Dynamic Car Dispatching and Pricing: Revenue and Fairness for Ridesharing Platforms

Zishuo Zhao[1], Xi Chen[2], Xuefeng Zhang[1], Yuan Zhou[1]

University of Illinois Urbana-Champaign[1]
New York University[2]

- Ridesharing:
  A modern topic of sharing economy, drawing wide concern especially in COVID times.

- Ridesharing:
  A modern topic of sharing economy, drawing wide concern especially in COVID times.
- Shortcomings of existing work: at least one of

- Ridesharing:
  A modern topic of sharing economy, drawing wide concern especially in COVID times.
- Shortcomings of existing work: at least one of
  - Myopic, only considering current orders but not incoming ones. (mostly as combinatorial optimization algorithms)

- Ridesharing:
  A modern topic of sharing economy, drawing wide concern especially in COVID times.
- Shortcomings of existing work: at least one of

  - Myopic, only considering current orders but not incoming ones. (mostly as combinatorial optimization algorithms)
  - Only computing centralized plans, without ensuring drivers would be willing to follow the plan.

- Ridesharing:
  A modern topic of sharing economy, drawing wide concern especially in COVID times.
- Shortcomings of existing work: at least one of

  - Myopic, only considering current orders but not incoming ones. (mostly as combinatorial optimization algorithms)
  - Only computing centralized plans, without ensuring drivers would be willing to follow the plan.
  - Using over-simplified models (equidistant, specified graphs, etc)

- Ridesharing:
  A modern topic of sharing economy, drawing wide concern especially in COVID times.
- Shortcomings of existing work: at least one of

  - Myopic, only considering current orders but not incoming ones. (mostly as combinatorial optimization algorithms)
  - Only computing centralized plans, without ensuring drivers would be willing to follow the plan.
  - Using over-simplified models (equidistant, specified graphs, etc)
  - No revenue or fairness guarantees. (Will companies use and will drivers and riders be satisfied?)

Our three-fold contribution:

Our three-fold contribution:

- We propose a versatile generalized network flow model for the task, with theoretical guarantees on optimal revenue.

Our three-fold contribution:

- We propose a versatile generalized network flow model for the task, with theoretical guarantees on optimal revenue.
- We propose a novel two-phase pricing mechanism decoupling prices on drivers' and riders' sides to meet misaligned interests and ensure fairness.

Our three-fold contribution:

- We propose a versatile generalized network flow model for the task, with theoretical guarantees on optimal revenue.
- We propose a novel two-phase pricing mechanism decoupling prices on drivers' and riders' sides to meet misaligned interests and ensure fairness.
- We consider the stochastic nature of ridesharing orders and perform online learning on incomplete information to balance the exploration-exploitation trade-off.

Non-linear network flow model:

Basic example:

Non-linear network flow model:

- Each edge corresponding to a spatiotemporal path.

Basic example:

Phase 1: Max-Revenue Car Dispatching

Non-linear network flow model

Non-linear network flow model:

- Each edge corresponding to a spatiotemporal path.
- Edge reward function corresponding to revenue with k dispatched drivers.

Basic example:

Non-linear network flow model:

- Each edge corresponding to a spatiotemporal path.
- Edge reward function corresponding to revenue with k dispatched drivers.

Basic example:

- An edge (A, B), cost: $c(A, B) = 2$.

Non-linear network flow model:

- Each edge corresponding to a spatiotemporal path.
- Edge reward function corresponding to revenue with k dispatched drivers.

Basic example:

- An edge (A, B), cost: $c(A, B) = 2$.
- 3 riders, valuations $v_1 = 10$, $v_2 = 9$, $v_3 = 8$ (assuming known)

Non-linear network flow model:

- Each edge corresponding to a spatiotemporal path.
- Edge reward function corresponding to revenue with k dispatched drivers.

Basic example:

- An edge (A, B), cost: $c(A, B) = 2$.
- 3 riders, valuations $v_1 = 10$, $v_2 = 9$, $v_3 = 8$ (assuming known)
- If we dispatch 1 driver, revenue $r_{(A,B)}(1) = 1 \cdot (v_1 - c) = 8$.

Non-linear network flow model:

- Each edge corresponding to a spatiotemporal path.
- Edge reward function corresponding to revenue with k dispatched drivers.

Basic example:

- An edge (A, B), cost: $c(A, B) = 2$.
- 3 riders, valuations $v_1 = 10$, $v_2 = 9$, $v_3 = 8$ (assuming known)
- If we dispatch 1 driver, revenue $r_{(A,B)}(1) = 1 \cdot (v_1 - c) = 8$.
- $r_{(A,B)}(2) = 2 \cdot (v_2 - c) = 14$, $r_{(A,B)}(3) = 18$.

Non-linear network flow model:

- Each edge corresponding to a spatiotemporal path.
- Edge reward function corresponding to revenue with k dispatched drivers.

Basic example:

- An edge (A, B), cost: $c(A, B) = 2$.
- 3 riders, valuations $v_1 = 10$, $v_2 = 9$, $v_3 = 8$ (assuming known)
- If we dispatch 1 driver, revenue $r_{(A,B)}(1) = 1 \cdot (v_1 - c) = 8$.
- $r_{(A,B)}(2) = 2 \cdot (v_2 - c) = 14$, $r_{(A,B)}(3) = 18$.
- $r_{(A,B)}(3 + j) = 18 - 2j$, $j \geq 0$.

Phase 1: Max-Revenue Car Dispatching
Edge decomposition

Edge decomposition: decompose an non-linear edge into linear ones!

Phase 1: Max-Revenue Car Dispatching

Edge decomposition

Edge decomposition: decompose an non-linear edge into linear ones!

- Virtual Valuation:
  $v'(i) = r(i) - r(i-1) = i(v_i - c) - (i-1)(v_{i-1} - c)$.
  i.e. *marginal reward* of dispatching one more driver.

Phase 1: Max-Revenue Car Dispatching

Edge decomposition

Edge decomposition: decompose an non-linear edge into linear ones!

- Virtual Valuation:
  $v'(i) = r(i) - r(i-1) = i(v_i - c) - (i-1)(v_{i-1} - c)$.
  i.e. *marginal reward* of dispatching one more driver.
- Edge weights: Virtual Valuation

Phase 1: Max-Revenue Car Dispatching
Edge decomposition

Edge decomposition: decompose an non-linear edge into linear ones!

- Virtual Valuation:
  $v'(i) = r(i) - r(i-1) = i(v_i - c) - (i-1)(v_{i-1} - c)$.
  i.e. *marginal reward* of dispatching one more driver.
- Edge weights: Virtual Valuation
- e_1: capacity 1, weight $v'(1) = 8$.
  e_2: capacity 1, weight $v'(2) = 6$.
  e_3: capacity 1, weight $v'(3) = 4$.

Phase 1: Max-Revenue Car Dispatching

Edge decomposition

Edge decomposition: decompose an non-linear edge into linear ones!

- Virtual Valuation:
  $v'(i) = r(i) - r(i-1) = i(v_i - c) - (i-1)(v_{i-1} - c)$.
  i.e. *marginal reward* of dispatching one more driver.
- Edge weights: Virtual Valuation
- e_1: capacity 1, weight $v'(1) = 8$.
  e_2: capacity 1, weight $v'(2) = 6$.
  e_3: capacity 1, weight $v'(3) = 4$.
- e_{cruise}: capacity $+\infty$, weight $-c = -2$.

Phase 1: Max-Revenue Car Dispatching

Definition: Regularity

An instance is *regular* iff all edges have non-increasing marginal rewards.

Phase 1: Max-Revenue Car Dispatching

Definition: Regularity

An instance is *regular* iff all edges have non-increasing marginal rewards.

Theorem

With regularity condition, the Max-Revenue Car Dispatching can be solved with the maximum weighted flow problem.

- Network flow problem is *Totally Unimodular* so we can get an integer optimal solution!

Phase 1: Max-Revenue Car Dispatching

Definition: Regularity

An instance is *regular* iff all edges have non-increasing marginal rewards.

Theorem

With regularity condition, the Max-Revenue Car Dispatching can be solved with the maximum weighted flow problem.

- Network flow problem is *Totally Unimodular* so we can get an integer optimal solution!

Hardness

Without the regularity constraint, Max-Revenue Car Dispatching is NP-hard.

- We can construct a MRCD to solve Set Cover.

NP-Hardness: Proof Sketch

- Construct an arc with $r(x+1) = A \cdot [x \geq k] + C$:
  $v_i = A\frac{k+1}{1}, A\frac{k+1}{2}, \cdots, A\frac{k+1}{k+1}, A \;\; ; \;\; c = 0. \qquad (C = Ak + A)$

NP-Hardness: Proof Sketch

- Construct an arc with $r(x+1) = A \cdot [x \geq k] + C$:
  $v_i = A\frac{k+1}{1}, A\frac{k+1}{2}, \cdots, A\frac{k+1}{k+1}, A \;\; ; \;\; c = 0. \qquad (C = Ak + A)$
- For each such arc, assign one driver at its starting point.

NP-Hardness: Proof Sketch

- Construct an arc with $r(x+1) = A \cdot [x \geq k] + C$:
  $v_i = A\frac{k+1}{1}, A\frac{k+1}{2}, \cdots, A\frac{k+1}{k+1}, A \;\; ; \;\; c = 0. \qquad (C = Ak + A)$
- For each such arc, assign one driver at its starting point.
- Construct a virtual arc (multiple paths in parallel) with
  $r(x) = \max(x-1, 0) + C'$:
  $1 \cdot [x \geq 2] \;\; // \;\; 2 \cdot [x \geq 3] \;\; // \;\; 3 \cdot [x \geq 4] \;\; // \;\; \cdots$

NP-Hardness: Proof Sketch

- Construct an arc with $r(x+1) = A \cdot [x \geq k] + C$:
  $v_i = A\frac{k+1}{1}, A\frac{k+1}{2}, \cdots, A\frac{k+1}{k+1}, A \;\; ; \;\; c = 0. \qquad (C = Ak + A)$
- For each such arc, assign one driver at its starting point.
- Construct a virtual arc (multiple paths in parallel) with
  $r(x) = \max(x-1, 0) + C'$:
  $1 \cdot [x \geq 2] \;\; // \;\; 2 \cdot [x \geq 3] \;\; // \;\; 3 \cdot [x \geq 4] \;\; // \;\; \cdots$
- Reduce to Set Cover:
  If a set covers x items, it gets a reward of $\max(x-1, 0)$.
  If there are N items and the optimal total reward is $N - m$, the minimum cover number is m.

NP-Hardness: Proof Sketch

- Construct an arc with $r(x+1) = A \cdot [x \geq k] + C$:
  $v_i = A\frac{k+1}{1}, A\frac{k+1}{2}, \cdots, A\frac{k+1}{k+1}, A \ ; \ \ c = 0. \qquad (C = Ak + A)$
- For each such arc, assign one driver at its starting point.
- Construct a virtual arc (multiple paths in parallel) with
  $r(x) = \max(x-1, 0) + C'$:
  $1 \cdot [x \geq 2] \ // \ 2 \cdot [x \geq 3] \ // \ 3 \cdot [x \geq 4] \ // \ \cdots$
- Reduce to Set Cover:
  If a set covers x items, it gets a reward of $\max(x-1, 0)$.
  If there are N items and the optimal total reward is $N - m$, the minimum cover number is m.
- Q.E.D.

Phase 2: Fair Reward Re-allocation to Drivers

Issue: Taxi Drivers’ Dilemma

Example: Wuhan, a major city with Yangtze River running through.

Example: Wuhan, a major city with Yangtze River running through.

- In early 2000s, only two bridges crossed the river.

Example: Wuhan, a major city with Yangtze River running through.

- In early 2000s, only two bridges crossed the river.
- In rush hour, traffic crowded the bridges, making cross-river transportation hard.

Example: Wuhan, a major city with Yangtze River running through.

- In early 2000s, only two bridges crossed the river.
- In rush hour, traffic crowded the bridges, making cross-river transportation hard.
- Taxi pricing: mainly based on distances.
  Drivers complained about slow traffic and refused cross-river trips.

Phase 2: Fair Reward Re-allocation to Drivers

Example: Wuhan, a major city with Yangtze River running through.

- In early 2000s, only two bridges crossed the river.
- In rush hour, traffic crowded the bridges, making cross-river transportation hard.
- Taxi pricing: mainly based on distances.
  Drivers complained about slow traffic and refused cross-river trips.
- Can we raise prices of cross-river trips in rush hour?

Example: Wuhan, a major city with Yangtze River running through.

- In early 2000s, only two bridges crossed the river.
- In rush hour, traffic crowded the bridges, making cross-river transportation hard.
- Taxi pricing: mainly based on distances.
  Drivers complained about slow traffic and refused cross-river trips.
- Can we raise prices of cross-river trips in rush hour?
- Riders: "It is slow already, and you ask me to pay even more?"

Example: Wuhan, a major city with Yangtze River running through.

- In early 2000s, only two bridges crossed the river.
- In rush hour, traffic crowded the bridges, making cross-river transportation hard.
- Taxi pricing: mainly based on distances.
  Drivers complained about slow traffic and refused cross-river trips.
- Can we raise prices of cross-river trips in rush hour?
- Riders: "It is slow already, and you ask me to pay even more?"
- How to resolve misaligned incentives via pricing?

Phase 2: Fair Reward Re-allocation to Drivers

Within the same total budget, we can allocate rewards differently to drivers.

Phase 2: Fair Reward Re-allocation to Drivers

Within the same total budget, we can allocate rewards differently to drivers.

- Budget balance: total collected income is equal to total amount distributed to drivers when every driver follows the dispatch (omitting amount platform keeps for profit).

Phase 2: Fair Reward Re-allocation to Drivers

Within the same total budget, we can allocate rewards differently to drivers.

- Budget balance: total collected income is equal to total amount distributed to drivers when every driver follows the dispatch (omitting amount platform keeps for profit).
- Incentive compatibility: Each driver earns at least the cost to cover an edge.

Phase 2: Fair Reward Re-allocation to Drivers

Within the same total budget, we can allocate rewards differently to drivers.

- Budget balance: total collected income is equal to total amount distributed to drivers when every driver follows the dispatch (omitting amount platform keeps for profit).
- Incentive compatibility: Each driver earns at least the cost to cover an edge.
- Subgame-perfectness: No driver is incentivized to deviate.

Phase 2: Fair Reward Re-allocation to Drivers

Within the same total budget, we can allocate rewards differently to drivers.

- Budget balance: total collected income is equal to total amount distributed to drivers when every driver follows the dispatch (omitting amount platform keeps for profit).
- Incentive compatibility: Each driver earns at least the cost to cover an edge.
- Subgame-perfectness: No driver is incentivized to deviate.
- Envy-freeness: No driver feels the mechanism is more favorable to others than themselves.

- Inspiration: potential energy, work independent to trajectory.

- Inspiration: potential energy, work independent to trajectory.
- Idea: define the potential as maximum possible net income a driver can get from a node to the end.

Phase 2: Fair Reward Re-allocation to Drivers

Potential method

- Inspiration: potential energy, work independent to trajectory.
- Idea: define the potential as maximum possible net income a driver can get from a node to the end.

Lemma

Given a routing plan $\mathcal{A}$, a reward re-allocation $y : S^2 \to \mathbb{R}^{\geq 0}$ is fair if and only if there exists a *potential function* $P : S \to \mathbb{R}^{\geq 0}$ such that

- For any $s \in S$ where $\mathcal{A}$ directs at least one driver to leave at state s (*terminal states*), it holds that $P(s) = 0$.
- $\forall (s, s') \in Q$, $y(s, s') - c(s, s') \leq P(s) - P(s')$.
- $\forall (s, s') \in Q : F(s, s') > 0$, $y(s, s') - c(s, s') = P(s) - P(s') \geq 0$.
- $\sum_{s \in S} P(s)(\deg_i(s) - \deg_o(s)) = \sum_{(s,s') \in Q} F(s, s')(p(s, s') - c(s, s'))$, where $\deg_i(s)$ and $\deg_o(s)$ are the number of drivers to enter and leave the platform at the state s respectively.

We prove that fair re-allocation plans always exist in non-degenerate scenarios:

Phase 2: Fair Reward Re-allocation to Drivers

We prove that fair re-allocation plans always exist in non-degenerate scenarios:

Theorem

Let $S_{\#} \subseteq S$ denote the set of terminal states. If there exist $s_1 \in S \setminus S_{\#}$ and $s_2 \in S$ such that $F(s_1, s_2) > 0$ and $\mathcal{I} \geq \sum_{(s,s') \in Q} F(s, s') \cdot c(s, s')$ ($\mathcal{I}$ is the total income collected from the riders), then there exists a fair reward allocation plan.

We prove that fair re-allocation plans always exist in non-degenerate scenarios:

Theorem

Let $S_{\#} \subseteq S$ denote the set of terminal states. If there exist $s_1 \in S \setminus S_{\#}$ and $s_2 \in S$ such that $F(s_1, s_2) > 0$ and $\mathcal{I} \geq \sum_{(s,s') \in Q} F(s, s') \cdot c(s, s')$ ($\mathcal{I}$ is the total income collected from the riders), then there exists a fair reward allocation plan.

- Proof sketch: Topological Ordering

We prove that fair re-allocation plans always exist in non-degenerate scenarios:

Theorem

Let $S_{\#} \subseteq S$ denote the set of terminal states. If there exist $s_1 \in S \setminus S_{\#}$ and $s_2 \in S$ such that $F(s_1, s_2) > 0$ and $\mathcal{I} \geq \sum_{(s,s') \in Q} F(s, s') \cdot c(s, s')$ ($\mathcal{I}$ is the total income collected from the riders), then there exists a fair reward allocation plan.

- Proof sketch: Topological Ordering
- We use Quadratic Programming to find a min-square-distortion feasible scheme, if multiple ones exist.

Stochastic demand setting: we only know the distribution of latent orders $\{\mathcal{D}(s, s')\}$ rather than exact ones.

- Setting:
  —Platform sets price p for an edge;
  —Riders accept the offer iff valuation is at least p *(qualified)*;
  —Drivers follow the dispatch and get the reward whether a rider shows up or not;

Stochastic demand setting: we only know the distribution of latent orders $\{\mathcal{D}(s, s')\}$ rather than exact ones.

- Setting:
  —Platform sets price p for an edge;
  —Riders accept the offer iff valuation is at least p *(qualified)*;
  —Drivers follow the dispatch and get the reward whether a rider shows up or not;
- We can still compute the edge reward function as *expected* rewards for k drivers dispatched on (s, s'), from the distribution $\mathcal{D}(s, s')$.

Stochastic demand setting: we only know the distribution of latent orders $\{\mathcal{D}(s, s')\}$ rather than exact ones.

- Setting:
  —Platform sets price p for an edge;
  —Riders accept the offer iff valuation is at least p *(qualified)*;
  —Drivers follow the dispatch and get the reward whether a rider shows up or not;
- We can still compute the edge reward function as *expected* rewards for k drivers dispatched on (s, s'), from the distribution $\mathcal{D}(s, s')$.
- We then perform Thompson Sampling to balance exploration and exploitation and update the estimated distributions.

Computing edge reward function:

Computing edge reward function:

- The probability mass function $\mathcal{P}(\cdot; s, s', p) : \mathbb{N} \to \mathbb{R}$ of the number of qualified orders:
  $\mathcal{P}(i; s, s', p) = \sum_{j=0}^{\infty} b(i, j; \Pr[v_t \geq p]) \Pr[x_{s,s'} = j]$, where
  $b(k, n; P) = \binom{n}{k} P^k (1 - P)^{n-k}$ computes the binomial distribution.

Computing edge reward function:

- The probability mass function $\mathcal{P}(\cdot; s, s', p) : \mathbb{N} \to \mathbb{R}$ of the number of qualified orders:
  $\mathcal{P}(i; s, s', p) = \sum_{j=0}^{\infty} b(i, j; \Pr[v_t \geq p]) \Pr[x_{s,s'} = j]$, where
  $b(k, n; P) = \binom{n}{k} P^k (1 - P)^{n-k}$ computes the binomial distribution.
- Let $\tilde{u}(n; s, s', p)$ be the number of the fulfilled orders; its expectation: $\mathbb{E}[\tilde{u}(n; s, s', p)] = \sum_{i=0}^{\infty} \mathcal{P}(i; s, s', p) \min\{i, n\}$.

Computing edge reward function:

- The probability mass function $\mathcal{P}(\cdot; s, s', p) : \mathbb{N} \to \mathbb{R}$ of the number of qualified orders:
  $\mathcal{P}(i; s, s', p) = \sum_{j=0}^{\infty} b(i, j; \Pr[v_t \geq p]) \Pr[x_{s,s'} = j]$, where $b(k, n; P) = \binom{n}{k} P^k (1 - P)^{n-k}$ computes the binomial distribution.
- Let $\tilde{u}(n; s, s', p)$ be the number of the fulfilled orders; its expectation: $\mathbb{E}[\tilde{u}(n; s, s', p)] = \sum_{i=0}^{\infty} \mathcal{P}(i; s, s', p) \min\{i, n\}$.
- The expected revenue at price p and n drivers:
  $\mathcal{R}(n, p; s, s') = p \cdot \mathbb{E}[\tilde{u}(n; s, s', p)] - c(s, s') \cdot n$.

Computing edge reward function:

- The probability mass function $\mathcal{P}(\cdot; s, s', p) : \mathbb{N} \to \mathbb{R}$ of the number of qualified orders:
  $\mathcal{P}(i; s, s', p) = \sum_{j=0}^{\infty} b(i, j; \Pr[v_t \geq p]) \Pr[x_{s,s'} = j]$, where
  $b(k, n; P) = \binom{n}{k} P^k (1 - P)^{n-k}$ computes the binomial distribution.
- Let $\tilde{u}(n; s, s', p)$ be the number of the fulfilled orders; its expectation: $\mathbb{E}[\tilde{u}(n; s, s', p)] = \sum_{i=0}^{\infty} \mathcal{P}(i; s, s', p) \min\{i, n\}$.
- The expected revenue at price p and n drivers:
  $\mathcal{R}(n, p; s, s') = p \cdot \mathbb{E}[\tilde{u}(n; s, s', p)] - c(s, s') \cdot n$.
- The expected revenue function at optimal price:
  $r(n; e_{s,s'}^{(\mathrm{w})}) = \max_{p \in \mathbb{R}^{\geq 0}} \{\mathcal{R}(n, p; s, s')\}$

We do need Thompson Sampling because the riders' valuations are NOT disclosed!

We do need Thompson Sampling because the riders' valuations are NOT disclosed!

- At the begin of day t, we sample parameters $\{\hat{\mathcal{D}}(s, s')\}_t$ from the prior distribution and compute dispatching and pricing.

We do need Thompson Sampling because the riders' valuations are NOT disclosed!

- At the begin of day t, we sample parameters $\{\hat{\mathcal{D}}(s, s')\}_t$ from the prior distribution and compute dispatching and pricing.
- We collect the data of riders' responses to offered prices and compute posterior distributions of parameters.

We do need Thompson Sampling because the riders' valuations are NOT disclosed!

- At the begin of day t, we sample parameters $\{\hat{\mathcal{D}}(s, s')\}_t$ from the prior distribution and compute dispatching and pricing.
- We collect the data of riders' responses to offered prices and compute posterior distributions of parameters.
- We set the posterior distributions as the prior of day $t + 1$.

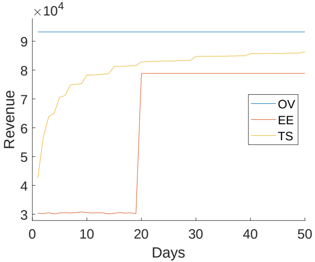
Phase 1: Max-Revenue Car Dispatching

Phase 1: Max-Revenue Car Dispatching

- Offline: 23% increased revenue than fixed price on DiDi dataset.

Phase 1: Max-Revenue Car Dispatching

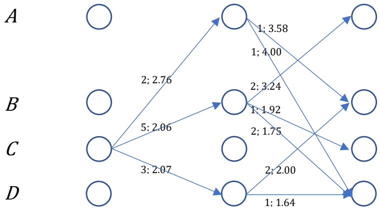
- Offline: 23% increased revenue than fixed price on DiDi dataset.
- Online: Significantly lower regret than explore-and-exploit scheme.



Phase 2: Fair Reward Re-Allocation

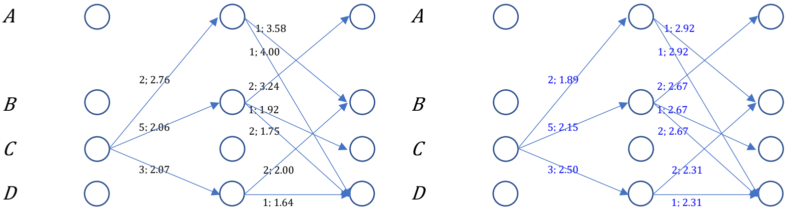
Phase 2: Fair Reward Re-Allocation

- $C \rightarrow A \rightarrow D$ has significantly higher reward than $C \rightarrow D \rightarrow D$.



Phase 2: Fair Reward Re-Allocation

- $C \rightarrow A \rightarrow D$ has significantly higher reward than $C \rightarrow D \rightarrow D$.
- After Phase 2: both 4.81.



Acknowledgement

- Thank Shiyuan Wang for discussion of relevant topics on mechanism design.

Q/A