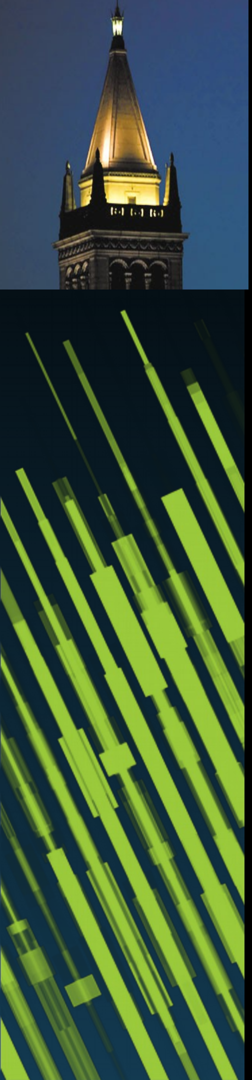


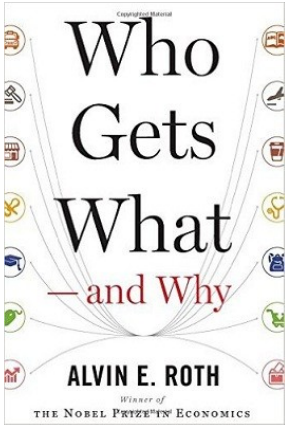
DS 102: Two-sided markets and bandits

Lecture 20

Michael Jordan | Horia Mania | Lydia T. Liu

University of California, Berkeley

Data, Decisions...and Economics



Copyrighted Material

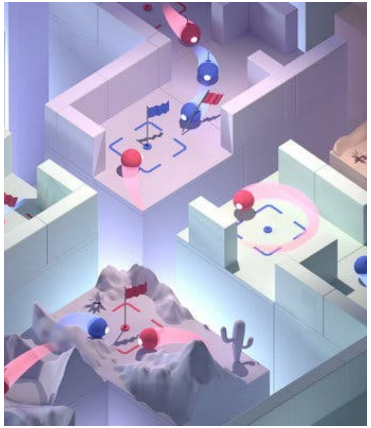
Who Gets What
—and Why

ALVIN E. ROTH

Winner of
THE NOBEL PRIZE IN ECONOMICS

Matching Markets

Auctions



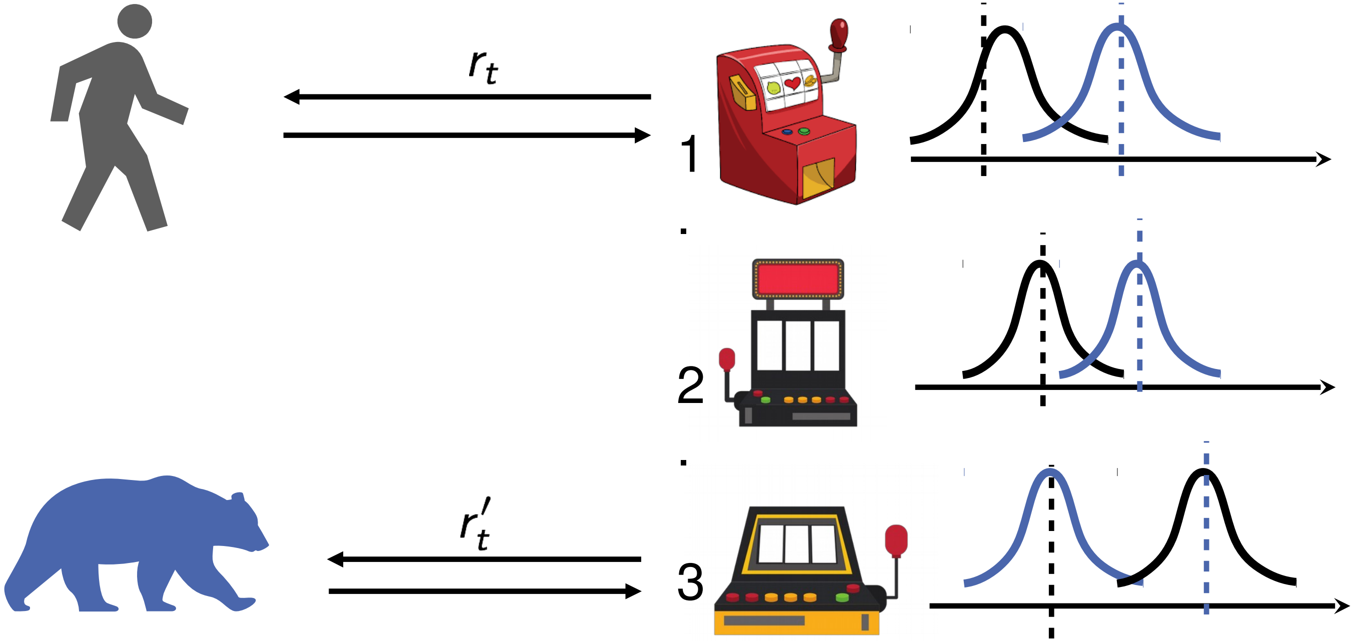
Multiplayer games

Decisions and Learning

- So far we have been concerned with single decision makers.
- In this lecture we would like to understand interactions between multiple decision makers.
- We first discuss market decisions in the absence of learning, when participants in the market have complete information.
- Finally, we discuss exploration-exploitation tradeoffs in markets.

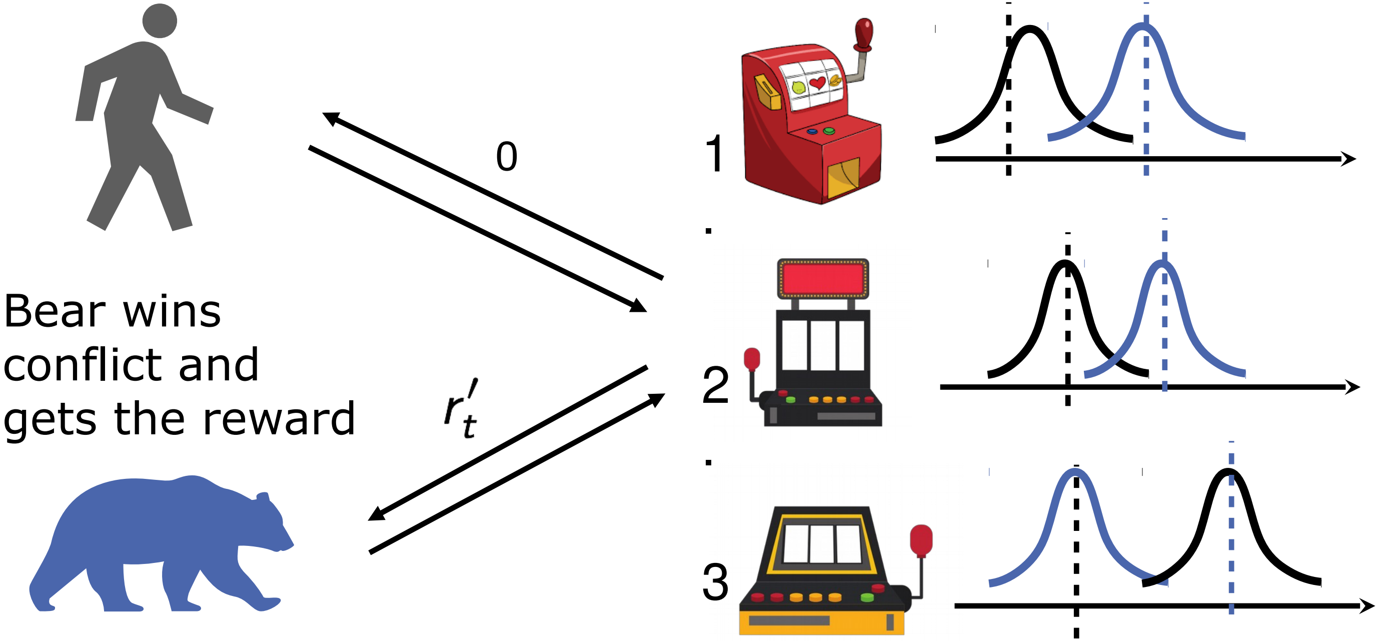
Multi-Player Bandits

Let’s add a competing player!



Bandits

Let’s add a competing player!

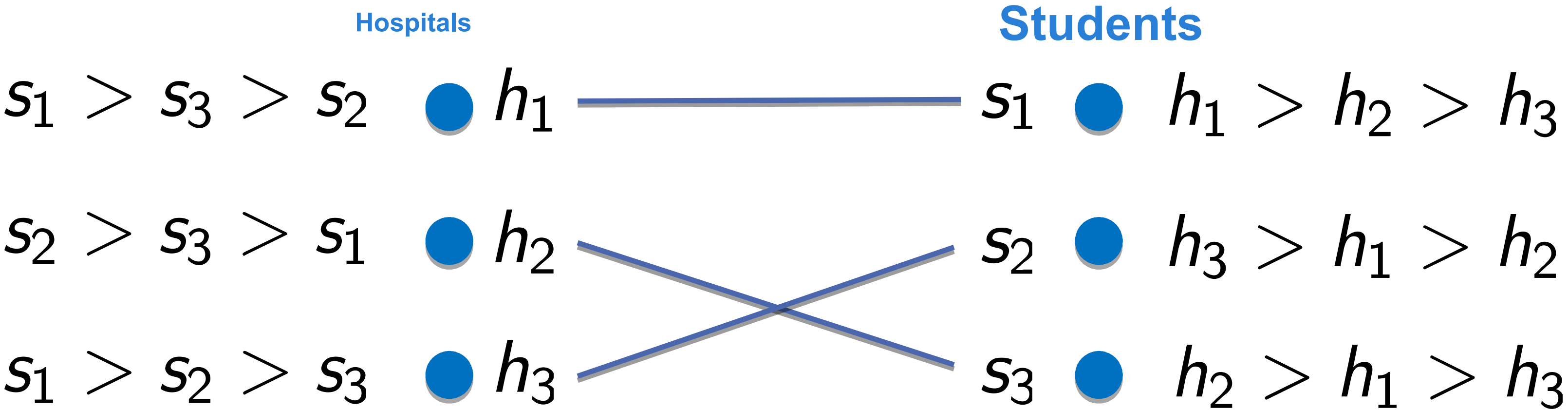


Bear wins conflict and gets the reward

Two-sided markets
(med-school students and hospitals)

- Markets often have two sides, supply and demand, which must be matched.
- Matching med-school students and hospitals is a classic example (Roth 1984).
- Med-school students have preferences over hospitals and hospitals have preferences over med-school students.

Two-sided markets (med-school students and hospitals)



Hospitals

$s_1 > s_3 > s_2$ h_1

$s_2 > s_3 > s_1$ h_2

$s_1 > s_2 > s_3$ h_3

Students

s_1 $h_1 > h_2 > h_3$

s_2 $h_3 > h_1 > h_2$

s_3 $h_2 > h_1 > h_3$

How should we match supply and demand?

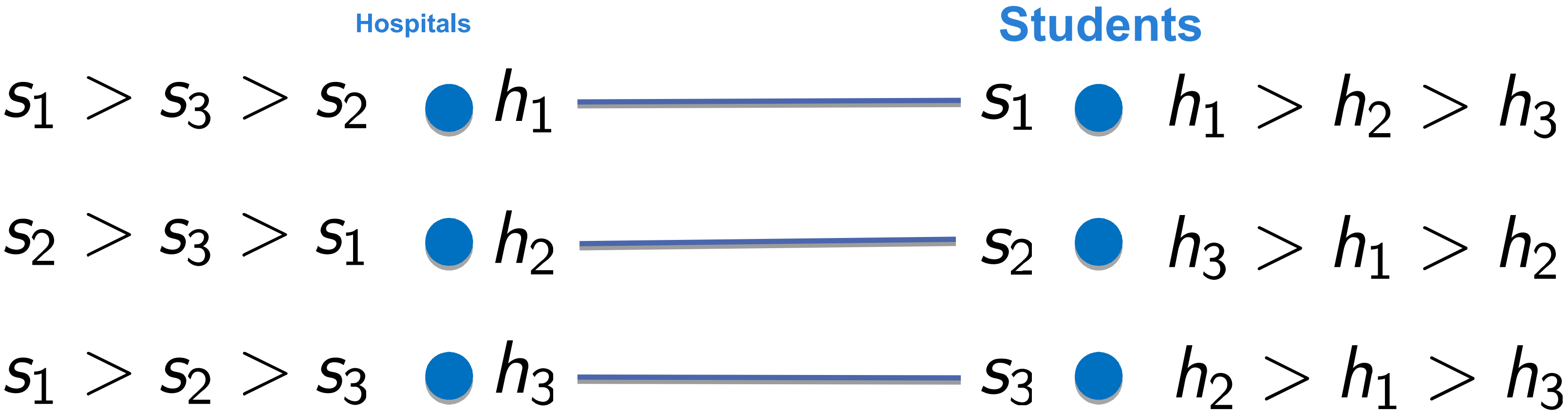
- **Definition:** a **matching** **M** is a set of pairs **(h, s)** such that
  - Each hospital **h** appears in at most one pair of **M**.
  - Each students **s** appears in at most one pair of **M**.

- **Goal:** given a set of preferences among supply and demand, determine a matching that is an equilibrium.

- **Definition:** a **blocking pair (h, s)** is a pair such that:
  - **h** prefers **s** to its current match
  - **s** prefers **h** to its current match
  - (if someone is not matched with anyone, we think of them as being matched with themselves)

Stable matchings

- **Definition:** a **stable matching** is a matching with no blocking pairs.
- **Question:** is the following matching a stable matching?



Hospitals

Students

$s_1 > s_3 > s_2$ h_1 — s_1 $h_1 > h_2 > h_3$

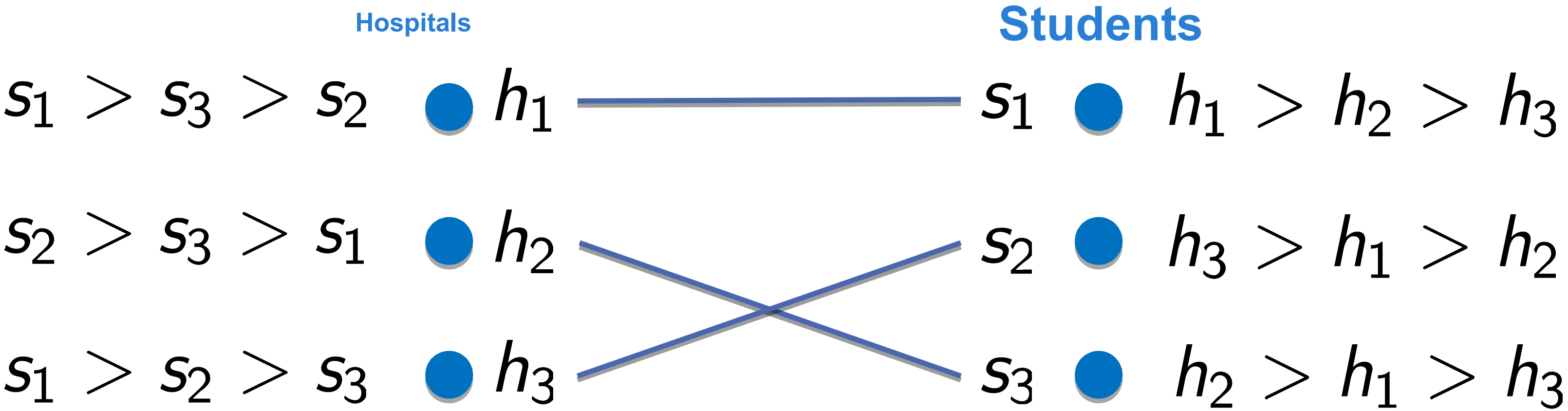
$s_2 > s_3 > s_1$ h_2 — s_2 $h_3 > h_1 > h_2$

$s_1 > s_2 > s_3$ h_3 — s_3 $h_2 > h_1 > h_3$

- **Answer: No,** (h_3, s_2) is a blocking pair.

Finding stable matchings

- **Observation:** the following matching is a **stable matching**.



Hospitals

Students

$s_1 > s_3 > s_2$ h_1 — s_1 $h_1 > h_2 > h_3$

$s_2 > s_3 > s_1$ h_2 — s_2 $h_3 > h_1 > h_2$

$s_1 > s_2 > s_3$ h_3 — s_3 $h_2 > h_1 > h_3$

- **Question:** can we always find a stable matching?

- **Answer:** In 1962 Gale and Shapley showed that a natural algorithm always finds a stable matching.

The Gale-Shapley deferred acceptance algorithm

```
GS (preference lists of hospitals and students):

INITIALIZE M to empty matching
WHILE (some hospital h is unmatched and hasn’t proposed to every student)
    s ← first student on h’s list to whom h has not yet proposed.
    IF (s is unmatched and s prefers to be matched with h)
        Add (h, s) to matching M.
    ELSE IF (s prefers h to current matching h’)
        Replace (h’, s) with (h, s) in matching M.
    ELSE
        s rejects h.
```

(From slides for Algorithm Design by Kleinberg and Tardos)

The GS algorithm finds stable matches

- **Claim:** the GS algorithm **always** outputs a stable matching, **regardless of the problem instance.**

- **Proof:** there can be at most (number of hospitals) x (number of students) proposals, so the algorithm will terminate.

We must show that there are no unstable pairs **(h, s)** as a consequence of the outputted matching **M**.

Let **(h, s)** be a pair not in **M**.

Case 1. If **h** never proposed to **s**, then **h** prefers its match in **M** over **s** because **h** proposed in order of its preferences.

Case 2. If **h** proposed to **s**, then **s** prefers its match in **M** over **h** because **s** always improves their match when they switch.

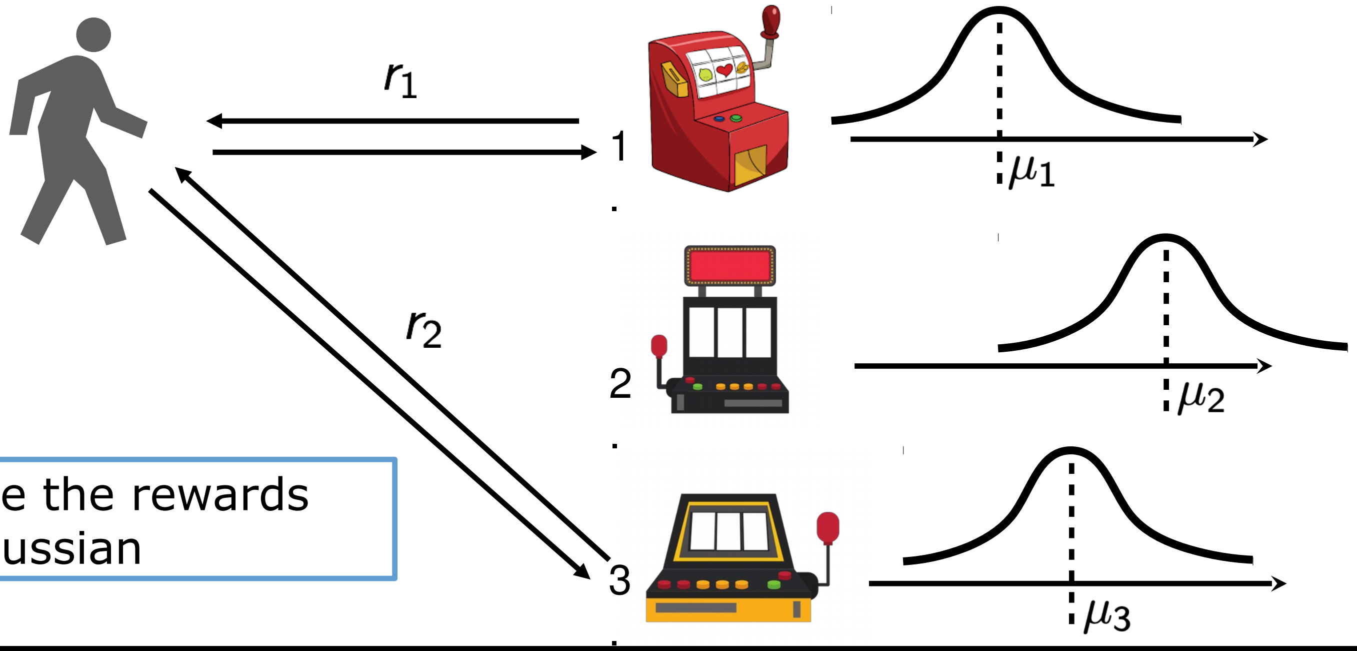
So there cannot be any unstable pairs produced by **M**.

Optimality of the GS match

- A problem instance can admit multiple stable matches.
- **Definition**: **(h, s)** are **valid partners** if there exists a stable matching in which **h** and **s** are matched.
- **Theorem**: the matching produced by GS matches each member of the **proposing side (hospitals) with their best valid partner** and matches each member of the **passive side (students) with their worst valid partner**.
- **Proof**: homework exercise. Proceed by contradiction: let h be the first hospital to be rejected by a valid match s (one must exist if the final match is not hospital-optimal).

Exploitation

Multi-armed bandits provide a natural framework to understand exploration / exploitation trade-offs.



Assume the rewards are Gaussian

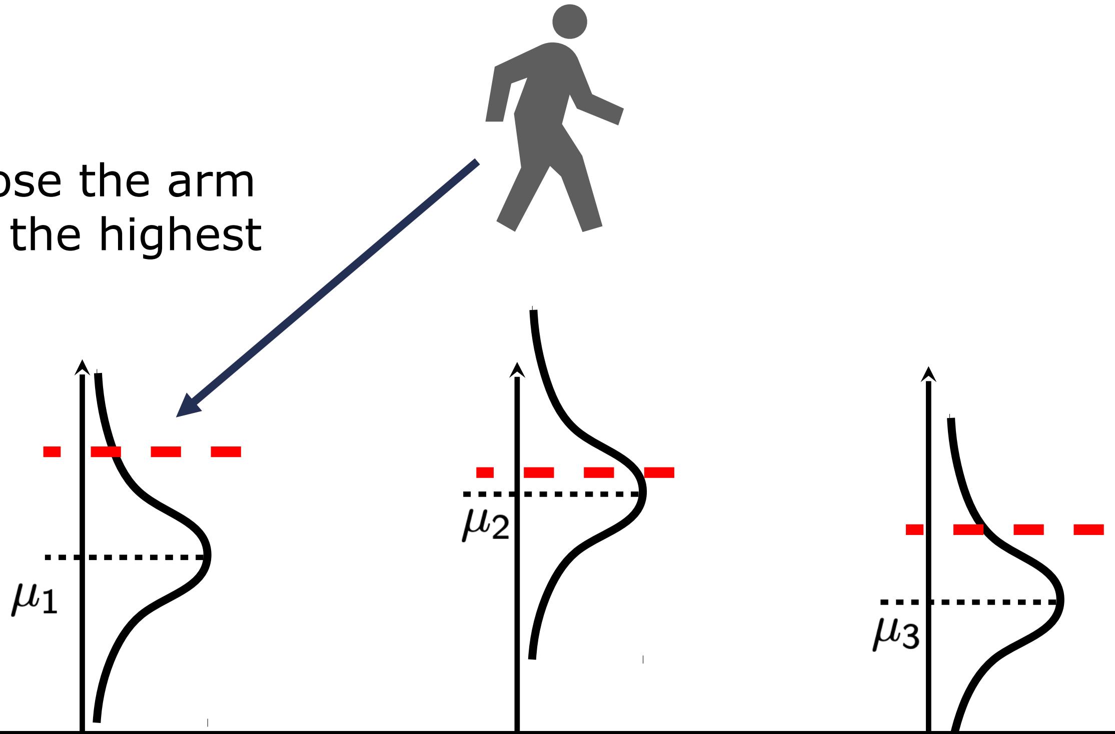
Recap: Regret

- Let *n* be the horizon (number of rounds)
- Let K be the number of arms.
- Let $X_i(t)$ be the reward of arm i at time t.
- Let a_t be the arm chosen at time t.
- The goal is to minimize (expected) regret:

$$R(n) = \underbrace{\max_{i \in \{1,2,\dots,K\}} \mathbb{E}\Big[\sum_{t=1}^{n} X_i(t)}_{\text{Total reward of best arm } \textit{in hindsight}} - \underbrace{\sum_{t=1}^{n} X_{a_t}(t)}_{\text{Reward obtained}}\Big]$$

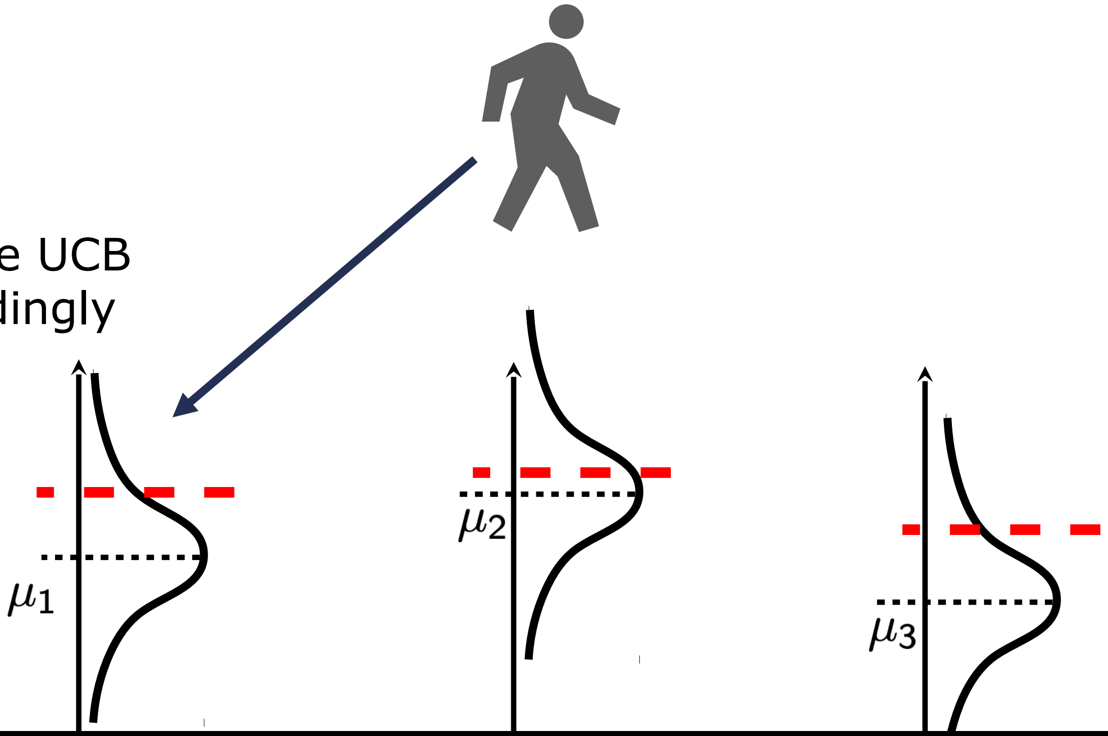
Upper Confidence Bound

Choose the arm with the highest UCB

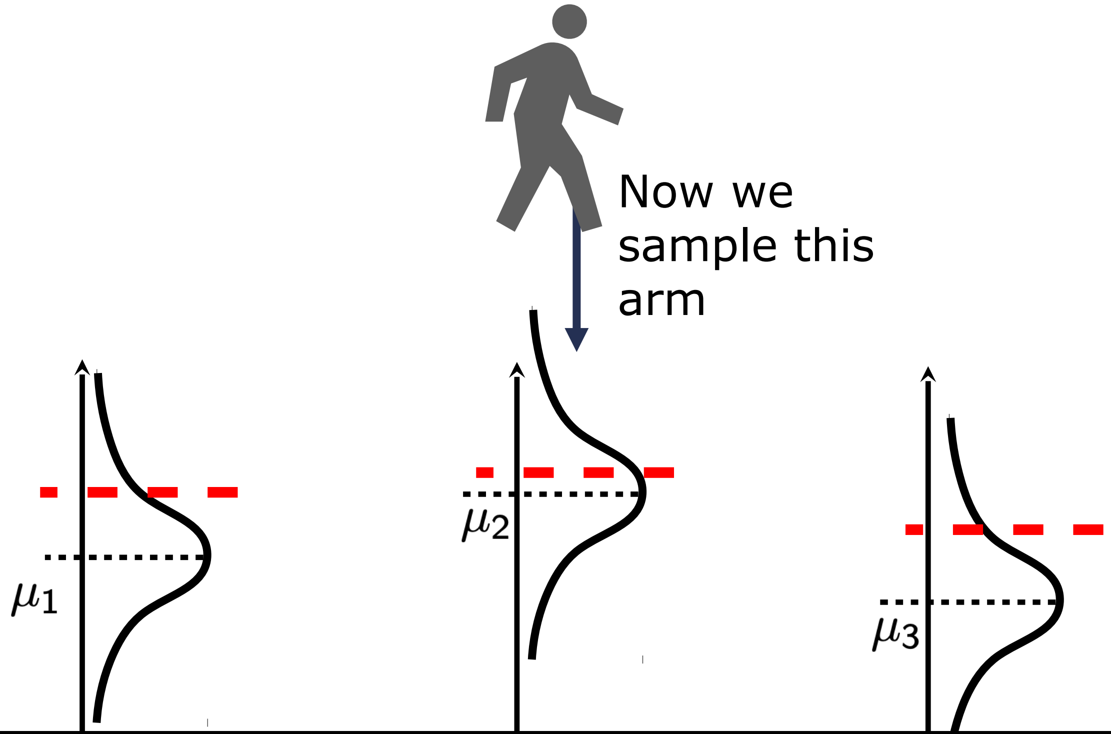


Upper Confidence Bound

Update UCB accordingly



Upper Confidence Bound



Now we sample this arm

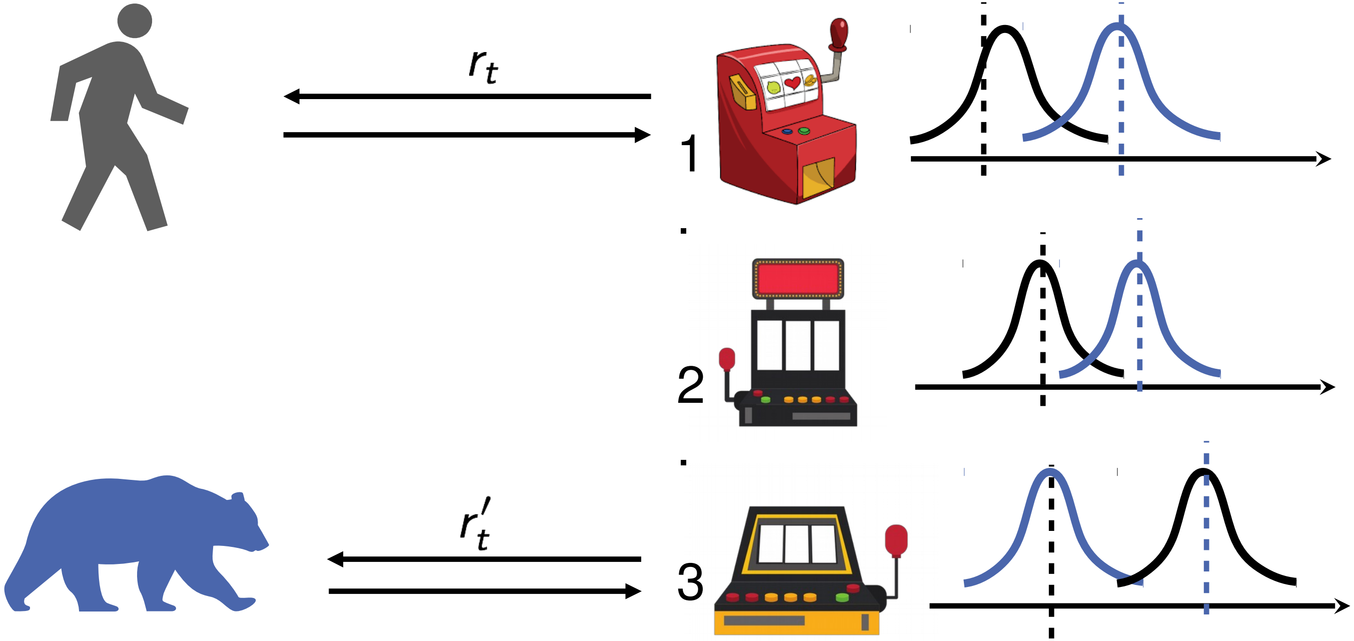
Regret of UCB

- Suppose arm 1 has the highest mean reward.
- Let $\Delta_i = \mu_1 - \mu_i$, called reward gap.
- Then, the regret of UCB satisfies

$$R(n) = \mathcal{O}\left(\log(n) \sum_{i=2}^{K} \frac{1}{\Delta_i}\right)$$

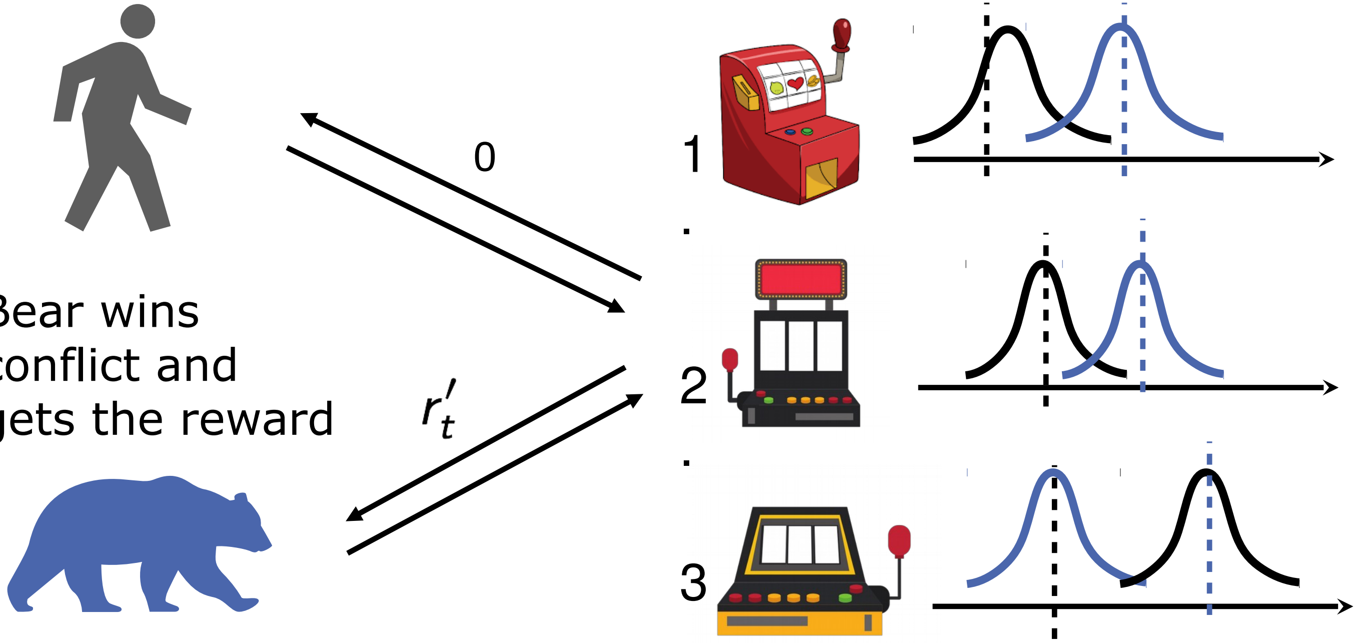
Multi-Player Bandits

Let’s add a competing player!



Bandits

Let’s add a competing player!



Bear wins conflict and gets the reward

Competing Bandits in Matching Markets

In summary: we consider a bandits market with players on one side, arms on the other.

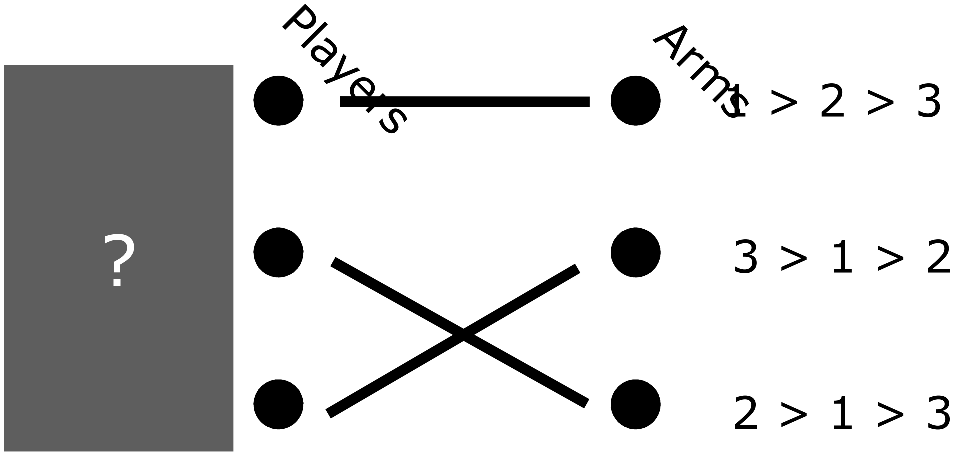
Players get noisy rewards when they pull arms. Same arm has different mean reward for different players.

Arms have known preferences over players (these preferences can also express agents’ skill levels).

When multiple players pull the same arm only the most preferred player gets a reward (competition).

Regret in Matching Markets

- Suppose that there is some unique stable matching.



Players should always choose their stable match in hindsight!

Regret in Matching Markets

- Let $m(i)$ be the stable match of player p_i.
- Let m_t be the matching played by all the players at time t.
- Let $\mu_i(j)$ be the mean reward of arm j for player p_i .

- Let be the stable match of player .
- Let be the matching played by all the players at time t.
- Let be the mean reward of arm j for player .

Define the ***stable regret*** of agent ***i*** up to time ***n*** as:

$$R_i(n) = n\underbrace{\mu_i(m(i))}_{\text{Mean reward of stable match}} - \sum_{t=1}^{n} \mathbb{E}\underbrace{X_{i,m_t}(t)}_{\text{Reward at time t}}$$

Optimal vs Pessimal regret

- Stable match may not be unique.
- Pessimal Stable regret

$$\underline{R}_i(n) = \underline{n\mu_i(\underline{m}(i))} - \sum_{t=1}^{n} \mathbb{E}X_{i,m_t}(t)$$

Mean reward of worst stable match

- Optimal Stable regret

$$\overline{R}_i(n) = \underline{n\mu_i(\overline{m}(i))} - \sum_{t=1}^{n} \mathbb{E}X_{i,m_t}(t)$$

Mean reward of best stable match

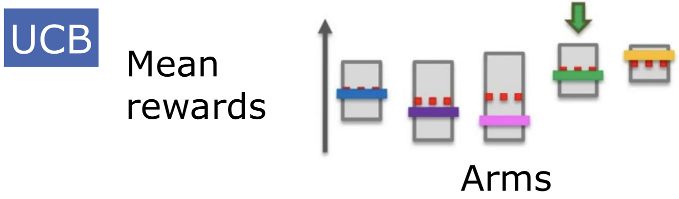
How to achieve a sequence of matchings that has low stable regret for all Players?

Algorithm: GS-UCB

Involves a Matching Platform that communicates with all Players.

At every round:

1. Players rank Arms by the UCBs of each Arm’s mean reward for themselves.
2. Platform runs the Gale-Shapley algorithm to match Players and Arms.
3. Players receive rewards from matched Arms and update their UCB for the Arm.



Regret of GS-UCB

Theorem (informal): If there are N players and K arms and GS-UCB is run, the *pessimal* stable regret of player *i* satisfies

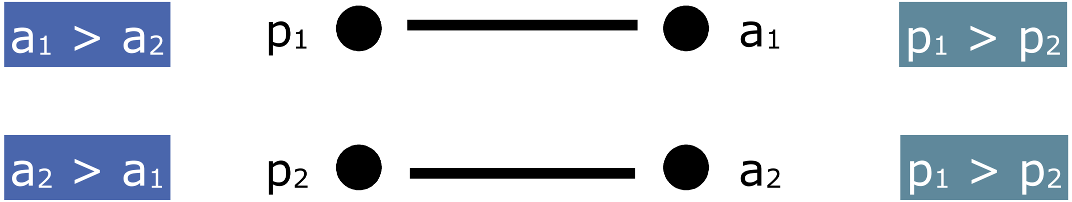
$$\underline{R}_i(n) = \mathcal{O}\left(\frac{NK\log(n)}{\boxed{\Delta^2}}\right)$$

Minimum gap of arms' rewards for all players.

In other words, if one player has to explore more, another player incurs higher stable regret.

Dependence on $1/\Delta^2$

Dependence on



- In order for p₂ to be matched with their stable arm a₂, p₁ must correctly determine that they prefer over .
- This requires rounds of exploration. is 's gap, which can be small.
- p₂'s gap can be large.

- In order for p₂ to be matched with their stable arm a₂, p₁ must correctly determine that they prefer a_1 over a_2.
- This requires $\Omega(1/\overline{\Delta}_1^2)$ rounds of exploration. $\overline{\Delta}_1$ is p_1's gap, which can be small.
- p₂'s gap $\overline{\Delta}_2$ can be large.

A puzzle

- **Gale-Shapley under known preferences** guarantees the **Player-optimal** matching
- **Gale-Shapley with preference learning (UCB)** only has guarantees for the **Player-pessimal** stable regret.
  - Player-optimal regret can be linear in the worst case
- Why?

Special case: Global preferences

- N Players all have the same ranking over Arms
- K Arms all have the same ranking over Players
- Arm/Player 1 is the most preferred etc.
  (everyone has the preference 1 > 2 > 3 > … > N)

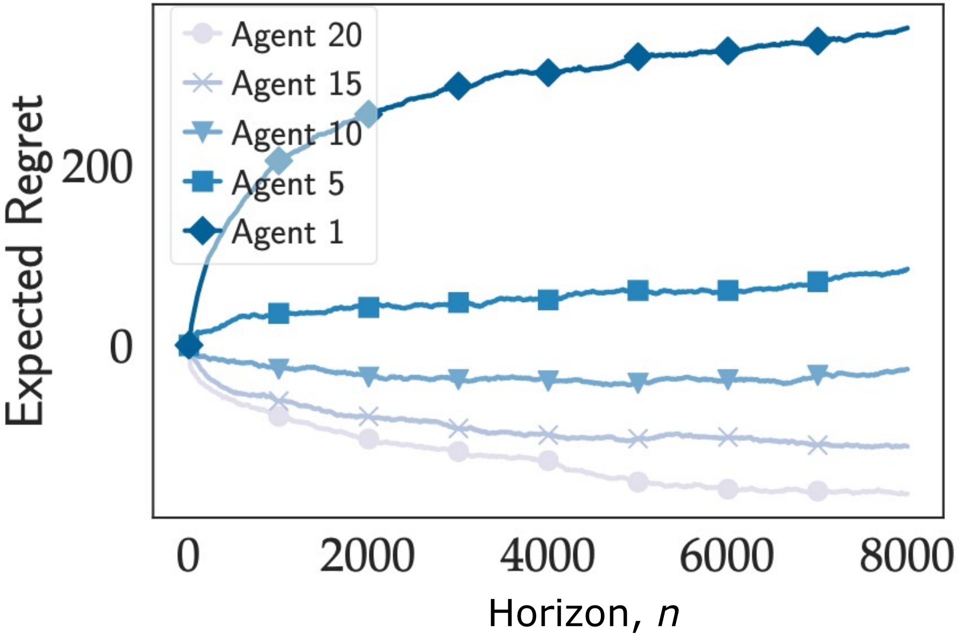
- In this setting, the pessimal regret of Player *i* is as follows:

$$\underline{R}_i(n) \leq 5i \sum_{\ell=i+1}^{K} \underline{\Delta}_{i,\ell} + \sum_{\ell=i+1}^{K} \frac{6i\log(n)}{\boxed{\underline{\Delta}_{i,\ell}}}.$$

Gap between Arm *ℓ* and pessimal stable Arm of Player *i*

Global preferences (N=K=20)

- Worst-ranked agent has negative regret (upper bound is 0)



Incentive compatibility

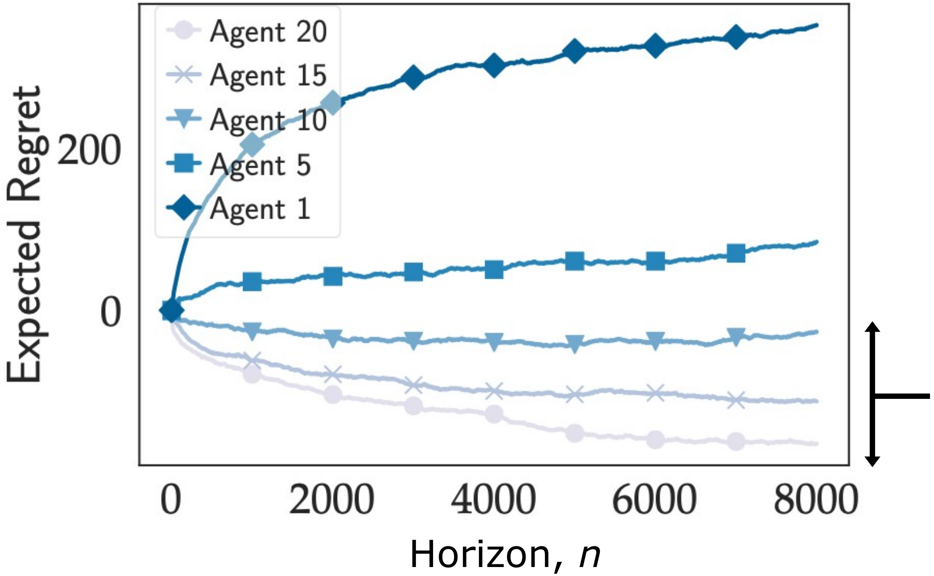
- Players must match with the Arm assigned by the Platform, but they can potentially submit preferences other than those based on their UCBs.
- Q: How much can a Player improve their stable regret, if all other Players are submitting UCB preferences and the Platform is running GS?
- A: Their (*optimal)* stable regret can be lower bounded by

$$\overline{R}_i(n) \geq -O(\log n)$$

- Intuition: Player can only do better than their optimal stable arm if other players make ranking mistakes.

Incentive compatibility

- *-O(log n)* is achievable. Example: Global preferences



worst ranked agents have negative *log n* stable regret

Summary

- Many decision problems involve learning and economic thinking
- Looked at a multi-Player bandit problem in the setting of matching markets
  - Notion of Stable Regret
- GS-UCB Algorithm
  - Log(n) Player-Pessimal Stable Regret
  - Limited gains from Player manipulation of preferences
- Many open problems remain!

Proof Sketches

Single-agent UCB proof sketch

- Number of times player pulled arm *i* after first *s* rounds

$$T_i(s) = \sum_{t=1}^{s} \mathbf{1}\{I_t = i\}$$

- Regret decomposition

$$\begin{aligned}
\bar{R}_n &= \max_{i=1,\cdots,K} \mathbb{E}\left[\sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}\right] \\
&= n\mu^* - \sum_{t=1}^{n} \mathbb{E}\left[\mu_{I_t}\right] \\
&= \sum_{i=1}^{K} \boxed{\Delta_i} \mathbb{E}[T_i(n)]
\end{aligned}$$

Gap between Arm *i* and Best Arm

Single-agent UCB proof sketch

If $I_t = i$, then one of the following must be true

A. Sample mean of best arm is too low

$$\hat{\mu}_{i^*,T_{i^*}(t-1)} + \sqrt{\frac{3\log t}{2T_{i^*}(t-1)}} \leq \mu^*$$

B. Sample mean of arm *i* is too high

$$\hat{\mu}_{i,T_i(t-1)} > \mu_i + \sqrt{\frac{3\log t}{2T_i(t-1)}}$$

C. Pulled arm *i* too few times, given how small the gap is

$$T_i(t-1) < \frac{6}{\Delta_i^2}\log(t)$$

Single-agent UCB proof sketch

- Pulling arm i because of Event C can at most happen

  $$\frac{6}{\Delta_i^2}\log(n)$$

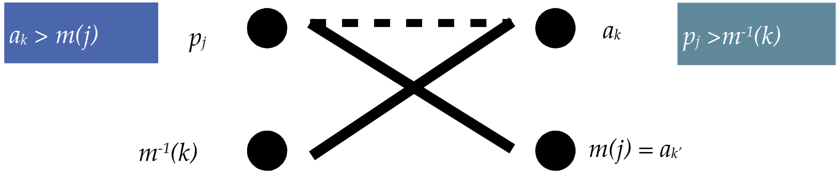
  times.

- Probability of Event A and B can be bounded by a concentration inequality (Hoeffding)
  - can happen at most a constant number of times in expectation.

Reference: Bubeck and Cesa-Bianchi, 2012

Main theorem preliminaries

- matching m: set of players → set of arms
- blocking pair (p_j, a_k)



- ***blocking triplet*** $(p_j, a_k, a_{k'})$
- Set of all matchings <u>blocked</u> by a triplet $(p_j, a_k, a_{k'})$ is $B_{j,j,k'}$

Main theorem preliminaries

- Given a set S of matchings, a set Q of triplets ***cover*** S if

$$\bigcup_{(p_j, a_k, a_{k'}) \in Q} B_{j,k,k'} \supseteq S.$$

- A matching m is ***full*** if all players are matched (N < K)
- $M_{i,l}$: set of full matchings m such that $m(p_i)=a_l$
- **Minimal covering** of $M_{i,l}$: smallest set of blocking triplets that cover $M_{i,l}$

Main theorem preliminaries

- Notation for reward gaps:
  - Mean reward gap between pessimal stable arm of Player i and Arm l for Player i , $\underline{\Delta}_{i,l}$.
  - Mean reward gap between Arm k and k' for Player j $\Delta_{j,k,k'}$.
- **Main Theorem**. The pessimal stable regret of GS-UCB is

$$\underline{R}_i(n) \leq \sum_{\ell:\ \underline{\Delta}_{i,\ell}>0} \underline{\Delta}_{i,\ell} \left[\min_{Q\in\mathcal{C}(M_{i,\ell})} \sum_{(p_j,a_k,a_{k'})\in Q} \left(5 + \frac{6\log(n)}{\Delta^2_{j,k,k'}}\right)\right]$$

Main theorem proof

- We may decompose regret by the number of times a matching happens.

$$\underline{R}_i(n) \leq \sum_{\ell:\, \underline{\Delta}_{i,\ell}>0} \underline{\Delta}_{i,\ell} \left(\sum_{m \in M_{i,\ell}} \mathbb{E} T_m(n) \right)$$

- $L_{j,k,k'}(n)$ is the number of times $(p_j, a_k, a_{k'})$ is a blocking triplet, i.e. p_j pulls $a_{k'}$, and p_j ranks $a_{k'}$ above a_k by mistake.

$$\sum_{m \in B_{j,k,k'}} T_m(n) = L_{j,k,k'}(n)$$

Main theorem proof

- By the usual argument for UCB, we have

$$\mathbb{E}L_{j,k,k'}(n) \leq 5 + \frac{6\log(n)}{\Delta_{j,k,k'}^{2}}.$$

For more details, see [Liu et al., 2019].

Corollary

- Consider the covering: all (j,k,k’) where Player j prefers k to k’
- This trivially covers $M_{i,l}$ for all i and l.
- Corollary.

$$\underline{R}_i(n) \leq \max_{\ell} \underline{\Delta}_{i,\ell} \left(6NK^2 + 12 \frac{NK\log(n)}{\boxed{\Delta}^2} \right).$$

Minimum gap between any pair of Arms’ rewards for any Player.

Proof sketch for incentive compatibility

- Number of times Player i pulls Arm l where l is preferred to its optimal stable arm

$$\mathbb{E}[T_l^i(n)] \le \min_{Q\in\mathcal{C}(M_{i,\ell})} \sum_{(j,k,k')\in Q} \left(5 + \frac{6\log(n)}{\Delta^2_{j,k,k'}}\right)$$

- Upper bound on Player-optimal regret

$$\overline{R}_i(n) \ge \sum_{\ell:\,\overline{\Delta}_{i,l}<0} \boxed{\overline{\Delta}_{i,l}} \left[\min_{Q\in\mathcal{C}(M_{i,\ell})} \sum_{(j,k,k')\in Q} \left(5 + \frac{6\log(n)}{\Delta^2_{j,k,k'}}\right)\right].$$

Gap between Arm l and optimal stable Arm of Player i

Negative!