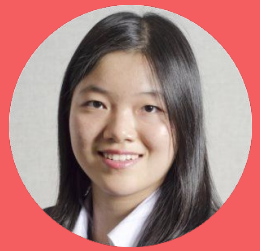
Learning Equilibria in Matching Markets from Bandit Feedback

Meena Jagadeesan (UC Berkeley)

Joint work w/ Alexander Wei, Yixin Wang, Michael I. Jordan, and Jacob Steinhardt (UC Berkeley)

(Appeared at NeurIPS 2021)

Data-Driven Matching Platforms

Data-Driven Matching Platforms

There are **two "sides" of the marketplace**:

- Think of one side as *customers (riders)* and the other side as *providers (drivers)*.

Data-Driven Matching Platforms

There are **two "sides" of the marketplace**:

- Think of one side as *customers (riders)* and the other side as *providers (drivers)*.

Both sides have **preferences** over each other.

- E.g., a driver may prefer certain types of routes; a rider may prefer drivers who are closer

Data-Driven Matching Platforms

There are **two "sides" of the marketplace**:

- Think of one side as *customers (riders)* and the other side as *providers (drivers)*.

Both sides have **preferences** over each other.

- E.g., a driver may prefer certain types of routes; a rider may prefer drivers who are closer

The platform **recommends matchings (and suggests $ transfers)** between participants.

- E.g. Uber suggests a match between a driver and a rider along with a $ transfer

Data-Driven Matching Platforms

There are **two "sides" of the marketplace**:

- Think of one side as *customers (riders)* and the other side as *providers (drivers)*.

Both sides have **preferences** over each other.

- E.g., a driver may prefer certain types of routes; a rider may prefer drivers who are closer

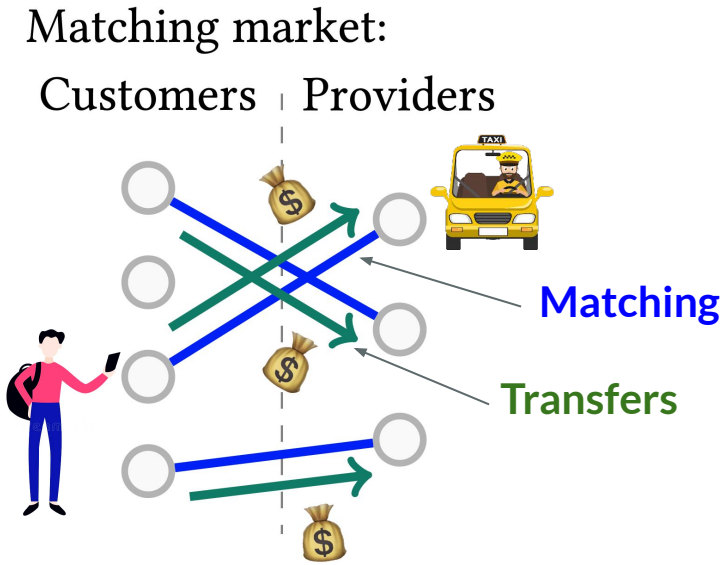
The platform **recommends matchings (and suggests $ transfers)** between participants.

- E.g. Uber suggests a match between a driver and a rider along with a $ transfer

Data-driven nature of recommendations:

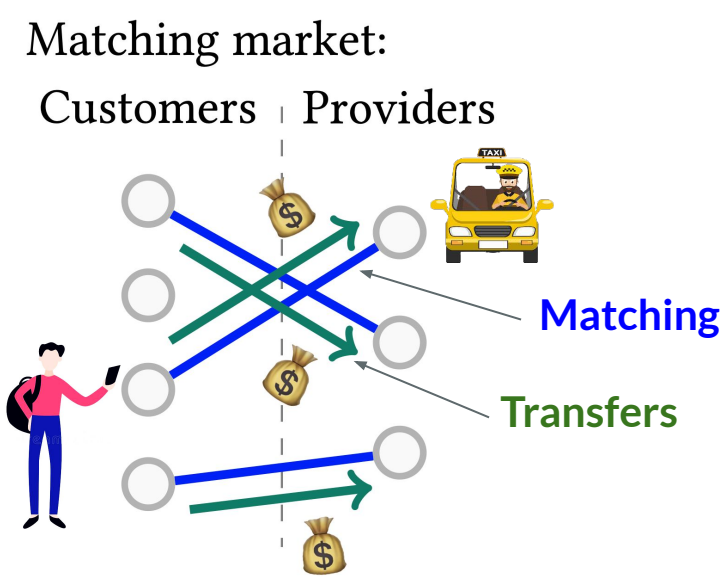
- *Learn* preferences from past data
- *Make decisions under uncertainty* when recommending a matching between participants

Two-Sided Matching Platform with Transfers

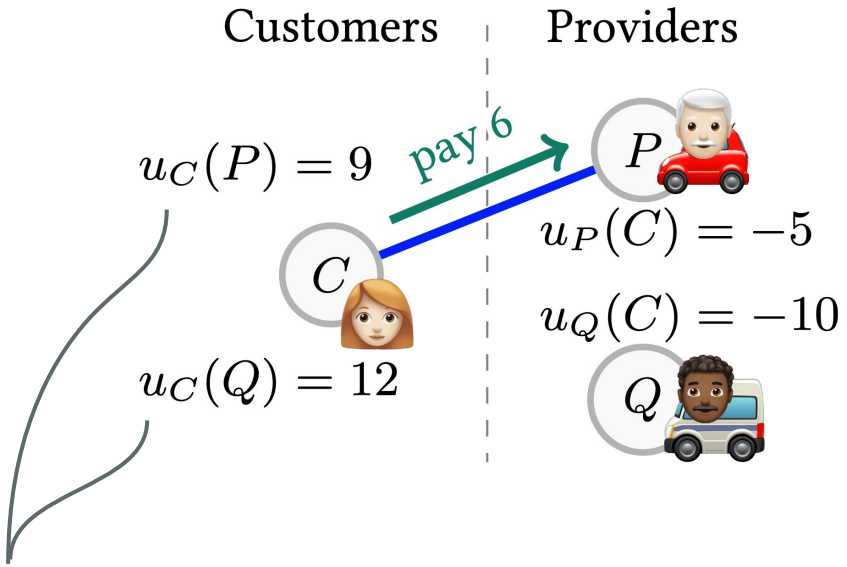


Market outcome = matching + transfers

Two-Sided Matching Platform with Transfers

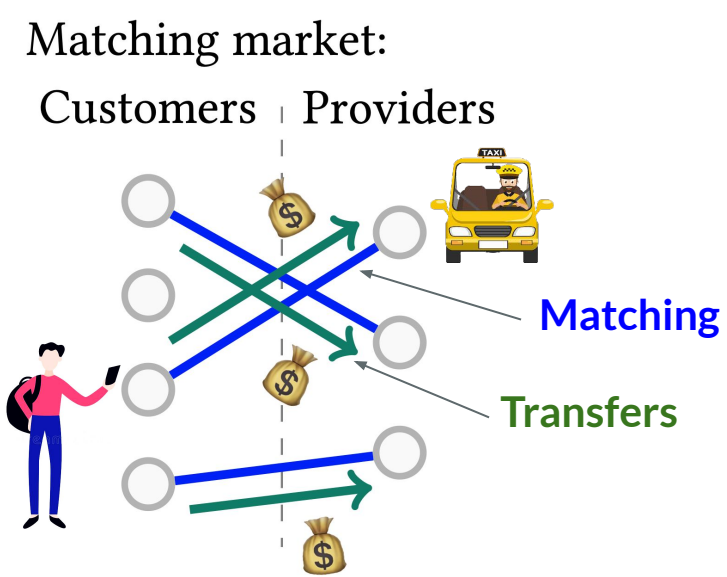


Market outcome = matching + transfers

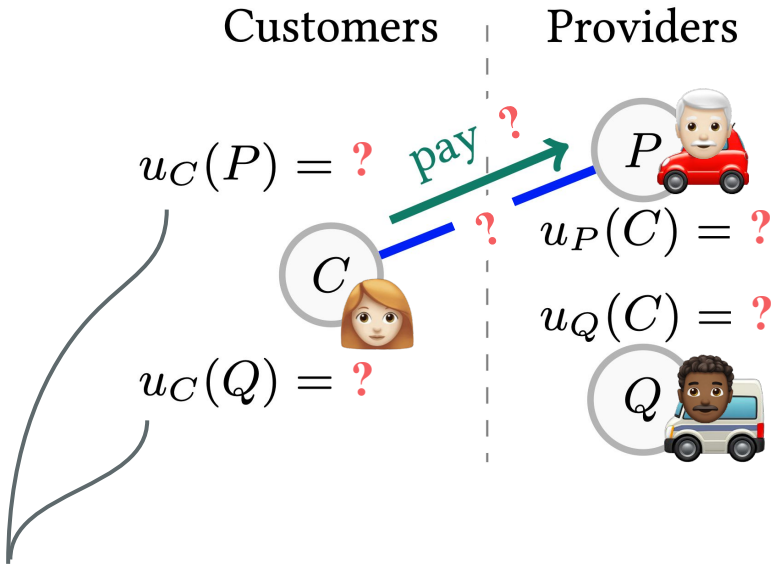


Utility values of agent *C* for matching with agents *P* and *Q*

Two-Sided Matching Platform with Transfers

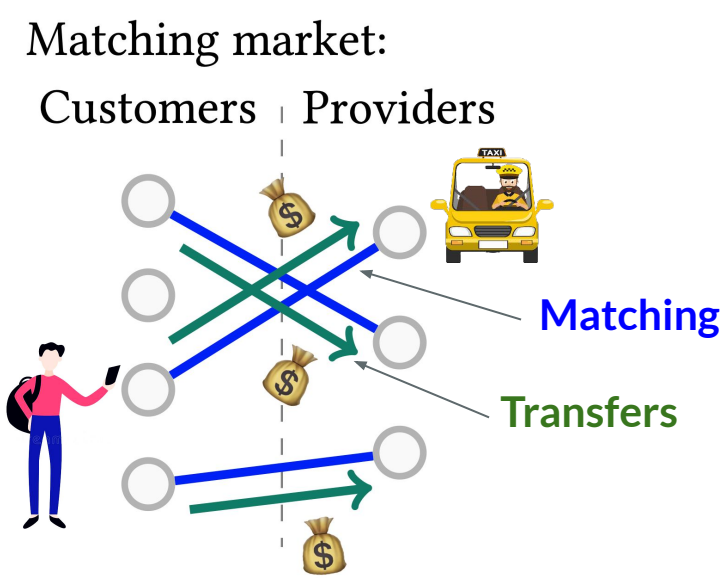


Market outcome = matching + transfers

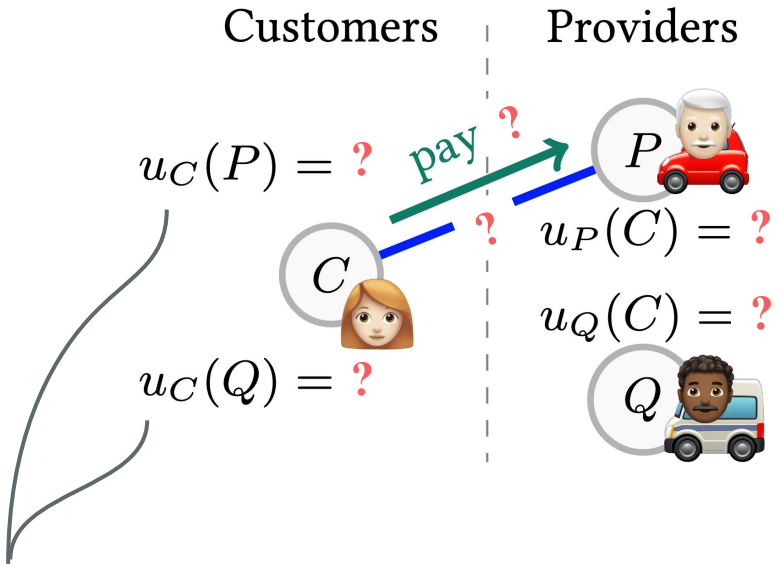


Utility values of agent *C* for matching with agents *P* and *Q*

Two-Sided Matching Platform with Transfers



Market outcome = matching + transfers



Utility values of agent *C* for matching with agents *P* and *Q*

Our contribution: incentive-aware learning framework for matching markets

Matching + Learning: Challenges

Platform wants to align market outcomes with agent preferences

- To keep agents, must offer desirable matches at fair prices

But agent preferences are unknown…

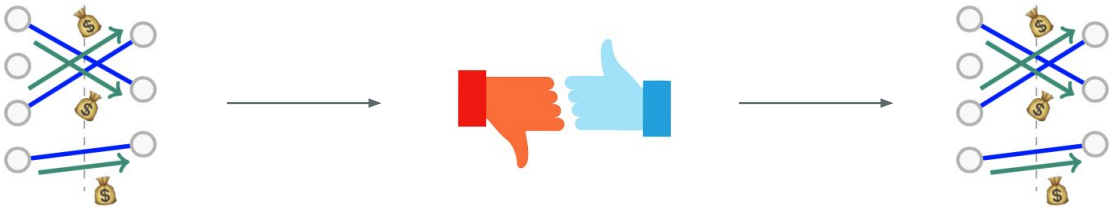
Matching + Learning: Challenges

Platform wants to align market outcomes with agent preferences

- To keep agents, must offer desirable matches at fair prices

But agent preferences are unknown…

Approach: learn agent preferences from repeated feedback!



Existing Approaches

Classical economic theory:

- Focuses on *known* agent preferences
- Deferred acceptance etc. finds stable outcomes given fixed preferences

Classical multi-armed bandits:

- Combinatorial bandits maximizes total welfare of the matching, an *incentive-unaware* objective

Existing Approaches

Classical economic theory:

- Focuses on *known* agent preferences
- Deferred acceptance etc. finds stable outcomes given fixed preferences

learning and exploration?

uncertainty about preferences?

Classical multi-armed bandits:

- Combinatorial bandits maximizes total welfare of the matching, an *incentive-unaware* objective

preference structures?

incentive-awareness?

Existing Approaches

Classical economic theory:

- Focuses on *known* agent preferences
- Deferred acceptance etc. finds stable outcomes given fixed preferences

Classical multi-armed bandits:

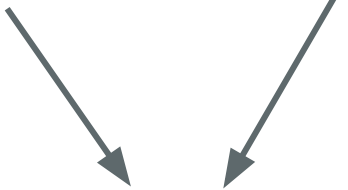
- Combinatorial bandits maximizes total welfare of the matching, an *incentive-unaware* objective

learning and exploration?

uncertainty about preferences?

preference structures?

incentive-awareness?



How should stable market outcomes be learned?
And what structures make efficient learning possible?

Existing Approaches

Classical economic theory:

Classical multi-armed bandits:

This work: Brings together **market design** + **multi-armed bandits** to answer these questions

preferences

learning and exploration?

preference structures?

uncertainty about preferences?

incentive-awareness?

How should stable market outcomes be learned?
And what structures make efficient learning possible?

Our Contribution

I. Framework

- Develop bandit framework for learning stable outcomes in matching markets
- Introduce Subset Instability as an learning objective

II. Algorithm Design

- Design no-regret algorithms for learning stable market outcomes
- Give examples of how preference structure impacts regret

III. Extensions

- Interpret learning objective as platform profit
- Extend framework and algorithms to matching without transfers

Related Work

Matching with known preferences:

e.g., Gale and Shapley (AMM, 1962), Shapley and Shubik (IJGT, 1971)

Matching with bandit feedback:

e.g., Das, Kamenica (IJCAI '05), Liu, Mania, Jordan (AISTATS '20), Johari, Kamble, Kanoria (Operations Research, '21), Kasy, Teytelboym (Econometrics Journal, 2022), Sankararaman, Basu, Sankararaman (AISTATS 2021), Cen, Shah (AISTATS 2022)

Learning stable matchings when preferences can evolve:

e.g., (Min, Wang, Xu, Wang, Jordan, Yang, 2022)

Other perspectives on learning stable matchings:

e.g., Ashlagi, Braverman, Kanoria, Shi (Management Science, '20), Emamjomeh-Zadeh, Gonczarowski, Kempe (EC '20)

Combinatorial bandits:

e.g., Cesa-Bianchi, Lugosi (J. CS Sci. 2012), Gai, Krishnamachari, Jain (Trans. Netw., '12), Chen, Wang, Yuan (ICML '13)

Outline for the rest of the talk

1. Background on stable matchings with transfers (Shapley, Shubik ‘71)

2. Our framework for learning stable market outcomes

3. Designing algorithms to learn stable market outcomes

4. Extensions and Conclusion

Outline for the rest of the talk

1. **Background on stable matchings with transfers (Shapley, Shubik ‘71)**

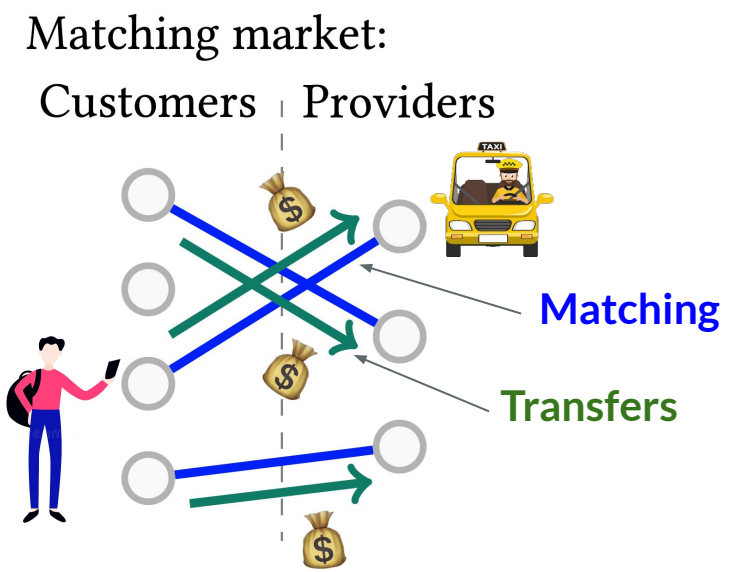
2. Our framework for learning stable market outcomes

3. Designing algorithms to learn stable market outcomes

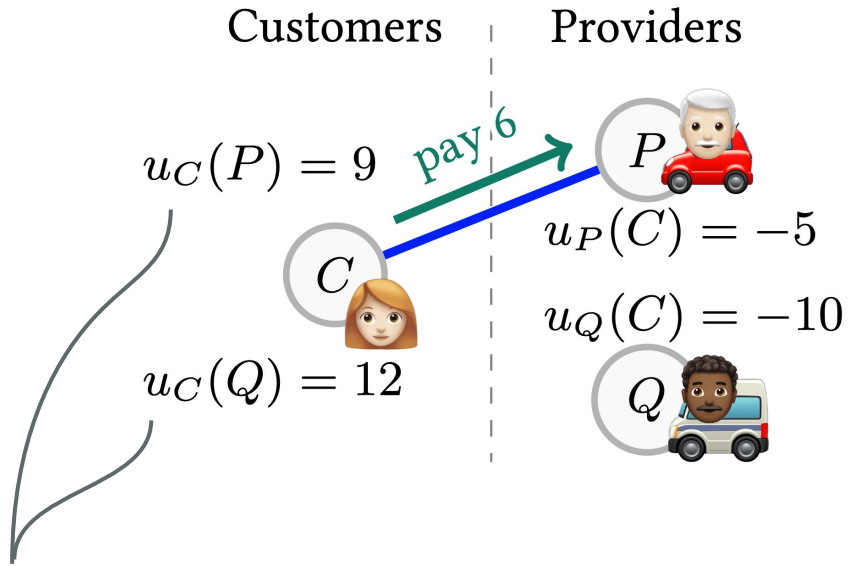
4. Extensions and Conclusion

Stable Matchings with Transfers (Shapley, Shubik ‘71)

Two-Sided Matching Market with Transfers

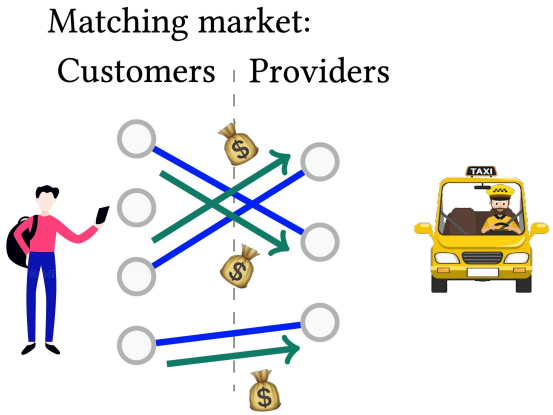


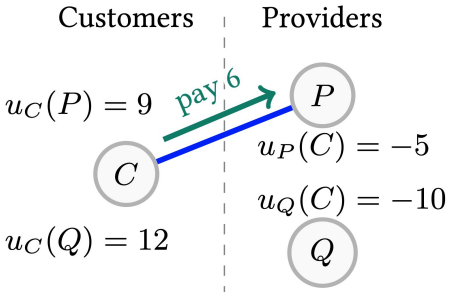
Market outcome = matching + transfers



Utility values of agent *C* for matching with agents *P* and *Q*

Two-Sided Matchings with Transfers

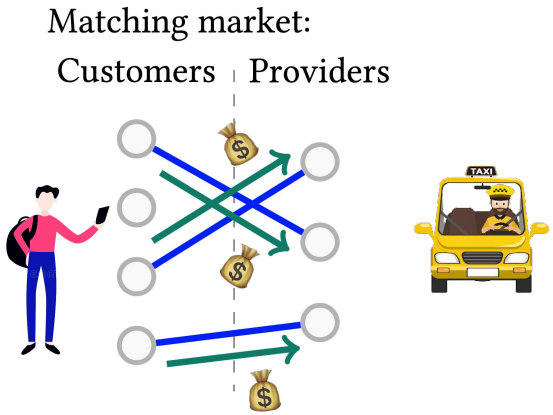


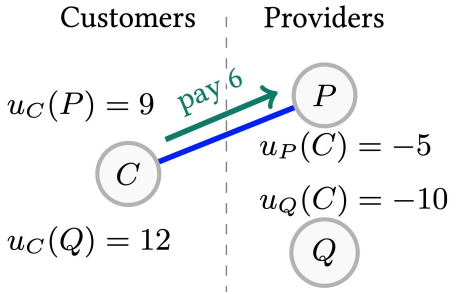


Two-Sided Matchings with Transfers

Participants:

- Set I of customers
- Set J of providers



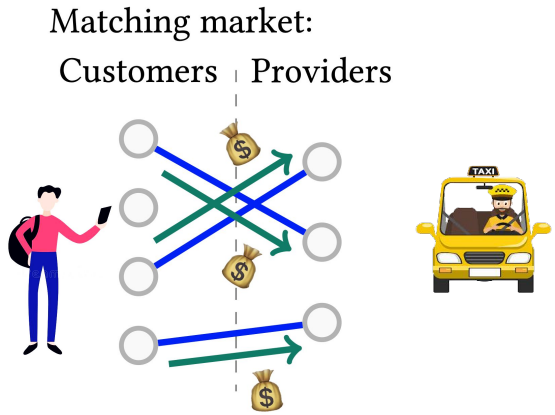


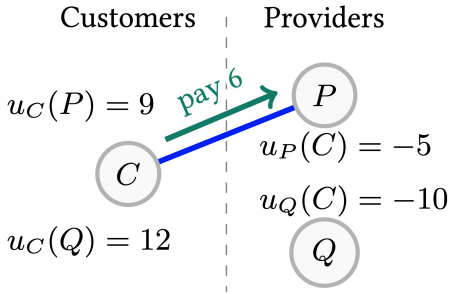
Two-Sided Matchings with Transfers

Participants:

- Set I of customers
- Set J of providers

Market outcome:





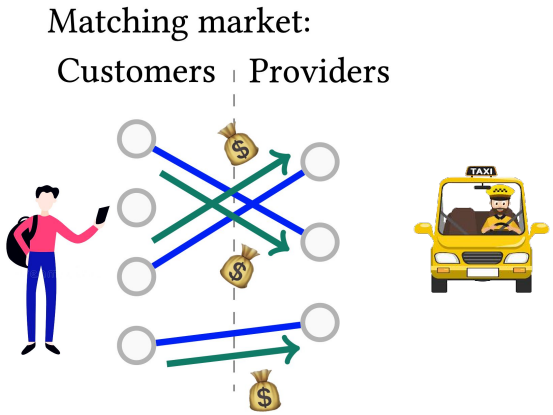
Two-Sided Matchings with Transfers

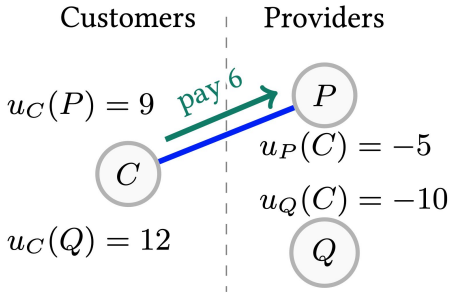
Participants:

- Set I of customers
- Set J of providers

Market outcome:

- Matching $\mu : I \cup J \to I \cup J$ that maps each agent to their match





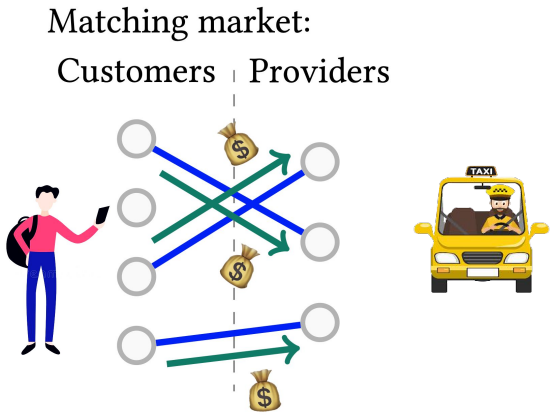
Two-Sided Matchings with Transfers

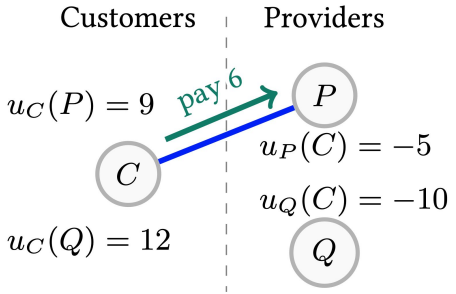
Participants:

- Set I of customers
- Set J of providers

Market outcome:

- Matching $\mu : I \cup J \to I \cup J$ that maps each agent to their match
- Transfers $\tau \in \mathbb{R}^{I \cup J}$, with agent a receiving transfer τ_a





Two-Sided Matchings with Transfers

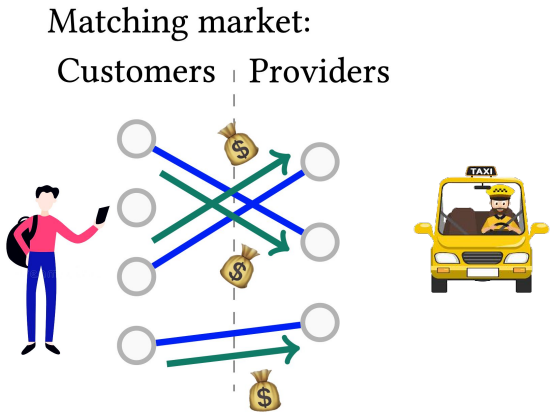
Participants:

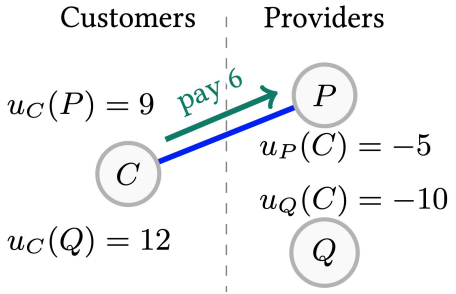
- Set I of customers
- Set J of providers

Market outcome:

- Matching $\mu : I \cup J \to I \cup J$ that maps each agent to their match
- Transfers $\tau \in \mathbb{R}^{I \cup J}$, with agent a receiving transfer τ_a

Utilities: agent a receives utility $u_a(\mu(a)) + \tau_a$ from market outcome (μ, τ)





Stability of a market outcome

Notation: matching $\mu : I \cup J \to I \cup J$, *transfers* $\tau \in \mathbb{R}^{I \cup J}$, *utility is* $u_a(\mu(a)) + \tau_a$

Stability captures the alignment of a market outcome with preferences.

Stability of a market outcome

Notation: matching $\mu : I \cup J \rightarrow I \cup J$, *transfers* $\tau \in \mathbb{R}^{I \cup J}$, *utility is* $u_a(\mu(a)) + \tau_a$

Stability captures the alignment of a market outcome with preferences.

Definition (Shapley, Shubik, 1971): A market outcome (μ, τ) is **stable** if (1) there are no blocking pairs and (2) it satisfies individually rationality.

No blocking pairs: no pair (i, j) would prefer each other over (μ, τ). That is:

$$for\ all\ (i,j) : u_i(j) + u_j(i) \leq (u_i(\mu(i)) + \tau_i) + (u_j(\mu(j)) + \tau_j).$$

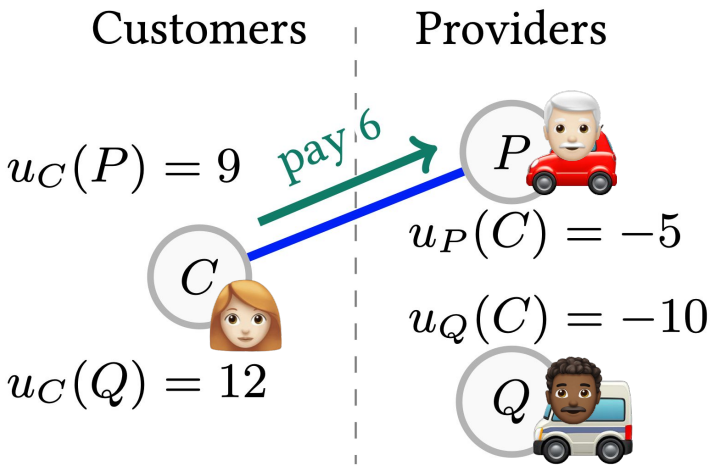
Individual rationality: no agent would prefer to quit the platform. That is,

$$for\ all\ a : u_a(\mu(a)) + \tau_a \geq 0.$$

Example of a stable market outcome

No blocking pairs: $u_i(j) + u_j(i) \leq (u_i(\mu(i)) + \tau_i) + (u_j(\mu(j)) + \tau_j)$ *for all pairs* (i, j)

Individual rationality: $u_a(\mu(a)) + \tau_a \geq 0$ *for every agent* a



No blocking pairs: 12 + (-10) < (9 - 6)

Individual rationality: 9 - 6 > 0 and -5 + 6 > 0

Properties of stable matchings *with transfers*

1. **Maximum matching:** for any stable market outcome (μ, τ), the matching μ maximizes total utility

$$\mu \in \text{argmax } \{\sum u_a(\mu^*(a)) \mid \mu^* \text{ is a matching over agents}\}.$$

(Shapley, Shubik, 1971)

Properties of stable matchings *with transfers*

1. **Maximum matching:** for any stable market outcome (μ, τ), the matching μ maximizes total utility

$$\mu \in \text{argmax } \{\sum u_a(\mu^*(a)) \mid \mu^* \text{ is a matching over agents}\}.$$

2. **Transfers are not unique:** For any maximum matching μ, there are typically *infinitely many* transfers τ for which (μ, τ) is a stable market outcome.

(Shapley, Shubik, 1971)

Properties of stable matchings *with transfers*

1. **Maximum matching:** for any stable market outcome (μ, τ), the matching μ maximizes total utility

$$\mu \in \text{argmax } \{\sum u_a(\mu^*(a)) \mid \mu^* \text{ is a matching over agents}\}.$$

2. **Transfers are not unique:** For any maximum matching μ, there are typically *infinitely many* transfers τ for which (μ, τ) is a stable market outcome.

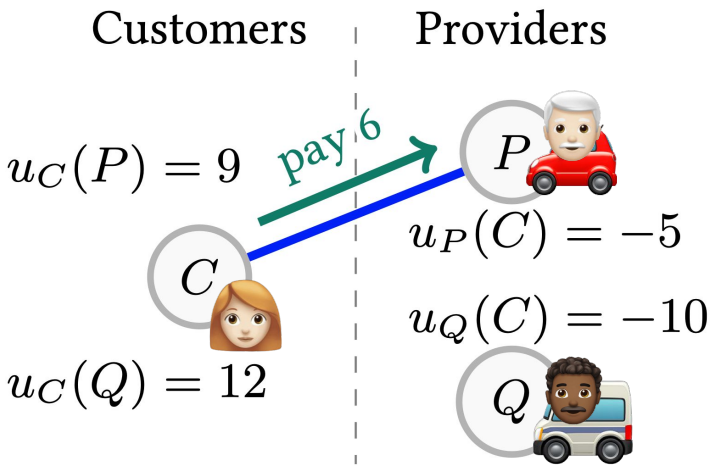
3. **Linear program formulation**: Stability can be formulated as an LP with primal variables are matchings μ and the dual variables are related to transfers τ.

(Shapley, Shubik, 1971)

Example of a stable market outcome

No blocking pairs: $u_i(j) + u_j(i) \leq (u_i(\mu(i)) + \tau_i) + (u_j(\mu(j)) + \tau_j)$ *for all pairs* (i, j)

Individual rationality: $u_a(\mu(a)) + \tau_a \geq 0$ *for every agent* a



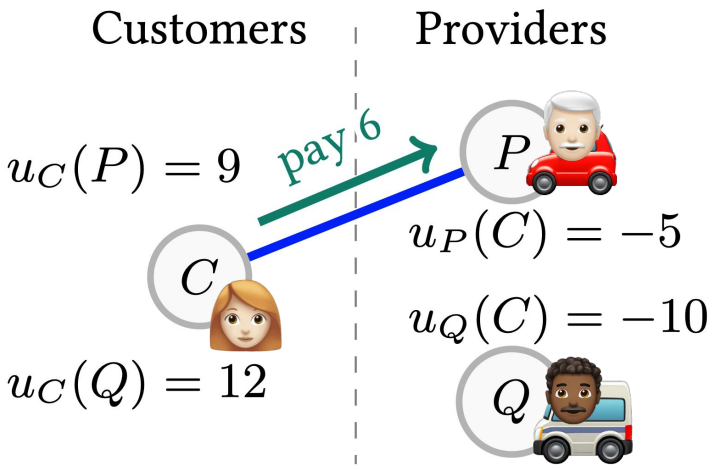
No blocking pairs: 12 + (-10) < (9 - 6)

Individual rationality: 9 - 6 > 0 and -5 + 6 > 0

Example of a stable market outcome

No blocking pairs: $u_i(j) + u_j(i) \leq (u_i(\mu(i)) + \tau_i) + (u_j(\mu(j)) + \tau_j)$ *for all pairs* (i, j)

Individual rationality: $u_a(\mu(a)) + \tau_a \geq 0$ *for every agent* a



No blocking pairs: 12 + (-10) < (9 - 6)

Individual rationality: 9 - 6 > 0 and -5 + 6 > 0

Note that:

1. (C, P) is a maximum matching
2. (C, P) with "pay 7" is also stable.

Outline for the rest of the talk

1. Background on stable matchings with transfers (Shapley, Shubik ‘71)

2. **Our framework for learning stable market outcomes**

3. Designing algorithms to learn stable market outcomes

4. Extensions and Conclusion

Our Framework

Learning and Feedback Model

Learning takes place over T rounds. In the t-th round:

Learning and Feedback Model

Learning takes place over T rounds. In the t-th round:

- Agents $I^t \subseteq I, J^t \subseteq J$ arrive to the market.

Learning and Feedback Model

Learning takes place over T rounds. In the t-th round:

- Agents $I^t \subseteq I, J^t \subseteq J$ arrive to the market.
- Platform selects a market outcome: a matching with transfers (μ^t, τ^t)

Learning and Feedback Model

Learning takes place over T rounds. In the t-th round:

- Agents $I^t \subseteq I, J^t \subseteq J$ arrive to the market.
- Platform selects a market outcome: a matching with transfers (μ^t, τ^t)
- Platform observes bandit feedback: noisy utilities $u_a(\mu^t(a)) + \varepsilon$ for each agent a

Learning and Feedback Model

Learning takes place over T rounds. In the t-th round:

- Agents $I^t \subseteq I, J^t \subseteq J$ arrive to the market.
- Platform selects a market outcome: a matching with transfers (μ^t, τ^t)
- Platform observes bandit feedback: noisy utilities $u_a(\mu^t(a)) + \varepsilon$ for each agent a
- The platform incurs loss given by instability $\mathrm{Instab}(\mu^t, \tau^t)$ of the selected outcome.

Learning and Feedback Model

Learning takes place over T rounds. In the t-th round:

- Agents $I^t \subseteq I, J^t \subseteq J$ arrive to the market.
- Platform selects a market outcome: a matching with transfers (μ^t, τ^t)
- Platform observes bandit feedback: noisy utilities $u_a(\mu^t(a)) + \varepsilon$ for each agent a
- The platform incurs loss given by instability $\mathrm{Instab}(\mu^t, \tau^t)$ of the selected outcome.

Regret = the cumulative loss over time $\sum_{t=1}^{T} \mathrm{Instab}(\mu^t, \tau^t)$

Why do we need an instability measure?

Stability is a binary notion, but we need a continuous notion.

- With uncertainty, impossible to guarantee exact stability.

Goal: design a measure of "distance from stability" that is:

(1) tractable for learning, and

(2) economically meaningful.

A naive measure of instability: utility difference

A naive measure of quality: total utility $\sum u_a(\mu(a)) = \sum u_a(\mu(a)) + \tau_a$ across agents

A naive measure of instability: utility difference

A naive measure of quality: total utility $\sum u_a(\mu(a)) = \sum u_a(\mu(a)) + \tau_a$ across agents

Utility difference = difference in utility achieved by μ and max possible utility

$$\max \{\sum u_a(\mu^*(a)) \mid \mu^* \text{ is a matching over agents}\} - \sum u_a(\mu(a))$$

A naive measure of instability: utility difference

A naive measure of quality: total utility $\sum u_a(\mu(a)) = \sum u_a(\mu(a)) + \tau_a$ across agents

Utility difference = difference in utility achieved by μ and max possible utility

$$\max \{\sum u_a(\mu^*(a)) \mid \mu^* \text{ is a matching over agents}\} - \sum u_a(\mu(a))$$

Utility difference is the standard loss function in *combinatorial bandits*:

e.g. (Cesa-Bianchi, Lugosi, 2012), (Gai, Krishnamachari, Jain, 2012), (Chen, Wang, Yuan, 2013), etc.

A naive measure of instability: utility difference

A naive measure of quality: total utility $\sum u_a(\mu(a)) = \sum u_a(\mu(a)) + \tau_a$ across agents

Utility difference = difference in utility achieved by μ and max possible utility

$$\max \{\sum u_a(\mu^*(a)) \mid \mu^* \text{ is a matching over agents}\} - \sum u_a(\mu(a))$$

Utility difference is the standard loss function in *combinatorial bandits*:

e.g. (Cesa-Bianchi, Lugosi, 2012), (Gai, Krishnamachari, Jain, 2012), (Chen, Wang, Yuan, 2013), etc.

But for matching with transfers: utility difference entirely ignores the transfers!

Our notion of instability: Subset Instability

Definition: The Subset Instability of a market outcome (μ, τ) is defined to be:

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S} \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Our notion of instability: Subset Instability

Definition: The Subset Instability of a market outcome (μ, τ) is defined to be:

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S} \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Maximum over coalitions S

Our notion of instability: Subset Instability

Definition: The Subset Instability of a market outcome (μ, τ) is defined to be:

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S} \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Maximum over coalitions S

Max utility that agents in S can generate by themselves

Our notion of instability: Subset Instability

Definition: The Subset Instability of a market outcome (μ, τ) is defined to be:

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S} \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Maximum over coalitions S

Max utility that agents in S can generate by themselves

Utility that agents in S receive from the platform

Our notion of instability: Subset Instability

Definition: The Subset Instability of a market outcome (μ, τ) is defined to be:

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S} \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Maximum over coalitions S

Max utility that agents in S can generate by themselves

Utility that agents in S receive from the platform

Subset Instability = $\max$ gain that any "coalition" of agents S could obtain by deviating from the given outcome (μ, τ) and only matching within S

Properties of Subset Instability

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over } S \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Properties of Subset Instability

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over } S \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Subset Instability is 0 if and only if (μ, τ) is stable

Properties of Subset Instability

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over } S \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Subset Instability is 0 if and only if (μ, τ) is stable

Subset Instability is *at least* the utility difference of μ.

Properties of Subset Instability

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S} \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Subset Instability is 0 if and only if (μ, τ) is stable

Subset Instability is *at least* the utility difference of μ.

Subset Instability = the "minimum stabilizing subsidy"

- How much a platform has to "pay" participants for the matching to be stable

Properties of Subset Instability

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S} \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Subset Instability is 0 if and only if (μ, τ) is stable

Subset Instability is *at least* the utility difference of μ.

Subset Instability = the "minimum stabilizing subsidy"

- How much a platform has to "pay" participants for the matching to be stable

Subset Instability is *Lipschitz in the utility function* u for any market outcome (μ, τ) .

- Guarantees that Subset Instability is tractable for learning stable matchings

Main algorithmic question

Minimize regret given by cumulative instability over time:

$$\text{Regret} := \sum_{t=1}^{T} \text{Instab}(\mu^t, \tau^t)$$

$$\text{Instab}((\mu^t, \tau^t)) := \max_{S \subseteq I \cup J} \left(\max_{\mu^*} \{ \sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over } S \} - \sum_{a \in S} (u_a(\mu(a)) + \tau_a) \right)$$

Outline for the rest of the talk

1. Background on stable matchings with transfers (Shapley, Shubik ‘71)

2. Our framework for learning stable market outcomes

3. **Designing algorithms to learn stable market outcomes**

4. Extensions and Conclusion

Algorithm Design

Results: No-Regret Algorithms

Theorem 1: There exists an algorithm incurring $\tilde{O}(N^{3/2}T^{1/2})$ instance-independent regret with N agents over T rounds.

Results: No-Regret Algorithms

Theorem 1: There exists an algorithm incurring $\tilde{O}(N^{3/2}T^{1/2})$ instance-independent regret with N agents over T rounds.

We further show that the instance-independent regret bound in Theorem 1 is *optimal* up to log factors (see paper).

Results: No-Regret Algorithms

Theorem 1: There exists an algorithm incurring $\tilde{O}(N^{3/2}T^{1/2})$ instance-independent regret with N agents over T rounds.

We further show that the instance-independent regret bound in Theorem 1 is *optimal* up to log factors (see paper).

We can also achieve instance-dependent regret bounds:

Theorem 2: The same algorithm incurs $\tilde{O}(N^5 \log(T)/\Delta^2)$ regret with N agents over T rounds where Δ is a measure of instance-dependent gap.

Intuition for why no-regret learning is possible

We can easily adapt ExploreThenCommit to this setting.

Explore phase: obtains estimate u_{est} of u with $||u_{est} - u||_\infty \leq \varepsilon$ within $\tilde{O}(N/\varepsilon^2)$ rounds.

Commit phase: "commit" to stable matching (μ, τ) with respect to u_{est}

- Lipschitzness of Subset Instability implies $\text{Instab}(\mu, \tau) \leq L\varepsilon$

This algorithm achieves regret $\tilde{O}(N^{4/3}T^{2/3})$.

Intuition for why no-regret learning is possible

We can easily adapt ExploreThenCommit to this setting.

Explore phase: obtains estimate u_{est} of u with $||u_{est} - u||_\infty \leq \varepsilon$ within $\tilde{O}(N/\varepsilon^2)$ rounds.

Commit phase: "commit" to stable matching (μ, τ) with respect to u_{est}

- Lipschitzness of Subset Instability implies $\text{Instab}(\mu, \tau) \leq L\varepsilon$

This algorithm achieves regret $\tilde{O}(N^{4/3}T^{2/3})$.

Key algorithmic question: Can we achieve $\tilde{O}(T^{1/2})$ regret?

An algorithmic meta-approach (MatchUCB)

Idea: adopt “optimism in the face of uncertainty” principle

An algorithmic meta-approach (MatchUCB)

Idea: adopt "optimism in the face of uncertainty" principle

(1) For each pair (i, j), keep track of confidence sets $C_{i,j}$ for $u_i(j)$ and $C_{j,i}$ for $u_j(i)$.
- Confidence sets should contain true utility values with high probability.

An algorithmic meta-approach (MatchUCB)

Idea: adopt "optimism in the face of uncertainty" principle

(1) For each pair (i, j), keep track of confidence sets $C_{i,j}$ for $u_i(j)$ and $C_{j,i}$ for $u_j(i)$.
- Confidence sets should contain true utility values with high probability.

(2) At each round:

An algorithmic meta-approach (MatchUCB)

Idea: adopt "optimism in the face of uncertainty" principle

(1) For each pair (i, j), keep track of confidence sets $C_{i,j}$ for $u_i(j)$ and $C_{j,i}$ for $u_j(i)$.
- Confidence sets should contain true utility values with high probability.

(2) At each round:
- Compute "upper confidence bound" estimates $u_i^{\mathrm{UCB}}(j) = \max C_{i,j}$ and $u_j^{\mathrm{UCB}}(i) = \max C_{j,i}$ for all pairs (i, j).

An algorithmic meta-approach (MatchUCB)

Idea: adopt "optimism in the face of uncertainty" principle

(1) For each pair (i, j), keep track of confidence sets $C_{i,j}$ for $u_i(j)$ and $C_{j,i}$ for $u_j(i)$.
- Confidence sets should contain true utility values with high probability.

(2) At each round:
- Compute "upper confidence bound" estimates $u_i^{\mathrm{UCB}}(j) = \max C_{i,j}$ and $u_j^{\mathrm{UCB}}(i) = \max C_{j,i}$ for all pairs (i, j).
- Compute any stable market outcome with respect to u^{UCB} over I^t and J^t.

An algorithmic meta-approach (MatchUCB)

Idea: adopt "optimism in the face of uncertainty" principle

(1) For each pair (i, j), keep track of confidence sets $C_{i,j}$ for $u_i(j)$ and $C_{j,i}$ for $u_j(i)$.
- Confidence sets should contain true utility values with high probability.

(2) At each round:
- Compute "upper confidence bound" estimates $u_i^{\mathrm{UCB}}(j) = \max C_{i,j}$ and $u_j^{\mathrm{UCB}}(i) = \max C_{j,i}$ for all pairs (i, j).
- Compute any stable market outcome with respect to u^{UCB} over I^t and J^t.

Key intuition behind MatchUCB

Instant-independent analysis of classical UCB boils down to:

(1) Show that regret is bounded by sum of the sizes of the confidence sets.

(2) Bound the sum of the sizes of the confidence sets.

Key intuition behind MatchUCB

Instant-independent analysis of classical UCB boils down to:

(1) Show that regret is bounded by sum of the sizes of the confidence sets.

(2) Bound the sum of the sizes of the confidence sets.

Key intuition behind MatchUCB

Instant-independent analysis of UCB boils down to:

(1) Show that regret is bounded by sum of the sizes of the confidence sets.

(2) Bound the sum of the sizes of the confidence sets.

Main lemma: Given confidence sets $C_{i,j}$ and $C_{i,j}$ for each pair (i, j), let (μ, τ) be a stable outcome with respect to the upper confidence bound estimate u^{UCB}. Then:

$$\mathrm{Instab}(\mu, \tau) \leq \sum_{(i,j)\in\mu} |\max C_{i,j} - \min C_{i,j}|.$$

Proof sketch of the main lemma

Write $\text{Instab}(\mu, \tau)$ as $\max_{S \subseteq I \cup J} f(\mu, \tau, S, u)$ where:

$$f(\mu, \tau, S, u) := \max_{\mu^*}\{\sum_{a \in S}(u_a(\mu^*(a))|\ \mu^* \text{ matching over } S\} - \sum_{a \in S}(u_a(\mu(a)) + \tau_a).$$

Proof sketch of the main lemma

Write $\text{Instab}(\mu, \tau)$ as $\max_{S \subseteq I \cup J} f(\mu, \tau, S, u)$ where:

$$f(\mu, \tau, S, u) := \max_{\mu^*}\{\sum_{a \in S}(u_a(\mu^*(a))|\ \mu^* \text{ matching over S}\} - \sum_{a \in S}(u_a(\mu(a)) + \tau_a).$$

Note that:

$$f(\mu, \tau, S, u) = f(\mu, \tau, S, u) - f(\mu, \tau, S, u^{\text{UCB}})$$

Proof sketch of the main lemma

Write Instab(μ, τ) as $\max_{S \subseteq I \cup J} f(\mu, \tau, S, u)$ where:

$$f(\mu, \tau, S, u) := \max_{\mu^*}\{\sum_{a \in S}(u_a(\mu^*(a))| \mu^* \text{ matching over S}\} - \sum_{a \in S}(u_a(\mu(a)) + \tau_a).$$

Note that:

$$f(\mu, \tau, S, u) = f(\mu, \tau, S, u) - f(\mu, \tau, S, u^{UCB})$$

$$= \max_{\mu^*}\{\sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S}\} - \max_{\mu^*}\{\sum_{a \in S} u_a^{UCB}(\mu^*(a)) \mid \mu^* \text{ matching over S}\}$$

$$+ \sum_{a \in S}(u_a^{UCB}(\mu(a)) + \tau_a) - \sum_{a \in S}(u_a(\mu(a)) + \tau_a)$$

Proof sketch of the main lemma

Write Instab(μ, τ) as $\max_{S \subseteq I \cup J} f(\mu, \tau, S, u)$ where:

$$f(\mu, \tau, S, u) := \max_{\mu^*}\{\sum_{a \in S}(u_a(\mu^*(a))|\ \mu^* \text{ matching over S}\} - \sum_{a \in S}(u_a(\mu(a)) + \tau_a).$$

Note that:

$$f(\mu, \tau, S, u) = f(\mu, \tau, S, u) - f(\mu, \tau, S, u^{UCB})$$

$$= \max_{\mu^*}\{\sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S}\} - \max_{\mu^*}\{\sum_{a \in S} u_a{}^{UCB}(\mu^*(a)) \mid \mu^* \text{ matching over S}\}$$

$$+ \sum_{a \in S}(u_a{}^{UCB}(\mu(a)) + \tau_a) - \sum_{a \in S}(u_a(\mu(a)) + \tau_a)$$

This is nonpositive.

Proof sketch of the main lemma

Write $\text{Instab}(\mu, \tau)$ as $\max_{S \subseteq I \cup J} f(\mu, \tau, S, u)$ where:

$$f(\mu, \tau, S, u) := \max_{\mu^*}\{\sum_{a \in S}(u_a(\mu^*(a))|\ \mu^* \text{ matching over S}\} - \sum_{a \in S}(u_a(\mu(a)) + \tau_a).$$

Note that:

$$f(\mu, \tau, S, u) = f(\mu, \tau, S, u) - f(\mu, \tau, S, u^{\text{UCB}})$$

$$= \max_{\mu^*}\{\sum_{a \in S} u_a(\mu^*(a)) \mid \mu^* \text{ matching over S}\} - \max_{\mu^*}\{\sum_{a \in S} u_a{}^{\text{UCB}}(\mu^*(a)) \mid \mu^* \text{ matching over S}\}$$

$$+ \sum_{a \in S}(u_a{}^{\text{UCB}}(\mu(a)) + \tau_a) - \sum_{a \in S}(u_a(\mu(a)) + \tau_a)$$

$$\leq \sum_{(i,j) \in \mu} |\max C_{i,j} - \min C_{i,j}|$$

This is nonpositive.

Bounded by confidence set size

This also yields a bound on $\text{Instab}(\mu, \tau) = \max_{S \subseteq I \cup J} f(\mu, \tau, S, u)$.

Key intuition behind MatchUCB

Instant-independent analysis of UCB boils down to:

(1) Show that regret is bounded by sum of the sizes of the confidence sets.

(2) Bound the sum of the sizes of the confidence sets.

Main lemma: Given confidence sets $C_{i,j}$ and $C_{i,j}$ for each pair (i, j), let (μ, τ) be a stable outcome with respect to the upper confidence bound estimate u^{UCB}. Then:

$$\mathrm{Instab}(\mu, \tau) \leq \sum_{(i,j)\in\mu} |\max C_{i,j} - \min C_{i,j}|.$$

Key intuition behind MatchUCB

Instant-independent analysis of UCB boils down to:

(1) Show that regret is bounded by sum of the sizes of the confidence sets.

(2) Bound the sum of the sizes of the confidence sets **[use classical approach]**

Main lemma: Given confidence sets $C_{i,j}$ and $C_{i,j}$ for each pair (i, j), let (μ, τ) be a stable outcome with respect to the upper confidence bound estimate u^{UCB}. Then:

$$\mathrm{Instab}(\mu, \tau) \leq \sum_{(i,j)\in\mu} |\max C_{i,j} - \min C_{i,j}|.$$

Results: No-Regret Algorithms

Theorem 1: There exists an algorithm incurring $\tilde{O}(N^{3/2}T^{1/2})$ instance-independent regret with N agents over T rounds.

We further show that the instance-independent regret bound in Theorem 1 is *optimal* up to log factors.

We can also achieve instance-dependent regret bounds:

Theorem 2: The same algorithm incurs $\tilde{O}(N^5 \log(T)/\Delta^2)$ regret with N agents over T rounds where Δ is a measure of instance-dependent gap.

Results: No-Regret Algorithms

Theorem 1: There exists an algorithm incurring $\tilde{O}(N^{3/2}T^{1/2})$ instance-independent regret with N agents over T rounds.

We further show that the instance-independent regret bound in Theorem 1 is *optimal* up to log factors.

We can also achieve instance-dependent regret bounds:

Theorem 2: The same algorithm incurs $\tilde{O}(N^5 \log(T)/\Delta^2)$ regret with N agents over T rounds where Δ is a measure of instance-dependent gap.

Proof ideas for instance-dependent regret

High-level approach: count the number of “mistakes” that the algorithm makes.

Proof ideas for instance-dependent regret

High-level approach: count the number of “mistakes” that the algorithm makes.

Challenge: mistakes arise if the matching is correct but the transfers are wrong.

Proof ideas for instance-dependent regret

High-level approach: count the number of "mistakes" that the algorithm makes.

Challenge: mistakes arise if the matching is correct but the transfers are wrong.

Obtaining the "right" transfers requires more structure than in MatchUCB.

An algorithmic meta-approach (MatchUCB)

Idea: adopt "optimism in the face of uncertainty" principle

(1) For each pair (i,j), keep track of confidence sets $C_{i,j}$ for $u_i(j)$ and $C_{j,i}$ for $u_j(i)$.
- Confidence sets should contain true utility values with high probability.

(2) At each round:
- Compute "upper confidence bound" estimates $u_i^{\mathrm{UCB}}(j) = \max C_{i,j}$ and $u_j^{\mathrm{UCB}}(i) = \max C_{j,i}$ for all pairs (i, j).
- Compute **any** stable market outcome with respect to u^{UCB} over I^t and J^t.

Proof ideas for instance-dependent regret

High-level approach: count the number of "mistakes" that the algorithm makes.

Challenge: mistakes arise if the matching is correct but the transfers are wrong.

Obtaining the "right" transfers requires more structure than in MatchUCB.

We need to choose a *specific* stable market outcome for UCB preferences

Proof ideas for instance-dependent regret

High-level approach: count the number of "mistakes" that the algorithm makes.

Challenge: mistakes arise if the matching is correct but the transfers are wrong.

Obtaining the "right" transfers requires more structure than in MatchUCB.

We need to choose a *specific* stable market outcome for UCB preferences

- Leverage the primal-dual formulation of stable matchings (Shapley and Shubik, 1971).

Proof ideas for instance-dependent regret

High-level approach: count the number of "mistakes" that the algorithm makes.

Challenge: mistakes arise if the matching is correct but the transfers are wrong.

Obtaining the "right" transfers requires more structure than in MatchUCB.

We need to choose a *specific* stable market outcome for UCB preferences

- Leverage the primal-dual formulation of stable matchings (Shapley and Shubik, 1971).
- Leave slack in constraints to find a "robust" dual solution (and "robust" transfers).

Results: No-Regret Algorithms

Theorem 1: There exists an algorithm incurring $\tilde{O}(N^{3/2}T^{1/2})$ instance-independent regret with N agents over T rounds.

We further show that the instance-independent regret bound in Theorem 1 is *optimal* up to log factors (see paper).

We can also achieve instance-dependent regret bounds:

Theorem 2: The same algorithm incurs $\tilde{O}(N^5 \log(T)/\Delta^2)$ regret with N agents over T rounds where Δ is a measure of instance-dependent gap.

Results: No-Regret Algorithms

Theorem 1: There exists an algorithm incurring $\tilde{O}(N^{3/2}T^{1/2})$ instance-independent regret with N agents over T rounds.

We further show that the instance-independent regret bound in Theorem 1 is *optimal* up to log factors (see paper).

We can also achieve instance-dependent regret bounds:

Theorem 2: The same algorithm incurs $\tilde{O}(N^5 \log(T)/\Delta^2)$ regret with N agents over T rounds where Δ is a measure of instance-dependent gap.

What about the dependence on N?

Results: Preference Structure

For arbitrary preferences, regret grows *super-linearly* in the size N of the market…

When can we do better (i.e., obtain $\propto N$ regret)?

Results: Preference Structure

For arbitrary preferences, regret grows *super-linearly* in the size N of the market…

When can we do better (i.e., obtain $\propto N$ regret)?

Consider two models of preference structure:

1. **Typed preferences**: each agent belongs to one of finitely many *types*; all agents of the same type have the same preferences
    - Can implement MatchUCB with confidence bounds that account for “delayed” feedback

Results: Preference Structure

For arbitrary preferences, regret grows *super-linearly* in the size N of the market…

When can we do better (i.e., obtain $\propto N$ regret)?

Consider two models of preference structure:

1. **Typed preferences**: each agent belongs to one of finitely many *types*; all agents of the same type have the same preferences
    - Can implement MatchUCB with confidence bounds that account for "delayed" feedback
2. **Linear preferences**: each agent's utilities are a linear function of the opposite side's context
    - Can implement with MatchUCB with confidence bounds based on LinUCB

Regret bounds for different preference structures

N =# of agents on platform; *n* = *# of agents who arrive each round*;
C is set of types; *d is preference dimension*

Unstructured preferences	$\tilde{O}(N n^{1/2} T^{1/2})$
Typed preferences	$\tilde{O}(\lvert C \rvert n^{1/2} T^{1/2})$
Linear preferences	$\tilde{O}(d N^{1/2} n^{1/2} T^{1/2})$

Outline for the rest of the talk

1. Background on stable matchings with transfers (Shapley, Shubik ‘71)

2. Our framework for learning stable market outcomes

3. Designing algorithms to learn stable market outcomes

4. **Extensions and Conclusion**

Extensions and Conclusion

Extension 1: Connection to Platform Profit

We interpret learning objective as platform profit.

Idea: use the interpretation of Subset Instability as "minimum stabilizing subsidies"

- Min amount the platforms needs to pay agents to make stable
- We envision that this is the "cost" of the platform to retaining agents.

We introduce "search frictions" (agents have to pay a cost to find alternate matchings).

Takeaway: the platform can earn a profit in the long-run.

Extension 2: Matching without Transfers

We extend our framework to matching without transfers (Gale, Shapley '62).

We propose a measure of instability based on the "minimum stability subsidies".

- Equals 0 for *any* stable matching
- Has significant advantages over "utility difference" considered in prior work (*e.g., Das, Kamenica (IJCAI '05), Liu, Mania, Jordan (AISTATS '20)*).

We show a variant of MatchUCB yields $\tilde{O}(N^{3/2}T^{1/2})$ regret in this setting.

Takeaway: our framework & algorithms apply to matchings without transfers

Summary of our contributions

We presented **an incentive-aware framework** for learning stable matchings that builds on the bandits literature.

(1) We proposed **Subset Instability** as a learning objective for the platform.
(2) We constructed an **algorithmic meta-approach** based on UCB, and we achieved **instance-independent and instance-dependent regret bounds**.
(3) We applied this algorithm principle to **different preference structures**.
(4) We also connected our results to platform profit and investigated matchings without transfers.

How can equilibria be efficiently learned?

Our core insight: in a stochastic setting, UCB-based algorithms can be adapted to learn stable matchings.

General question: In what other settings (and with what other algorithms) can equilibria be learned?

Direction 1: Learn stable matchings in more general environments

- Setting where utilities can dynamically change? (e.g. (Min, Wang, Xu, Wang, Jordan, Yang, 2022))
- Adversarial settings?
- A platform objective that captures other aspects of the marketplace (e.g. competing platforms)?

Direction 2: Learning equilibria in more general matching markets or market settings

- Learning competitive equilibria? (e.g. (Gu, Kandaswamy, Jordan, 2021), (Liu, Lu, Wang, Jordan, Yang, 2022))
- Learning equilibria in many-to-many matching markets?