Motivation: Auction

• suppose we want to auction off a single item to one of $n$ potential buyers in $U$
• every bidder $i \in U$ has a valuation $v_i$ for receiving the item
• valuation is only known to $i$ and not to the auctioneer
• every bidder $i$ announces a bid $b_i$

Mechanism: protocol that based on the bids determines a winner of the auction and a selling price $p$
Motivation: Auction

**Selfishness:** every player wants to maximize his net gain 
\((v_i - p)q_i\), where \(q_i = 1\) if \(i\) is the winner and \(q_i = 0\) otherwise.

**Goal:** economic efficiency, i.e., sell the item to the buyer with maximum valuation.

**Question:** Can efficiency be achieved although valuations are private?
Vickrey’s Truthful Mechanism

First-Price Auction: sell the item to the buyer with the highest bid and charge his bid

- buyers have an incentive to underbid

Second-Price Auction (Vickrey Auction ’61): sell the item to the buyer with the highest bid and charge the second-highest bid

- buyers bid their valuations truthfully, i.e., $b_i = v_i$
- economic efficiency is achieved
Vickrey’s Truthful Mechanism

**First-Price Auction:** sell the item to the buyer with the highest bid and charge his bid
- buyers have an incentive to underbid

**Second-Price Auction (Vickrey Auction ’61):** sell the item to the buyer with the highest bid and charge the second-highest bid
- buyers bid their valuations truthfully, i.e., $b_i = v_i$
- economic efficiency is achieved
Vickrey’s Truthful Mechanism

**First-Price Auction**: sell the item to the buyer with the highest bid and charge his bid
- buyers have an incentive to **underbid**

**Second-Price Auction (Vickrey Auction ’61)**: sell the item to the buyer with the highest bid and charge the second-highest bid
- buyers bid their valuations **truthfully**, i.e., $b_i = v_i$
- **economic efficiency** is achieved
Vickrey’s Truthful Mechanism

**First-Price Auction:** sell the item to the buyer with the highest bid and charge his bid
- buyers have an incentive to underbid

**Second-Price Auction (Vickrey Auction ’61):** sell the item to the buyer with the highest bid and charge the second-highest bid
- buyers bid their valuations truthfully, i.e., \( b_i = v_i \)
- economic efficiency is achieved
Group-Strategyproof Cost Sharing Mechanisms
Cooperative Cost Sharing

Setting:

• set of players are interested in receiving some service
• provision of service incurs a (player-set dependent) cost that needs to be shared among the players
• players act strategically: aim at receiving service at low individual price
• players can coordinate their strategies

Applications: sharing the cost of public investments, access to network, etc.

Goal: design selection and payment scheme such that
• it is in every player’s self-interest to behave truthfully
• payments of selected players cover the service cost
• player selection is “socially efficient”
Motivating Example

Given:
- network $N = (V, E, c)$
- set of players $U = [n]$
- player $i \in U$ requests connection between $s_i, t_i$

Cost Function:
$C(S) = \min \text{ cost to satisfy all requests of players in } S \subseteq U$

Example: $C\left(\{1, 3, 4\}\right) = 5$
Motivating Example

Given:
- network \( N = (V, E, c) \)
- set of players \( U = [n] \)
- player \( i \in U \) requests connection between \( s_i, t_i \)

Cost Function:
\[
C(S) = \min \text{ cost to satisfy all requests of players in } S \subseteq U
\]

Example: \( C(\{1, 3, 4\}) = 5 \)
Motivating Example

Given:
- network $N = (V, E, c)$
- set of players $U = [n]$
- player $i \in U$ requests connection between $s_i, t_i$

Cost Function:
$C(S) = \text{min. cost to satisfy all requests of players in } S \subseteq U$

Example: $C(\{1, 3, 4\}) = 5$
Motivating Example

Given:
- network $N = (V, E, c)$
- set of players $U = [n]$
- player $i \in U$ requests connection between $s_i, t_i$

Cost Function:
$C(S) = \text{min. cost to satisfy all requests of players in } S \subseteq U$

Example: $C(\{1, 3, 4\}) = 5$
Motivating Example

Given:
- network $N = (V, E, c)$
- set of players $U = [n]$
- player $i \in U$ requests connection between $s_i, t_i$

Cost Function:
$C(S) = \min$ cost to satisfy all requests of players in $S \subseteq U$

Example: $C(\{1, 3, 4\}) = 5$
Motivating Example

Given:
- network $N = (V, E, c)$
- set of players $U = [n]$
- player $i \in U$ requests connection between $s_i, t_i$

Cost Function:
$C(S) = \text{min. cost to satisfy all requests of players in } S \subseteq U$

Example: $C(\{1, 2, 3, 4\}) = 6$
Motivating Example

Player $i \in U$:
- valuation $v_i$ (private!)
- bid $b_i$ (public)
- goal: maximize $v_i - p_i$

Cost Sharing Mechanism:
- selects a set $Q$ of players whose requests are satisfied
- determines a payment $p_i$ for every $i \in Q$ to distribute the cost $C(Q)$
Motivating Example

Player $i \in U$:
- valuation $v_i$ (private!)
- bid $b_i$ (public)
- goal: maximize $v_i - p_i$

Cost Sharing Mechanism:
- selects a set $Q$ of players whose requests are satisfied
- determines a payment $p_i$ for every $i \in Q$ to distribute the cost $C(Q)$
Motivating Example

Player $i \in U$:
- valuation $v_i$ (private!)
- bid $b_i$ (public)
- goal: maximize $v_i - p_i$

Cost Sharing Mechanism:
- selects a set $Q$ of players whose requests are satisfied
- determines a payment $p_i$ for every $i \in Q$ to distribute the cost $C(Q)$
Motivating Example

Player $i \in U$:
- valuation $v_i$ (private!)
- bid $b_i$ (public)
- goal: maximize $v_i - p_i$

Cost Sharing Mechanism:
- selects a set $Q$ of players whose requests are satisfied
- determines a payment $p_i$ for every $i \in Q$ to distribute the cost $C(Q)$
Motivating Example

Player $i \in U$:
- valuation $v_i$ (private!)
- bid $b_i$ (public)
- goal: maximize $v_i - p_i$

Cost Sharing Mechanism:
- selects a set $Q$ of players whose requests are satisfied
- determines a payment $p_i$ for every $i \in Q$ to distribute the cost $C(Q)$
**Motivating Example**

**Objectives:**

1. **Truthfulness:** bidding truthfully is a dominant strategy for every player.

2. **Budget Balance:** payments recover solution cost.

3. **Efficiency:** selected player set realizes “social efficiency” objective.
Cost Sharing Model

**Given:**

- set $U$ of players (interested in some service)
- every player $i \in U$:
  - valuation $v_i$: value (private!) of the service
  - bid $b_i$: maximum amount he is willing to pay
- player-set dependent cost function $C : 2^U \to \mathbb{R}^+$
  - defined implicitly: cost function of combinatorial optimization problem $\mathcal{P}$ (e.g., Steiner forest, scheduling, etc.)
  - $C(S) = \text{optimal solution cost for player set } S \subseteq U$
Cost Sharing Mechanism

Cost Sharing Mechanism $M$: collects bids $(b_i)_{i \in U}$ from players and computes

- set $Q \subseteq U$ of players that receive service (selection scheme)

  Notation: $q_i = 1$ if $i \in Q$ and $q_i = 0$ otherwise

- payment $p_i$ for every player $i \in U$ to distribute the cost $C(Q)$ (payment scheme)

1. **No Positive Transfer**: $p_i \geq 0$ for all $i \in Q$

2. **Voluntary Participation**: $p_i = 0$ for all $i \notin Q$ and $p_i \leq b_i$ for all $i \in Q$

3. **Consumer Sovereignty**: for every $i \in U$ there exists a bid $b_i^*$ for which $i$ is guaranteed to receive service
**Truthfulness**

**Strategic Behavior:** every player $i \in U$ acts selfishly and attempts to maximize his quasi-linear utility function:

$$u_i(q, p) := q_i(v_i - p_i)$$

$\Rightarrow$ player $i$ will misreport his valuation ($b_i \neq v_i$) if this leads to larger utility

**Strategyproofness:** utility of every player $i \in U$ is maximized if he bids truthfully $b_i = v_i$ (independently of other players’ bids)

**Group-Strategyproofness:** same holds true even if players form coalitions to coordinate their bids
**Truthfulness**

**Strategic Behavior:** every player \( i \in U \) acts selfishly and attempts to maximize his quasi-linear utility function:

\[
  u_i(q, p) := q_i(v_i - p_i)
\]

\( \Rightarrow \) player \( i \) will misreport his valuation \( (b_i \neq v_i) \) if this leads to larger utility

**Strategyproofness:** utility of every player \( i \in U \) is maximized if he bids truthfully \( b_i = v_i \) (independently of other players’ bids)

**Group-Strategyproofness:** same holds true even if players form coalitions to coordinate their bids
**Truthfulness**

**Strategic Behavior:** every player $i \in U$ acts selfishly and attempts to maximize his quasi-linear utility function:

$$u_i(q, p) := q_i(v_i - p_i)$$

⇒ player $i$ will misreport his valuation ($b_i \neq v_i$) if this leads to larger utility

**Strategyproofness:** utility of every player $i \in U$ is maximized if he bids truthfully $b_i = v_i$ (independently of other players’ bids)

**Group-Strategyproofness:** same holds true even if players form coalitions to coordinate their bids
**Definition**

A cost sharing mechanism $M$ is **group-strategyproof** iff for all $S \subseteq U$

$$u_i(\tilde{q}, \tilde{p}) \geq u_i(q, p) \quad \forall i \in S \quad \Rightarrow \quad u_i(\tilde{q}, \tilde{p}) = u_i(q, p) \quad \forall i \in S$$

$(q, p)$: outcome if $b_i = v_i$ for every $i \in S$

$(\tilde{q}, \tilde{p})$: outcome if $b_i = \cdot$ for every $i \in S$
Illustration: Group-Strategyproofness

**Definition**

A cost sharing mechanism $M$ is **group-strategyproof** iff for all $S \subseteq U$

\[ \exists i \in S : \quad u_i(\tilde{q}, \tilde{p}) > u_i(q, p) \quad \Rightarrow \quad \exists j \in S : \quad u_j(\tilde{q}, \tilde{p}) < u_j(q, p) \]

$(q, p)$: outcome if $b_i = v_i$ for every $i \in S$
$(\tilde{q}, \tilde{p})$: outcome if $b_i = \cdot$ for every $i \in S$
Definition

A cost sharing mechanism $M$ is **group-strategyproof** iff for all $S \subseteq U$

$$\exists i \in S : \ u_i(\tilde{q}, \tilde{p}) > u_i(q, p) \ \Rightarrow \ \exists j \in S : \ u_j(\tilde{q}, \tilde{p}) < u_j(q, p)$$

$(q, p)$: outcome if $b_i = v_i$ for every $i \in S$

$(\tilde{q}, \tilde{p})$: outcome if $b_i = \cdot$ for every $i \in S$
**Definition**

A cost sharing mechanism $M$ is **group-strategyproof** iff for all $S \subseteq U$

$$\exists i \in S : \ u_i(\tilde{q}, \tilde{p}) > u_i(q, p) \ \Rightarrow \ \exists j \in S : \ u_j(\tilde{q}, \tilde{p}) < u_j(q, p)$$

$(q, p)$: outcome if $b_i = v_i$ for every $i \in S$

$(\tilde{q}, \tilde{p})$: outcome if $b_i = \cdot$ for every $i \in S$
“Classical” Objectives

1 **Budget Balance**: payments equal servicing cost

\[ \sum_{i \in Q} p_i = C(Q) \]

2 **Group-Strategyproofness**

3 **Efficiency**: assuming truthful bidding, selected player set maximizes social welfare

\[ \sum_{i \in Q} v_i - C(Q) = \max_{S \subseteq U} \sum_{i \in S} v_i - C(S) \]
Computational Issues

Want to design mechanisms that are computationally efficient

Problems:

1. underlying optimization problem \( \mathcal{P} \) is often computationally hard

2. truthfulness, budget balance and efficiency cannot be achieved simultaneously

Solutions:

1. use approximation algorithm to compute an approximate solution of cost \( \bar{C}(Q) \leq \beta \cdot C(Q) \) where \( \beta \geq 1 \)

2. consider different (but equivalent) social efficiency objective

[Green et al. ’76] [Roberts ’79] [Feigenbaum et al., TCS ’03]
Approximate Budget Balance: cost sharing mechanism \( M \) is \( \beta \)-budget balanced if

\[
\bar{C}(Q) \leq \sum_{i \in Q} p_i \leq \beta \cdot C(Q) \quad (\beta \geq 1)
\]

Define the social cost of a set \( S \subseteq U \) as

\[
\Pi(S) := \sum_{i \notin S} v_i + C(S) = \sum_{i \in U} v_i - \left( \sum_{i \in S} v_i - C(S) \right)
\]

Approximate Efficiency: cost sharing mechanism \( M \) is \( \alpha \)-approximate if, assuming truthful bidding,

\[
\sum_{i \notin Q} v_i + \bar{C}(Q) \leq \alpha \cdot \min_{S \subseteq U} \Pi(S) \quad (\alpha \geq 1)
\]

[Roughgarden and Sundararajan, JACM '09]
Approximate Budget Balance: cost sharing mechanism $M$ is $\beta$-budget balanced if

$$\bar{C}(Q) \leq \sum_{i \in Q} p_i \leq \beta \cdot C(Q) \quad (\beta \geq 1)$$

Define the social cost of a set $S \subseteq U$ as

$$\Pi(S) := \sum_{i \notin S} v_i + C(S) = \sum_{i \in U} v_i - \left(\sum_{i \in S} v_i - C(S)\right)$$

Approximate Efficiency: cost sharing mechanism $M$ is $\alpha$-approximate if, assuming truthful bidding,

$$\sum_{i \notin Q} v_i + \bar{C}(Q) \leq \alpha \cdot \min_{S \subseteq U} \Pi(S) \quad (\alpha \geq 1)$$

[Roughgarden and Sundararajan, JACM '09]
Approximate Budget Balance: cost sharing mechanism $M$ is $\beta$-budget balanced if

$$\overline{C}(Q) \leq \sum_{i \in Q} p_i \leq \beta \cdot C(Q) \quad (\beta \geq 1)$$

Define the social cost of a set $S \subseteq U$ as

$$\Pi(S) := \sum_{i \notin S} v_i + C(S) = \sum_{i \in U} v_i - \left( \sum_{i \in S} v_i - C(S) \right)$$

Approximate Efficiency: cost sharing mechanism $M$ is $\alpha$-approximate if, assuming truthful bidding,

$$\sum_{i \notin Q} v_i + \bar{C}(Q) \leq \alpha \cdot \min_{S \subseteq U} \Pi(S) \quad (\alpha \geq 1)$$

[Roughgarden and Sundararajan, JACM '09]
Objectives at a Glance

1. Computational Efficiency

2. Approximate Budget Balance:

\[ \bar{C}(Q) \leq \sum_{i \in Q} p_i \leq \beta \cdot C(Q) \quad (\beta \geq 1) \]

3. Group-Strategyproofness

4. Approximate Efficiency:

\[ \sum_{i \notin Q} v_i + \bar{C}(Q) \leq \alpha \cdot \min_{S \subseteq U} \left\{ \sum_{i \notin S} v_i + C(S) \right\} \quad (\alpha \geq 1) \]
How to achieve \( \beta \)-budget balance?

\[
\left( \bar{C}(Q) \leq \sum_{i \in Q} p_i \leq \beta \cdot C(Q) \right)
\]
How to achieve group-strategyproofness?

(Not everybody in the coalition is better off by misreporting his valuation.)
Moulin’s Framework

Cost Sharing Function: \( \xi : U \times 2^U \rightarrow \mathbb{R}^+ \)
\( \xi_i(S) = \text{cost share of player } i \text{ with respect to set } S \subseteq U \)

\[ \bar{C}(S) \leq \sum_{i \in S} \xi_i(S) \leq \beta \cdot C(S) \quad \forall S \subseteq U \]

\( \beta \)-Budget Balance:

Cross-Monotonicity: cost share of player \( i \) does not decrease if other players leave the game:

\( \forall S \subseteq T, \forall i \in S : \xi_i(S) \geq \xi_i(T) \)
Moulin’s Framework

**Cost Sharing Function:** \( \xi : U \times 2^U \rightarrow \mathbb{R}^+ \)

\( \xi_i(S) = \text{cost share of player } i \text{ with respect to set } S \subseteq U \)

**\( \beta \)-Budget Balance:**

\[
\bar{C}(S) \leq \sum_{i \in S} \xi_i(S) \leq \beta \cdot C(S) \quad \forall S \subseteq U
\]

**Cross-Monotonicity:** cost share of player \( i \) does not decrease if other players leave the game:

\[
\forall S \subseteq T, \forall i \in S : \quad \xi_i(S) \geq \xi_i(T)
\]
Moulin’s Framework

Moulin Mechanism $M(\xi)$:

1: Initialize: $Q \leftarrow U$
2: If for each player $i \in Q$: $\xi_i(Q) \leq b_i$ then STOP
3: Otherwise, remove from $Q$ all players with $\xi_i(Q) > b_i$ and repeat

Theorem

If $\xi$ is cross-monotonic and $\beta$-budget balanced, then the Moulin mechanism $M(\xi)$ is group-strategyproof and $\beta$-budget balanced.

[Moulin, SCW ’99]
How to achieve $\alpha$-approximability?

$$\left(\sum_{i \notin Q} v_i + \bar{C}(Q) \leq \alpha \cdot \min_{S \subseteq U} \sum_{i \notin S} v_i + C(S)\right)$$
Suppose we are given an arbitrary order $\sigma$ on the players in $U$. Order each subset $S \subseteq U$ according to $\sigma$:

$$S := \{i_1, \ldots, i_{|S|}\} \text{ with } i_j \prec_{\sigma} i_k \text{ for all } 1 \leq j < k \leq |S|.$$  

Let $S_j$ refer to the first $j$ players of $S$.

A cost sharing function $\xi$ is $\alpha$-summable if for every order $\sigma$ of the players in $U$

$$\forall S \subseteq U : \sum_{j=1}^{|S|} \xi_{ij}(S_j) \leq \alpha \cdot C(S)$$
Suppose we are given an arbitrary order $\sigma$ on the players in $U$. Order each subset $S \subseteq U$ according to $\sigma$:

$$S := \{i_1, \ldots, i_{|S|}\} \text{ with } i_j \prec_\sigma i_k \text{ for all } 1 \leq j < k \leq |S|.$$ 

Let $S_j$ refer to the first $j$ players of $S$.

A cost sharing function $\xi$ is $\alpha$-summable if for every order $\sigma$ of the players in $U$

$$\forall S \subseteq U : \sum_{j=1}^{|S|} \xi_{i_j}(S_j) \leq \alpha \cdot C(S)$$
Suppose we are given an arbitrary order $\sigma$ on the players in $U$. Order each subset $S \subseteq U$ according to $\sigma$:

$$S := \{i_1, \ldots, i_{|S|}\} \text{ with } i_j \prec_\sigma i_k \text{ for all } 1 \leq j < k \leq |S|.$$ 

Let $S_j$ refer to the first $j$ players of $S$.

A cost sharing function $\xi$ is $\alpha$-summable if for every order $\sigma$ of the players in $U$

$$\forall S \subseteq U : \sum_{j=1}^{\lfloor S \rfloor} \xi_i(S_j) \leq \alpha \cdot C(S)$$
Suppose we are given an arbitrary order $\sigma$ on the players in $U$. Order each subset $S \subseteq U$ according to $\sigma$:

$$S := \{i_1, \ldots, i_{|S|}\} \text{ with } i_j \prec_\sigma i_k \text{ for all } 1 \leq j < k \leq |S|.$$

Let $S_j$ refer to the first $j$ players of $S$.

A cost sharing function $\xi$ is $\alpha$-summable if for every order $\sigma$ of the players in $U$

$$\forall S \subseteq U : \sum_{j=1}^{|S|} \xi_i(S_j) \leq \alpha \cdot C(S).$$
Theorem

Let $\xi$ be a cross-monotonic cost sharing function and let $\alpha$, $\beta$ be the smallest numbers such that $\xi$ is $\alpha$-summable and $\beta$-budget balanced. Then the Moulin mechanism $M(\xi)$ is $(\alpha + \beta)$-approximate and no better than $\max\{\alpha, \beta\}$-approximate.

[Roughgarden, Sundararajan, JACM '09]
## Moulin Mechanisms: Known Results I

### Upper bounds

<table>
<thead>
<tr>
<th>Reference</th>
<th>Problem</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Moulin, Shenker, ET ’01]</td>
<td>submodular cost</td>
<td>1</td>
</tr>
<tr>
<td>[Jain, Vazirani, STOC ’01]</td>
<td>minimum spanning tree</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Steiner tree and TSP</td>
<td>2</td>
</tr>
<tr>
<td>[Pál, Tardos, FOCS ’03]</td>
<td>facility location</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>single-sink rent-or-buy</td>
<td>15</td>
</tr>
<tr>
<td>[Leonardi, Schäfer, EC ’03], [Gupta et al., APPROX ’04]</td>
<td>single-sink rent-or-buy</td>
<td>4</td>
</tr>
<tr>
<td>[Leonardi, Schäfer, EC ’03]</td>
<td>connected facility location</td>
<td>30</td>
</tr>
<tr>
<td>[Könemann, Leonardi, Schäfer, SODA ’05]</td>
<td>Steiner forest</td>
<td>2</td>
</tr>
<tr>
<td>[Gupta et al., SODA ’07]</td>
<td>price-collecting Steiner forest</td>
<td>3</td>
</tr>
<tr>
<td>[Bleischwitz, Monien, CIAC ’07]</td>
<td>makespan scheduling</td>
<td>2</td>
</tr>
</tbody>
</table>

### Lower bounds

<table>
<thead>
<tr>
<th>Reference</th>
<th>Problem</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Immorlica et al., SODA ’05]</td>
<td>set cover, vertex cover</td>
<td>( n^c )</td>
</tr>
<tr>
<td></td>
<td>facility location</td>
<td>3</td>
</tr>
<tr>
<td>[Könemann et al., SODA ’05]</td>
<td>Steiner tree</td>
<td>2</td>
</tr>
<tr>
<td>[Bleischwitz, Monien, CIAC ’07]</td>
<td>makespan scheduling</td>
<td>2</td>
</tr>
<tr>
<td>[Brenner, Schäfer, STACS ’07]</td>
<td>completion time scheduling, etc.</td>
<td>( n/c )</td>
</tr>
</tbody>
</table>
## Moulin Mechanisms: Known Results I

### Upper bounds

<table>
<thead>
<tr>
<th>Reference</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Moulin, Shenker, ET ’01]</td>
<td>submodular cost 1</td>
</tr>
<tr>
<td>[Jain, Vazirani, STOC ’01]</td>
<td>minimum spanning tree 1</td>
</tr>
<tr>
<td>[Pál, Tardos, FOCS ’03]</td>
<td>Steiner tree and TSP 2</td>
</tr>
<tr>
<td>[Leonardi, Schäfer, EC ’03], [Gupta et al., APPROX ’04]</td>
<td>facility location 3</td>
</tr>
<tr>
<td>[Leonardi, Schäfer, EC ’03]</td>
<td>single-sink rent-or-buy 15</td>
</tr>
<tr>
<td>[Könemann, Leonardi, Schäfer, SODA ’05]</td>
<td>connected facility location 30</td>
</tr>
<tr>
<td>[Gupta et al., SODA ’07]</td>
<td>Steiner forest 2</td>
</tr>
<tr>
<td>[Bleischwitz, Monien, CIAC ’07]</td>
<td>price-collecting Steiner forest 3</td>
</tr>
</tbody>
</table>

### Lower bounds

<table>
<thead>
<tr>
<th>Reference</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Immorlica et al., SODA ’05]</td>
<td>set cover, vertex cover $n^c$</td>
</tr>
<tr>
<td>[Könemann et al., SODA ’05]</td>
<td>facility location 3</td>
</tr>
<tr>
<td>[Bleischwitz, Monien, CIAC ’07]</td>
<td>Steiner tree 2</td>
</tr>
<tr>
<td>[Brenner, Schäfer, STACS ’07]</td>
<td>makespan scheduling 2</td>
</tr>
<tr>
<td></td>
<td>completion time scheduling, etc. $n/c$</td>
</tr>
</tbody>
</table>
| Reference | Problem | $\beta$ | $\alpha$  \\
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[Roughgarden, Sundararajan, STOC ’06]</td>
<td>submodular cost</td>
<td>1</td>
<td>$\Theta(\log n)$</td>
</tr>
<tr>
<td></td>
<td>Steiner tree</td>
<td>2</td>
<td>$\Theta(\log^2 n)$</td>
</tr>
<tr>
<td>[Chawla et al., WINE ’06]</td>
<td>Steiner forest</td>
<td>2</td>
<td>$\Theta(\log^2 n)$</td>
</tr>
<tr>
<td>[Roughgarden, Sundararajan, IPCO ’07]</td>
<td>facility location</td>
<td>3</td>
<td>$\Theta(\log n)$</td>
</tr>
<tr>
<td></td>
<td>SROB</td>
<td>4</td>
<td>$\Theta(\log^2 n)$</td>
</tr>
<tr>
<td>[Gupta et al., SODA ’07]</td>
<td>price-collecting SF</td>
<td>3</td>
<td>$\Theta(\log^2 n)$</td>
</tr>
<tr>
<td>[Brenner, Schäfer, STACS ’07]</td>
<td>makespan scheduling</td>
<td>2</td>
<td>$\Theta(\log n)$</td>
</tr>
<tr>
<td></td>
<td>cost-stable problems</td>
<td></td>
<td>$\Omega(\log n)$</td>
</tr>
</tbody>
</table>
Cross-Monotonic Cost Shares for Steiner Forest
Steiner Forest Game

**Goal:** design a cost sharing mechanism for the Steiner forest game

- graph $G = (V, E)$ with edge costs $c : E \rightarrow \mathbb{R}^+$
- player $i$ requests connection between terminals $s_i, t_i \in V$
- identify players with terminal pairs: $U = \{(s_1, t_1), \ldots, (s_n, t_n)\}$
- $C(S) =$ cost of a minimum cost Steiner forest connecting all terminal pairs in $S \subseteq U$

**Theorem**

*There is a cross-monotonic and 2-budget balanced cost sharing function for the Steiner forest game.*

[Könemann, Leonardi, Schäfer, van Zwam, SICOMP '08]
Fix a set $Q \subseteq U$ of terminal pairs. We sketch the primal-dual algorithm AKR($Q$) of [Agrawal, Klein, Ravi, SICOMP ’95] for the Steiner forest problem with terminal pair set $Q$.

A subset $S \subseteq V$ of nodes is a Steiner cut if it separates at least one terminal pair $(s, t) \in Q$. Let $S$ be the set of all such cuts.

**Observation:** for every Steiner cut $S \in S$, any feasible Steiner forest must contain at least one of the red edges

$$\delta(S) = \{uv \in E : u \in S, v \notin S\}$$
Undirected Cut Formulation

**Integer Program:**

\[
\begin{align*}
\text{min} & \quad \sum_{e \in E} c_e \cdot x_e \\
\text{s.t.} & \quad \sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \in S \\
& \quad x_e \in \{0, 1\} \quad \forall e \in E
\end{align*}
\]
Undirected Cut Formulation

**Primal LP:**

\[
\begin{align*}
\text{min} & \quad \sum_{e \in E} c_e \cdot x_e \\
\text{s.t.} & \quad \sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \in \mathcal{S} \\
& \quad x_e \geq 0 \quad \forall e \in E
\end{align*}
\]
Undirected Cut Formulation

**Primal LP:**
\[
\begin{align*}
\text{min} & \quad \sum_{e \in E} c_e \cdot x_e \\
\text{s.t.} & \quad \sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \in S \\
& \quad x_e \geq 0 \quad \forall e \in E
\end{align*}
\]

**Dual LP:**
\[
\begin{align*}
\text{max} & \quad \sum_{S \in S} y_S \\
\text{s.t.} & \quad \sum_{S : e \in \delta(S)} y_S \leq c_e \quad \forall e \in E \\
& \quad y_S \geq 0 \quad \forall S \in S
\end{align*}
\]
The dual $y_S$ of Steiner cut $S$ is visualized as moat around $S$ of radius $y_S$.

An edge $e$ is said to be tight if its corresponding dual constraint is tight:

$$\sum_{S: e \in \delta(S)} y_S = c_e$$
The dual $y_S$ of Steiner cut $S$ is visualized as moat around $S$ of radius $y_S$.

An edge $e$ is said to be tight if its corresponding dual constraint is tight:

$$\sum_{S : e \in \delta(S)} y_S = c_e$$
Visualizing the Dual

The dual $y_S$ of Steiner cut $S$ is visualized as moat around $S$ of radius $y_S$

An edge $e$ is said to be tight if its corresponding dual constraint is tight:

$$
\sum_{S \colon e \in \delta(S)} y_S = c_e
$$
Execution of AKR can be seen as a process over time $\tau$:

- $(F^\tau, y^\tau) =$ forest and dual packing
- terminal $v$ is active if it is separated from its mate in $F^\tau$
- $\bar{F}^\tau =$ subgraph induced by tight edges with respect to $y^\tau$
- connected components of $\bar{F}^\tau$ are called moats
- moat is active if it contains an active terminal

Algorithm AKR:

1: $F^0 = \emptyset$, $y^0 = 0$

2: repeat

3: simultaneously increase duals of all active moats until some path $P$ between two active terminals becomes tight

4: add tight segments of $P$ to the current forest $F^\tau$

5: until all terminals are inactive
\[ \tau = 0.0 \]
$\tau = 0.3$
Illustration: AKR

\[ \tau = 1.0 \]
\( \tau = 1.0 \)
\[ \tau = 1.5 \]
Illustration: AKR

\[ \tau = 1.5 \]
Illustration: AKR

\[ \tau = 2.0 \]
\[ \tau = 2.5 \]
\( \tau = 3.5 \)
\[ \tau = 5.0 \]
Illustration: AKR

\[ \tau = 5.0 \]
Approximation Guarantee

Theorem

The algorithm AKR(Q) computes a feasible forest F for terminal pair set Q and a feasible dual \((y_S)_{S \in S}\) such that

\[
c(F) \leq \left(2 - \frac{1}{k}\right) \sum_{S \in S} y_S \leq \left(2 - \frac{1}{k}\right) \text{OPT}(Q),
\]

where \(k\) is the number of terminal pairs in Q.

[Idea: run AKR and distribute (twice) the total dual among the terminals]

[Agrawal, Klein, Ravi, SICOMP '95]
Example:

- all terminals are active
- grow active moats by $\epsilon$
- growth of each moat is shared evenly among active terminals:

$$s_1 : \epsilon/3$$
$$t_2 : \epsilon/2$$
$$t_1 : \epsilon$$
Sharing the Dual Growth

Example:

- all terminals are active
- grow active moats by $\epsilon$
- growth of each moat is shared evenly among active terminals:

  \[ s_1 : \epsilon/3 \]
  \[ t_2 : \epsilon/2 \]
  \[ t_1 : \epsilon \]
Sharing the Dual Growth

Example:

- all terminals are active
- grow active moats by $\epsilon$
- growth of each moat is shared evenly among active terminals:

$$s_1 : \epsilon/3$$
$$t_2 : \epsilon/2$$
$$t_1 : \epsilon$$
$a^\tau_v = \text{number of active terminals in the moat containing } v \text{ at time } \tau$

Suppose terminal $v \in Q$ becomes inactive at time $T$. Define the cost share of $v$ as

$$\xi_v(Q) = \int_0^T \frac{1}{a^\tau_v} d\tau$$

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

$$\xi_{st}(Q) = 2 \cdot (\xi_s(Q) + \xi_t(Q))$$
Problem: Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

Example: $Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}$, $Q_0 = Q \setminus \{(s_3, t_3)\}$

$$\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
\end{array}$$

$\tau = 0.0$
Problem: Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

Example: \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}, \quad Q_0 = Q \setminus \{(s_3, t_3)\} \)

AKR(\(Q\))

<table>
<thead>
<tr>
<th>s_1</th>
<th>s_2</th>
<th>s_3</th>
<th>t_3</th>
<th>t_2</th>
<th>t_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

\(\tau = 0.5\)
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\} \), \( Q_0 = Q \setminus \{(s_3, t_3)\} \)

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\
\end{array}
\]

\( \tau = 0.5 \)
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}, \ Q_0 = Q \setminus \{(s_3, t_3)\} \)

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\
\end{array}
\]

\( \tau = 0.5 \)
Problem: Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

Example: \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}, \ Q_0 = Q \setminus \{(s_3, t_3)\} \)

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
3.5 & 0.5 & 0.5 & 0.5 & 0.5 & 3.5 \\
\end{array}
\]

\( \tau = 3.5 \)
Problem: Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

Example: \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\} \), \( Q_0 = Q \setminus \{(s_3, t_3)\} \)

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
3.5 & 0.5 & 0.5 & 0.5 & 0.5 & 3.5
\end{array}
\]

\( \tau = 3.5 \)
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:**

\[ Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}, \ Q_0 = Q \setminus \{(s_3, t_3)\}\]

<table>
<thead>
<tr>
<th>(\xi_{s_1})</th>
<th>(\xi_{s_2})</th>
<th>(\xi_{s_3})</th>
<th>(\xi_{t_3})</th>
<th>(\xi_{t_2})</th>
<th>(\xi_{t_1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>3.5</td>
</tr>
</tbody>
</table>

\(\tau = 3.5\)
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\} \), \( Q_0 = Q \setminus \{(s_3, t_3)\} \)

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
3.5 & 0.5 & 0.5 & 0.5 & 0.5 & 3.5 \\
\end{array}
\]

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
0.0 & 0.0 & - & - & 0.0 & 0.0 \\
\end{array}
\]

\( \tau = 0.0 \)
Sharing the Dual Growth

**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** \[ Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}, \quad Q_0 = Q \setminus \{(s_3, t_3)\} \]

<table>
<thead>
<tr>
<th>( \xi_{s_1} )</th>
<th>( \xi_{s_2} )</th>
<th>( \xi_{s_3} )</th>
<th>( \xi_{t_2} )</th>
<th>( \xi_{t_1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>3.5</td>
</tr>
</tbody>
</table>

\[ \text{AKR}(Q) \]

<table>
<thead>
<tr>
<th>( \xi_{s_1} )</th>
<th>( \xi_{s_2} )</th>
<th>( \xi_{s_3} )</th>
<th>( \xi_{t_3} )</th>
<th>( \xi_{t_2} )</th>
<th>( \xi_{t_1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>–</td>
<td>–</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

\[ \text{AKR}(Q_0) \]

\[ \tau = 0.5 \]
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}, \quad Q_0 = Q \setminus \{(s_3, t_3)\} \)

<table>
<thead>
<tr>
<th></th>
<th>(\xi_{s_1})</th>
<th>(\xi_{s_2})</th>
<th>(\xi_{s_3})</th>
<th>(\xi_{t_2})</th>
<th>(\xi_{t_1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q)\</td>
<td>3.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>3.5</td>
</tr>
<tr>
<td>(Q_0)\</td>
<td>1.5</td>
<td>1.5</td>
<td>–</td>
<td>–</td>
<td>1.5</td>
</tr>
</tbody>
</table>

\(\tau = 1.5\)
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** $Q = \{ (s_1, t_1), (s_2, t_2), (s_3, t_3) \}$, $Q_0 = Q \setminus \{ (s_3, t_3) \}$

<table>
<thead>
<tr>
<th>$\xi_{s_1}$</th>
<th>$\xi_{s_2}$</th>
<th>$\xi_{s_3}$</th>
<th>$\xi_{t_3}$</th>
<th>$\xi_{t_2}$</th>
<th>$\xi_{t_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>3.5</td>
</tr>
</tbody>
</table>

$\text{AKR}(Q_0)$

<table>
<thead>
<tr>
<th>$\xi_{s_1}$</th>
<th>$\xi_{s_2}$</th>
<th>$\xi_{s_3}$</th>
<th>$\xi_{t_3}$</th>
<th>$\xi_{t_2}$</th>
<th>$\xi_{t_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>1.5</td>
<td>–</td>
<td>–</td>
<td>1.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>

$\tau = 1.5$
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** $Q = \{ (s_1, t_1), (s_2, t_2), (s_3, t_3) \}$, $Q_0 = Q \setminus \{ (s_3, t_3) \}$

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
3.5 & 0.5 & 0.5 & 0.5 & 0.5 & 3.5 \\
\hline
\text{AKR}(Q_0) & \xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
1.5 & 1.5 & - & - & 1.5 & 1.5 \\
\end{array}
\]

$\tau = 1.5$
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}, \ Q_0 = Q \setminus \{(s_3, t_3)\} \)

<table>
<thead>
<tr>
<th>AKR((Q))</th>
<th>(\xi_{s_1})</th>
<th>(\xi_{s_2})</th>
<th>(\xi_{s_3})</th>
<th>(\xi_{t_1})</th>
<th>(\xi_{t_2})</th>
<th>(\xi_{t_1})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>3.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AKR((Q_0))</th>
<th>(\xi_{s_1})</th>
<th>(\xi_{s_2})</th>
<th>(\xi_{s_3})</th>
<th>(\xi_{t_1})</th>
<th>(\xi_{t_2})</th>
<th>(\xi_{t_1})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.5</td>
<td>1.5</td>
<td>–</td>
<td>–</td>
<td>1.5</td>
<td>2.5</td>
</tr>
</tbody>
</table>

\(\tau = 2.5\)
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}, \ Q_0 = Q \setminus \{(s_3, t_3)\} \)

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_2} & \xi_{t_1} \\
3.5 & 0.5 & 0.5 & 0.5 & 3.5 \\
\end{array}
\]

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_2} & \xi_{t_1} \\
2.5 & 1.5 & - & 1.5 & 2.5 \\
\end{array}
\]

\( \tau = 2.5 \)
**Problem:** Activity time of terminal may depend on presence of other terminal pairs. Impossible to achieve cross-monotonicity.

**Example:** \( Q = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\} \), \( Q_0 = Q \setminus \{(s_3, t_3)\} \):

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
3.5 & 0.5 & 0.5 & 0.5 & 0.5 & 3.5 \\
\end{array}
\]

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
2.5 & 1.5 & - & - & 1.5 & 2.5 \\
\end{array}
\]

\( \tau = 2.5 \)
**Question:** How long would a terminal pair need to connect if all other terminal pairs were absent?

**Death time:** for each terminal pair \((s, t) \in U\) define

\[
d(s) = d(t) = d(s, t) := \frac{1}{2} c(s, t),
\]

where \(c(s, t)\) is cost of minimum-cost \(s, t\)-path.
Cross-Monotonic Primal-Dual Algorithm

**New Activity Notion:** terminals $s, t$ are active until time $d(s, t)$

**Primal-Dual Algorithm KLS:** as before, but with modified activity notion

**Cost Shares:** define cost share of terminal $v \in Q$ as:

$$
\xi_v(Q) = \int_0^{d(v)} \frac{1}{a^\tau_v} \, d\tau
$$

**Theorem**

The cost shares $\xi$ computed by KLS are **cross-monotonic** and **2-budget balanced**.

[Könemann, Leonardi, Schäfer, van Zwam, SICOMP '08]
Example

\[ \text{KLS}(Q) \]

<table>
<thead>
<tr>
<th>( \xi_{s_1} )</th>
<th>( \xi_{s_2} )</th>
<th>( \xi_{s_3} )</th>
<th>( \xi_{t_3} )</th>
<th>( \xi_{t_2} )</th>
<th>( \xi_{t_1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

\( \tau = 0.0 \)
Example

KLS(Q) | \( \xi_{s_1} \) | \( \xi_{s_2} \) | \( \xi_{s_3} \) | \( \xi_{t_3} \) | \( \xi_{t_2} \) | \( \xi_{t_1} \) \\
---|---|---|---|---|---|---
0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5

\[ \tau = 0.5 \]
Example

KLS(Q) | $\xi_{s_1}$ | $\xi_{s_2}$ | $\xi_{s_3}$ | $\xi_{t_3}$ | $\xi_{t_2}$ | $\xi_{t_1}$ |
-------|-------------|-------------|-------------|-------------|-------------|-------------|
       | 0.5         | 0.5         | 0.5         | 0.5         | 0.5         | 0.5         |

$\tau = 0.5$

Guido Schäfer Cost Sharing and Approximation Algorithms 39
Example

<table>
<thead>
<tr>
<th>$\xi_{s_1}$</th>
<th>$\xi_{s_2}$</th>
<th>$\xi_{s_3}$</th>
<th>$\xi_{t_3}$</th>
<th>$\xi_{t_2}$</th>
<th>$\xi_{t_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

$\tau = 0.5$
Example

KLS(Q) | $\xi_{s_1}$ | $\xi_{s_2}$ | $\xi_{s_3}$ | $\xi_{t_3}$ | $\xi_{t_2}$ | $\xi_{t_1}$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.5</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
</tr>
</tbody>
</table>

$\tau = 1.5$
Example

KLS($Q$) \[ \begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
1.5 & 1.0 & 0.5 & 0.5 & 1.0 & 1.5 \\
\end{array} \]

\[ \tau = 1.5 \]
### Example

**KLS**($Q$) \[ \begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
2.5 & 1.0 & 0.5 & 0.5 & 1.0 & 2.5 \\
\end{array} \]

\[ \tau = 2.5 \]
Example

KLS\( (Q) \)

<table>
<thead>
<tr>
<th>( \xi_{s_1} )</th>
<th>( \xi_{s_2} )</th>
<th>( \xi_{s_3} )</th>
<th>( \xi_{t_3} )</th>
<th>( \xi_{t_2} )</th>
<th>( \xi_{t_1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
<td>2.5</td>
</tr>
</tbody>
</table>

\( \tau = 2.5 \)

Guido Schäfer
Cost Sharing and Approximation Algorithms
### Example

The table below represents the costs associated with the KLS(Q) function, where $\tau = 5.5$.

<table>
<thead>
<tr>
<th>$\xi_{s_1}$</th>
<th>$\xi_{s_2}$</th>
<th>$\xi_{s_3}$</th>
<th>$\xi_{t_3}$</th>
<th>$\xi_{t_2}$</th>
<th>$\xi_{t_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>

![Diagram](guido-schafer-cost-sharing-and-approximation-algorithms-39.png)
Example

\[ \xi_{s_1} \quad \xi_{s_2} \quad \xi_{s_3} \quad \xi_{t_3} \quad \xi_{t_2} \quad \xi_{t_1} \]

\[
\begin{array}{cccccc}
4.0 & 1.0 & 0.5 & 0.5 & 1.0 & 4.0 \\
\end{array}
\]

\[ \tau = 5.5 \]
### Example

<table>
<thead>
<tr>
<th></th>
<th>$\xi_{s_1}$</th>
<th>$\xi_{s_2}$</th>
<th>$\xi_{s_3}$</th>
<th>$\xi_{t_3}$</th>
<th>$\xi_{t_2}$</th>
<th>$\xi_{t_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLS($Q$)</td>
<td>4.0</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
<td>4.0</td>
</tr>
<tr>
<td>KLS($Q_0$)</td>
<td>0.0</td>
<td>0.0</td>
<td>–</td>
<td>–</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

$\tau = 0.0$
Example

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
4.0 & 1.0 & 0.5 & 0.5 & 1.0 & 4.0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
0.5 & 0.5 & - & - & 0.5 & 0.5 \\
\end{array}
\]

\[\tau = 0.5\]
Example

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
4.0 & 1.0 & 0.5 & 0.5 & 1.0 & 4.0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
1.5 & 1.5 & - & - & 1.5 & 1.5 \\
\end{array}
\]

\[\tau = 1.5\]
**Example**

<table>
<thead>
<tr>
<th>( \xi_{s_1} )</th>
<th>( \xi_{s_2} )</th>
<th>( \xi_{s_3} )</th>
<th>( \xi_{t_3} )</th>
<th>( \xi_{t_2} )</th>
<th>( \xi_{t_1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>

\( \tau = 1.5 \)

<table>
<thead>
<tr>
<th>( \xi_{s_1} )</th>
<th>( \xi_{s_2} )</th>
<th>( \xi_{s_3} )</th>
<th>( \xi_{t_3} )</th>
<th>( \xi_{t_2} )</th>
<th>( \xi_{t_1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>1.5</td>
<td>-</td>
<td>-</td>
<td>1.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Guido Schäfer

Cost Sharing and Approximation Algorithms
Example

\begin{align*}
\text{KLS}(Q) & \begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
4.0 & 1.0 & 0.5 & 0.5 & 1.0 & 4.0 \\
\end{array} \\
\text{KLS}(Q_0) & \begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
1.5 & 1.5 & - & - & 1.5 & 1.5 \\
\end{array}
\end{align*}

\[ \tau = 1.5 \]
Example

<table>
<thead>
<tr>
<th>KLS(Q)</th>
<th>(\xi_{s_1})</th>
<th>(\xi_{s_2})</th>
<th>(\xi_{s_3})</th>
<th>(\xi_{t_3})</th>
<th>(\xi_{t_2})</th>
<th>(\xi_{t_1})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.0</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KLS(Q_0)</th>
<th>(\xi_{s_1})</th>
<th>(\xi_{s_2})</th>
<th>(\xi_{s_3})</th>
<th>(\xi_{t_3})</th>
<th>(\xi_{t_2})</th>
<th>(\xi_{t_1})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.5</td>
<td>1.5</td>
<td>-</td>
<td>-</td>
<td>1.5</td>
<td>2.5</td>
</tr>
</tbody>
</table>

\(\tau = 2.5\)
Example

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
4.0 & 1.0 & 0.5 & 0.5 & 1.0 & 4.0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
2.5 & 1.5 & - & - & 1.5 & 2.5 \\
\end{array}
\]

\(\tau = 2.5\)
Example

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
4.0 & 1.0 & 0.5 & 0.5 & 1.0 & 4.0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
\xi_{s_1} & \xi_{s_2} & \xi_{s_3} & \xi_{t_3} & \xi_{t_2} & \xi_{t_1} \\
4.0 & 1.5 & - & - & 1.5 & 4.0 \\
\end{array}
\]

\( \tau = 5.5 \)

Guido Schäfer Cost Sharing and Approximation Algorithms 40
Example

<table>
<thead>
<tr>
<th>KLS(Q)</th>
<th>$\xi_{s_1}$</th>
<th>$\xi_{s_2}$</th>
<th>$\xi_{s_3}$</th>
<th>$\xi_{t_3}$</th>
<th>$\xi_{t_2}$</th>
<th>$\xi_{t_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.0</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KLS(Q$_0$)</th>
<th>$\xi_{s_1}$</th>
<th>$\xi_{s_2}$</th>
<th>$\xi_{s_3}$</th>
<th>$\xi_{t_3}$</th>
<th>$\xi_{t_2}$</th>
<th>$\xi_{t_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.0</td>
<td>1.5</td>
<td>–</td>
<td>–</td>
<td>1.5</td>
<td>4.0</td>
</tr>
</tbody>
</table>

$\tau = 5.5$
Lemma

The cost shares $\xi$ computed by KLS are \textit{cross-monotonic}.

Proof (sketch):

\[ M^\tau(v) = \text{moat of } v \text{ at time } \tau \text{ in KLS}(Q), \ Q \subseteq U \]
\[ M^\tau_0(v) = \text{moat of } v \text{ at time } \tau \text{ in KLS}(Q_0), \ Q_0 = Q \setminus \{(s, t)\} \]

\textbf{Obs.:} death-times of terminals are instance independent!

\[ M^\tau_0(v) \text{ active } \implies M^\tau(v) \text{ active} \]
\[ \implies M^\tau_0(v) \subseteq M^\tau(v) \]
\[ \implies a^\tau_0(v) \leq a^\tau(v) \]

\[ \xi_v(Q) = \int_0^{d(v)} \frac{1}{a^\tau(v)} \, d\tau \leq \int_0^{d(v)} \frac{1}{a^\tau_0(v)} \, d\tau = \xi_v(Q_0) \]
Lemma

The cost shares $\xi$ computed by KLS are cross-monotonic.

Proof (sketch):

$M^\tau(v) = \text{moat of } v \text{ at time } \tau \text{ in KLS}(Q)$, $Q \subseteq U$

$M^\tau_0(v) = \text{moat of } v \text{ at time } \tau \text{ in KLS}(Q_0)$, $Q_0 = Q \setminus \{(s, t)\}$

Obs.: death-times of terminals are instance independent!

$M^\tau_0(v) \text{ active } \Rightarrow M^\tau(v) \text{ active}$

$\Rightarrow M^\tau_0(v) \subseteq M^\tau(v)$

$\Rightarrow a^\tau_0(v) \leq a^\tau(v)$

$\xi_v(Q) = \int_0^{d(v)} \frac{1}{a^\tau(v)} \, d\tau \leq \int_0^{d(v)} \frac{1}{a^\tau_0(v)} \, d\tau = \xi_v(Q_0)$
 Lemma

The cost shares \( \xi \) computed by KLS are cross-monotonic.

Proof (sketch):
\( \mathcal{M}^\tau(v) = \) moat of \( v \) at time \( \tau \) in KLS(\( Q \)), \( Q \subseteq U \)
\( \mathcal{M}^\tau_0(v) = \) moat of \( v \) at time \( \tau \) in KLS(\( Q_0 \)), \( Q_0 = Q \setminus \{(s, t)\} \)

Obs.: death-times of terminals are instance independent!

\[ \mathcal{M}^\tau_0(v) \text{ active } \Rightarrow \mathcal{M}^\tau(v) \text{ active } \]
\[ \Rightarrow \mathcal{M}^\tau_0(v) \subseteq \mathcal{M}^\tau(v) \]
\[ \Rightarrow a^\tau_0(v) \leq a^\tau(v) \]

\[ \xi_v(Q) = \int_0^{d(v)} \frac{1}{a^\tau(v)} \, d\tau \leq \int_0^{d(v)} \frac{1}{a^\tau_0(v)} \, d\tau = \xi_v(Q_0) \]
Lemma

The cost shares $\xi$ computed by KLS are cross-monotonic.

Proof (sketch):

$\mathcal{M}^\tau(v) = \text{moat of } v \text{ at time } \tau \text{ in KLS}(Q), \ Q \subseteq U$

$\mathcal{M}_0^\tau(v) = \text{moat of } v \text{ at time } \tau \text{ in KLS}(Q_0), \ Q_0 = Q \setminus \{(s, t)\}$

Obs.: death-times of terminals are instance independent!

$\mathcal{M}_0^\tau(v) \text{ active } \Rightarrow \mathcal{M}^\tau(v) \text{ active}$

$\Rightarrow \mathcal{M}_0^\tau(v) \subseteq \mathcal{M}^\tau(v)$

$\Rightarrow a_0^\tau(v) \leq a^\tau(v)$

$$
\xi_v(Q) = \int_0^{d(v)} \frac{1}{a^\tau(v)} \, d\tau \leq \int_0^{d(v)} \frac{1}{a_0^\tau(v)} \, d\tau = \xi_v(Q_0)
$$
Lemma

The cost shares $\xi$ computed by KLS are cross-monotonic.

Proof (sketch):

$M^\tau(v) = \text{moat of } v \text{ at time } \tau \text{ in } \text{KLS}(Q), \ Q \subseteq U$

$M^\tau_0(v) = \text{moat of } v \text{ at time } \tau \text{ in } \text{KLS}(Q_0), \ Q_0 = Q \setminus \{(s, t)\}$

Obs.: death-times of terminals are instance independent!

$M^\tau_0(v) \text{ active } \Rightarrow M^\tau(v) \text{ active}$

$\Rightarrow M^\tau_0(v) \subseteq M^\tau(v)$

$\Rightarrow a^\tau_0(v) \leq a^\tau(v)$

$\xi_v(Q) = \int_0^{d(v)} \frac{1}{a^\tau(v)} \, d\tau \leq \int_0^{d(v)} \frac{1}{a^\tau_0(v)} \, d\tau = \xi_v(Q_0)$
Proving Cross-Monotonicity

**Lemma**

The cost shares $\xi$ computed by KLS are *cross-monotonic*.

**Proof (sketch):**

$M_\tau(v) = \text{moat of } v \text{ at time } \tau \text{ in } \text{KLS}(Q)$, $Q \subseteq U$

$M_{\tau_0}(v) = \text{moat of } v \text{ at time } \tau \text{ in } \text{KLS}(Q_0)$, $Q_0 = Q \setminus \{(s, t)\}$

**Obs.:** death-times of terminals are instance independent!

$M_{\tau_0}(v) \text{ active } \Rightarrow M_\tau(v) \text{ active}$

$\Rightarrow M_{\tau_0}(v) \subseteq M_\tau(v)$

$\Rightarrow a_{\tau_0}(v) \leq a_\tau(v)$

$\xi_v(Q) = \int_0^{d(v)} \frac{1}{a_\tau(v)} d_\tau \leq \int_0^{d(v)} \frac{1}{a_{\tau_0}(v)} d_\tau = \xi_v(Q_0)$
Lemma

The cost shares $\xi$ computed by KLS are cross-monotonic.

Proof (sketch):

$M^\tau(v) = \text{moat of } v \text{ at time } \tau \text{ in } KLS(Q), \ Q \subseteq U$

$M^\tau_0(v) = \text{moat of } v \text{ at time } \tau \text{ in } KLS(Q_0), \ Q_0 = Q \setminus \{(s, t)\}$

Obs.: death-times of terminals are instance independent!

$M^\tau_0(v) \text{ active } \Rightarrow M^\tau(v) \text{ active}$

$\Rightarrow M^\tau_0(v) \subseteq M^\tau(v)$

$\Rightarrow a^\tau_0(v) \leq a^\tau(v)$

$\xi_v(Q) = \int_0^{d(v)} \frac{1}{a^\tau(v)} \, d^\tau \leq \int_0^{d(v)} \frac{1}{a^\tau_0(v)} \, d^\tau = \xi_v(Q_0)$
Lemma

The cost shares $\xi$ computed by KLS are $2$-budget balanced.

Proof (sketch):

$(F, y) = \text{forest and dual computed by KLS(}Q, Q \subseteq U \text{). Then}

$$c(F) \leq 2 \sum_{S} y_{S} = \sum_{i \in Q} \xi_{s_{i}t_{i}}$$

But: $y$ is not dual feasible since some active moats do not correspond to Steiner cuts. Can still show that $\sum y_{S} \leq \text{OPT}(Q)$!

Idea: charge dual growth against an optimal forest $F^{*}$ for $Q$. 
**Lemma**

The cost shares $\xi$ computed by KLS are 2-budget balanced.

**Proof (sketch):**

$(F, y) =$ forest and dual computed by KLS($Q$), $Q \subseteq U$. Then

$$c(F) \leq 2 \sum_S y_S = \sum_{i \in Q} \xi_{si_i}$$

But: $y$ is not dual feasible since some active moats do not correspond to Steiner cuts. Can still show that $\sum y_S \leq OPT(Q)$!

Idea: charge dual growth against an optimal forest $F^*$ for $Q$. 
Lemma

The cost shares $\xi$ computed by KLS are 2-budget balanced.

Proof (sketch):

$(F, y) = \text{forest and dual computed by KLS}(Q), \; Q \subseteq U$. Then

$$c(F) \leq 2 \sum_S y_S = \sum_{i \in Q} \xi_{s_it_i}$$

But: $y$ is not dual feasible since some active moats do not correspond to Steiner cuts. Can still show that $\sum y_S \leq OPT(Q)$!

Idea: charge dual growth against an optimal forest $F^*$ for $Q$. 

Lemma

The cost shares $\xi$ computed by KLS are 2-budget balanced.

Proof (sketch):

$(F, y) =$ forest and dual computed by KLS$(Q)$, $Q \subseteq U$. Then

$$c(F) \leq 2 \sum_S y_S = \sum_{i \in Q} \xi_{s_it_i}$$

But: $y$ is not dual feasible since some active moats do not correspond to Steiner cuts. Can still show that $\sum y_S \leq OPT(Q)$!

Idea: charge dual growth against an optimal forest $F^*$ for $Q$. 
The cost shares $\xi$ computed by KLS are 2-budget balanced.

Proof (sketch):

$(F, y) =$ forest and dual computed by KLS$(Q), \ Q \subseteq U$. Then

$$c(F) \leq 2 \sum_S y_S = \sum_{i \in Q} \xi_{s_i t_i}$$

But: $y$ is not dual feasible since some active moats do not correspond to Steiner cuts. Can still show that $\sum y_S \leq \text{OPT}(Q)$!

Idea: charge dual growth against an optimal forest $F^*$ for $Q$. 

Guido Schäfer
Lemma

The cost shares $\xi$ computed by KLS are 2-budget balanced.

Proof (sketch):

$(F, y) =$ forest and dual computed by KLS($Q$), $Q \subseteq U$. Then

$$c(F) \leq 2 \sum_S y_S = \sum_{i \in Q} \xi_{s_i t_i}$$

But: $y$ is not dual feasible since some active moats do not correspond to Steiner cuts. Can still show that $\sum y_S \leq OPT(Q)$!

Idea: charge dual growth against an optimal forest $F^*$ for $Q$. 
Let $Q = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ such that
\[ d(s_1, t_1) \leq \cdots \leq d(s_k, t_k) \]

Define precedence order on terminals:
\[ s_1 \prec t_1 \prec s_2 \prec t_2 \prec \cdots \prec s_k \prec t_k \]

Terminal $v$ is responsible at time $\tau$ if $u \prec v$ for all $u \in M^\tau(v)$. Define $r^\tau(v) = 1$ if $v$ is responsible at time $\tau$ and $r^\tau(v) = 0$ otherwise. Let the responsibility time of $v$ be
\[ r(v) = \int_0^{d(v)} r^\tau(v) \, d\tau \]

Intuition: No sharing of dual growth; the responsible terminal gets it all! Suffices to bound total responsibility time by $OPT(Q)$. 

Guido Schäfer

Cost Sharing and Approximation Algorithms

43
Let \( Q = \{(s_1, t_1), \ldots, (s_k, t_k)\} \) such that

\[
d(s_1, t_1) \leq \cdots \leq d(s_k, t_k)
\]

Define precedence order on terminals:

\[
s_1 \prec t_1 \prec s_2 \prec t_2 \prec \cdots \prec s_k \prec t_k
\]

Terminal \( v \) is responsible at time \( \tau \) if \( u \prec v \) for all \( u \in M^\tau(v) \).

Define \( r^\tau(v) = 1 \) if \( v \) is responsible at time \( \tau \) and \( r^\tau(v) = 0 \) otherwise. Let the responsibility time of \( v \) be

\[
r(v) = \int_0^{d(v)} r^\tau(v) \, d\tau
\]

Intuition: No sharing of dual growth; the responsible terminal gets it all! Suffices to bound total responsibility time by \( OPT(Q) \).
Let \( Q = \{(s_1, t_1), \ldots, (s_k, t_k)\} \) such that
\[
d(s_1, t_1) \leq \cdots \leq d(s_k, t_k)
\]

Define precedence order on terminals:
\[
s_1 \prec t_1 \prec s_2 \prec t_2 \prec \cdots \prec s_k \prec t_k
\]

Terminal \( v \) is responsible at time \( \tau \) if \( u \prec v \) for all \( u \in M^\tau(v) \).
Define \( r^\tau(v) = 1 \) if \( v \) is responsible at time \( \tau \) and \( r^\tau(v) = 0 \) otherwise. Let the responsibility time of \( v \) be
\[
r(v) = \int_0^{d(v)} r^\tau(v) \, d\tau
\]

Intuition: No sharing of dual growth; the responsible terminal gets it all! Suffices to bound total responsibility time by \( OPT(Q) \).
Let $Q = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ such that
\[d(s_1, t_1) \leq \cdots \leq d(s_k, t_k)\]

Define precedence order on terminals:
\[s_1 \prec t_1 \prec s_2 \prec t_2 \prec \cdots \prec s_k \prec t_k\]

Terminal $v$ is responsible at time $\tau$ if $u \prec v$ for all $u \in M^{\tau}(v)$. Define $r^{\tau}(v) = 1$ if $v$ is responsible at time $\tau$ and $r^{\tau}(v) = 0$ otherwise. Let the responsibility time of $v$ be
\[r(v) = \int_0^{d(v)} r^{\tau}(v) d\tau\]

Intuition: No sharing of dual growth; the responsible terminal gets it all! Suffices to bound total responsibility time by $OPT(Q)$. 

Guido Schäfer Cost Sharing and Approximation Algorithms 43
Proving Budget Balance

Let $Q = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ such that

$$d(s_1, t_1) \leq \cdots \leq d(s_k, t_k)$$

Define precedence order on terminals:

$$s_1 \prec t_1 \prec s_2 \prec t_2 \prec \cdots \prec s_k \prec t_k$$

Terminal $v$ is responsible at time $\tau$ if $u \prec v$ for all $u \in M^\tau(v)$. Define $r^\tau(v) = 1$ if $v$ is responsible at time $\tau$ and $r^\tau(v) = 0$ otherwise. Let the responsibility time of $v$ be

$$r(v) = \int_0^{d(v)} r^\tau(v) \, d\tau$$

Intuition: No sharing of dual growth; the responsible terminal gets it all! Suffices to bound total responsibility time by $OPT(Q)$. 

Guido Schäfer Cost Sharing and Approximation Algorithms 43
Let \( Q = \{ (s_1, t_1), \ldots, (s_k, t_k) \} \) such that
\[
d(s_1, t_1) \leq \cdots \leq d(s_k, t_k)
\]

Define precedence order on terminals:
\[
s_1 \prec t_1 \prec s_2 \prec t_2 \prec \cdots \prec s_k \prec t_k
\]

Terminal \( v \) is responsible at time \( \tau \) if \( u \prec v \) for all \( u \in M^\tau(v) \).
Define \( r^\tau(v) = 1 \) if \( v \) is responsible at time \( \tau \) and \( r^\tau(v) = 0 \) otherwise. Let the responsibility time of \( v \) be
\[
r(v) = \int_0^{d(v)} r^\tau(v) d\tau
\]

**Intuition:** No sharing of dual growth; the responsible terminal gets it all! Suffices to bound total responsibility time by \( OPT(Q) \).
Consider a tree $T \in F^*$ and assume that $T$ spans terminals $\{v_1, \ldots, v_p\}$.

Every terminal $v$ that is responsible at time $\tau$ loads a distinct part of $T$. **Note:** argument applies if there are at least two responsible terminals at time $\tau$.

Let $v_p$ be the terminal with highest responsibility time. Then

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

**Note:** $v_p$'s mate is in $T$ as well.

$\Rightarrow r(v_p) \leq d(v_p) \leq \frac{1}{2} c(T)$
Consider a tree $T \in F^*$ and assume that $T$ spans terminals $\{v_1, \ldots, v_p\}$.
Every terminal $v$ that is responsible at time $\tau$ loads a distinct part of $T$. Note: argument applies if there are at least two responsible terminals at time $\tau$.
Let $v_p$ be the terminal with highest responsibility time. Then
\[
\sum_{i=1}^{p-1} r(v_i) \leq c(T).
\]
Note: $v_p$’s mate is in $T$ as well.
$\Rightarrow r(v_p) \leq d(v_p) \leq \frac{1}{2} c(T)$.
Consider a tree $T \in F^*$ and assume that $T$ spans terminals $\{v_1, \ldots, v_p\}$. Every terminal $v$ that is responsible at time $\tau$ loads a distinct part of $T$. **Note:** argument applies if there are at least two responsible terminals at time $\tau$.

Let $v_p$ be the terminal with highest responsibility time. Then

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

**Note:** $v_p$’s mate is in $T$ as well. 

$$\Rightarrow r(v_p) \leq d(v_p) \leq \frac{1}{2} c(T)$$
Consider a tree $T \in F^*$ and assume that $T$ spans terminals $\{v_1, \ldots, v_p\}$. Every terminal $v$ that is responsible at time $\tau$ loads a distinct part of $T$. **Note:** argument applies if there are at least two responsible terminals at time $\tau$. Let $v_p$ be the terminal with highest responsibility time. Then

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

**Note:** $v_p$’s mate is in $T$ as well. 

$\Rightarrow r(v_p) \leq d(v_p) \leq \frac{1}{2} c(T)$
Consider a tree $T \in F^*$ and assume that $T$ spans terminals $\{v_1, \ldots, v_p\}$. Every terminal $v$ that is responsible at time $\tau$ loads a distinct part of $T$. **Note:** argument applies if there are at least two responsible terminals at time $\tau$.

Let $v_p$ be the terminal with highest responsibility time. Then

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

**Note:** $v_p$’s mate is in $T$ as well.

$$
\Rightarrow r(v_p) \leq d(v_p) \leq \frac{1}{2} c(T)
$$
Consider a tree \( T \in F^* \) and assume that \( T \) spans terminals \( \{v_1, \ldots, v_p\} \). Every terminal \( v \) that is responsible at time \( \tau \) loads a distinct part of \( T \). **Note:** argument applies if there are at least two responsible terminals at time \( \tau \).

Let \( v_p \) be the terminal with highest responsibility time. Then

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

**Note:** \( v_p \)'s mate is in \( T \) as well.

\[
\Rightarrow r(v_p) \leq d(v_p) \leq \frac{1}{2} c(T).
\]
Further Consequences

Suppose our modified Steiner forest algorithm produces forest $F$ and (infeasible) dual $y$ for terminal pair set $Q$.

Surprisingly, can still show

$$c(F) \leq (2 - 1/k) \cdot OPT(Q)$$

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

<table>
<thead>
<tr>
<th>$OPT(Q)$</th>
<th>$2k - 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard AKR-dual</td>
<td>$k$</td>
</tr>
<tr>
<td>Our dual</td>
<td>$2k - 1$</td>
</tr>
</tbody>
</table>
**Recall:** death-times induce precedence order $\prec$ on terminals

$$s_1 \prec t_1 \prec s_2 \prec t_2 \prec \cdots \prec s_k \prec t_k$$

Associate each cut $S \subseteq V$ with a terminal

**Example:** $v \prec \bar{v} \prec w \prec \bar{w}$

Guido Schäfer

Cost Sharing and Approximation Algorithms

46
Lifted-Cut Dual for Steiner Forests

\[ \text{OPT}_{\text{LC}} = \max \sum_{S \subseteq V} y_S \]

s.t. \[ \sum_{S \subseteq V: e \in \delta(S)} y_S \leq c_e \quad \forall e \in E \]
\[ \sum_{S \in \mathcal{S}_v} y_S + \sum_{S \in \mathcal{N}_v} y_S \leq d(v) \quad \forall v \in R \]
\[ y_S \geq 0 \quad \forall S \subseteq V \]

**Theorem**

1. \( \text{OPT}_{\text{UC}} \leq \text{OPT}_{\text{LC}} \leq \text{OPT} \)
2. IP/LC gap is about 2
3. Additional strength of LC can be used to prove better approximation ratio of AKR for certain instances

[Könemann, Leonardi, Schäfer, van Zwan, SICOMP '08]
There is no \((2 - \epsilon)\)-budget balance cross-monotonic cost sharing scheme for the **Steiner tree problem**  
[Könemann, Leonardi, Schäfer, van Zwam, SICOMP '08]

KLS is \(\Theta(\log^2 n)\)-approximate with respect to social cost  
[Chawla, Roughgarden, Sundararajan, WINE '06]

Similar idea yields 3-budget balanced, \(\Theta(\log^2 n)\)-approximate, cross-monotonic cost sharing function for the **price-collecting Steiner forest problem**  
[Gupta, Könemann, Leonardi, Ravi, Schäfer, SODA '07]
Idea:

- every player $i$ has a cut-demand function $f_i : 2^V \rightarrow \{0, 1\}$
- model general connectivity game via the following LP

\[
\begin{align*}
\text{min} & \quad \sum_{e \in E} c_e \cdot x_e \\
\text{s.t.} & \quad \sum_{e \in \delta(S)} x_e \geq f_i(S) \quad \forall S \subseteq V, \forall i \in U \\
& \quad x_e \in \{0, 1\} \quad \forall e \in E
\end{align*}
\]

- adapt approximation framework by Goemans and Williamson to obtain $O(1)$-budget balance, cross-monotonic cost sharing function

[Könemann, Leonardi, Schäfer, Wheatley, manuscript]
Conclusions and Open Problems
Conclusions

Moulin’s framework enables to derive group-strategyproof cost sharing mechanisms through cross-monotonic cost sharing functions.

Have techniques at hand to bound social cost efficiency of Moulin mechanisms.

Trade-off between budget balance and social cost approximation guarantees of Moulin mechanisms are well-understood for several fundamental optimization problems.

Designing cross-monotonic cost sharing functions may lead to new insights that are useful in other contexts.
Characterization of GSP Mechanisms

Group-Strategyproof Cost Sharing Mechanisms
Characterization of GSP Mechanisms

Group-Strategyproof Cost Sharing Mechanisms

Moulin Mechanisms
Characterization of GSP Mechanisms

Group-Strategyproof Cost Sharing Mechanisms

Moulin Mechanisms
Characterization of GSP Mechanisms

Group-Strategyproof Cost Sharing Mechanisms

Characterization has recently been given [Pountourakis and Vidali, ESA ’10]

Moulin Mechanisms
Open Problem: Can we exploit the characterization of group-strategyproof cost sharing mechanisms algorithmically?

Open Problem: Are there other general techniques to derive group-strategyproof cost sharing mechanisms?

Open Problem: What are the trade-offs between group-strategyproofness and budget balance and social cost approximation guarantees?