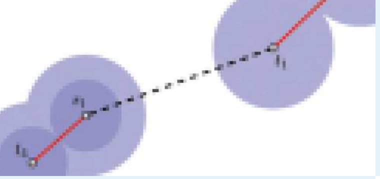
- Assume cut $U \in U_w$ violated. Cut $V/U \in U_{\bar{w}}$ is also violated.

# Optimal Integral Solution is a Steiner Forest



$$\begin{aligned} \min \quad & \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w \\ \text{s.t.} \quad & \sum_{e \in \delta(U)} x_e + x_w \geq 1 \quad \forall U \in \mathcal{U}_w,\ \forall w \in R \\ & \textstyle\sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall U \in \mathcal{U}_{w,\bar{w}},\ \forall (w, \bar{w}) \in R \\ & x \geq 0 \end{aligned}$$

- Assume cut $U \in U_w$ violated. Cut $V/U \in U_{\bar{w}}$ is also violated.
- Feasible integral solution assigns $x_w = x_{\bar{w}} = 1$



# Optimal Integral Solution is a Steiner Forest


- Talk Outline

Cost-Sharing Mechanisms

Facility location

Steiner Forests

Steiner Forest CS-Mechanism

Lifted-Cut Dual Relaxation
- Approximating Steiner Forests
- Lifted Cut-Dual for SF
- Lifted-Cut Primal for SF
- Optimal Integral Solution
- Relaxation of LC Primal

Lower Bounds


$$\min \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w$$

$$\text{s.t.} \quad \sum_{e \in \delta(U)} x_e + x_w \geq 1 \quad \forall U \in \mathcal{U}_w,\ \forall w \in R$$

$$\sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall U \in \mathcal{U}_{w,\bar{w}},\ \forall (w, \bar{w}) \in R$$

$$x \geq 0$$

- Assume cut $U \in U_w$ violated. Cut $V/U \in U_{\bar{w}}$ is also violated.
- Feasible integral solution assigns $x_w = x_{\bar{w}} = 1$
- Cost $(x_w + x_{\bar{w}})$, $d(w) = c(w, \bar{w})$ pays for the cost of connecting $w$ to $\bar{w}$.

# Relaxation of LC Primal and Steiner forests

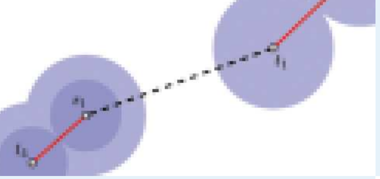


$$\min \quad \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w$$

$$\text{s.t.} \quad \sum_{e \in \delta(U)} x_e + x_w \geq 1 \quad \forall U \in \mathcal{U}_w, \ \forall w \in R$$

$$\sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall U \in \mathcal{U}_{w,\bar{w}}, \ \forall (w, \bar{w}) \in R$$

$$x \geq 0$$

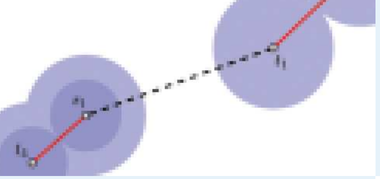


# Relaxation of LC Primal and Steiner forests



$$
\begin{aligned}
\text{min} \quad & \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w \\
\text{s.t.} \quad & \sum_{e \in \delta(U)} x_e + x_w \geq 1 \quad \forall U \in \mathcal{U}_w, \ \forall w \in R \\
& \textstyle\sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall U \in \mathcal{U}_{w,\bar{w}}, \ \forall (w, \bar{w}) \in R \\
& x \geq 0
\end{aligned}
$$

- **Theorem:** $\mathtt{opt}_{LP} \leq \mathtt{opt}_{LC} \leq \mathtt{opt}_R$.



# Relaxation of LC Primal and Steiner forests



$$\begin{aligned} \text{min} \quad & \sum_{e\in E} c_e \cdot x_e + \sum_{w\in R} \mathsf{d}(w) x_w \\ \text{s.t.} \quad & \sum_{e\in\delta(U)} x_e + x_w \geq 1 \quad \forall U \in \mathcal{U}_w,\ \forall w \in R \\ & \textstyle\sum_{e\in\delta(U)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall U \in \mathcal{U}_{w,\bar{w}},\ \forall (w,\bar{w}) \in R \\ & x \geq 0 \end{aligned}$$

- **Theorem:** $\mathtt{opt}_{LP} \leq \mathtt{opt}_{LC} \leq \mathtt{opt}_R$.
- Consider each tree $T$ of the optimal forest $F^*$.

# Relaxation of LC Primal and Steiner forests



$$
\begin{aligned}
\text{min} \quad & \sum_{e\in E} c_e \cdot x_e + \sum_{w\in R} \mathsf{d}(w) x_w & & \\
\text{s.t.} \quad & \sum_{e\in\delta(U)} x_e + x_w & \geq \quad 1 \quad & \forall U \in \mathcal{U}_w,\ \forall w \in R \\
& \textstyle\sum_{e\in\delta(U)} x_e + x_w + x_{\bar{w}} & \geq \quad 1 \quad & \forall U \in \mathcal{U}_{w,\bar{w}},\ \forall (w,\bar{w}) \in R \\
& x & \geq \quad 0 \quad &
\end{aligned}
$$

- **Theorem:** $\mathtt{opt}_{LP} \leq \mathtt{opt}_{LC} \leq \mathtt{opt}_R$.
- Consider each tree $T$ of the optimal forest $F^*$.
- $(w, \bar{w})$: responsible pair for $T$.

# Relaxation of LC Primal and Steiner forests



$$\begin{aligned}
\text{min} \quad & \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w \\
\text{s.t.} \quad & \sum_{e \in \delta(U)} x_e + x_w \geq 1 \quad \forall U \in \mathcal{U}_w, \ \forall w \in R \\
& \textstyle\sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall U \in \mathcal{U}_{w,\bar{w}}, \ \forall (w, \bar{w}) \in R \\
& x \geq 0
\end{aligned}$$

- **Theorem:** $\texttt{opt}_{LP} \leq \texttt{opt}_{LC} \leq \texttt{opt}_R$.
- Consider each tree $T$ of the optimal forest $F^*$.
- $(w, \bar{w})$: responsible pair for $T$.
- Path $P$ connects $w$ to $\bar{w}$ in $T$.

# Relaxation of LC Primal and Steiner forests



$$
\begin{aligned}
\text{min} \quad & \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w \\
\text{s.t.} \quad & \sum_{e \in \delta(U)} x_e + x_w \geq 1 \quad \forall U \in \mathcal{U}_w,\ \forall w \in R \\
& \textstyle\sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall U \in \mathcal{U}_{w,\bar{w}},\ \forall (w, \bar{w}) \in R \\
& x \geq 0
\end{aligned}
$$

- **Theorem:** $\mathtt{opt}_{LP} \leq \mathtt{opt}_{LC} \leq \mathtt{opt}_R$.
- Consider each tree $T$ of the optimal forest $F^*$.
- $(w, \bar{w})$: responsible pair for $T$.
- Path $P$ connects $w$ to $\bar{w}$ in $T$.
- Feasible solution $\mathcal{S}$:

# Relaxation of LC Primal and Steiner forests



$$\begin{array}{lrcll} \text{min} & \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w & & & \\ \text{s.t.} & \sum_{e \in \delta(U)} x_e + x_w & \geq & 1 & \forall U \in \mathcal{U}_w,\ \forall w \in R \\ & \sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} & \geq & 1 & \forall U \in \mathcal{U}_{w,\bar{w}},\ \forall (w, \bar{w}) \in R \\ & x & \geq & 0 & \end{array}$$

- **Theorem:** $\texttt{opt}_{LP} \leq \texttt{opt}_{LC} \leq \texttt{opt}_R$.
- Consider each tree $T$ of the optimal forest $F^*$.
- $(w, \bar{w})$: responsible pair for $T$.
- Path $P$ connects $w$ to $\bar{w}$ in $T$.
- Feasible solution $\mathcal{S}$:
  - Set $x_e = 1/2, \forall e \in P$ and $x_e = 1, \forall e \in T/P$

# Relaxation of LC Primal and Steiner forests



$$
\begin{array}{lrcll}
\text{min} & \sum_{e\in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w & & & \\
\text{s.t.} & \sum_{e\in\delta(U)} x_e + x_w & \geq & 1 & \forall U \in \mathcal{U}_w,\ \forall w \in R \\
 & \sum_{e\in\delta(U)} x_e + x_w + x_{\bar{w}} & \geq & 1 & \forall U \in \mathcal{U}_{w,\bar{w}},\ \forall (w,\bar{w}) \in R \\
 & x & \geq & 0 &
\end{array}
$$

- **Theorem:** $\mathtt{opt}_{LP} \leq \mathtt{opt}_{LC} \leq \mathtt{opt}_{R}$.
- Consider each tree $T$ of the optimal forest $F^*$.
- $(w, \bar{w})$: responsible pair for $T$.
- Path $P$ connects $w$ to $\bar{w}$ in $T$.
- Feasible solution $\mathcal{S}$:
  - Set $x_e = 1/2$, $\forall e \in P$ and $x_e = 1$, $\forall e \in T/P$
  - Set $x_w = x_{\bar{w}} = 1/2$ and $x_v = 0$, $\forall v \in V(T)/\{w, \bar{w}\}$.

# Relaxation of LC Primal and Steiner forests



$$\min \quad \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathrm{d}(w) x_w$$

$$\text{s.t.} \quad \sum_{e \in \delta(U)} x_e + x_w \geq 1 \quad \forall U \in \mathcal{U}_w, \ \forall w \in R$$

$$\textstyle\sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall U \in \mathcal{U}_{w,\bar{w}}, \ \forall (w, \bar{w}) \in R$$

$$x \geq 0$$

- **Theorem:** $\mathtt{opt}_{LP} \leq \mathtt{opt}_{LC} \leq \mathtt{opt}_R$.
- Consider each tree $T$ of the optimal forest $F^*$.
- $(w, \bar{w})$: responsible pair for $T$.
- Path $P$ connects $w$ to $\bar{w}$ in $T$.
- Feasible solution $\mathcal{S}$:
  - Set $x_e = 1/2, \forall e \in P$ and $x_e = 1, \forall e \in T/P$
  - Set $x_w = x_{\bar{w}} = 1/2$ and $x_v = 0, \forall v \in V(T)/\{w, \bar{w}\}$.
- $c(S) = c(T) - 1/2c(P) + 1/2(d(w) + d(\bar{w})) \leq c(T)$

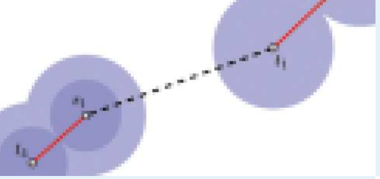
# Relaxation of LC Primal and Steiner forests



$$\begin{array}{llcll}
\text{min} & \sum_{e \in E} c_e \cdot x_e + \sum_{w \in R} \mathsf{d}(w) x_w & & & \\
\text{s.t.} & \sum_{e \in \delta(U)} x_e + x_w & \geq & 1 & \forall U \in \mathcal{U}_w,\ \forall w \in R \\
 & \sum_{e \in \delta(U)} x_e + x_w + x_{\bar{w}} & \geq & 1 & \forall U \in \mathcal{U}_{w,\bar{w}},\ \forall (w, \bar{w}) \in R \\
 & x & \geq & 0 &
\end{array}$$

- **Theorem:** $\mathtt{opt}_{LP} \leq \mathtt{opt}_{LC} \leq \mathtt{opt}_{R}$.
- Consider each tree $T$ of the optimal forest $F^*$.
- $(w, \bar{w})$: responsible pair for $T$.
- Path $P$ connects $w$ to $\bar{w}$ in $T$.
- Feasible solution $\mathcal{S}$:
  - Set $x_e = 1/2, \forall e \in P$ and $x_e = 1, \forall e \in T/P$
  - Set $x_w = x_{\bar{w}} = 1/2$ and $x_v = 0, \forall v \in V(T)/\{w, \bar{w}\}$.
- $c(S) = c(T) - 1/2 c(P) + 1/2(d(w) + d(\bar{w})) \leq c(T)$
- Solution $S$ is feasible.

# Lower bounds for cross-monotonic cost-sharing mechanisms



# Lower Bound for Cross-Monotonicity


- Talk Outline

Cost-Sharing Mechanisms

Facility location

Steiner Forests

Steiner Forest CS-Mechanism

Lifted-Cut Dual Relaxation

Lower Bounds
- Lower Bound for Cross-Monotonicity
- Lower Bound for Steiner Trees
- Limitations of Moulin mechanisms
- Objectives
- Known Results - Social Cost
- Summary
- Open Issues


- [Immorlica, Mahdian, Mirrokni ’05]: Give bounds on budget balance of cross-monotonic cost-sharing methods for facility location (3), vertex cover ($n^{1/3}$) and edge cover (2).

# Lower Bound for Cross-Monotonicity



- [Immorlica, Mahdian, Mirrokni '05]: Give bounds on budget balance of cross-monotonic cost-sharing methods for facility location (3), vertex cover ($n^{1/3}$) and edge cover (2).
- We prove a lower bound of $2$ for Steiner trees.

# Lower Bound for Cross-Monotonicity



- [Immorlica, Mahdian, Mirrokni '05]: Give bounds on budget balance of cross-monotonic cost-sharing methods for facility location (3), vertex cover ($n^{1/3}$) and edge cover (2).
- We prove a lower bound of $2$ for Steiner trees.
- $\Rightarrow$ our result for Steiner forest is tight.

# Lower Bound for Cross-Monotonicity



- [Immorlica, Mahdian, Mirrokni '05]: Give bounds on budget balance of cross-monotonic cost-sharing methods for facility location (3), vertex cover ($n^{1/3}$) and edge cover (2).
- We prove a lower bound of $2$ for Steiner trees.
- $\Rightarrow$ our result for Steiner forest is tight.

- Lower bounds are irrespective of time complexity.

# Lower Bound for Cross-Monotonicity



- [Immorlica, Mahdian, Mirrokni '05]: Give bounds on budget balance of cross-monotonic cost-sharing methods for facility location (3), vertex cover ($n^{1/3}$) and edge cover (2).
- We prove a lower bound of $2$ for Steiner trees.
- $\Rightarrow$ our result for Steiner forest is tight.

- Lower bounds are irrespective of time complexity.
- Proofs exploit the core property (weaker than cross-monotonicity):

$$\forall Q \subseteq V, \sum_{j \in Q} \xi_V(j) \leq \texttt{opt}_Q$$

# Lower Bound for Cross-Monotonicity



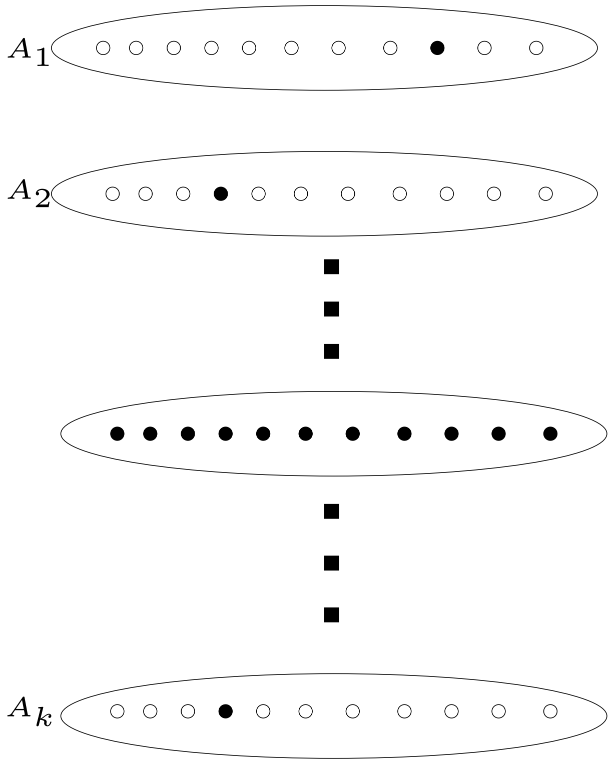
- [Immorlica, Mahdian, Mirrokni '05]: Give bounds on budget balance of cross-monotonic cost-sharing methods for facility location (3), vertex cover ($n^{1/3}$) and edge cover (2).
- We prove a lower bound of $2$ for Steiner trees.
- $\Rightarrow$ our result for Steiner forest is tight.

- Lower bounds are irrespective of time complexity.
- Proofs exploit the core property (weaker than cross-monotonicity):

$$\forall Q \subseteq V, \sum_{j \in Q} \xi_V(j) \leq \texttt{opt}_Q$$

- Turns into a lower bound on budget-balance of group-strategyproof methods only if there are no free riders.

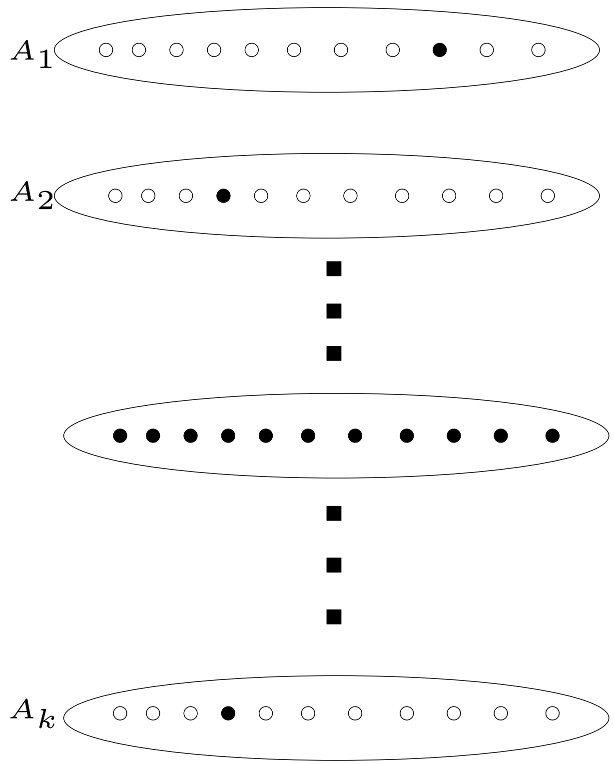
- $k$ pairwise disjoint classes $A_i$ of $m$ vertices.



# Lower Bound for Steiner Trees



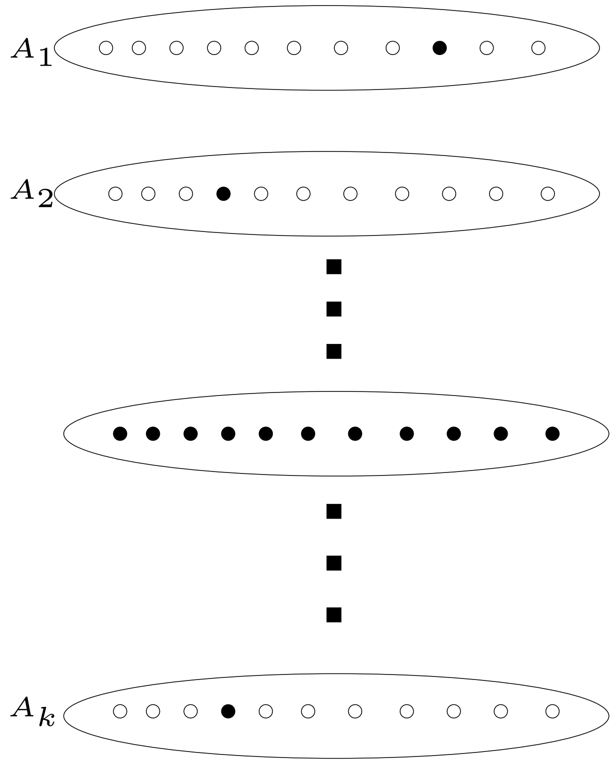
- $k$ pairwise disjoint classes $A_i$ of $m$ vertices.
- Select a random class $A_i = \{c_1, \ldots, c_m\}$.



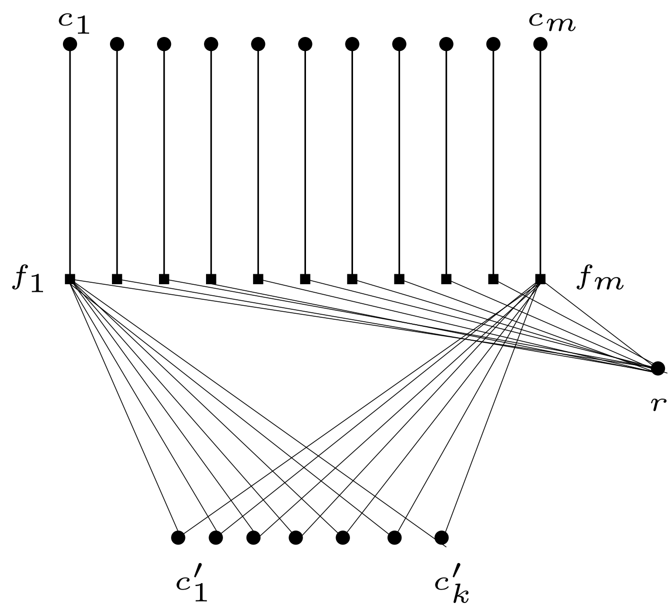
# Lower Bound for Steiner Trees



- $k$ pairwise disjoint classes $A_i$ of $m$ vertices.
- Select a random class $A_i = \{c_1, \ldots, c_m\}$.
- For each class $j \neq i$ select a random vertex $a_j$.



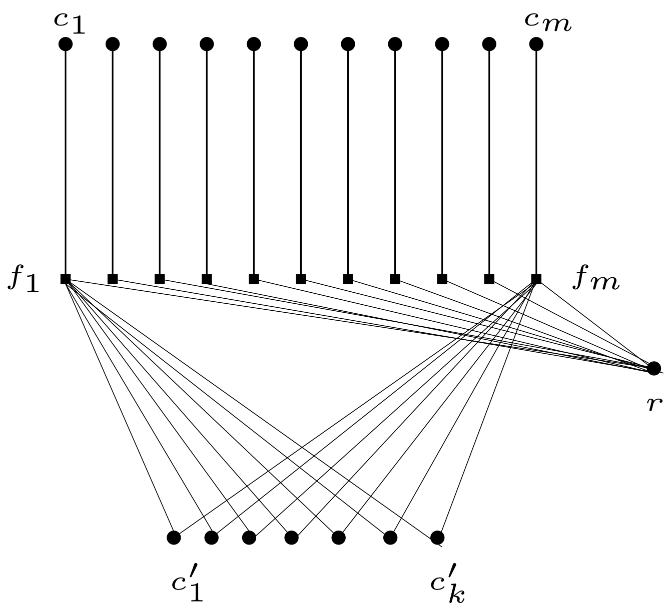
- $\mathcal{B} := \{\{a_1, \ldots, a_k\} : a_i \in A_i,\ i = 1, \ldots, k\}$.



# Lower Bound for Steiner Trees



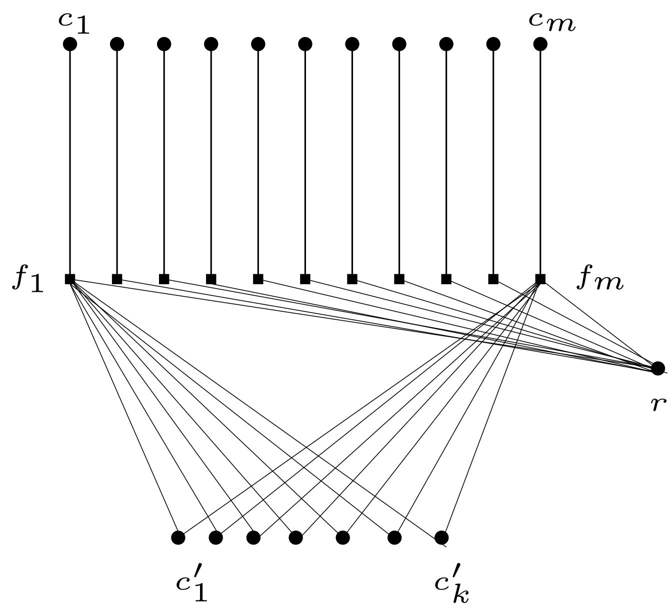
- $\mathcal{B} := \{\{a_1, \ldots, a_k\} : a_i \in A_i,\ i = 1, \ldots, k\}$.
- For each $B \in \mathcal{B}$: vertex $f_B$ with distance 1 to all vertices in $B$.



# Lower Bound for Steiner Trees

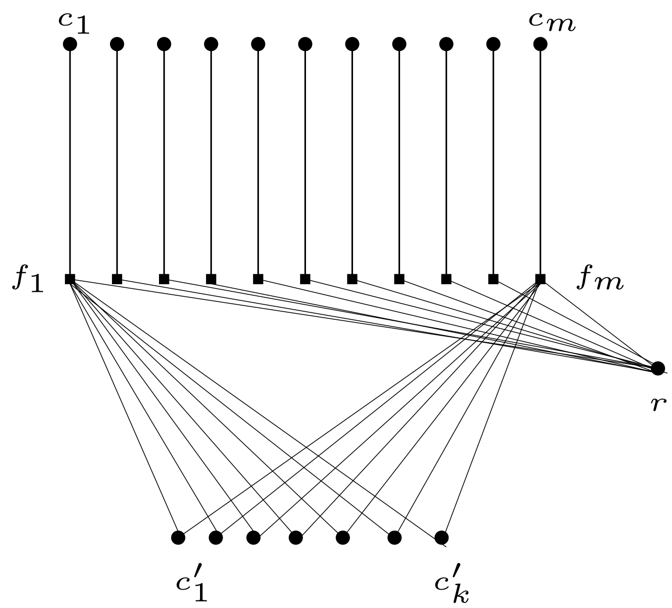


- $\mathcal{B} := \{\{a_1, \ldots, a_k\} : a_i \in A_i,\ i = 1, \ldots, k\}$.
- For each $B \in \mathcal{B}$: vertex $f_B$ with distance 1 to all vertices in $B$.
- $f_B$ is connected to the root $r$, with edges of length 3.



# Lower Bound for Steiner Trees

- $\mathcal{B} := \{\{a_1, \ldots, a_k\} : a_i \in A_i,\ i = 1, \ldots, k\}$.
- For each $B \in \mathcal{B}$: vertex $f_B$ with distance 1 to all vertices in $B$.
- $f_B$ is connected to the root $r$, with edges of length 3.
- $f_B$ has distance 3 to vertices not in $B$.



# Lower Bound for Steiner Trees


- Talk Outline

Cost-Sharing Mechanisms

Facility location

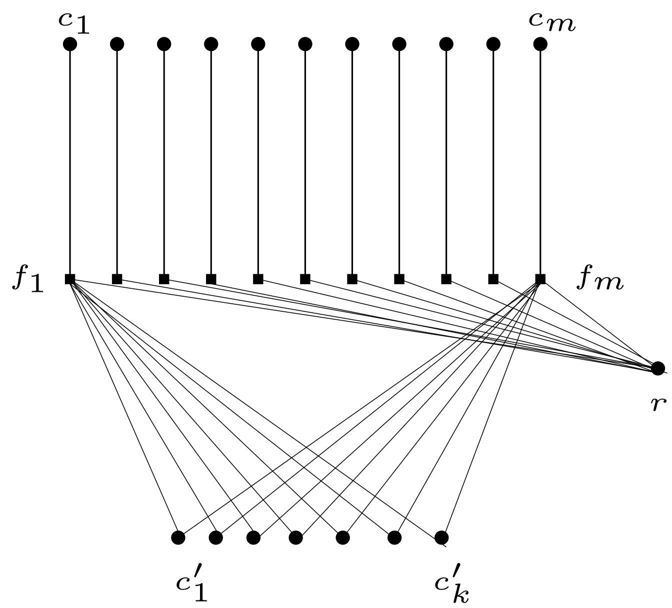
Steiner Forests

Steiner Forest CS-Mechanism

Lifted-Cut Dual Relaxation

Lower Bounds
- Lower Bound for Cross-Monotonicity
- Lower Bound for Steiner Trees
- Limitations of Moulin mechanisms
- Objectives
- Known Results - Social Cost
- Summary
- Open Issues


- $\mathcal{B} := \{\{a_1, \ldots, a_k\} : a_i \in A_i,\ i = 1, \ldots, k\}$.
- For each $B \in \mathcal{B}$: vertex $f_B$ with distance 1 to all vertices in $B$.
- $f_B$ is connected to the root $r$, with edges of length 3.
- $f_B$ has distance 3 to vertices not in $B$.
- For each $c_l$, $l = 1, \ldots, m$, $c(\{a_1, \ldots, a_{i-1}, c_l, a_{i+1}, a_k\}) = k + 3$ implies $\xi(c_l) = \frac{k+3}{k}$

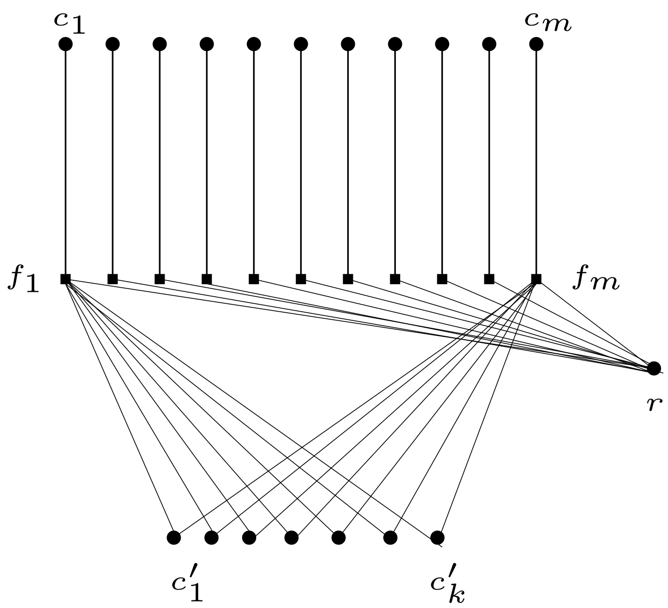


# Lower Bound for Steiner Trees



- $\mathcal{B} := \{\{a_1, \ldots, a_k\} : a_i \in A_i, \ i = 1, \ldots, k\}$.
- For each $B \in \mathcal{B}$: vertex $f_B$ with distance 1 to all vertices in $B$.
- $f_B$ is connected to the root $r$, with edges of length 3.
- $f_B$ has distance 3 to vertices not in $B$.
- For each $c_l$, $l = 1, \ldots, m$, $c(\{a_1, \ldots, a_{i-1}, c_l, a_{i+1}, a_k\}) = k + 3$ implies $\xi(c_l) = \frac{k+3}{k}$
- Total cost share:

$$\sum_{c \in A_i} \xi(c) + \sum_{j \neq i} \xi(a_j) \leq m \times \frac{k+3}{k} + k + 2$$

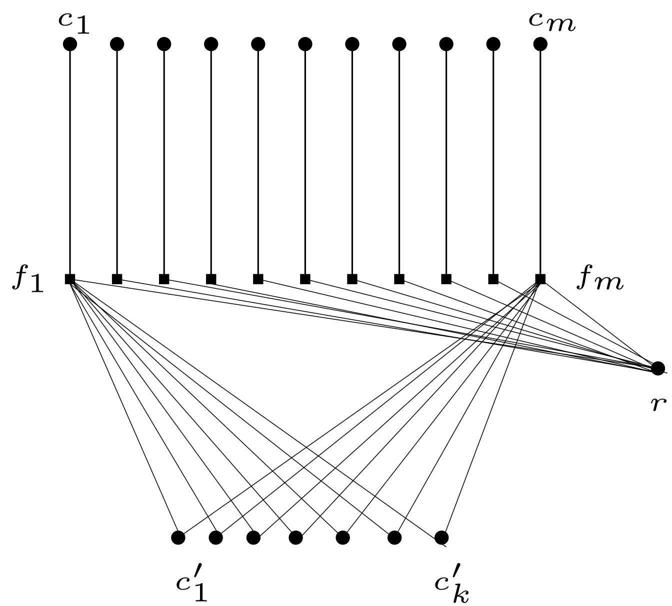


# Lower Bound for Steiner Trees



- $\mathcal{B} := \{\{a_1, \ldots, a_k\} : a_i \in A_i, \ i = 1, \ldots, k\}$.
- For each $B \in \mathcal{B}$: vertex $f_B$ with distance 1 to all vertices in $B$.
- $f_B$ is connected to the root $r$, with edges of length 3.
- $f_B$ has distance 3 to vertices not in $B$.
- For each $c_l$, $l = 1, \ldots, m$, $c(\{a_1, \ldots, a_{i-1}, c_l, a_{i+1}, a_k\}) = k + 3$ implies $\xi(c_l) = \frac{k+3}{k}$
- Total cost share:

$$\sum_{c \in A_i} \xi(c) + \sum_{j \neq i} \xi(a_j) \leq m \times \frac{k+3}{k} + k + 2$$




- $\texttt{opt} \geq 2m + k + 3$

# Cost-Sharing Mechanisms



## Objectives:

- **Strategyproofness:** Dominant strategy for each user is to bid true utility.
- **Group-Strategyproofness:** Same holds even if users collaborate. No side payments between users.
- **Cost Recovery or Budget Balance:** $\sum_{j \in Q} p_j \geq c(Q)$.
- **Competitiveness:** $\sum_{j \in Q} p_j \leq \texttt{opt}_Q$.
- $\alpha$**-Efficiency approximate maximum social welfare:**

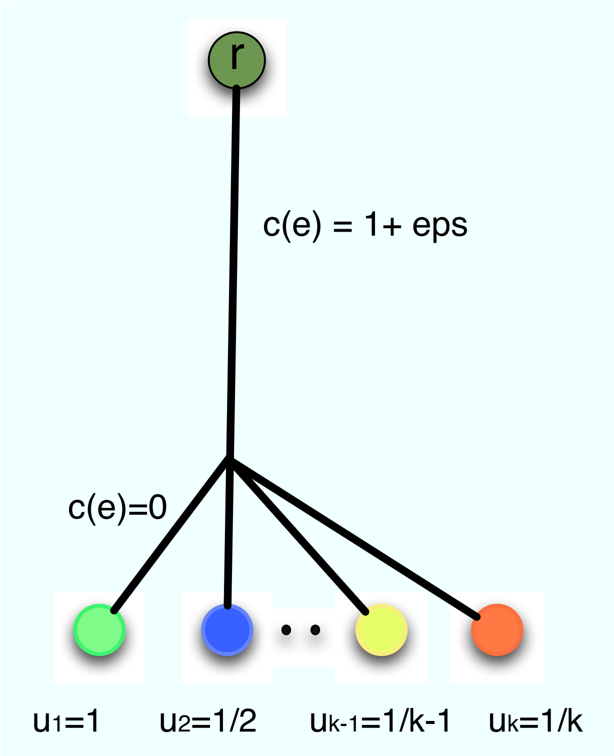
$$u(Q) - c(Q) \geq \frac{1}{\alpha} \cdot \max_{S \subseteq U} [u(S) - C(S)], \quad \alpha \geq 1$$

No mechanism can achieve (approximate) budget balance, truthfullness and efficiency [Feigenbaum et al. '01]

# Limitations of Moulin mechanisms



- Moulin mechanism ends with dropping all players
- (1+$\epsilon$)-budget balance solution achieves $H(k)$ social welfare.



# Objectives



1. $\beta$-budget balance: approximate total cost

$$\frac{1}{\beta}c(Q) \leq p(Q) \leq \texttt{opt}_Q, \quad \beta \geq 1$$

2. Group-strategyproofness: bidding truthfully $b_i = u_i$ is a dominant strategy for every user $i \in U$, even if users cooperate
3. $\alpha$-approximate: approximate minimum social cost

$$\Pi(Q) \leq \alpha \cdot \min_{S \subseteq U} \Pi(S), \quad \alpha \geq 1$$

where $\Pi(S) := u(U \setminus S) + C(S)$

[Roughgarden and Sundararajan '06]

# Known Results - Social Cost


- Talk Outline

Cost-Sharing Mechanisms

Facility location

Steiner Forests

Steiner Forest CS-Mechanism

Lifted-Cut Dual Relaxation

Lower Bounds
- Lower Bound for Cross-Monotonicity
- Lower Bound for Steiner Trees
- Limitations of Moulin mechanisms
- Objectives
- Known Results - Social Cost
- Summary
- Open Issues


| **Authors** | **Problem** | $\beta$ | $\alpha$ |
|---|---|---|---|
| [Roughgarden, Sundararajan '06] | submodular cost | $1$ | $\Theta(\log n)$ |
| | Steiner tree | $2$ | $\Theta(\log^2 n)$ |
| [Chawla, Roughgarden, Sundararajan '06] | Steiner forest | $2$ | $\Theta(\log^2 n)$ |
| [Roughgarden, Sundararajan ] | facility location | $3$ | $\Theta(\log n)$ |
| | SRoB | $4$ | $\Theta(\log^2 n)$ |
| [Gupta, Könemann, Leonardi, Ravi, Schäfer '07] | prize-collecting Steiner forest | $3$ | $\Theta(\log^2 n)$ |
| [Goyal, Gupta, Leonardi, Ravi '07] | 2-stage Stochastic Steiner Tree | $O(1)$ | $\Theta(\log^2 n)$ |

# Summary

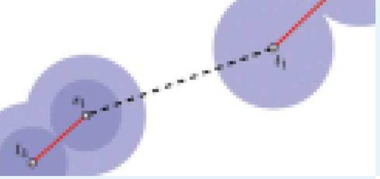


- Introduced cost-sharing mechanisms for network design problems
- Presented a group-strategyproof mechanism for Steiner forests that is $2$-budget balance.
- Presented a new undirected cut relaxation for Steiner forests, strictly stronger than the classical undirected cut relaxation.
- Presented a lower bound of $2$ on the budget balance approximation of cross-monotonic algorithms for Steiner trees.

# Open Issues



- Can we use our infeasible dual to give better primal-dual approximation algorithms for Steiner forests/trees?



# Open Issues



- Can we use our infeasible dual to give better primal-dual approximation algorithms for Steiner forests/trees?
- Give cross-monotonic cost-sharing methods for more network design problems.

# Open Issues



- Can we use our infeasible dual to give better primal-dual approximation algorithms for Steiner forests/trees?
- Give cross-monotonic cost-sharing methods for more network design problems.
- Characterize classes of problems yielding mechanisms with good cost recovery.

# Open Issues



- Can we use our infeasible dual to give better primal-dual approximation algorithms for Steiner forests/trees?
- Give cross-monotonic cost-sharing methods for more network design problems.
- Characterize classes of problems yielding mechanisms with good cost recovery.
- A more satisfactory definition of group-strategyproofness.