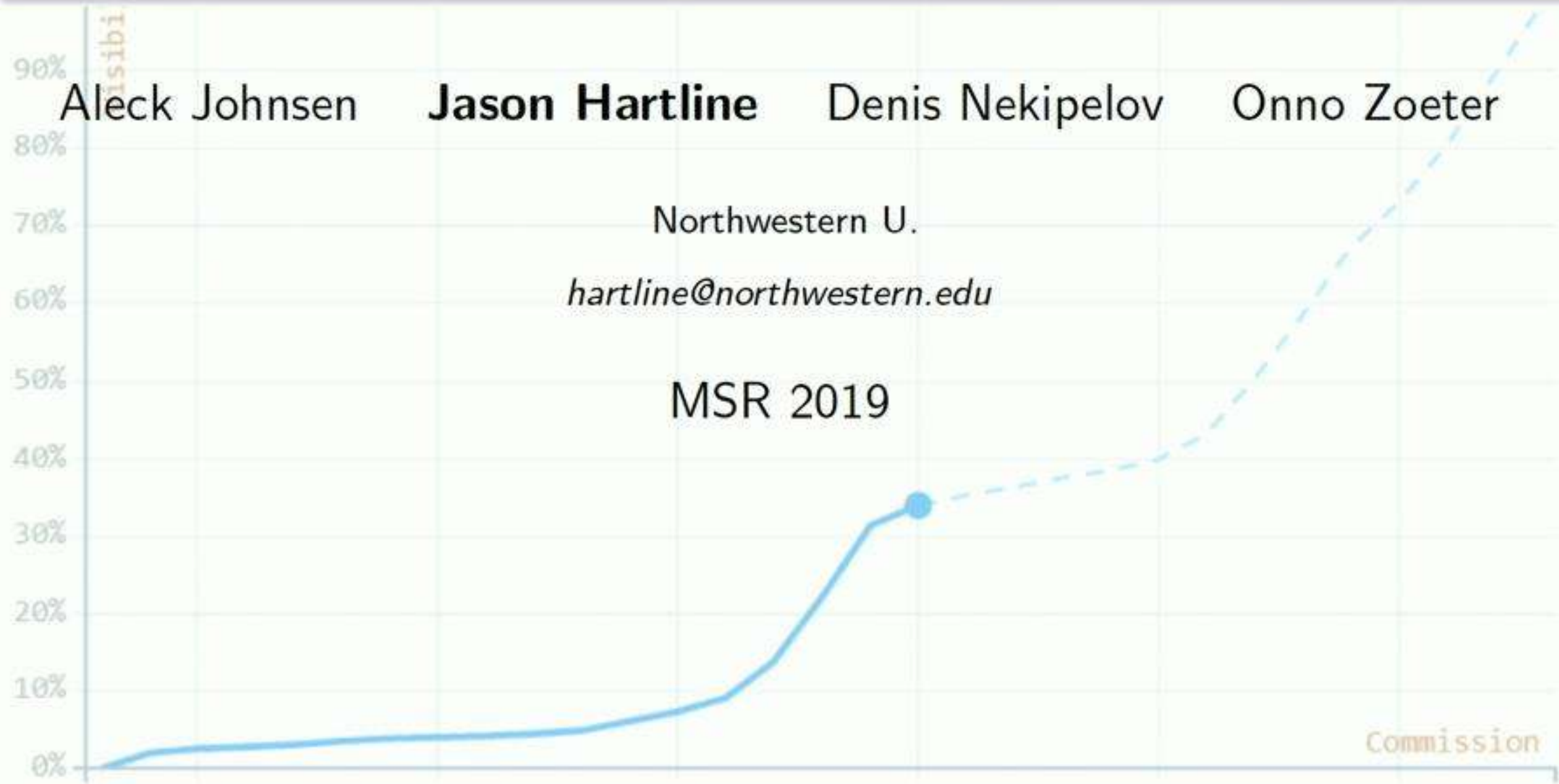


Dashboard Mechanisms for Online Marketplaces¹



¹<https://arxiv.org/abs/1905.05750>

Online Marketplaces:

- short-lived **users** matched to strategic long-lived **agents**.
- matching **algorithm**:
 - marketplace prioritize agents
 - users select agents

Online Marketplaces:

- short-lived **users** matched to strategic long-lived **agents**.
- matching **algorithm**:
 - marketplace prioritize agents
 - users select agents

Examples: ad auctions, booking.com, eBay, etc.

Algorithm Design vs. Mechanism Design

Algorithm Design: given input (agents' values), determine good output (agents to prioritize); goal: good welfare for input.

Algorithm Design vs. Mechanism Design

Algorithm Design: given input (agents' values), determine good output (agents to prioritize); goal: good welfare for input.

Mechanism Design: solicit bids from agents, determine outcome; goal: good welfare in equilibrium.

Algorithm Design vs. Mechanism Design

Algorithm Design: given input (agents' values), determine good output (agents to prioritize); goal: good welfare for input.

Mechanism Design: solicit bids from agents, determine outcome; goal: good welfare in equilibrium.

Reducing Mechanism Design to Algorithm Design:

- VCG mechanism, cf. second-price auction. [Vickrey '61, Clarke '71, Groves '73]
- unbiased payment mechanism [Archer, Tardos '01]
- implicit payment mechanism [Babaioff, Klienber, Slivkins '10]

Algorithm Design vs. Mechanism Design

Algorithm Design: given input (agents' values), determine good output (agents to prioritize); goal: good welfare for input.

Mechanism Design: solicit bids from agents, determine outcome; goal: good welfare in equilibrium.

Reducing Mechanism Design to Algorithm Design:

- VCG mechanism, cf. second-price auction. [Vickrey '61, Clarke '71, Groves '73]
- unbiased payment mechanism [Archer, Tardos '01]
- implicit payment mechanism [Babaioff, Klienber, Slivkins '10]

Main idea: carefully constructed **truthful payment format**.

Algorithm Design vs. Mechanism Design

Algorithm Design: given input (agents' values), determine good output (agents to prioritize); goal: good welfare for input.

Mechanism Design: solicit bids from agents, determine outcome; goal: good welfare in equilibrium.

Reducing Mechanism Design to Algorithm Design:

- VCG mechanism, cf. second-price auction. [Vickrey '61, Clarke '71, Groves '73]
- unbiased payment mechanism [Archer, Tardos '01]
- implicit payment mechanism [Babaioff, Kleinberg, Slivkins '10]

Main idea: carefully constructed **truthful payment format**.

Challenge: almost all online markets have **non-truthful payment format**.

Algorithm Design vs. Mechanism Design

Algorithm Design: given input (agents' values), determine good output (agents to prioritize); goal: good welfare for input.

Mechanism Design: solicit bids from agents, determine outcome; goal: good welfare in equilibrium.

Reducing Mechanism Design to Algorithm Design:

- VCG mechanism, cf. second-price auction. [Vickrey '61, Clarke '71, Groves '73]
- unbiased payment mechanism [Archer, Tardos '01]
- implicit payment mechanism [Babaioff, Klienber, Slivkins '10]

Main idea: carefully constructed **truthful payment format**.

Challenge: almost all online markets have **non-truthful payment format**.

Goal for Talk: sequential non-truthful mech. \approx sequential truthful mech.

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm.
- 3 winners pay their bids.

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which algorithms are good?
- 3 winners pay their bids.

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which algorithms are good?
- 3 winners pay their bids.

Definition (Highest-bids-win)

allocate to the feasible set of agents with highest total bid.

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which algorithms are good?
- 3 winners pay their bids.

Definition (Highest-bids-win)

allocate to the feasible set of agents with highest total bid.

Example (exclusive ad space)

- 1 item, 2 agents, winner-pays-bid highest-bids-win mechanism

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which algorithms are good?
- 3 winners pay their bids.

Definition (Highest-bids-win)

allocate to the feasible set of agents with highest total bid.

Example (exclusive ad space)

- 1 item, 2 agents, winner-pays-bid highest-bids-win mechanism
- values: 101 and 100.

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which algorithms are good?
- 3 winners pay their bids.

Definition (Highest-bids-win)

allocate to the feasible set of agents with highest total bid.

Example (exclusive ad space)

- 1 item, 2 agents, winner-pays-bid highest-bids-win mechanism
- **values:** 101 and 100.
- **equilibrium bids:*** 100.01 and 100

* approximate Nash equilibrium.

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which algorithms are good?
- 3 winners pay their bids.

Definition (Highest-bids-win)

allocate to the feasible set of agents with highest total bid.

Example (exclusive ad space)

- 1 item, 2 agents, winner-pays-bid highest-bids-win mechanism
 - **values:** 101 and 100.
 - **equilibrium bids:*** 100.01 and 100
 - **outcome:** 101 wins at 100.01; **welfare:** 101; **optimal welfare:** 101.
- * approximate Nash equilibrium.

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which algorithms are good?
- 3 winners pay their bids.

Definition (Highest-bids-win)

allocate to the feasible set of agents with highest total bid.

Example (exclusive or shared ad space [cf., Dütting, Kesselheim '15])

- 1 exclusive item or 3 shared items.

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which bid wins

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which algorithms are good?
- 3 winners pay their bids.

Definition (Highest-bids-win)

allocate to the feasible set of agents with highest total bid.

Example (exclusive or shared ad space [cf., Dütting, Kesselheim '15])

- 1 exclusive item or 3 shared items.
- 2 exclusive agents, **values:** 101 and 100.
- 3 shared agents, **values:** 99, 98, 97.
- **equilibrium bids:*** 100.01, 100.00, 0, 0, 0.

Winner-pays-bid Mechanisms

Definition (Winner-pays-bid Mechanism)

- 1 solicit bids.
- 2 run allocation algorithm. ← which algorithms are good?
- 3 winners pay their bids.

Definition (Highest-bids-win)

allocate to the feasible set of agents with highest total bid.

Example (exclusive or shared ad space [cf., Dütting, Kesselheim '15])

- 1 exclusive item or 3 shared items.
- 2 exclusive agents, **values:** 101 and 100.
- 3 shared agents, **values:** 99, 98, 97.
- **equilibrium bids:*** 100.01, 100.00, 0, 0, 0.
- **outcome:** 101 wins at 100.01; **welfare:** 101; **optimal welfare:** 294.

Basic Challenge (for Winner-pays-bid Mechanism Design)

- agent strategies are non-trivial.
- equilibria can be very bad.

Basic Challenge (for Winner-pays-bid Mechanism Design)

- agent strategies are non-trivial.
- equilibria can be very bad.

Our Solution: Bidding Dashboards (estimated price-quantity curve)

- agents can easily optimize bids for dashboard.
- design mechanism that is aware of dashboard.

Basic Challenge (for Winner-pays-bid Mechanism Design)

- agent strategies are non-trivial.
- equilibria can be very bad.

Our Solution: Bidding Dashboards (estimated price-quantity curve)

- agents can easily optimize bids for dashboard. **what dashboard?**
- design mechanism that is aware of dashboard. **what mechanism?**

Informal Definition (Dashboard Mechanism)

The **dashboard mechanism** is (for given dashboard and allocation alg):

- 1 publish **dashboard** (estimated bid-allocation rule)
- 2 solicit bids.
- 3 **infer** values for bids (as best response to dashboard)
- 4 execute desired **allocation algorithm** on values.
- 5 charge winners their bids.

Informal Definition (Dashboard Mechanism)

The **dashboard mechanism** is (for given dashboard and allocation alg):

- 1 publish **dashboard** (estimated bid-allocation rule)
- 2 solicit bids.
- 3 **infer** values for bids (as best response to dashboard)
- 4 execute desired **allocation algorithm** on values.
- 5 charge winners their bids.

Main Question

Informal Definition (Dashboard Mechanism)

The **dashboard mechanism** is (for given dashboard and allocation alg):

- 1 publish **dashboard** (estimated bid-allocation rule)
- 2 solicit bids.
- 3 **infer** values for bids (as best response to dashboard)
- 4 execute desired **allocation algorithm** on values.
- 5 charge winners their bids.

Main Question

This seems wrong!

Dashboard Mechanisms

Informal Definition (Dashboard Mechanism)

The **dashboard mechanism** is (for given dashboard and allocation alg):

- 1 publish **dashboard** (estimated bid-allocation rule)
- 2 solicit bids.
- 3 **infer** values for bids (as best response to dashboard)
- 4 execute desired **allocation algorithm** on values.
- 5 charge winners their bids.

Main Question

This seems wrong! For what dashboards is dashboard mechanism “right”?

Dashboard Mechanisms

Informal Definition (Dashboard Mechanism)

The **dashboard mechanism** is (for given dashboard and allocation alg):

- 1 publish **dashboard** (estimated bid-allocation rule)
- 2 solicit bids.
- 3 **infer** values for bids (as best response to dashboard)
- 4 execute desired **allocation algorithm** on values.
- 5 charge winners their bids.

Main Question

This seems wrong! For what dashboards is dashboard mechanism “right”?

Note: Allocation is correct if agents follow dashboard. Issue: payments?

Overview of Results

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Overview of Results:

- 1 construct dashboards.

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Overview of Results:

- 1 construct dashboards.
- 2 static environment, static values

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Overview of Results:

- 1 construct dashboards.
- 2 static environment, static values
 \Rightarrow following dashboard converges to equilibrium.

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Overview of Results:

- ① construct dashboards.
- ② static environment, static values
 \Rightarrow following dashboard converges to equilibrium.
- ③ dynamic environment, static value

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Overview of Results:

- ① construct dashboards.
- ② static environment, static values
 \Rightarrow following dashboard converges to equilibrium.
- ③ dynamic environment, static value
 \Rightarrow following dashboard: average payments converge to correct payments

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Overview of Results:

- ① construct dashboards.
- ② static environment, static values
 \Rightarrow following dashboard converges to equilibrium.
- ③ dynamic environment, static value
 \Rightarrow following dashboard: average payments converge to correct payments
- ④ dynamic environment, dynamic values

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Overview of Results:

- ① construct dashboards.
- ② static environment, static values
 \Rightarrow following dashboard converges to equilibrium.
- ③ dynamic environment, static value
 \Rightarrow following dashboard: average payments converge to correct payments
- ④ dynamic environment, dynamic values
 \Rightarrow approx. strategic equivalence to sequential truthful mechanism.

Outline

- 1 Introduction and Motivation
- 2 Single-agent Winner-pays-bid Mechanisms
- 3 Dashboard Construction and Analysis
- 4 Single-call Dashboard Mechanisms
- 5 Discussion and Directions

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Overview of Results:

- ① construct dashboards.
- ② static environment, static values
 \Rightarrow following dashboard converges to equilibrium.
- ③ dynamic environment, static value
 \Rightarrow following dashboard: average payments converge to correct payments
- ④ dynamic environment, dynamic values
 \Rightarrow approx. strategic equivalence to sequential truthful mechanism.

Informal Definition (Dashboard Mechanism)

The **dashboard mechanism** is (for given dashboard and allocation alg):

- 1 publish **dashboard** (estimated bid-allocation rule)
- 2 solicit bids.
- 3 **infer** values for bids (as best response to dashboard)
- 4 execute desired **allocation algorithm** on values.
- 5 charge winners their bids.

Main Question

This seems wrong! For what dashboards is dashboard mechanism “right”?

Dashboard Mechanisms

Informal Definition (Dashboard Mechanism)

The **dashboard mechanism** is (for given dashboard and allocation alg):

- 1 publish **dashboard** (estimated bid-allocation rule)
- 2 solicit bids.
- 3 **infer** values for bids (as best response to dashboard)
- 4 execute desired **allocation algorithm** on values.
- 5 charge winners their bids.

Main Question

This seems wrong! For what dashboards is dashboard mechanism “right”?

Note: Allocation is correct if agents follow dashboard. Issue: payments?

Overview of Results

Recall Goal: sequential non-truthful mech. \approx sequential truthful mech.

Outline

- 1 Introduction and Motivation
- 2 Single-agent Winner-pays-bid Mechanisms
- 3 Dashboard Construction and Analysis
- 4 Single-call Dashboard Mechanisms
- 5 Discussion and Directions

Single-agent: Characterization and Revelation Principle

Notation: single agent with value v

Single-agent: Characterization and Revelation Principle

Notation: single agent with value v

- truthful mechanism (x, p) defines: allocation rule x ; payment rule p .

Single-agent: Characterization and Revelation Principle

Notation: single agent with value v

- truthful mechanism (x, p) defines: allocation rule x ; payment rule p .
- winner-pays-bid mechanism (\tilde{x}, \tilde{p}) defines:
bid allocation rule \tilde{x} ; bid payment rule \tilde{p} with $\tilde{p}(b) = b \tilde{x}(b)$.

Single-agent: Characterization and Revelation Principle

Notation: single agent with value v

- truthful mechanism (x, p) defines: allocation rule x ; payment rule p .
- winner-pays-bid mechanism (\tilde{x}, \tilde{p}) defines:
bid allocation rule \tilde{x} ; bid payment rule \tilde{p} with $\tilde{p}(b) = b \tilde{x}(b)$.
- bid strategy b maps values v to bids b .

Single-agent: Characterization and Revelation Principle

Notation: single agent with value v

- truthful mechanism (x, p) defines: allocation rule x ; payment rule p .
- winner-pays-bid mechanism (\tilde{x}, \tilde{p}) defines:
bid allocation rule \tilde{x} ; bid payment rule \tilde{p} with $\tilde{p}(b) = b \tilde{x}(b)$.
- bid strategy b maps values v to bids b .

Revelation Principle: if b is optimal for (\tilde{x}, \tilde{p}) then (x, p) with $x(v) = \tilde{x}(b(v))$ and $p(v) = \tilde{p}(b(v))$ is truthful.

Single-agent: Characterization and Revelation Principle

Notation: single agent with value v

- truthful mechanism (x, p) defines: **allocation rule** x ; **payment rule** p .
- winner-pays-bid mechanism (\tilde{x}, \tilde{p}) defines:
bid allocation rule \tilde{x} ; **bid payment rule** \tilde{p} with $\tilde{p}(b) = b \tilde{x}(b)$.
- **bid strategy** b maps values v to bids b .

Revelation Principle: if b is optimal for (\tilde{x}, \tilde{p}) then (x, p) with $x(v) = \tilde{x}(b(v))$ and $p(v) = \tilde{p}(b(v))$ is truthful.

Theorem (Myerson '81)

A single-agent mechanism (x, p) is **truthful** if and only if

- monotonicity:** allocation rule $x(\cdot)$ is monotonically non-decreasing.
- payment identity:** payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.

Single-agent: Characterization and Revelation Principle

Notation: single agent with value v

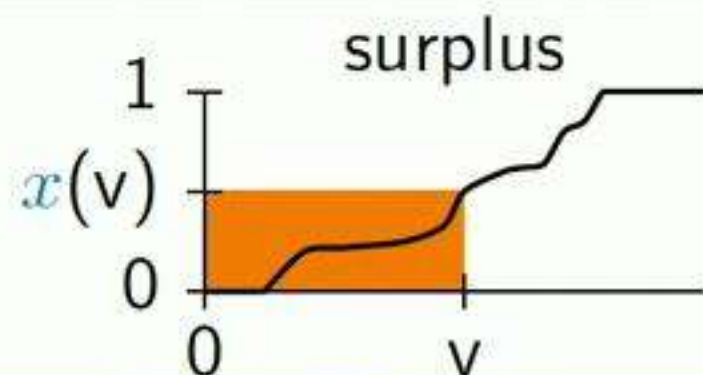
- truthful mechanism (x, p) defines: **allocation rule** x ; **payment rule** p .
- winner-pays-bid mechanism (\tilde{x}, \tilde{p}) defines:
bid allocation rule \tilde{x} ; **bid payment rule** \tilde{p} with $\tilde{p}(b) = b \tilde{x}(b)$.
- **bid strategy** b maps values v to bids b .

Revelation Principle: if b is optimal for (\tilde{x}, \tilde{p}) then (x, p) with $x(v) = \tilde{x}(b(v))$ and $p(v) = \tilde{p}(b(v))$ is truthful.

Theorem (Myerson '81)

A single-agent mechanism (x, p) is **truthful** if and only if

- monotonicity:** allocation rule $x(\cdot)$ is monotonically non-decreasing.
- payment identity:** payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.



Single-agent: Characterization and Revelation Principle

Notation: single agent with value v

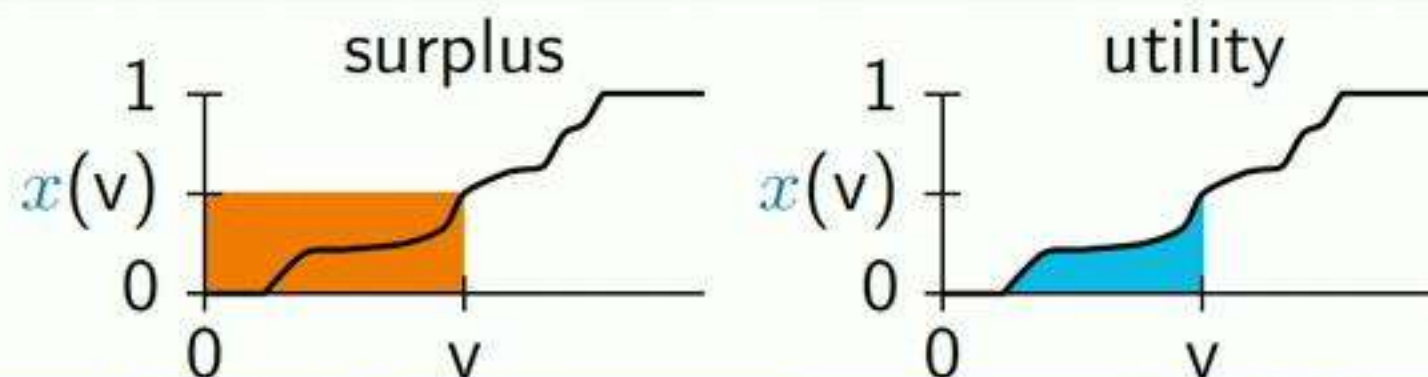
- truthful mechanism (x, p) defines: **allocation rule** x ; **payment rule** p .
- winner-pays-bid mechanism (\tilde{x}, \tilde{p}) defines:
bid allocation rule \tilde{x} ; **bid payment rule** \tilde{p} with $\tilde{p}(b) = b \tilde{x}(b)$.
- **bid strategy** b maps values v to bids b .

Revelation Principle: if b is optimal for (\tilde{x}, \tilde{p}) then (x, p) with $x(v) = \tilde{x}(b(v))$ and $p(v) = \tilde{p}(b(v))$ is truthful.

Theorem (Myerson '81)

A single-agent mechanism (x, p) is **truthful** if and only if

- monotonicity:** allocation rule $x(\cdot)$ is monotonically non-decreasing.
- payment identity:** payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.



Single-agent: Characterization and Revelation Principle

Notation: single agent with value v

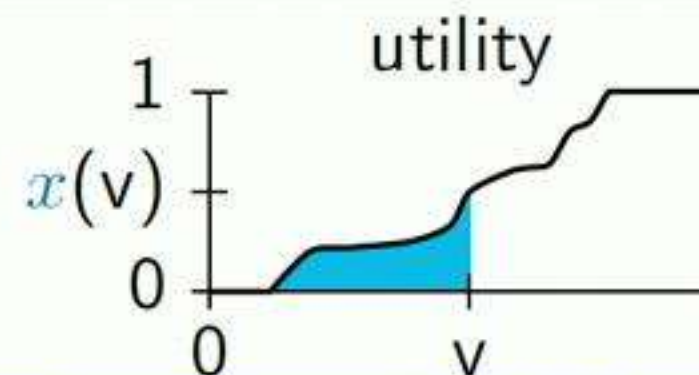
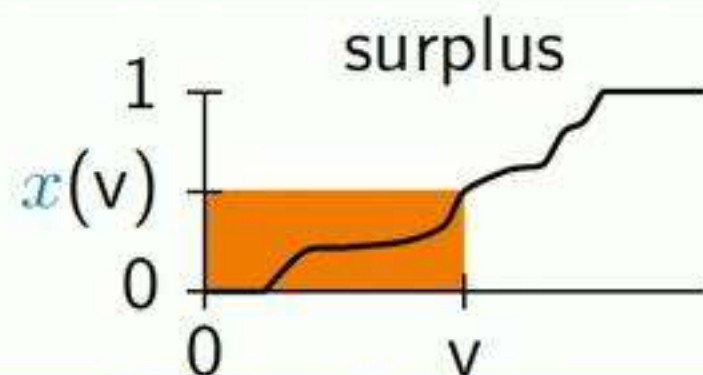
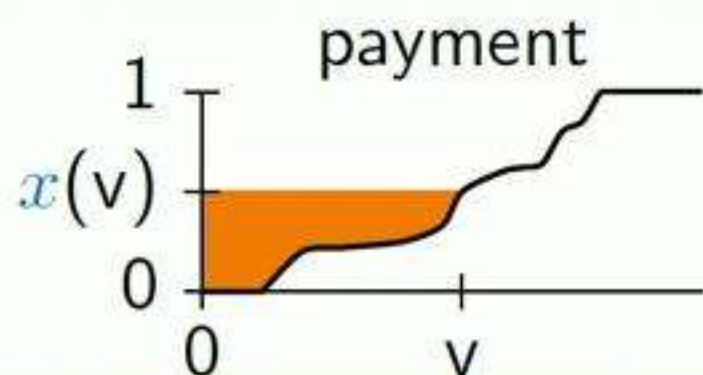
- truthful mechanism (x, p) defines: **allocation rule** x ; **payment rule** p .
- winner-pays-bid mechanism (\tilde{x}, \tilde{p}) defines:
bid allocation rule \tilde{x} ; **bid payment rule** \tilde{p} with $\tilde{p}(b) = b \tilde{x}(b)$.
- **bid strategy** b maps values v to bids b .

Revelation Principle: if b is optimal for (\tilde{x}, \tilde{p}) then (x, p) with $x(v) = \tilde{x}(b(v))$ and $p(v) = \tilde{p}(b(v))$ is truthful.

Theorem (Myerson '81)

A single-agent mechanism (x, p) is **truthful** if and only if

- monotonicity:** allocation rule $x(\cdot)$ is monotonically non-decreasing.
- payment identity:** payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.

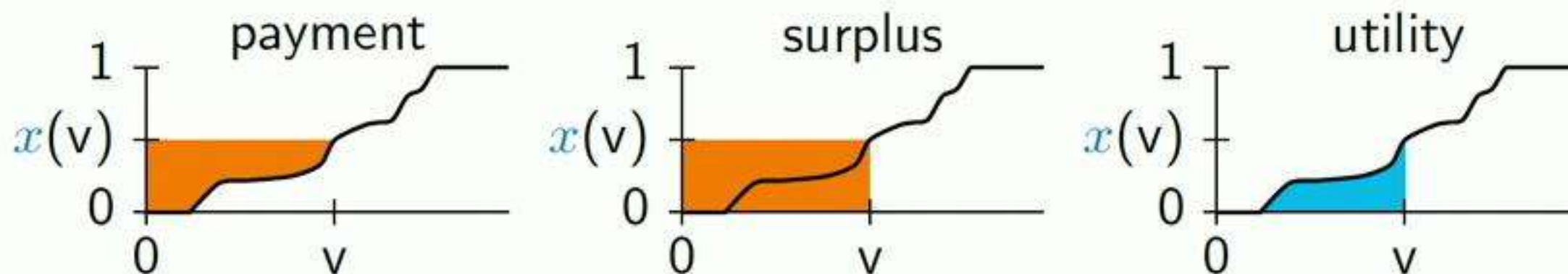


Single-agent: Truthful vs Winner-pays-bid

Theorem (Myerson '81)

A single-agent mechanism (x, p) is *truthful* if and only if

- a) *monotonicity*: allocation rule $x(\cdot)$ is monotonically non-decreasing.
- b) *payment identity*: payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.

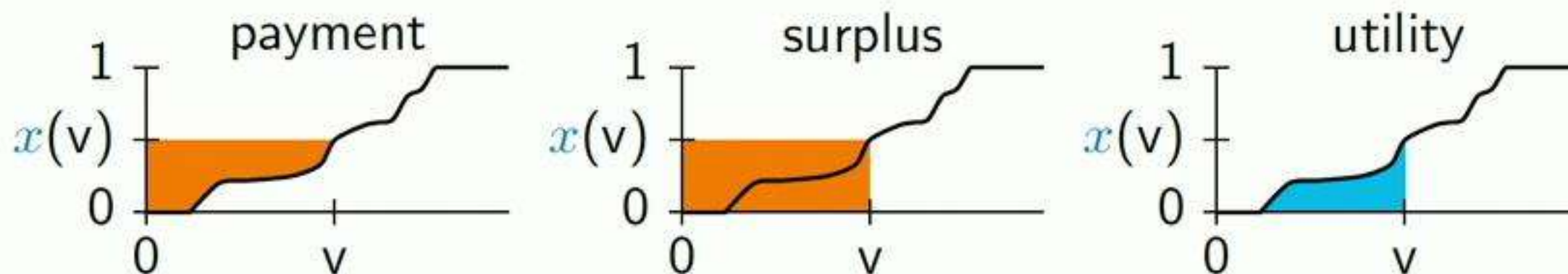


Single-agent: Truthful vs Winner-pays-bid

Theorem (Myerson '81)

A single-agent mechanism (x, p) is **truthful** if and only if

- a) **monotonicity**: allocation rule $x(\cdot)$ is monotonically non-decreasing.
- b) **payment identity**: payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.



Definition

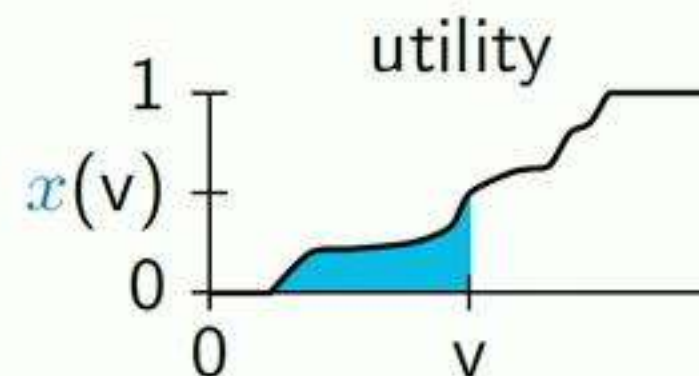
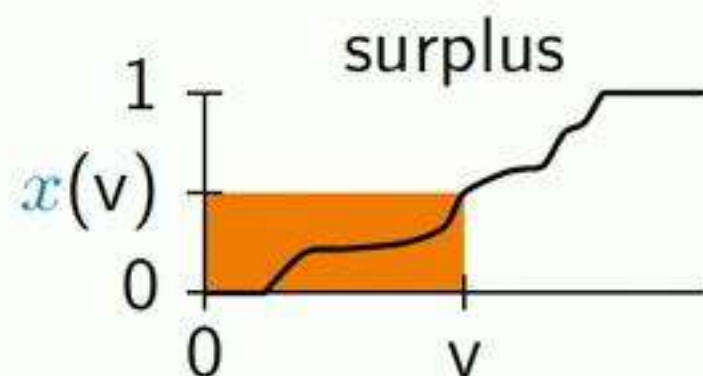
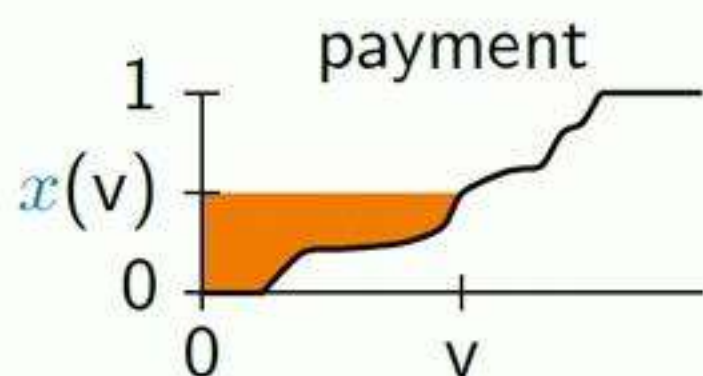
The single-agent **winner-pays-bid** implementation of x is \tilde{x} satisfying:

Single-agent: Truthful vs Winner-pays-bid

Theorem (Myerson '81)

A single-agent mechanism (x, p) is *truthful* if and only if

- a) *monotonicity*: allocation rule $x(\cdot)$ is monotonically non-decreasing.
- b) *payment identity*: payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.

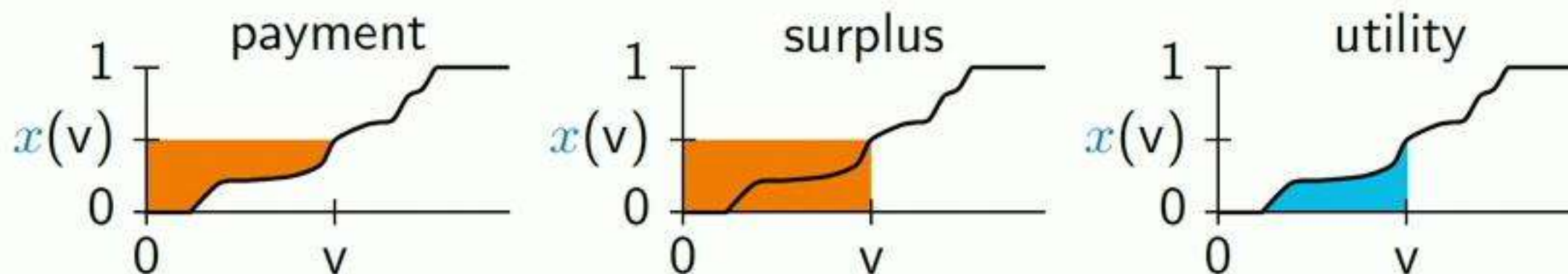


Single-agent: Truthful vs Winner-pays-bid

Theorem (Myerson '81)

A single-agent mechanism (x, p) is **truthful** if and only if

- a) **monotonicity**: allocation rule $x(\cdot)$ is monotonically non-decreasing.
- b) **payment identity**: payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.



Definition

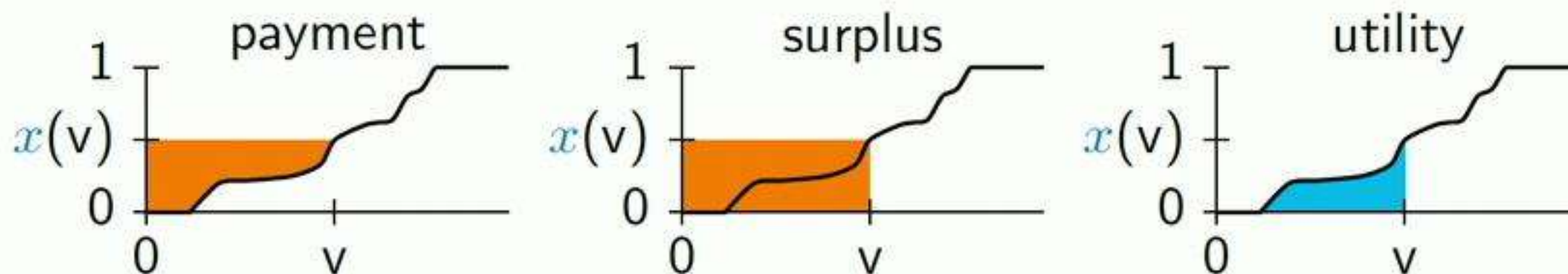
The single-agent **winner-pays-bid** implementation of x is \tilde{x} satisfying:

Single-agent: Truthful vs Winner-pays-bid

Theorem (Myerson '81)

A single-agent mechanism (x, p) is **truthful** if and only if

- a) **monotonicity**: allocation rule $x(\cdot)$ is monotonically non-decreasing.
- b) **payment identity**: payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.



Definition

The single-agent **winner-pays-bid** implementation of x is \tilde{x} satisfying:

- agent's **bid strategy**:* $b(v) = p(v)/x(v) = v - \frac{1}{x(v)} \int_0^v x(z) dz + \frac{p(0)}{x(v)}$.

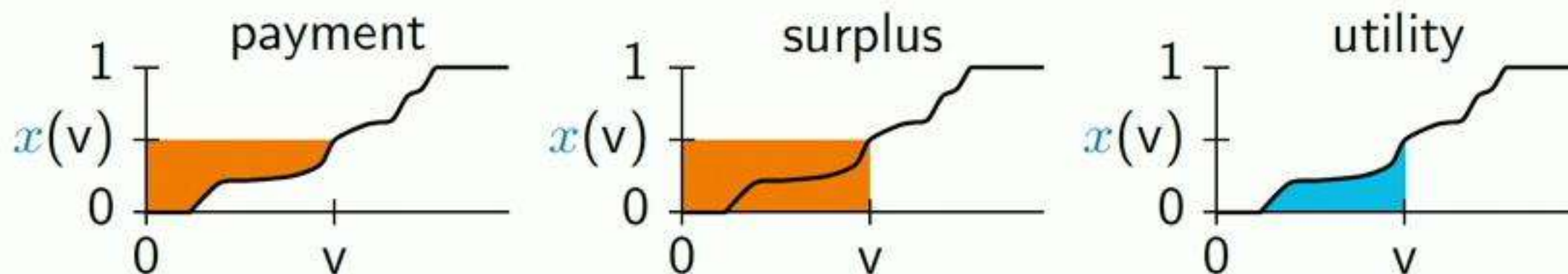
* monotone iff $p(0) \leq 0$

Single-agent: Truthful vs Winner-pays-bid

Theorem (Myerson '81)

A single-agent mechanism (x, p) is **truthful** if and only if

- a) **monotonicity**: allocation rule $x(\cdot)$ is monotonically non-decreasing.
- b) **payment identity**: payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.



Definition

The single-agent **winner-pays-bid** implementation of x is \tilde{x} satisfying:

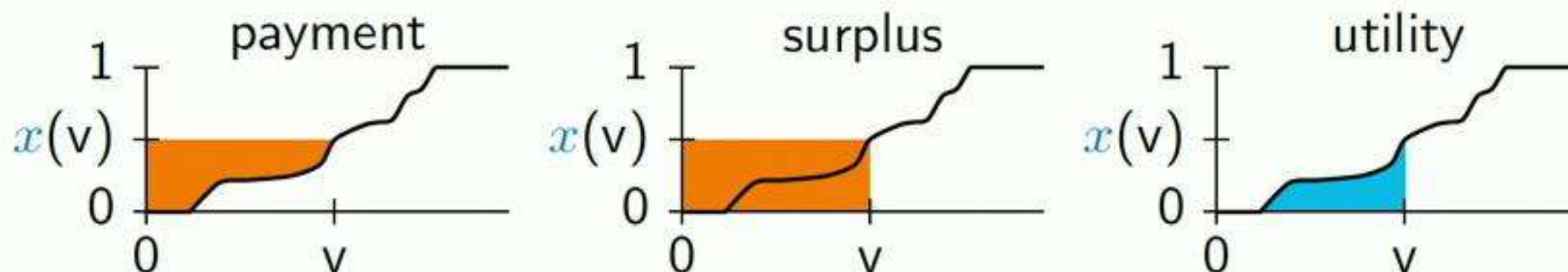
- agent's **bid strategy**:* $b(v) = p(v)/x(v) = v - \frac{1}{x(v)} \int_0^v x(z) dz + \frac{p(0)}{x(v)}$.
- **bid allocation rule**: $\tilde{x}(b) = x(b^{-1}(b))$ * monotone iff $p(0) \leq 0$

Single-agent: Truthful vs Winner-pays-bid

Theorem (Myerson '81)

A single-agent mechanism (x, p) is **truthful** if and only if

- a) **monotonicity**: allocation rule $x(\cdot)$ is monotonically non-decreasing.
- b) **payment identity**: payment rule $p_i(v) = v x(v) - \int_0^v x(z) dz + p_i(0)$.



Definition

The single-agent **winner-pays-bid** implementation of x is \tilde{x} satisfying:

- agent's **bid strategy**:* $b(v) = p(v)/x(v) = v - \frac{1}{x(v)} \int_0^v x(z) dz + \frac{p(0)}{x(v)}$.
- **bid allocation rule**: $\tilde{x}(b) = x(b^{-1}(b))$ * monotone iff $p(0) \leq 0$

Main Challenge: inverting bids in multi-agent settings.

Outline

- 1 Introduction and Motivation
- 2 Single-agent Winner-pays-bid Mechanisms
- 3 Dashboard Construction and Analysis**
- 4 Single-call Dashboard Mechanisms
- 5 Discussion and Directions

Main Ideas

- ① use historical allocation rule as dashboard.

Main Ideas

- ① use historical allocation rule as dashboard.
- ② infer agents' values from response to dashboard.

- ① use historical allocation rule as dashboard.
- ② infer agents' values from response to dashboard.
⇒ “following dashboard” converges to Nash.

Main Ideas

- ① use historical allocation rule as dashboard.
- ② infer agents' values from response to dashboard.
⇒ "following dashboard" converges to Nash.
- ③ update dashboard only when agent is allocated.

- ① use historical allocation rule as dashboard.
- ② infer agents' values from response to dashboard.
⇒ "following dashboard" converges to Nash.
- ③ update dashboard only when agent is allocated.
⇒ payments converge to truthful payments for static valued agents

Main Ideas

- ① use historical allocation rule as dashboard.
- ② infer agents' values from response to dashboard.
⇒ “following dashboard” converges to Nash.
- ③ update dashboard only when agent is allocated.
⇒ payments converge to truthful payments for static valued agents
- ④ rebalance incorrect payments in subsequent stages (\approx “add to $p(0)$ ”)

Main Ideas

- ① use historical allocation rule as dashboard.
- ② infer agents' values from response to dashboard.
⇒ "following dashboard" converges to Nash.
- ③ update dashboard only when agent is allocated.
⇒ payments converge to truthful payments for static valued agents
- ④ rebalance incorrect payments in subsequent stages (\approx "add to $p(0)$ ")
⇒ approx. strategic equivalent to sequential truthful mech.

Main Ideas

- ① use historical allocation rule as dashboard.
- ② infer agents' values from response to dashboard.
⇒ "following dashboard" converges to Nash.
- ③ update dashboard only when agent is allocated.
⇒ payments converge to truthful payments for static valued agents
- ④ rebalance incorrect payments in subsequent stages (\approx "add to $p(0)$ ")

Main Ideas

- ① use historical allocation rule as dashboard.
- ② infer agents' values from response to dashboard.
⇒ "following dashboard" converges to Nash.
- ③ update dashboard only when agent is allocated.
⇒ payments converge to truthful payments for static valued agents
- ④ rebalance incorrect payments in subsequent stages (\approx "add to $p(0)$ ")
⇒ approx. strategic equivalent to sequential truthful mech.

Dynamic model:

- dynamic iterated environment, in stage $s \in \{1, \dots, t\}$:
 - n agent valuation profile: $\mathbf{v}^{(s)} = (v_1^{(s)}, \dots, v_n^{(s)})$
 - stochastic allocation algorithm: $\mathbf{x}^{(s)} : \mathbb{R}^n \rightarrow [0, 1]^n$.
- linear utility: $u_i = \sum_{s=1}^t [v_i^{(s)} x_i^{(s)} - p_i^{(s)}]$.
- payment format: winner-pays-bid.

Dynamic model:

- dynamic iterated environment, in stage $s \in \{1, \dots, t\}$:
 - n agent valuation profile: $\mathbf{v}^{(s)} = (v_1^{(s)}, \dots, v_n^{(s)})$
 - stochastic allocation algorithm: $\mathbf{x}^{(s)} : \mathbb{R}^n \rightarrow [0, 1]^n$.
- linear utility: $u_i = \sum_{s=1}^t [v_i^{(s)} x_i^{(s)} - p_i^{(s)}]$.
- payment format: winner-pays-bid.

Static model: $\mathbf{v}^{(s)} = \mathbf{v}$ and $\mathbf{x}^{(s)} = \mathbf{x}$ for all stages s .

Dynamic model:

- dynamic iterated environment, in stage $s \in \{1, \dots, t\}$:
 - n agent valuation profile: $\mathbf{v}^{(s)} = (v_1^{(s)}, \dots, v_n^{(s)})$
 - stochastic allocation algorithm: $\mathbf{x}^{(s)} : \mathbb{R}^n \rightarrow [0, 1]^n$.
- linear utility: $u_i = \sum_{s=1}^t [v_i^{(s)} x_i^{(s)} - p_i^{(s)}]$.
- payment format: winner-pays-bid.

Static model: $\mathbf{v}^{(s)} = \mathbf{v}$ and $\mathbf{x}^{(s)} = \mathbf{x}$ for all stages s .

Goal: sequential winner-pays-bid mechanism to implement $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(t)}$.

Dynamic model:

- dynamic iterated environment, in stage $s \in \{1, \dots, t\}$:
 - n agent valuation profile: $\mathbf{v}^{(s)} = (v_1^{(s)}, \dots, v_n^{(s)})$
 - stochastic allocation algorithm: $\mathbf{x}^{(s)} : \mathbb{R}^n \rightarrow [0, 1]^n$.
- linear utility: $u_i = \sum_{s=1}^t [v_i^{(s)} x_i^{(s)} - p_i^{(s)}]$.
- payment format: winner-pays-bid.

Static model: $\mathbf{v}^{(s)} = \mathbf{v}$ and $\mathbf{x}^{(s)} = \mathbf{x}$ for all stages s .

Goal: sequential winner-pays-bid mechanism to implement $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(t)}$.

Note: for single and single stage analysis, will drop i and s .

Definition (Dashboard)

A **dashboard** $\tilde{y} : (\mathbb{R} \rightarrow [0, 1])^n$ specifies a bid allocation rule for each agent.

Definition (Dashboard)

A **dashboard** $\tilde{y} : (\mathbb{R} \rightarrow [0, 1])^n$ specifies a bid allocation rule for each agent.

Notation:

- for algorithm: x, p, \tilde{x}, b .
- for dashboard: y, q, \tilde{y}, c .

Dashboard Mechanisms

Definition (Dashboard)

A **dashboard** $\tilde{y} : (\mathbb{R} \rightarrow [0, 1])^n$ specifies a bid allocation rule for each agent.

Notation:

- for algorithm: x, p, \tilde{x}, b .
- for dashboard: y, q, \tilde{y}, c .

Definition (Dashboard Mechanism)

The **dashboard mechanism** $\tilde{x} : \mathbb{R}^n \rightarrow [0, 1]^n$ for dashboard \tilde{y} , algo x is:

- 1 **publish** dashboard \tilde{y} ; **solicit** bids \mathbf{b} .
- 2 **infer** values $\hat{\mathbf{v}}$ as $\hat{v}_i = c_i^{-1}(b_i)$ for \tilde{y}_i .
- 3 **run** x on $\hat{\mathbf{v}}$ to get allocation $\mathbf{x} \sim x(\hat{\mathbf{v}})$ and payments \mathbf{p} as $p_i = b_i x_i$.

Definition (Last-stage Dashboard)

In stage s :

- last stage inferred values: $\hat{\mathbf{v}}^{(s-1)}$
- last stage allocation rules: $\mathbf{y}^{(s)}$ as $y_i^{(s)}(z) = x_i^{(s-1)}(z, \hat{\mathbf{v}}_{-i}^{(s-1)})$.
- dashboard: $\tilde{\mathbf{y}}^{(s)}$ with $\tilde{y}_i^{(s)}$ corresponding to $y_i^{(s)}$.

Last-stage Dashboard

Definition (Last-stage Dashboard)

In stage s :

- last stage inferred values: $\hat{\mathbf{v}}^{(s-1)}$
- last stage allocation rules: $\mathbf{y}^{(s)}$ as $y_i^{(s)}(z) = x_i^{(s-1)}(z, \hat{\mathbf{v}}_{-i}^{(s-1)})$.
- dashboard: $\tilde{\mathbf{y}}^{(s)}$ with $\tilde{y}_i^{(s)}$ corresponding to $y_i^{(s)}$.

Theorem (Static Analysis)

For last-stage dashboard and static environment, the action profile for “follow the dashboard” strategy converges (in two rounds) to Nash equilibrium of stage game.

Last-winning-stage Dashboard

Definition (Last-winning-stage Dashboard)

dashboard is allocation rule from last winning stage.

Last-winning-stage Dashboard

Definition (Last-winning-stage Dashboard)

dashboard is allocation rule from last winning stage.

Theorem (Mixed Analysis)

For last-winning-stage dashboard and dynamic environment, the average outcome from following the dashboard with static value converges to the average outcome of the sequential truthful mechanism.

Last-winning-stage Dashboard

Definition (Last-winning-stage Dashboard)

dashboard is allocation rule from last winning stage.

Theorem (Mixed Analysis)

For last-winning-stage dashboard and dynamic environment, the average outcome from following the dashboard with static value converges to the average outcome of the sequential truthful mechanism.

Proof.

- fix value v .

Last-winning-stage Dashboard

Definition (Last-winning-stage Dashboard)

dashboard is allocation rule from last winning stage.

Theorem (Mixed Analysis)

For last-winning-stage dashboard and dynamic environment, the average outcome from following the dashboard with static value converges to the average outcome of the sequential truthful mechanism.

Proof.

- fix value v .
- consider successive winning stages $s_{k-1} < s_k$:

Last-winning-stage Dashboard

Definition (Last-winning-stage Dashboard)

dashboard is allocation rule from last winning stage.

Theorem (Mixed Analysis)

For last-winning-stage dashboard and dynamic environment, the average outcome from following the dashboard with static value converges to the average outcome of the sequential truthful mechanism.

Proof.

- fix value v .
- consider successive winning stages $s_{k-1} < s_k$:
 - \Rightarrow bid in stage s_k is correct for stage s_{k-1} , i.e., $c^{(s_k)}(v) = b^{(s_{k-1})}(v)$.

Last-winning-stage Dashboard

Definition (Last-winning-stage Dashboard)

dashboard is allocation rule from last winning stage.

Theorem (Mixed Analysis)

For last-winning-stage dashboard and dynamic environment, the average outcome from following the dashboard with static value converges to the average outcome of the sequential truthful mechanism.

Proof.

- fix value v .
- consider successive winning stages $s_{k-1} < s_k$:
 - \Rightarrow bid in stage s_k is correct for stage s_{k-1} , i.e., $c^{(s_k)}(v) = b^{(s_{k-1})}(v)$.
- payment difference summed over τ winning stages:

Last-winning-stage Dashboard

Definition (Last-winning-stage Dashboard)

dashboard is allocation rule from last winning stage.

Theorem (Mixed Analysis)

For last-winning-stage dashboard and dynamic environment, the average outcome from following the dashboard with static value converges to the average outcome of the sequential truthful mechanism.

Proof.

- fix value v .
- consider successive winning stages $s_{k-1} < s_k$:
 - \Rightarrow bid in stage s_k is correct for stage s_{k-1} , i.e., $c^{(s_k)}(v) = b^{(s_{k-1})}(v)$.
- payment difference summed over τ winning stages:
 - \Rightarrow total difference from truthful is: $c^{(s_1)}(v) - b^{(s_\tau)}(v)$

Intuition for Payment Rebalancing

Goal: dashboard robust to changing values and environment.

Last-winning-stage Dashboard

Definition (Last-winning-stage Dashboard)

dashboard is allocation rule from last winning stage.

Theorem (Mixed Analysis)

For last-winning-stage dashboard and dynamic environment, the average outcome from following the dashboard with static value converges to the average outcome of the sequential truthful mechanism.

Proof.

- fix value v .
- consider successive winning stages $s_{k-1} < s_k$:
 - \Rightarrow bid in stage s_k is correct for stage s_{k-1} , i.e., $c^{(s_k)}(v) = b^{(s_{k-1})}(v)$.
- payment difference summed over τ winning stages:
 - \Rightarrow total difference from truthful is: $c^{(s_1)}(v) - b^{(s_\tau)}(v) \leq v$. □

Intuition for Payment Rebalancing

Goal: dashboard robust to changing values and environment.

Intuition: For empirical values dashboard:

- When $x_i^{(s)} \neq x_i^{(s-1)}$:
 - allocation is correct (for estimated value $\hat{v}_i^{(s)}$)
 - payment is incorrect.
- calculate error in payment (positive or negative), add to **balance**.
- future dashboards include lump sum payment to reduce balance.

Payment Rebalancing Dashboards

Fix a stage s and agent i (dropping superscript and subscript):

- **inferred value:** \hat{v} ; **implemented allocation:** $x(\hat{v})$.

Payment Rebalancing Dashboards

Fix a stage s and agent i (dropping superscript and subscript):

- inferred value: \hat{v} ; implemented allocation: $x(\hat{v})$.
- actual bid: $c(\hat{v})$; desired bid: $b(\hat{v})$

Payment Rebalancing Dashboards

Fix a stage s and agent i (dropping superscript and subscript):

- inferred value: \hat{v} ; implemented allocation: $x(\hat{v})$.
- actual bid: $c(\hat{v})$; desired bid: $b(\hat{v})$
- payment residual: $[b(\hat{v}) - c(\hat{v})] x(\hat{v})$.

Payment Rebalancing Dashboards

Fix a stage s and agent i (dropping superscript and subscript):

- inferred value: \hat{v} ; implemented allocation: $x(\hat{v})$.
- actual bid: $c(\hat{v})$; desired bid: $b(\hat{v})$
- payment residual: $[b(\hat{v}) - c(\hat{v})] x(\hat{v})$.

Approach

Payment Rebalancing Dashboards

Fix a stage s and agent i (dropping superscript and subscript):

- inferred value: \hat{v} ; implemented allocation: $x(\hat{v})$.
- actual bid: $c(\hat{v})$; desired bid: $b(\hat{v})$
- payment residual: $[b(\hat{v}) - c(\hat{v})] x(\hat{v})$.

Approach

- add payment residual to balance L

Payment Rebalancing Dashboards

Fix a stage s and agent i (dropping superscript and subscript):

- **inferred value:** \hat{v} ; **implemented allocation:** $x(\hat{v})$.
- **actual bid:** $c(\hat{v})$; **desired bid:** $b(\hat{v})$
- **payment residual:** $[b(\hat{v}) - c(\hat{v})] x(\hat{v})$.

Approach

- add payment residual to balance L
- adjust dashboard payment rule to partially resolve balance as $q(0)$.

Payment Rebalancing Dashboards

Fix a stage s and agent i (dropping superscript and subscript):

- **inferred value:** \hat{v} ; **implemented allocation:** $x(\hat{v})$.
- **actual bid:** $c(\hat{v})$; **desired bid:** $b(\hat{v})$
- **payment residual:** $[b(\hat{v}) - c(\hat{v})] x(\hat{v})$.

Approach

- add payment residual to balance L
- adjust dashboard payment rule to partially resolve balance as $q(0)$.
- (cannot actually add $q(0) > 0$, but can approximate, details omitted)

Payment Rebalancing Dashboards

Fix a stage s and agent i (dropping superscript and subscript):

- **inferred value:** \hat{v} ; **implemented allocation:** $x(\hat{v})$.
- **actual bid:** $c(\hat{v})$; **desired bid:** $b(\hat{v})$
- **payment residual:** $[b(\hat{v}) - c(\hat{v})] x(\hat{v})$.

Approach

- add payment residual to balance L
- adjust dashboard payment rule to partially resolve balance as $q(0)$.
- (cannot actually add $q(0) > 0$, but can approximate, details omitted)

Definition (Rebalancing Dashboard)

The **rebalancing dashboard** for dashboard \tilde{y} , rebalancing rate $\eta \in (0, 1]$, and outstanding balance L is \tilde{y}^\dagger for payment rule and bid strategy

$$q^\dagger(v) = q(v) + L\eta \qquad c^\dagger(v) = c(v) + L\eta/y(v).$$

Rebalancing Dashboard Analysis

Recall Definition (Rebalancing Dashboard)

The **rebalancing dashboard** for dashboard \tilde{y} , rebalancing rate $\eta \in (0, 1]$, and outstanding balance L is \tilde{y}^\dagger for payment rule and bid strategy

$$q^\dagger(v) = q(v) + L\eta \qquad c^\dagger(v) = c(v) + L\eta/y(v).$$

Rebalancing Dashboard Analysis

Recall Definition (Rebalancing Dashboard)

The **rebalancing dashboard** for dashboard \tilde{y} , rebalancing rate $\eta \in (0, 1]$, and outstanding balance L is \tilde{y}^\dagger for payment rule and bid strategy

$$q^\dagger(v) = q(v) + L\eta \qquad c^\dagger(v) = c(v) + L\eta/y(v).$$

Lemmas

- The payment residual for inferred value \hat{v} is in $[-\hat{v}, \hat{v}]$.
- For $\eta \leq y(0)$, the balance resolved in winning stage is in $[L\eta, L]$.
- Unresolved payment residual after k winning stages $\leq (1 - \eta)^k \hat{v}$.

Rebalancing Dashboard Analysis

Recall Definition (Rebalancing Dashboard)

The **rebalancing dashboard** for dashboard \tilde{y} , rebalancing rate $\eta \in (0, 1]$, and outstanding balance L is \tilde{y}^\dagger for payment rule and bid strategy

$$q^\dagger(v) = q(v) + L\eta \qquad c^\dagger(v) = c(v) + L\eta/y(v).$$

Lemmas

- The payment residual for inferred value \hat{v} is in $[-\hat{v}, \hat{v}]$.
- For $\eta \leq y(0)$, the balance resolved in winning stage is in $[L\eta, L]$.
- Unresolved payment residual after k winning stages $\leq (1 - \eta)^k \hat{v}$.

Theorem

The **unresolved balance** at any stage t is at most $\frac{\bar{v}}{\eta}$ (for values $\leq \bar{v}$).

Rebalancing Dashboard Analysis

Recall Definition (Rebalancing Dashboard)

The **rebalancing dashboard** for dashboard \tilde{y} , rebalancing rate $\eta \in (0, 1]$, and outstanding balance L is \tilde{y}^\dagger for payment rule and bid strategy

$$q^\dagger(v) = q(v) + L\eta \qquad c^\dagger(v) = c(v) + L\eta/y(v).$$

Lemmas

- The payment residual for inferred value \hat{v} is in $[-\hat{v}, \hat{v}]$.
- For $\eta \leq y(0)$, the balance resolved in winning stage is in $[L\eta, L]$.
- Unresolved payment residual after k winning stages $\leq (1 - \eta)^k \hat{v}$.

Theorem

The **unresolved balance** at any stage t is at most $\frac{\bar{v}}{\eta}$ (for values $\leq \bar{v}$).

Corollary (Dynamic Analysis)

For payment rebalancing dashboard and dynamic environment, seq. dashboard mech. is $\frac{\bar{v}}{\eta t}$ -approx. strat. equiv. to seq. truthful mech.

Outline

- 1 Introduction and Motivation
- 2 Single-agent Winner-pays-bid Mechanisms
- 3 Dashboard Construction and Analysis
- 4 Single-call Dashboard Mechanisms**
- 5 Discussion and Directions

Definition (Single-call Model)

Only access to algorithm x is by implementing its outcome, i.e., $x \sim x(\hat{v})$.

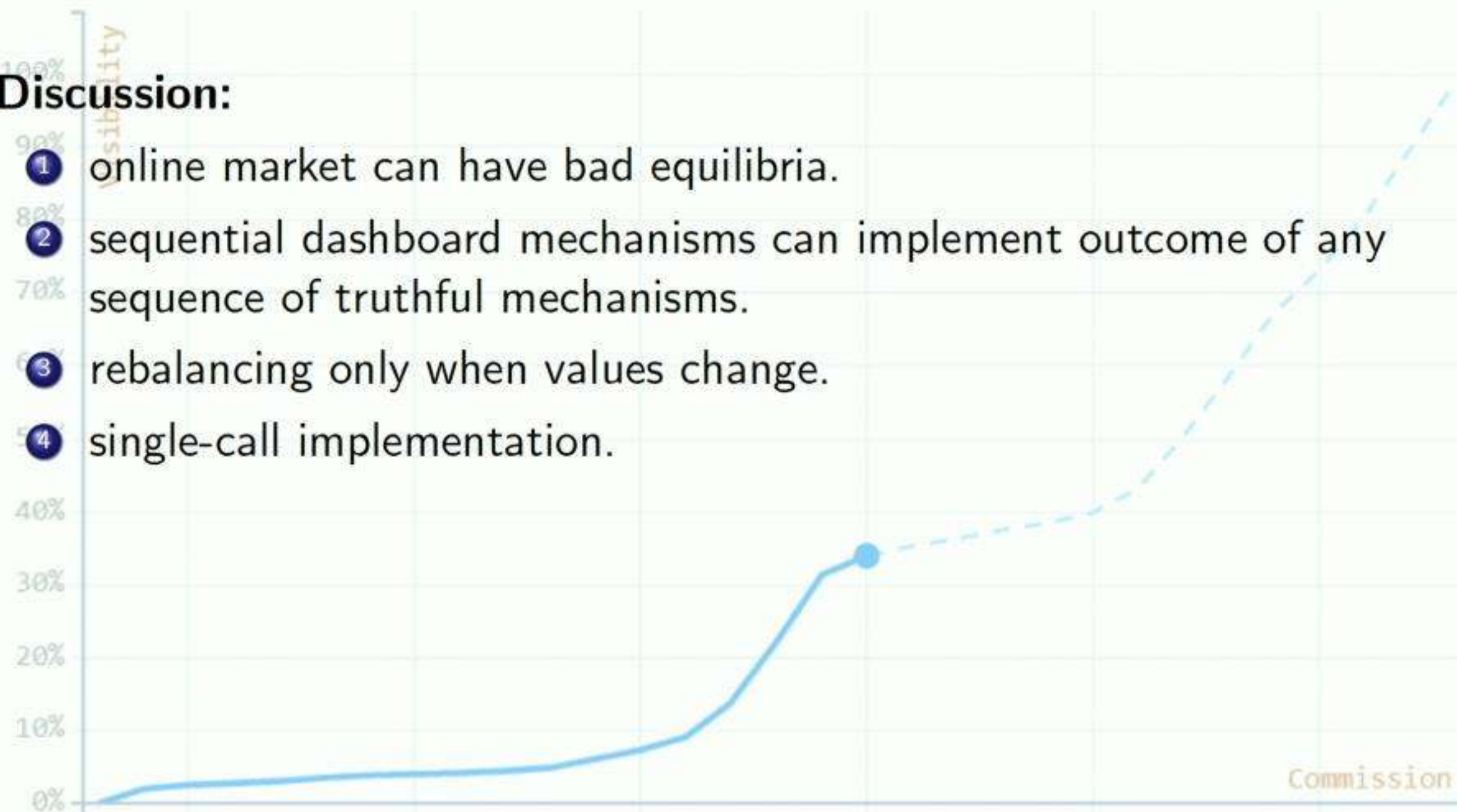
Recall Motivation: online markets:

- short-lived users matched to long-lived agents.
- matching algorithm, e.g.:
 - marketplace prioritize agents
 - users select agents

Discussion & Directions

Discussion:

- 1 online market can have bad equilibria.
- 2 sequential dashboard mechanisms can implement outcome of any sequence of truthful mechanisms.
- 3 rebalancing only when values change.
- 4 single-call implementation.

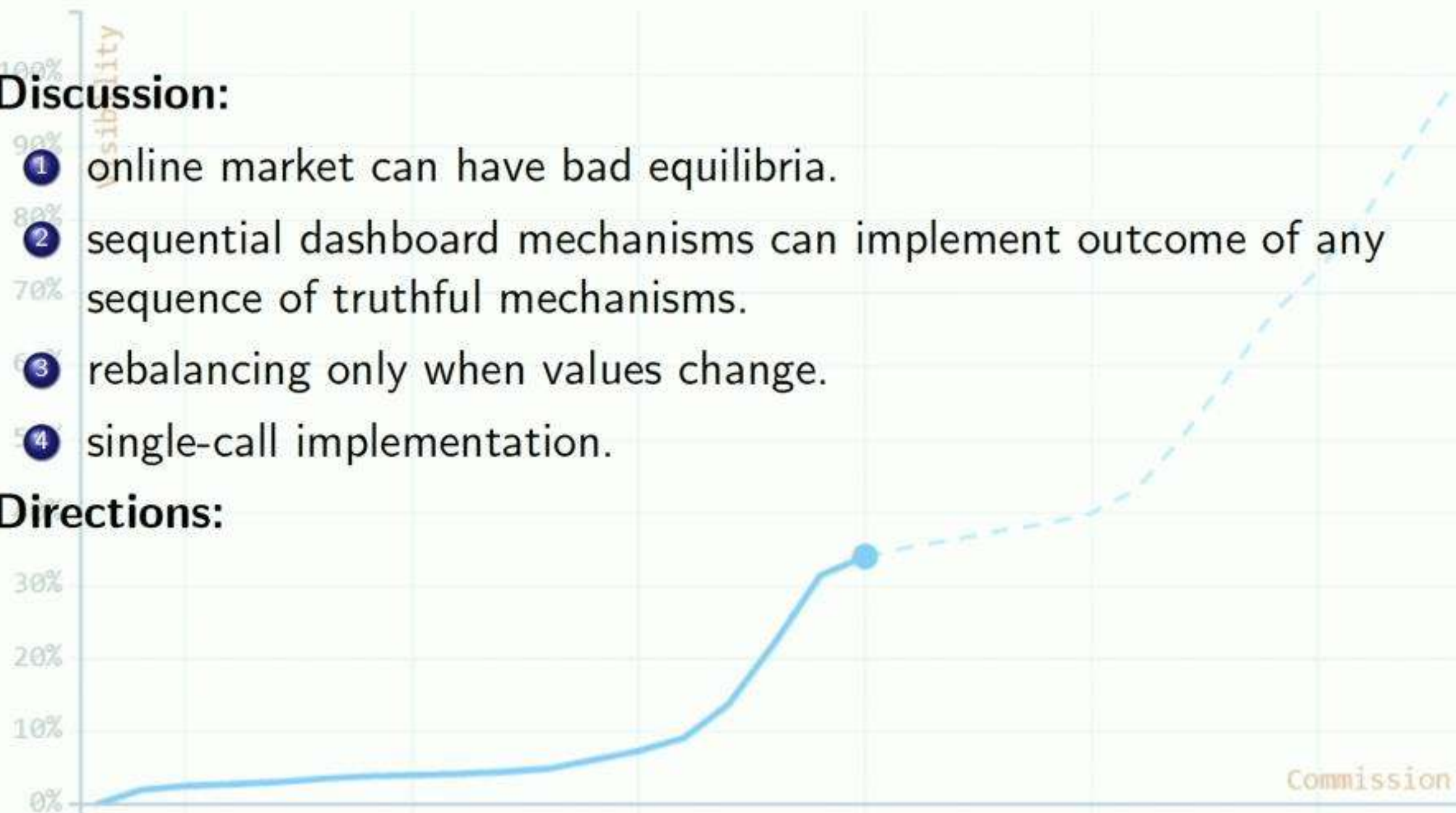


Discussion & Directions

Discussion:

- 1 online market can have bad equilibria.
- 2 sequential dashboard mechanisms can implement outcome of any sequence of truthful mechanisms.
- 3 rebalancing only when values change.
- 4 single-call implementation.

Directions:



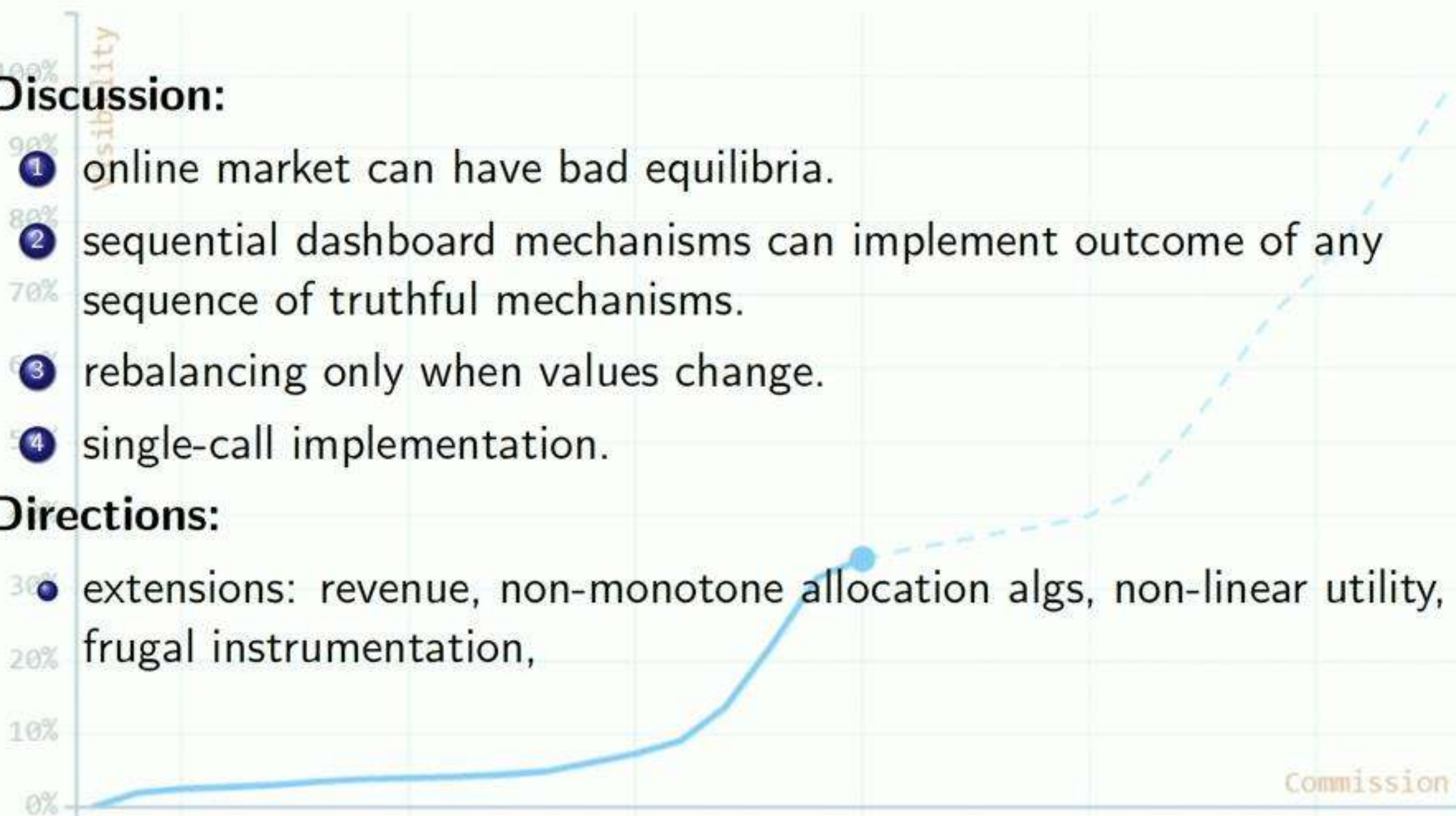
Discussion & Directions

Discussion:

- 1 online market can have bad equilibria.
- 2 sequential dashboard mechanisms can implement outcome of any sequence of truthful mechanisms.
- 3 rebalancing only when values change.
- 4 single-call implementation.

Directions:

- 1 extensions: revenue, non-monotone allocation algs, non-linear utility, frugal instrumentation,



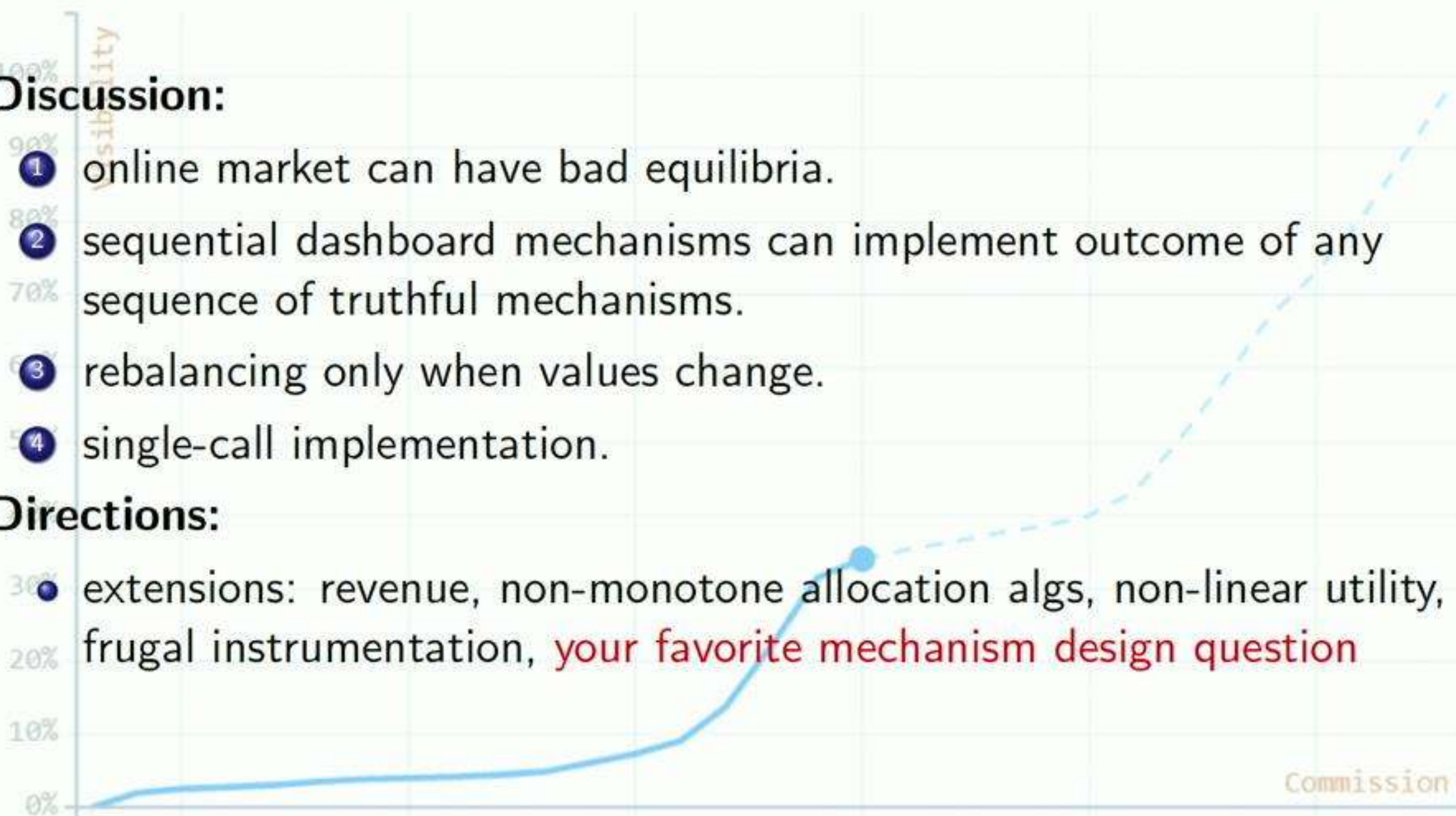
Discussion & Directions

Discussion:

- 1 online market can have bad equilibria.
- 2 sequential dashboard mechanisms can implement outcome of any sequence of truthful mechanisms.
- 3 rebalancing only when values change.
- 4 single-call implementation.

Directions:

- 1 extensions: revenue, non-monotone allocation algs, non-linear utility, frugal instrumentation, **your favorite mechanism design question**



Discussion & Directions

Discussion:

- 1 online market can have bad equilibria.
- 2 sequential dashboard mechanisms can implement outcome of any sequence of truthful mechanisms.
- 3 rebalancing only when values change.
- 4 single-call implementation.

Directions:

- 30% ● extensions: revenue, non-monotone allocation algs, non-linear utility, frugal instrumentation, **your favorite mechanism design question**
- 16% ● broad theory for non-truthful mechanism design. [cf. Hartline, Taggart '19]

