Truthful Mechanisms for Scheduling Problems

Project B16: Mechanisms for Network Design Problems

Guido Schäfer
Technische Universität Berlin

joint work with Janina Brenner
Algorithmic Game Theory

- Economics
- Game Theory
- Algorithms
- Theoretical CS/ Applied Maths
Algorithmic Game Theory
Motivation

The diagram illustrates the concept of $C_{\text{max}}$, which represents the maximum completion time in scheduling problems. The bars indicate different tasks, each with a specified duration and associated cost.

- The first bar has a duration of 5€ and a cost of 10€.
- The second bar has a duration of 10€ and a cost of 31€.
- The third bar has a duration of 18€ and a cost of 1€.
- The fourth bar has a duration of 12€ and a cost of 1€.

The arrows indicate the scheduling process, aiming to minimize $C_{\text{max}}$ while considering the costs associated with each task.
Motivation
1. Cost sharing model, objectives, truthfulness
2. Moulin mechanisms
   - mechanism
   - state of affairs
   - limitations and new trade-offs
3. Singleton mechanisms
   - mechanism
   - framework: approximation algorithm → singleton mechanism
   - three example applications
4. Concluding Remarks
Setting:

- service provider offers some service
- set $U$ of potential users, interested in service
- cost function $C : 2^U \to \mathbb{R}^+$
  
  $C(S) =$ optimal cost to serve user-set $S \subseteq U$
- every user $i \in U$:
  - has a (private) valuation $v_i \geq 0$ for receiving the service
  - announces bid $b_i \geq 0$, the maximum amount he is willing to pay for the service
Cost sharing mechanism $M$:

- collects all bids $(b_i)_{i \in U}$ from users
- based on these bids:
  - decides a set $Q \subseteq U$ of users that receive service
  - computes (approximate) solution for $Q$ of cost $\bar{C}(Q)$
  - determines a cost share $\xi_i(Q) \leq b_i$ for every user $i \in Q$
Cost sharing mechanism $M$: 

- collects all bids $(b_i)_{i \in U}$ from users 
- based on these bids:
  - decides a set $Q \subseteq U$ of users that receive service 
  - computes (approximate) solution for $Q$ of cost $\tilde{C}(Q)$ 
  - determines a cost share $\xi_i(Q) \leq b_i$ for every user $i \in Q$

Strategic behaviour: every user $i \in U$ acts selfishly and attempts to maximize his utility:

- utility $u_i := v_i - \xi_i(Q)$ if served, $u_i := 0$ otherwise 
- user manipulates mechanism if advantageous by misreporting his valuation, i.e., $b_i \neq v_i$
Strategyproofness: utility of every user $i \in U$ is maximized if he bids truthfully $b_i = v_i$, independently of other users.
**Strategyproofness:** utility of every user $i \in U$ is maximized if he bids truthfully $b_i = v_i$, independently of other users.

**Group-strategyproofness:** same holds true even if users form coalitions and coordinate their biddings.
Illustration: Group-strategyproofness
Illustration: Group-strategyproofness
\( \beta \)-budget balance: cost shares approximate servicing cost

\[
\tilde{C}(Q) \leq \sum_{i \in Q} \xi_i(Q) \leq \beta \cdot C(Q), \quad \beta \geq 1
\]
\( \beta \)-budget balance: cost shares approximate servicing cost

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

Social cost: define minimum social cost

\[
\Pi^* := \min_{S \subseteq U} \left\{ \sum_{i \in S} v_i + C(S) \right\}
\]
**Objectives**

**β-budget balance:** cost shares approximate servicing cost

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

**Social cost:** define minimum social cost

\[
\Pi^* := \min_{S \subseteq U} \left\{ \sum_{i \notin S} v_i + C(S) \right\}
\]

**α-approximate:** computed solution approximates social cost

\[
\sum_{i \notin Q} v_i + \bar{C}(Q) \leq \alpha \cdot \Pi^*, \quad \alpha \geq 1
\]
Moulin’s Framework

Moulin mechanism $M(\xi)$:

1: Initialize: $Q \leftarrow U$
2: If for each user $i \in Q$: $\xi_i(Q) \leq b_i$ then STOP
3: Otherwise, remove from $Q$ all users with $\xi_i(Q) > b_i$ and repeat
Moulin’s Framework

Moulin mechanism $M(\xi)$:
1. Initialize: $Q \leftarrow U$
2. If for each user $i \in Q$: $\xi_i(Q) \leq b_i$ then STOP
3. Otherwise, remove from $Q$ all users with $\xi_i(Q) > b_i$ and repeat

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

[MoIlin, Shenker ’01]
[Jain, Vazirani ’01]
## Upper bounds

<table>
<thead>
<tr>
<th>Reference</th>
<th>Problem</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Moulin, Shenker '01]</td>
<td>submodular cost</td>
<td>1</td>
</tr>
<tr>
<td>[Jain, Vazirani '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>[Pal, Tardos '03]</td>
<td>facility location</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>single-commodity rent-or-buy</td>
<td>15</td>
</tr>
<tr>
<td>[Leonardi, S. '03], [Gupta et al. '03]</td>
<td>single-commodity rent-or-buy</td>
<td>4</td>
</tr>
<tr>
<td>[Leonardi, S. '03]</td>
<td>connected facility location</td>
<td>30</td>
</tr>
<tr>
<td>[Könemann, Leonardi, S. '05]</td>
<td>Steiner forest</td>
<td>2</td>
</tr>
<tr>
<td>[Gupta et al. '07]</td>
<td>price-collecting Steiner forest</td>
<td>3</td>
</tr>
<tr>
<td>[Bleischwitz, Monien '07]</td>
<td>makespan scheduling</td>
<td>2</td>
</tr>
</tbody>
</table>

## Lower bounds

<table>
<thead>
<tr>
<th>Reference</th>
<th>Problem</th>
<th>Lower Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Immorlica et al. '05]</td>
<td>set cover, edge cover</td>
<td>$\Omega(n)$</td>
</tr>
<tr>
<td></td>
<td>facility location</td>
<td>3</td>
</tr>
<tr>
<td>[Könemann et al. '05]</td>
<td>Steiner forest</td>
<td>2</td>
</tr>
<tr>
<td>[Bleischwitz, Monien '07]</td>
<td>makespan scheduling</td>
<td>2</td>
</tr>
<tr>
<td>[Brenner, S. '07]</td>
<td>completion time scheduling, etc.</td>
<td>$\Omega(n)$</td>
</tr>
<tr>
<td>Reference</td>
<td>Problem</td>
<td>$\beta$</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>[Roughgarden, Sundararajan '06]</td>
<td>submodular cost</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Steiner tree</td>
<td>2</td>
</tr>
<tr>
<td>[Chawla, Roughgarden, Sundararajan '06]</td>
<td>Steiner forest</td>
<td>2</td>
</tr>
<tr>
<td>[Roughgarden, Sundararajan '07]</td>
<td>facility location</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>SRoB</td>
<td>4</td>
</tr>
<tr>
<td>[Gupta et al. '07]</td>
<td>price-collecting SF</td>
<td>3</td>
</tr>
<tr>
<td>[Brenner, S. '07]</td>
<td>makespan scheduling</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>cost-stable problems</td>
<td></td>
</tr>
</tbody>
</table>
Group-strategyproofness:
- very strong notion of truthfulness
- often bottleneck in achieving good performance guarantees
- strong lower bounds exist, even if we allow exponential time
Group-strategyproofness:

- very strong notion of truthfulness
- often bottleneck in achieving good performance guarantees
- strong lower bounds exist, even if we allow exponential time

Idea: use (slightly) weaker notion of group-strategyproofness: weak group-strategyproofness [Mehta et al. '07]
Illustration: Weak Group-strategyproofness

utility

users
Illustration: Weak Group-strategyproofness

utility

coalition

users
Illustration: Weak Group-strategyproofness

- utility
- users
- coalition
Offer function: Let $\tau : U \times 2^U \rightarrow \mathbb{R}^+$ be an offer function $\tau(i, S) =$ offer time of user $i$ with respect to $S \subseteq U$
Offer function: Let \( \tau : U \times 2^U \rightarrow \mathbb{R}^+ \) be an offer function
\[ \tau(i, S) = \text{offer time of user } i \text{ with respect to } S \subseteq U \]

Singleton offer function: for every subset \( S \subseteq U \) and for every two users \( i, j \in S \): \( \tau(i, S) \neq \tau(j, S) \)
Offer function: Let $\tau : U \times 2^U \rightarrow \mathbb{R}^+$ be an offer function
$\tau(i, S) =$ offer time of user $i$ with respect to $S \subseteq U$

Singleton offer function: for every subset $S \subseteq U$ and for every two users $i, j \in S$: $\tau(i, S) \neq \tau(j, S)$

Singleton mechanism $M(\xi, \tau)$:

1. Initialize: $Q \leftarrow U$
2. If for each user $i \in Q$: $\xi_i(Q) \leq b_i$ then STOP
3. Otherwise: Among all users in $S$ with $\xi_i(S) > b_i$, let $i^*$ be the one with minimum offer time $\tau(i, S)$. Remove $i^*$ from $Q$ and repeat.
Thm: Let \textit{ALG} be a $\rho$-approximation algorithm that satisfies certain conditions. Then \textit{ALG} can be turned into a singleton mechanism that is weakly group-strategyproof and $\rho$-budget balanced.

[Brenner, S. ’08]
**Consistent singleton offer function:** for every $T \subseteq S$:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>
| $T$ | 1 | 2 | 3 | 5 | 6 | 8 | 9 | \n
($\tau(\cdot, S)$ order)
**Consistent singleton offer function:** for every $T \subseteq S$:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$(\tau(\cdot, S) \text{ order})$

\[5 9 8 1 2 3 4 5 6 7 32 \quad 8 \quad 6 \quad 9\]
Consistent singleton offer function: for every $T \subseteq S$:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>$T$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$(\tau(\cdot, S) \text{ order})$

$(\tau(\cdot, T) \text{ order})$
Consistent singleton offer function: for every \( T \subseteq S \):

\[
\begin{array}{cccccccc}
S & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
T & 1 & 2 & 3 & 5 & 6 & 8 & 9 \\
T & 1 & 2 & 3 & 9 & 8 & 5 & 6
\end{array}
\]

(\( \tau(\cdot, S) \) order)

\( \tau \)-increasing: \( ALG \) is \( \tau \)-increasing if for every \( S \subseteq U \) and every \( i \in S \):

\[
\xi_i(S) := \bar{C}(S_i) - \bar{C}(S_{i-1}) \geq 0,
\]

where \( S_i \) is the set of the first \( i \) elements of \( S \) (ordered according to \( \tau(\cdot, S) \)).
Problem: parallel machines, no preemption, minimize makespan

Offer function: order jobs by non-increasing processing times (Graham’s rule)

Thm: There is a singleton mechanism that is weakly group-strategyproof and 4/3-budget balanced.

Contrast: lower bound for Moulin mechanisms: 2 (budget balance)
Problem: parallel machines, no preemption, minimize sum of weighted completion times

Offer function: order jobs by non-increasing weight per processing time (Smith’s rule)

Thm: There is a singleton mechanism that is weakly group-strategyproof, 1.21-budget balanced, and 2.42-approximate.

Contrast: lower bound for Moulin mechanisms: \( \Omega(n) \) (budget balance)
**Problem:** single machine, release dates, preemption, minimize sum of completion times

**Offer function:** order jobs by increasing completion times in the shortest remaining processing time schedule

**Thm:** There is a singleton mechanism that is weakly group-strategyproof, 1-budget balanced, and 4-approximate.

**Contrast:** lower bound for Moulin mechanisms: $\Omega(n)$ (budget balance)
developed framework to convert approximation algorithms into weakly group-strategyproof mechanisms (if only some mild conditions are satisfied)
- developed framework to convert approximation algorithms into weakly group-strategyproof mechanisms (if only some mild conditions are satisfied)
- slightly relaxed notion of truthfulness leads to significant improvement in performance guarantees; still allowing coalitions
- developed framework to convert approximation algorithms into weakly group-strategyproof mechanisms (if only some mild conditions are satisfied)
- slightly relaxed notion of truthfulness leads to significant improvement in performance guarantees; still allowing coalitions
- framework yields good approximation ratios in scheduling context
- developed framework to convert approximation algorithms into weakly group-strategyproof mechanisms (if only some mild conditions are satisfied)
- slightly relaxed notion of truthfulness leads to significant improvement in performance guarantees; still allowing coalitions
- framework yields good approximation ratios in scheduling context
- by-product: obtain approximation algorithms for respective scheduling problems with rejection
developed framework to convert approximation algorithms into weakly group-strategyproof mechanisms (if only some mild conditions are satisfied)

slightly relaxed notion of truthfulness leads to significant improvement in performance guarantees; still allowing coalitions

framework yields good approximation ratios in scheduling context

by-product: obtain approximation algorithms for respective scheduling problems with rejection

Thank you!