上海交通大学
约翰·霍普克罗夫特
计算机科学中心
John Hopcroft Center for Computer Science

Player-optimal Stable Regret for Bandit Learning in Matching Markets

(Accepted at SODA 2023)

Fang Kong, Shuai Li

Emerging online matching platforms

upwork™

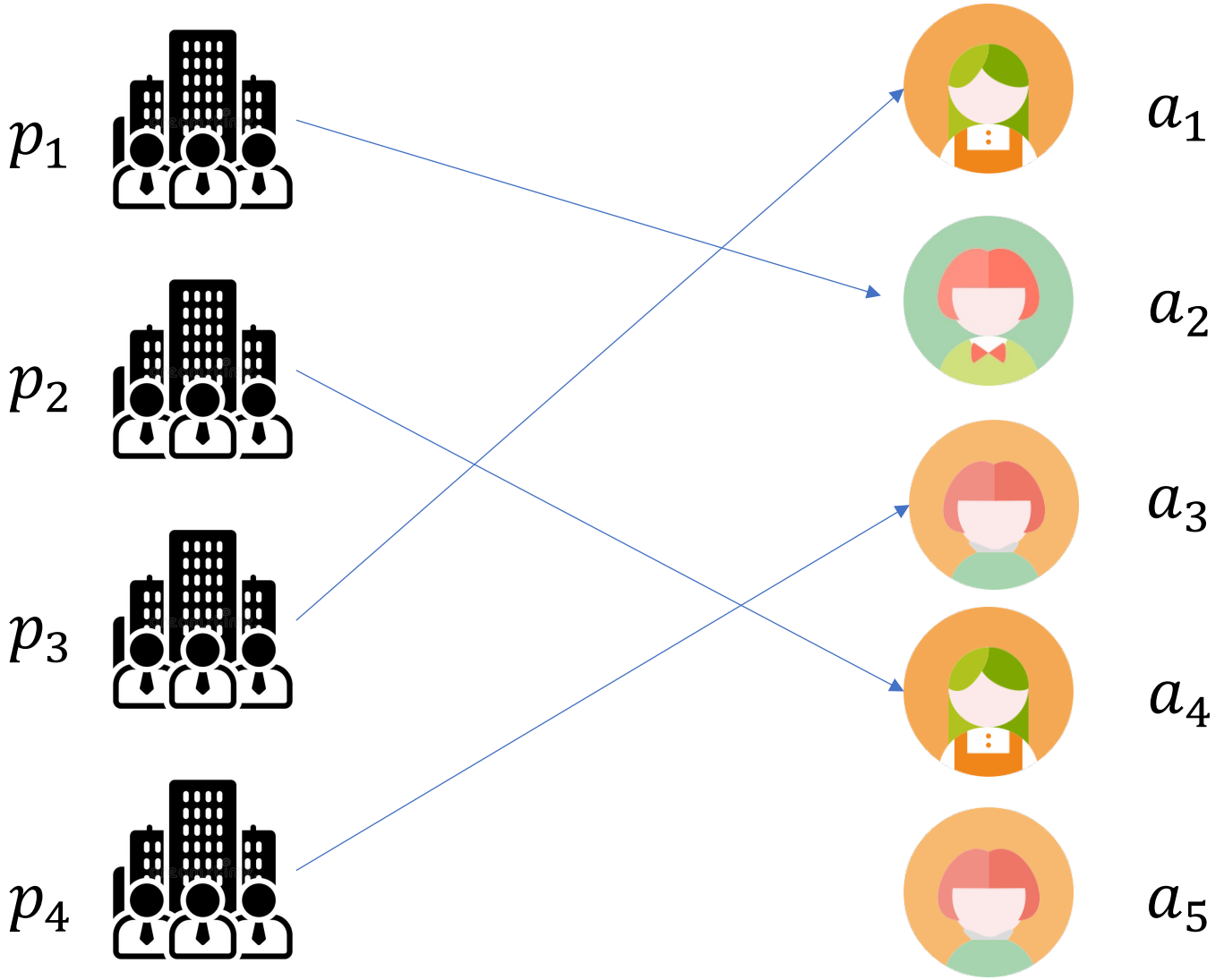
amazon beta mechanical turk

match.

TaskRabbit

Two-sided matching markets

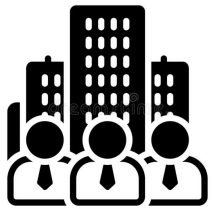
employers
(players)

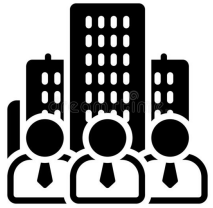


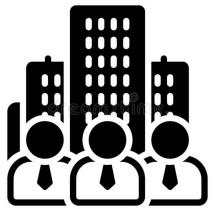
workers
(arms)

Players' preferences (based on the skill levels of workers

p_1  $a_1 > a_2 > a_3 > a_4 > a_5$

p_2  $a_2 > a_1 > a_4 > a_3 > a_5$

p_3  $a_3 > a_1 > a_2 > a_5 > a_4$

p_4  $a_4 > a_5 > a_1 > a_2 > a_3$

Arms' preferences (based on payment or prior familiarity of the task

a_1  $p_2 > p_3 > p_1 > p_4$

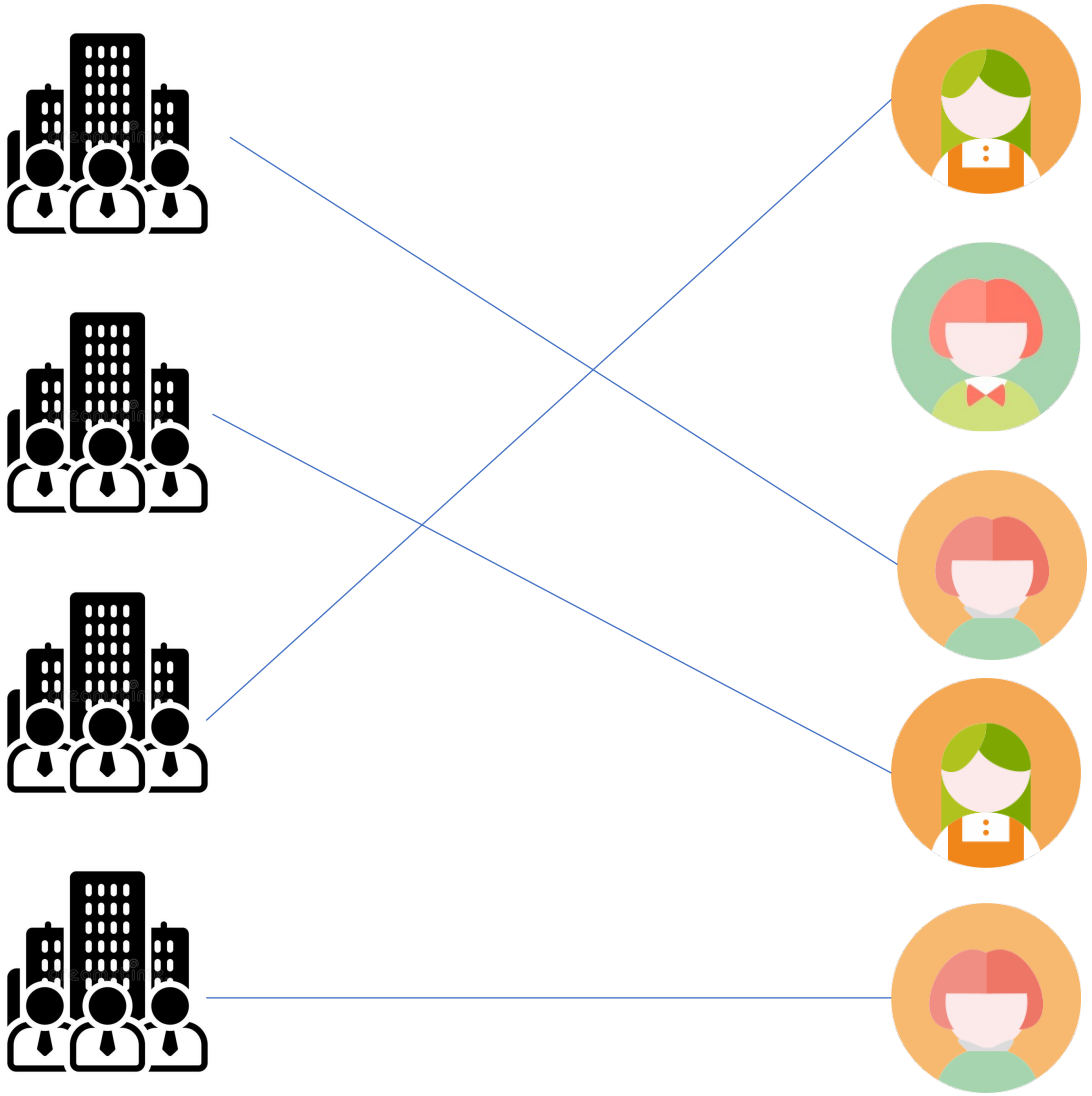
a_2  $p_1 > p_2 > p_3 > p_4$

a_3  $p_3 > p_1 > p_2 > p_4$

a_4  $p_1 > p_2 > p_3 > p_4$

a_5  $p_1 > p_2 > p_3 > p_4$

Stable matching



Participants have no incentive to abandon their current partner,

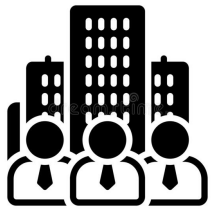
i.e.,

no pair of participants such that they both prefer to be matched with each other than their current partner

May be more than one stable matchings

p_1

$a_1 > a_2 > a_3$

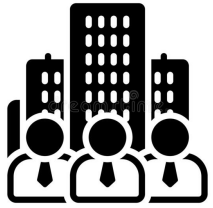


a_1

$p_2 > p_3 > p_1$

p_2

$a_2 > a_1 > a_3$

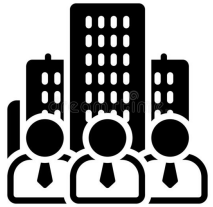


a_2

$p_1 > p_2 > p_3$

p_3

$a_3 > a_1 > a_2$



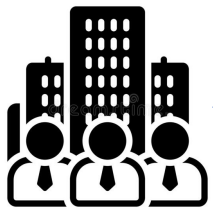
a_3

$p_3 > p_1 > p_2$

$$m_1 = \{(p_1, a_1), (p_2, a_2), (p_3, a_3)\}$$

$$m_2 = \{(p_1, a_2), (p_2, a_1), (p_3, a_3)\}$$

Player-optimal stable matching

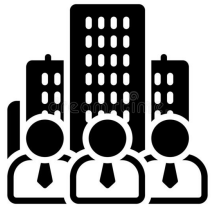


p_1

$a_1 > a_2 > a_3$

a_1

$p_2 > p_3 > p_1$

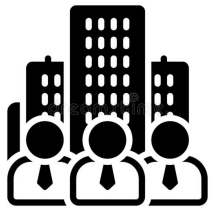


p_2

$a_2 > a_1 > a_3$

a_2

$p_1 > p_2 > p_3$



p_3

$a_3 > a_1 > a_2$

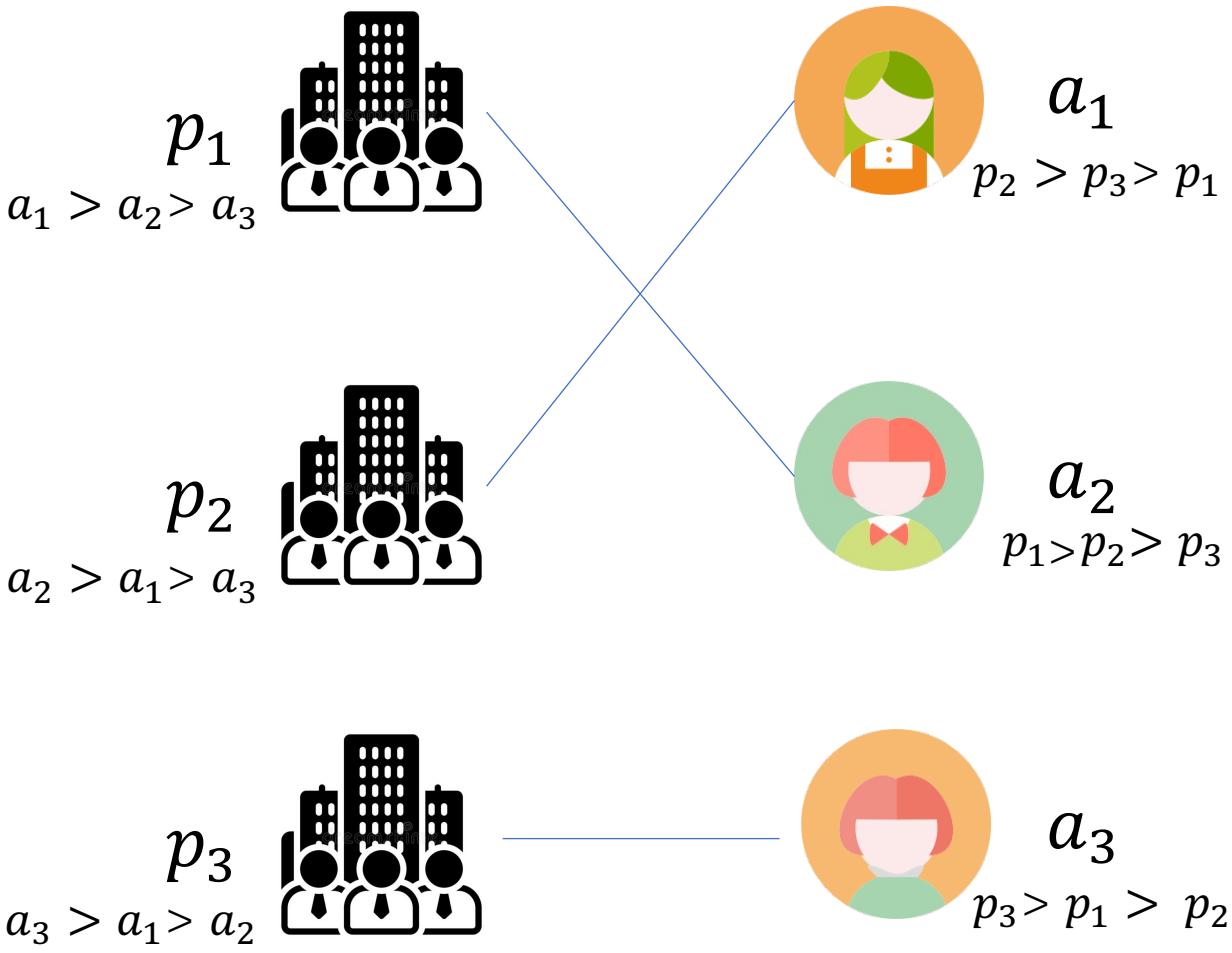
a_3

$p_3 > p_1 > p_2$

The player is matched with the most preferred arm among all stable matchings

$$\{(p_1, a_1), (p_2, a_2), (p_3, a_3)\}$$

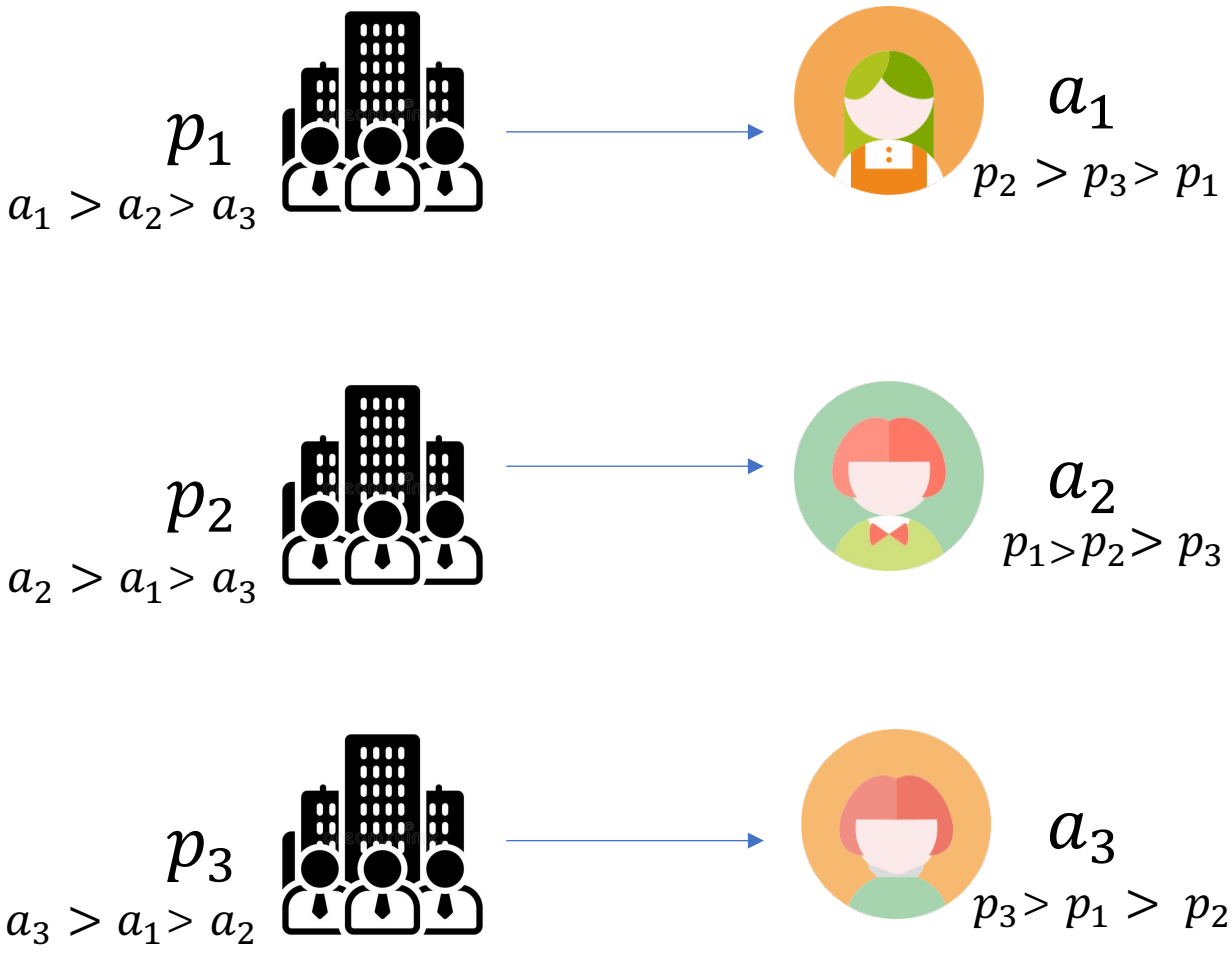
Player-pessimal stable matching



The player is matched with the least preferred arm among all stable matchings

$$\{(p_1, a_2), (p_2, a_1), (p_3, a_3)\}$$

How to find a stable matching?

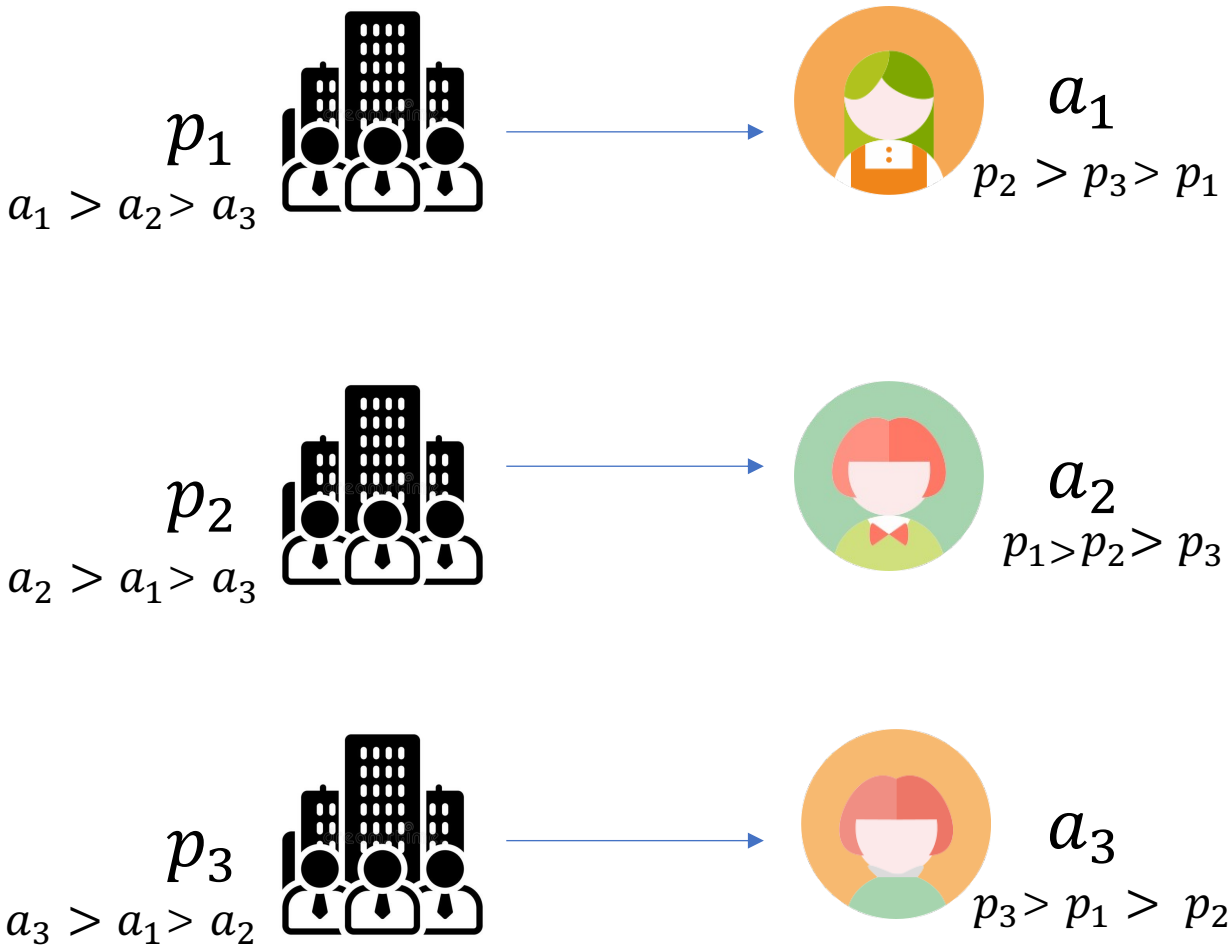


Gale-Shapley (GS) algorithm [1962]:

players independently propose to arms according to their preference order until no rejection happens

Also the player-optimal stable matching!

Gale-Shapley (GS) algorithm

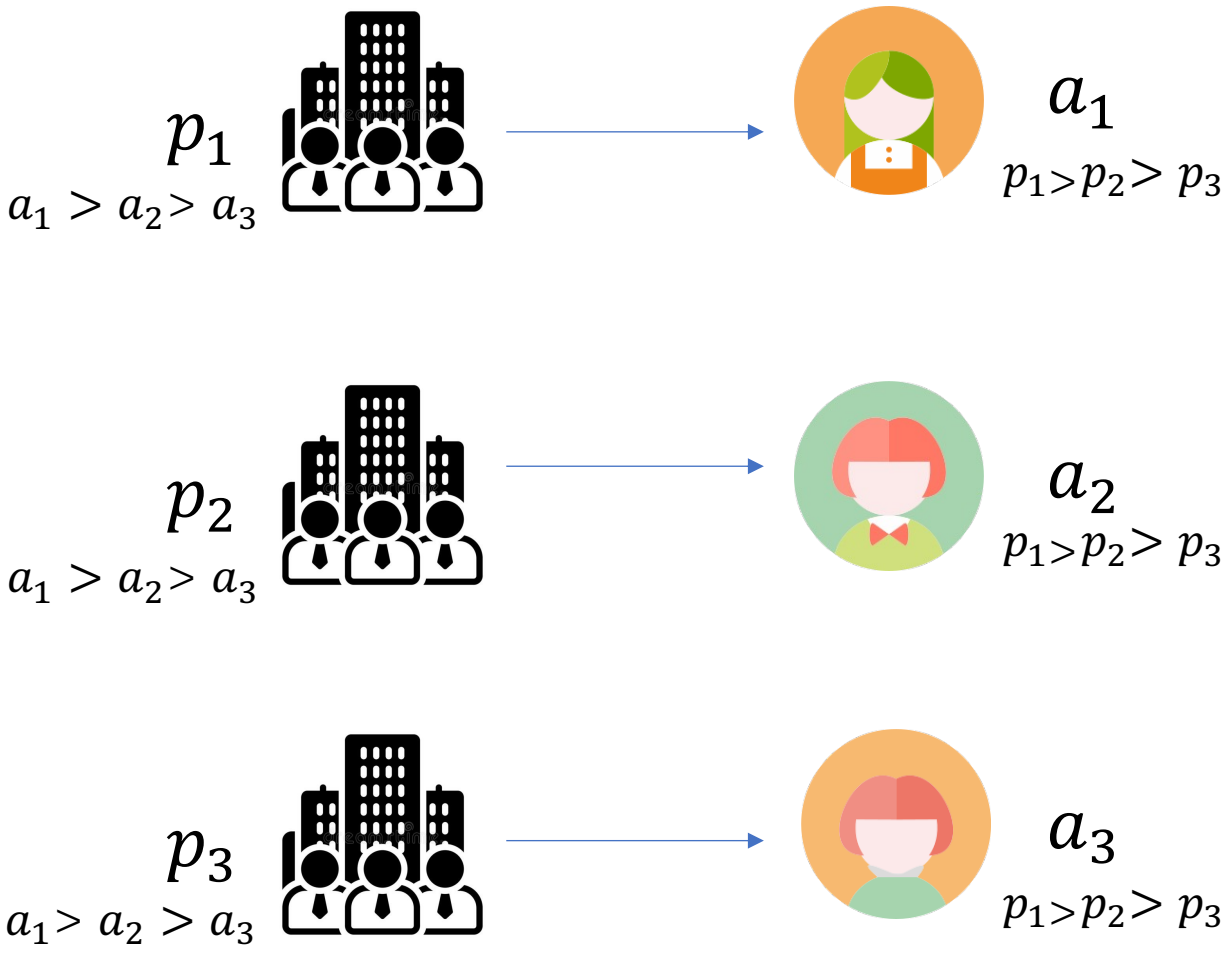


Step 1: p_1 selects a_1
p_2 selects a_2
p_3 selects a_3

No rejection happens!

Find the player-optimal stable matching

Gale-Shapley (GS) algorithm: Case 2



Step 1: p_1 selects a_1
p_2 selects a_1
p_3 selects a_1
[p_2 and p_3 are rejected]
Step 2: p_1 selects a_1
p_2 selects a_2
p_3 selects a_2
[p_3 is rejected]
Step 3: p_1 selects a_1
p_2 selects a_2
p_3 selects a_3
[no rejection happens]

Find the stable matching
$\{(p_1, a_1), (p_2, a_2), (p_3, a_3)\}$

Gale-Shapley (GS) algorithm 3

- Who proposes matters
  - Among all stable matchings
  - every player is happiest in the one produced by the player-proposal algorithm
  - every arm is happiest under the arm-proposal algorithm

- Each arm can reject each player for at most once
- At least one rejection happens at each step before stop
- Denote N as the number of players, K as the number of arms
- GS will stop in at most NK steps

But players usually have unknown preferences in practice

upwork™

match.

amazon mechanical turk beta

TaskRabbit

Can learn them from iterative interactions !

Online matching markets

- N players, K arms
- $\mu_{i,j} > 0$: the satisfaction of player p_i towards arm a_j
- For each player p_i
  - $\{\mu_{i,j}\}_{j\in[K]}$ forms its preference ranking
  - For simplicity, the preference values of a player are distinct
- For each round t:
  - Player p_i selects arm $A_i(t)$
  - If p_i is matched: receive $X_{i,A_i(t)}(t)$ with

$$\mathbb{E}\left[X_{i,A_i(t)}(t)\right] = \mu_{i,A_i(t)}$$

  - If p_i is not matched: receive $X_{i,A_i(t)}(t) = 0$

the satisfaction over this matching experience

Objective: Minimize the stable regret

- The player-optimal stable matching

$$m^* = \{(i, m_i^*) : i \in [N]\}$$

- The player-optimal stable regret of player p_i is

$$\mathrm{Reg}_i(T) = T\mu_{i,m_i^*} - \mathbb{E}\left[\sum_{t=1}^{T} X_{i,A_i(t)}(t)\right]$$

Multi-armed bandits (MAB)

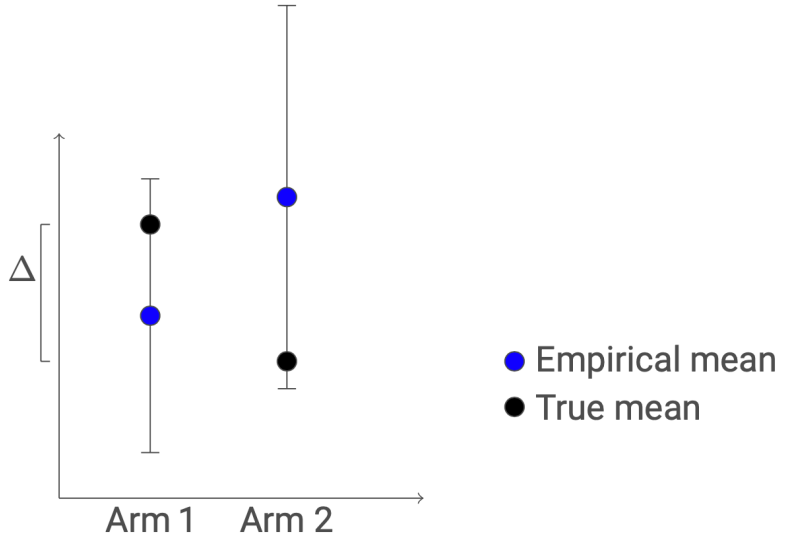
- A classic framework charactering the learning process from iterative interactions

Time	1	2	3	4	5	6	7	8	9	10	11	12
Left arm	$1	$0			$1	$1	$0					
Right arm			$1	$0								

- Can be regarded as a market with only $N = 1$ player and K arms

- Lattimore, Tor, and Csaba Szepesvári. Bandit Algorithms. Cambridge University Press, 2020

The UCB algorithm for MAB



- With high probability $\geq 1-\delta$, for each arm j

$$\mu_j \in \left[\hat{\mu}_j - \sqrt{\frac{\log 1/\delta}{T_j}}, \hat{\mu}_j + \sqrt{\frac{\log 1/\delta}{T_j}}\right]$$

- For each round t, select the arm

$$A(t) \in \operatorname{argmax}_{j\in[K]} \left\{\hat{\mu}_j + \sqrt{\frac{\log 1/\delta}{T_j}}\right\}$$

- Each sub-optimal arm $j \in [K]$ is chosen $O\left(\frac{\log T}{\Delta_j^2}\right)$ times, where $\Delta_j = \max_i \mu_i - \mu_j$
- $\mathrm{Reg}(T) = O(K \log T/\Delta)$

- Lattimore, Tor, and Csaba Szepesvári. Bandit Algorithms. Cambridge University Press, 2020

Challenge in online matching markets

Other players will block observations!

Centralized VS. Decentralized

- Centralized
  - All participants submit their estimations to the platform
  - The platform computes an assignment
  - All players follow this assignment

- Decentralized
  - Each player independently computes the target arm
  - Available information:
    - common index of arms, successful matching results in each round

Previous works for online matching markets

	Regret bound	Setting
Liu *et al.* [2020]	$O\left(K \log T/\Delta^2\right)$	player-optimal, centralized, known T, Δ
	$O\left(NK^3 \log T/\Delta^2\right)$	player-pessimal, centralized
Liu *et al.* [2021]	$O\left(\frac{N^5K^2 \log^2 T}{\varepsilon^{N^4}\Delta^2}\right)$	player-pessimal
Sankararaman *et al.* [2021]	$O\left(NK \log T/\Delta^2\right)$	unique stable matching
	$\Omega\left(N \log T/\Delta^2\right)$	
Basu *et al.* [2021]	$O\left(K \log^{1+\varepsilon} T + 2^{\left(\frac{1}{\Delta^2}\right)^{\frac{1}{\varepsilon}}}\right)$	player-optimal
	$O\left(NK \log T/\Delta^2\right)$	unique stable matching
Kong *et al.* [2022]	$O\left(\frac{N^5K^2 \log^2 T}{\varepsilon^{N^4}\Delta^2}\right)$	player-pessimal
Maheshwari *et al.* [2022]	$O\left(CNK \log T/\Delta^2\right)$	unique stable matching

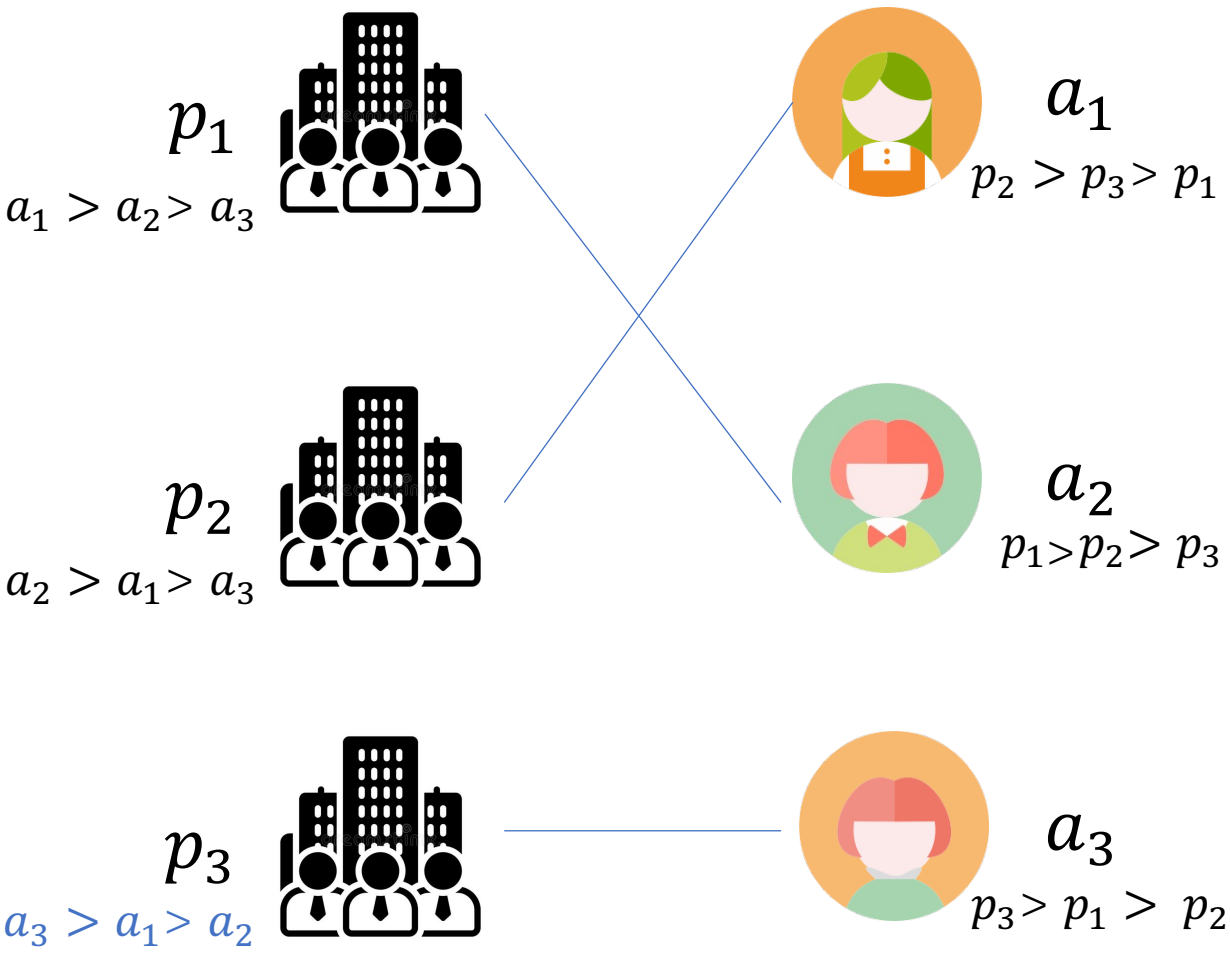
Δ is the minimum preference gap between different arms among all players, ε is the hyper-parameter of the algorithm, C is related to the unique stable matching condition and can grow exponentially in N

Why some previous work fail to achieve player-optimality?

- Example: Centralized UCB algorithm in Liu et al., [2020]

- For round $t = 1,2, \dots,$
  - Each player estimates a UCB ranking towards all arms
  - The GS platform returns an assignment under these UCB rankings
  - Each player selects the assigned arm

Analysis of failure to achieve player-optimal stable matching



- when p_3 lacks exploration on a_1 with $a_1 > a_3 > a_2$ on UCB, GS outputs the matching[1] $(p_1, a_2), (p_2, a_1), (p_3, a_3)$
- p_3 fails to observe a_1

- UCB vectors do not help on exploration here

- Not consistent with the principle of *optimism in face of uncertainty*

1. When p_1 and p_2 submit the correct rankings

Algorithm design idea

- Exploration-Exploitation trade-off
  - Exploitation goes though with correct rankings
  - Require enough exploration
- The UCB ranking does not guarantee enough exploration
- Perhaps design manually?
- To avoid other players' block: Arrange selections in a round-robin way

Algorithm design

- //Phase 1, exploration for good ranking
- For round $t = 1, 2, \ldots$
  - For each player p_i:
    - $A_i(t) = a_{(i+t) \bmod K}$ //arranged to be matched successfully
    - Observe $X_{i,A_i(t)}(t)$, update $\hat{\mu}_{i,A_i(t)}$ and $T_{i,A_i(t)}$
  - Break if all players have a good preference ranking → For any player p_i, there exists a ranking σ_i over arms such that $\mathrm{LCB}_{i,\sigma_{i,k}} > \mathrm{UCB}_{i,\sigma_{i,k+1}}$, for any $k \in [K-1]$
- //Phase 2, exploitation
- Follow GS with the estimated preference ranking

How to determine that all players have a good ranking?

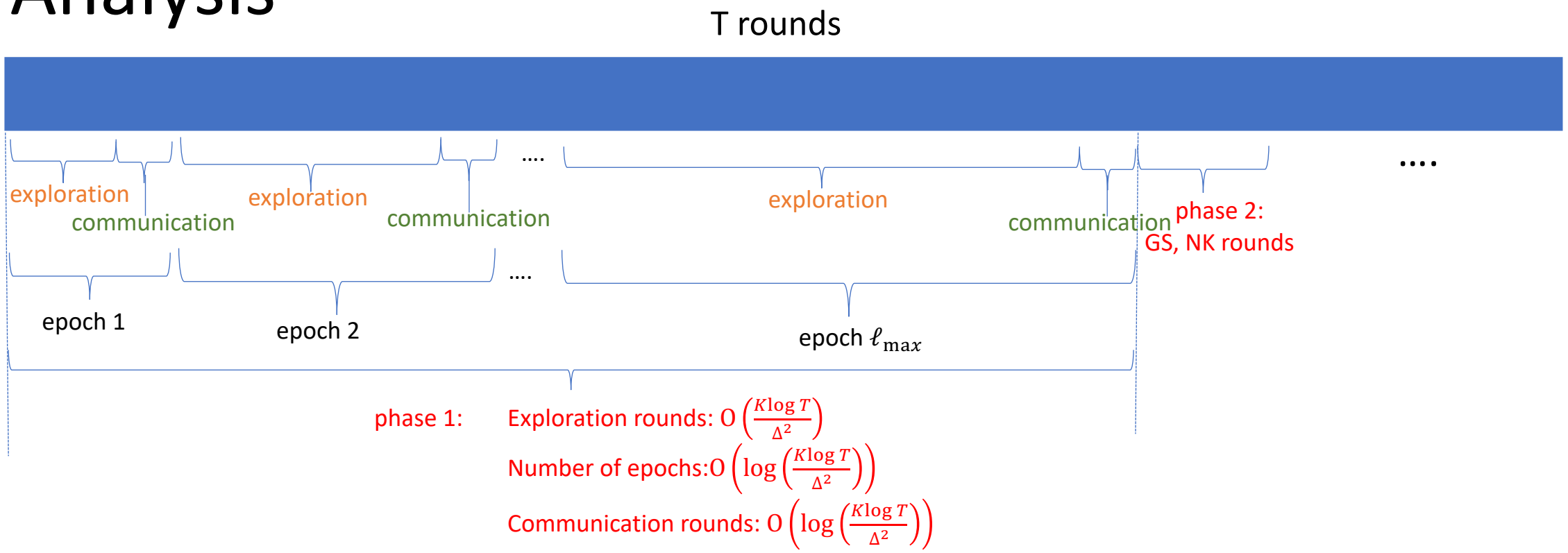
- //Phase 1, exploration for good ranking
- For epoch $\ell = 1,2, \ldots$
  - For round $t = 2^{\ell} + (\ell - 1), \ldots, 2^{\ell} + (\ell - 1) + 2^{\ell}$ do
    - $A_i(t) = a_{(i+t) \bmod K}$
    - Observe $X_{i,A_i(t)}(t)$, update $\hat{\mu}_{i,A_i(t)}$ and $T_{i,A_i(t)}$
  - Compute the $\mathrm{UCB}_{i,j}$ and $\mathrm{LCB}_{i,j}$ for each arm $j \in [K]$
  - At round $t = 2^{\ell} + (\ell - 1) + 2^{\ell} + 1$
    - If p_i has a good ranking σ_i: select arm a_i
    - Else: give up the chance of selecting arms

If p_i observes that all players have been matched with each arm for once in this round:
Go to next phase and set $\ell_{\max} = \ell$

Find the player-optimal stable matching with the estimated preference ranking

- //Phase 2, exploitation
- // Follow GS to find the stable matching with the estimated ranking σ
- Initialize $s_i = 1$ for each player p_i
  - //the ranking index of the most preferred arm who have not rejected p_i previously
- For $t = 2^{\ell_{\max}+1} + \ell_{\max}, \dots,$
  - For each player p_i
    - $A_i(t) = \sigma_{s_i}$
    - If p_i is not matched, $s_i = s_i + 1$

Analysis



The player-optimal stable regret of each player p_i over T rounds can be upper bounded as

$$\mathrm{Reg}_i(T) \leq O\left(\frac{K\log T}{\Delta^2} + \log\left(\frac{K\log T}{\Delta^2}\right) + NK\right) \cdot \Delta_{\mathrm{i,max}} = O\left(\frac{K\log T}{\Delta^2}\right)$$

$\Delta = \min_{i,j,j':\mu_{i,j}\neq\mu_{i,j'}} |\mu_{i,j} - \mu_{i,j'}|$ is the minimum preference gap between different arms among all players and

$\Delta_{\mathrm{i,max}} = \max_j \mu_{i,j}$ is the maximum regret that player p_i pays in each round

Results

	Regret bound	Setting
Liu *et al.* [2020]	$O\left(K \log T/\Delta^2\right)$ $O\left(NK^3 \log T/\Delta^2\right)$	player-optimal, centralized, known T, Δ player-pessimal, centralized
Liu *et al.* [2021]	$O\left(\frac{N^5 K^2 \log^2 T}{\varepsilon^{N^4} \Delta^2}\right)$	player-pessimal
Sankararaman *et al.* [2021]	$O\left(NK \log T/\Delta^2\right)$ $\Omega\left(N \log T/\Delta^2\right)$	unique stable matching
Basu *et al.* [2021]	$O\left(K \log^{1+\varepsilon} T + 2^{\left(\frac{1}{\Delta^2}\right)^{\frac{1}{\varepsilon}}}\right)$ $O\left(NK \log T/\Delta^2\right)$	player-optimal unique stable matching
Kong *et al.* [2022]	$O\left(\frac{N^5 K^2 \log^2 T}{\varepsilon^{N^4} \Delta^2}\right)$	player-pessimal
Maheshwari *et al.* [2022]	$O\left(CNK \log T/\Delta^2\right)$	unique stable matching
ours	$O\left(K \log T/\Delta^2\right)$	player-optimal

Future work

- ‘communication’-free algorithms to achieve player-optimal stable matching?
- Many-to-one matching markets (or combinatorial preferences)?

References

- [Gale and Shapley, 1962] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. The American Mathematical Monthly, 69(1):9–15, 1962
- [Liu et al., 2020] Lydia T Liu, Horia Mania, and Michael Jordan. Competing bandits in matching markets. In International Conference on Artificial Intelligence and Statistics, pages 1618–1628. PMLR, 2020.
- [Liu et al., 2021] Lydia T Liu, Feng Ruan, Horia Mania, and Michael I Jordan. Bandit learning in decentralized matching markets. Journal of Machine Learning Research, 22(211):1–34, 2021.
- [Sankararaman et al., 2021] Abishek Sankararaman, Soumya Basu, and Karthik Abinav Sankararaman. Dominate or delete: Decentralized competing bandits in serial dictatorship. In International Conference on Artificial Intelligence and Statistics, pages 1252–1260. PMLR, 2021.
- [Basu et al., 2021] Soumya Basu, Karthik Abinav Sankararaman, and Abishek Sankararaman. Beyond log2(t) regret for decentralized bandits in matching markets. In International Conference on Machine Learning, pages 705–715, 2021.