# Multiplayer bandits, overview and perspectives

Etienne Boursier
Celeste, INRIA Saclay
Laboratoire Mathématique d'Orsay

December 12th, 2023
From matchings to markets, CIRM

# Joint work with

Vianney Perchet

**CREST, ENSAE Paris**
**Criteo AI Lab**

Emilie Kaufmann

**Univ. Lille, CNRS**
**Scool, INRIA Lille**

Hugo Richard

**Criteo AI Lab**

Abbas Mehrabian

**DeepMind, Montréal**

Flore Sentenac

**HEC Paris**

# Outline

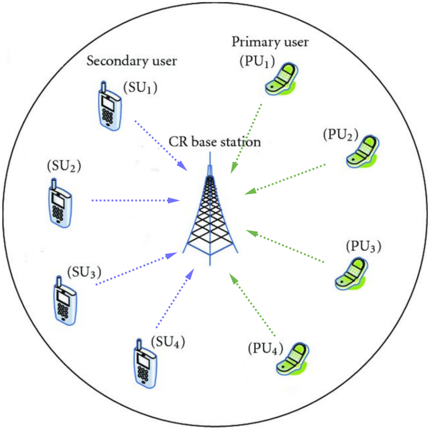
- Introduction
- Multi-armed bandits
- Multiplayer bandits: reaching centralized performance
- Towards a new formulation
- Decentralized queuing systems

# Introduction

Learning with multiple agents

- environment depends on others' actions
- harder to learn (non i.i.d. data)
- competition between agents

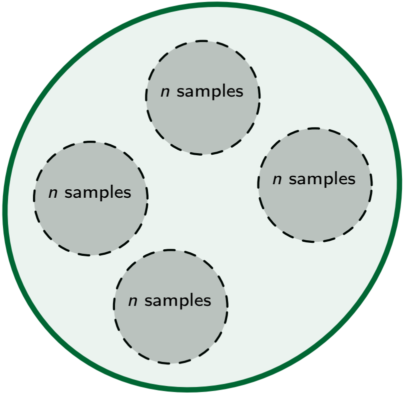
Cognitive radio networks: SUs learn channels with best transmission quality




→ **what interactions between learning agents?**

# Main challenges

**Centralized** $\longrightarrow$ **Decentralized**




gathering the data speeds learning up

**Cooperative** $\longrightarrow$ **Competitive**

| | | **Prisoner 2** | |
|---|---|---|---|
| | | Cooperate | Defect |
| **Prisoner 1** | Cooperate | 3,3 | 0,5 |
| | Defect | 5,0 | 1,1 |

Prisoner's dilemma (rewards)

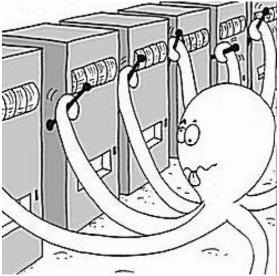
best selfish strategy = defect

# Multi-armed bandits

A 5 minutes course

# Multi-armed bandits (MAB)

- online learning problem
- widely used in online recommendation
- allows nice theory
- many existing variations



# Stochastic MAB

For $t = 1, \ldots, T$:

- pull arm $\pi(t)$ in $[K] := \{1, \ldots, K\}$, based on previous observations
- observe reward $X_{\pi(t)}(t) \in [0, 1]$ with $X_k$ of mean $\mu_k$ (drawn i.i.d.)

**Notation:** statistic order of means $\mu_{(1)} \geq \mu_{(2)} \geq \ldots \geq \mu_{(K)}$
**Goal:** maximize total reward or, equivalently, minimize regret

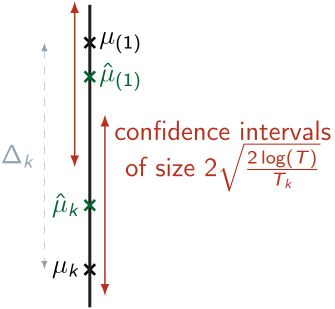
$$R_T = \mu_{(1)} T - \sum_{t=1}^{T} \mathbb{E}[X_{\pi(t)}]$$

**Exploration/exploitation** dilemma: only observe reward of pulled arm

- exploration: pull all arms to estimate $\boldsymbol{\mu}$
- exploitation: pull seemingly best arm to maximise short term reward

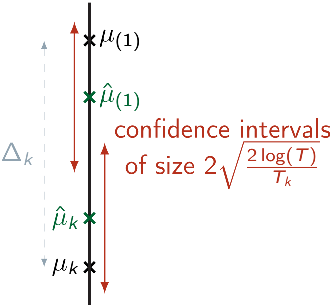
# Successive Eliminations algorithm

$\mathcal{A} = [K]$
**while** $\mathrm{card}(\mathcal{A}) > 1$ **do**
  Pull all arms in $\mathcal{A}$ once
  **for** *all* $k \in \mathcal{A}$ *such that* $\max_{i\in\mathcal{A}} \hat{\mu}_i + \sqrt{\frac{2\log(T)}{T_i}} \geq \hat{\mu}_k - \sqrt{\frac{2\log(T)}{T_k}}$ **do**
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{k\}$
  **end**
**end**
Pull best empirical arm until the end



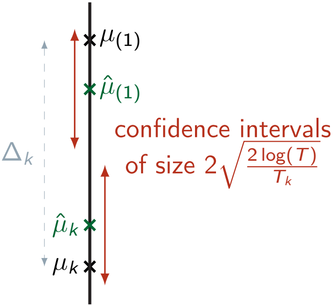
# Successive Eliminations algorithm

$\mathcal{A} = [K]$
**while** $\mathrm{card}(\mathcal{A}) > 1$ **do**
  Pull all arms in $\mathcal{A}$ once
  **for** *all* $k \in \mathcal{A}$ *such that* $\max_{i \in \mathcal{A}} \hat{\mu}_i + \sqrt{\frac{2\log(T)}{T_i}} \geq \hat{\mu}_k - \sqrt{\frac{2\log(T)}{T_k}}$ **do**
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{k\}$
  **end**
**end**
Pull best empirical arm until the end



# Successive Eliminations algorithm

$\mathcal{A} = [K]$
**while** $\mathrm{card}(\mathcal{A}) > 1$ **do**
  Pull all arms in $\mathcal{A}$ once
  **for** *all* $k \in \mathcal{A}$ *such that* $\max_{i \in \mathcal{A}} \hat{\mu}_i + \sqrt{\frac{2\log(T)}{T_i}} \geq \hat{\mu}_k - \sqrt{\frac{2\log(T)}{T_k}}$ **do**
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{k\}$
  **end**
**end**
Pull best empirical arm until the end




Arm $k$ is eliminated after $\approx \frac{\log(T)}{(\mu_{(1)} - \mu_k)^2}$ pulls **whp** (Hoeffding inequality)

$R_T \lesssim \sum_{k>2} \frac{\log(T)}{\mu_{(1)} - \mu_{(k)}}$

**Optimal regret bound**: no algorithm can do better

# Multiplayer bandits

## Reaching centralized performance

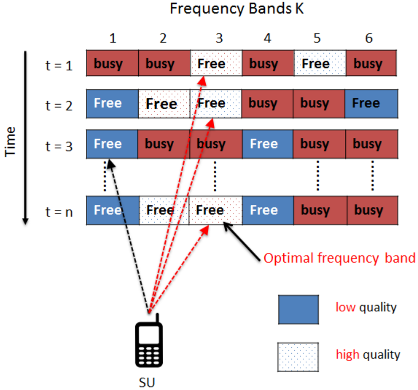
# Motivation: Cognitive Radios

- licensed bands: Opportunistic Spectrum Access

arm ↔ availability from primary users

- un-licensed bands: IoT communications

arm ↔ background traffic



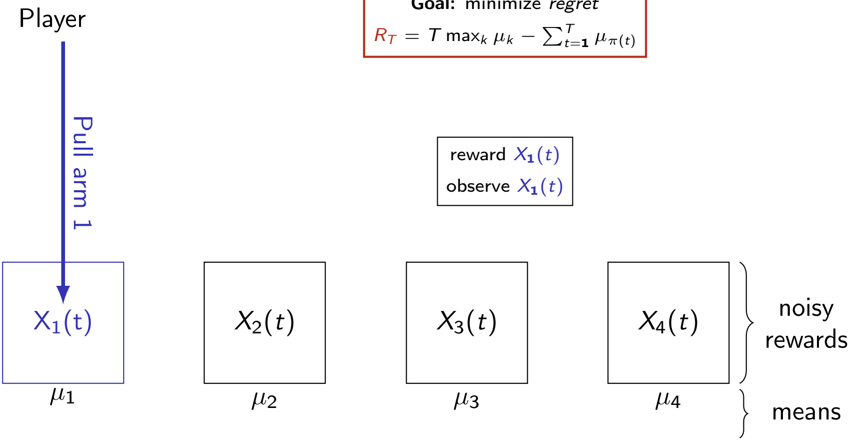

what about **multiple devices**?

→ several users cannot transmit on same channel

# Model: single player

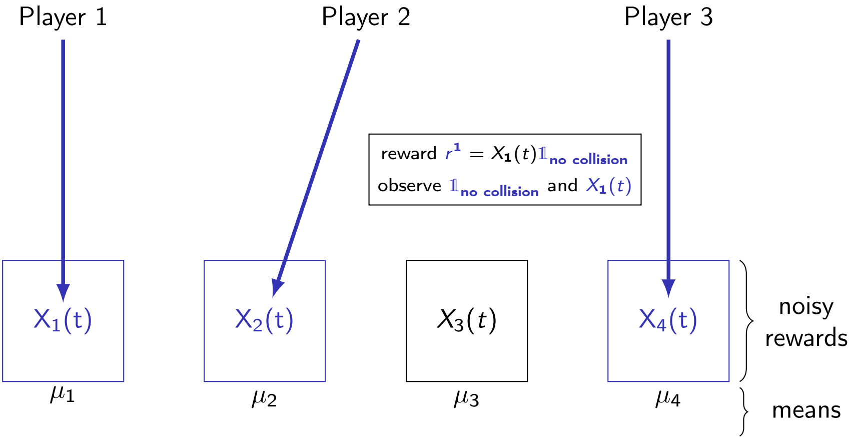
Stochastic bandits
$K$ arms (frequency bands)

For $t = 1, \dots, T$, pull $\pi(t)$
based on observations history
**Goal:** minimize *regret*
$R_T = T \max_k \mu_k - \sum_{t=1}^{T} \mu_{\pi(t)}$



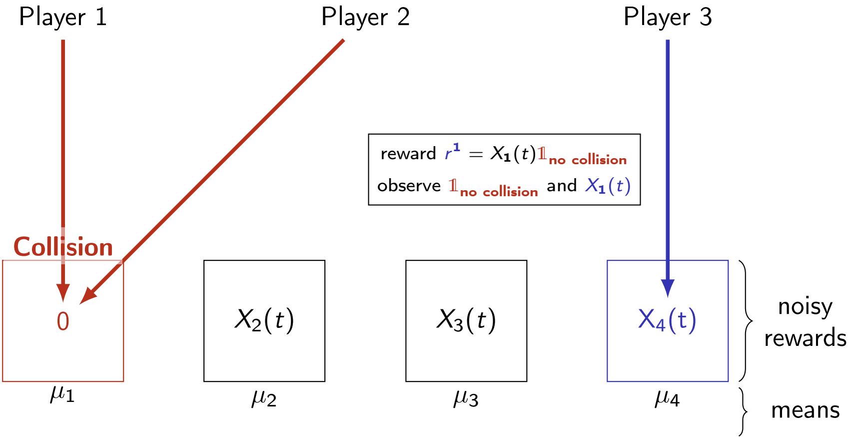
# Model: multiplayer

Stochastic bandits [**Multiplayer**]
$K$ arms (frequency bands), $M$ players (secondary users)



# Model: multiplayer

Stochastic bandits [**Multiplayer**]
$K$ arms (frequency bands), $M$ players (secondary users)



# Model

$M$ players pull arms $\pi^m(t)$ at each round $t = 1, \ldots, T \quad (m \in [M])$
$K$ arms with rewards $X_k(t) \overset{\text{i.i.d.}}{\sim} \mathrm{Bernoulli}(\mu_k) \quad (K \geq M)$

Observe **separately** $X_{\pi^m(t)}(t)$ and $\mathbb{1}_{\text{no collision on } \pi^m(t)}$

**Notation**: $\mu_{(1)} \geq \mu_{(2)} \geq \ldots \geq \mu_{(K)}$
**Goal**: minimize regret

$$R_T = \underbrace{T \sum_{m=1}^{M} \mu_{(m)}}_{\text{best possible reward}} - \mathbb{E}\left[\underbrace{\sum_{t=1}^{T} \sum_{m=1}^{M} \mu_{\pi^m(t)} \mathbb{1}_{\text{no collision on } \pi^m(t)}}_{\text{actual reward}}\right]$$

$\rightarrow$ find $M$ best arms

# First intuitions

**Centralized** optimal algorithms:

$$R_T \approx \sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}$$

Prior belief for **decentralized** case:

$$R_T \gtrsim M \sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}$$

# First intuitions

**Centralized** optimal algorithms:

$$R_T \approx \sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}$$

Prior belief for **decentralized** case:

$$R_T \gtrsim M \sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}$$

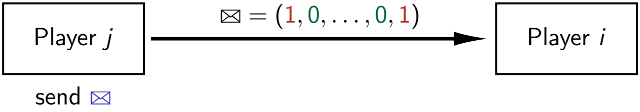
holds for algorithms without collisions
$\rightarrow$ recent optimal algorithms force many collisions

- collision $=$ **immediate** cost 1
- collision is an information bit: $\mathbb{1}_{\text{collision}} \in \{0, 1\}$
- single information bit can have a huge **long term** value

centralized bound achievable when enforcing collisions

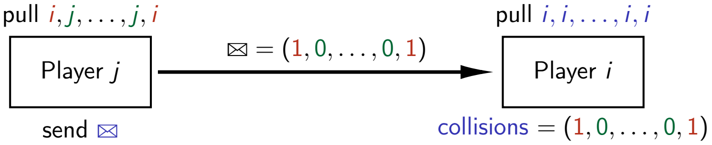
# Communication trick

**Feedback:** observe *separately* $X_{\pi^m(t)}(t)$ and $\mathbb{1}_{\text{no collision on } \pi^m(t)}$
$\mathbb{1}_{\text{collision}}$ = bit sent between players



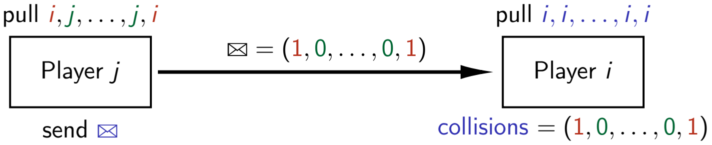
# Communication trick

**Feedback:** observe *separately* $X_{\pi^m(t)}(t)$ and $\mathbb{1}_{\text{no collision on } \pi^m(t)}$
$\mathbb{1}_{\text{collision}}$ = bit sent between players



# Communication trick

**Feedback:** observe *separately* $X_{\pi^m(t)}(t)$ and $\mathbb{1}_{\text{no collision on } \pi^m(t)}$
$\mathbb{1}_{\text{collision}}$ = bit sent between players




Communication Protocol

**input :** empirical means $(\hat{\mu}_k^m)_{k=1,\dots,K}$
**for** $i, j, k \in [M] \times [M] \times [K]$ **do**
  Player $j$ sends $\hat{\mu}_k^j$ in binary to player $i$ `// p bits for 2^p observations`
**end**

Enable communication between players
Gather statistics → **centralized** performance

# SIC-MMAB

| SIC-MMAB | |
|---|---|
| $m, M \leftarrow$ Initialize | // $K\log(T)$ rounds |
| **for** $p = 1, \dots, \infty$ ***until*** *M best arms found* **do** | |
| Pull each *active* arm $2^p$ times | // explore |
| Communication Protocol | // $M^2Kp$ rounds |
| Eliminate suboptimal arms | |
| **end** | |
| Pull $M$ best arms until $T$ | // exploit |

**Initialization:** estimate $M$ + assign unique ranks in $[M]$ to players

Eliminate $k$ when there are $M$ arms $i$ such that

$$\hat{\mu}_i - \underbrace{3\sqrt{\frac{\log(T)}{2T_i}}}_{\text{confidence bound}} \geq \hat{\mu}_k + 3\sqrt{\frac{\log(T)}{2T_k}}$$

# SIC-MMAB

Exploration ends after $\sim \frac{K \log(T)}{\Delta^2}$ rounds with $\Delta := \mu_{(M)} - \mu_{(M+1)}$
$\rightarrow N \sim \log\left(\frac{\log(T)}{\Delta^2}\right)$ epochs and $M^2KN^2$ communication rounds

**Theorem (SIC-MMAB[1])**

$$R_T \lesssim \underbrace{\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}}_{\text{exploration}} + \underbrace{MK\log(T)}_{\text{initialization}} + \underbrace{o(\log(T))}_{\text{communication}}$$

Wang et al. (2020) later improved the initialization and communication

**Same regret as centralized!**

[1] Boursier E. and Perchet V. SIC-MMAB: synchronisation involves communication in multiplayer multi-armed bandits. *NeurIPS 2019.*

# Heterogeneous case

**Heterogeneous**: arm means $\mu_k^m$ differ among the $M$ players

Utility of matching $\pi$: $U(\pi) = \sum_{m=1}^{M} \mu_{\pi(m)}^m$

**Goal**: find best player-arm matching $U^* = \max_\pi U(\pi)$

$$R_T = \underbrace{TU^*}_{\text{best possible reward}} - \mathbb{E}\left[\underbrace{\sum_{t=1}^{T}\sum_{m=1}^{M} \mu_{\pi^m(t)}^m \mathbb{1}_{\text{no collision on } \pi^m(t)}}_{\text{actual reward}}\right]$$

# Heterogeneous case

**Heterogeneous**: arm means $\mu_k^m$ differ among the $M$ players

Utility of matching $\pi$: $U(\pi) = \sum_{m=1}^{M} \mu_{\pi(m)}^m$

**Goal**: find best player-arm matching $U^* = \max_\pi U(\pi)$

$$R_T = \underbrace{TU^*}_{\text{best possible reward}} - \mathbb{E}\left[\underbrace{\sum_{t=1}^{T}\sum_{m=1}^{M} \mu_{\pi^m(t)}^m \mathbb{1}_{\text{no collision on } \pi^m(t)}}_{\text{actual reward}}\right]$$

$\rightarrow$ adapt SIC-MMAB with some tweaks

$$R_T \lesssim \frac{M^3 K \log(T)}{\Delta}$$

where $\Delta := U^* - \max_{U(\pi)<U^*} U(\pi)$

# Closing the gap between centralized and decentralized

- Homogeneous: Wang et al. (2020)
- Homogeneous + no sensing (only observe $X_k(t)\mathbb{1}_{\text{no collision on } k}$): Huang et al. (2021)
- Heterogeneous: Shi et al. (2021)

$\rightarrow$ **decentralized no harder than centralized in multiplayer bandits**

# Closing the gap between centralized and decentralized

- Homogeneous: Wang et al. (2020)
- Homogeneous + no sensing (only observe $X_k(t)\mathbb{1}_{\text{no collision on } k}$): Huang et al. (2021)
- Heterogeneous: Shi et al. (2021)

$\rightarrow$ **decentralized no harder than centralized in multiplayer bandits**

Hard communication **undesirable** in practice, but **best** in theory

Weakness in the current formulation

# Towards a new formulation

# Towards a new formulation

- Focus too much on dependence in $T$?
  - in large networks, dependence in $M, K$ can be more important than $\log(T)$

[2] Boursier E. and Perchet V. Selfish robustness and equilibria in multi-player bandits. *COLT 2020.*

# Towards a new formulation

- Focus too much on dependence in $T$?
  - in large networks, dependence in $M, K$ can be more important than $\log(T)$

- Players should not be cooperative?[2]

[2] Boursier E. and Perchet V. Selfish robustness and equilibria in multi-player bandits. *COLT 2020.*

# Selfish Players

**Goal:** small regret **and** robust to selfish behaviors ($\varepsilon$-Nash equilibrium)

**Definition ($\varepsilon$-Nash equilibrium)**

$\boldsymbol{s} = (s^1, \ldots, s^M)$ is an $\varepsilon$-Nash equilibrium if for any player $m$ and strategy $s'$

$$\mathrm{Rew}_T^m(s', \boldsymbol{s}^{-\boldsymbol{m}}) \leq \mathrm{Rew}_T^m(\boldsymbol{s}) + \varepsilon.$$

Unilaterally deviate from $\varepsilon$-Nash equilibrium $\implies$ earn at most $\varepsilon$ more (in $T$ rounds)

SIC-MMAB with additional tricks:

- robust initialization
- detection of malicious behavior when sending messages
- cut out extreme statistics from estimation
- trigger collective punishment if malicious behavior

# Selfish Players

**Goal:** small regret **and** robust to selfish behaviors ($\varepsilon$-Nash equilibrium)

**Definition ($\varepsilon$-Nash equilibrium)**

$\boldsymbol{s} = (s^1, \dots, s^M)$ is an $\varepsilon$-Nash equilibrium if for any player $m$ and strategy $s'$

$$\mathrm{Rew}_T^m(s', \boldsymbol{s}^{-\boldsymbol{m}}) \leq \mathrm{Rew}_T^m(\boldsymbol{s}) + \varepsilon.$$

Unilaterally deviate from $\varepsilon$-Nash equilibrium $\implies$ earn at most $\varepsilon$ more (in $T$ rounds)

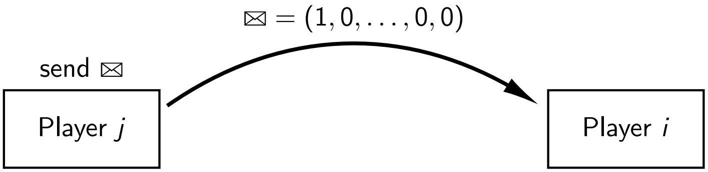
SIC-MMAB with additional tricks:

- robust initialization
- detection of malicious behavior when sending messages
- cut out extreme statistics from estimation
- trigger collective punishment if malicious behavior

# Selfish Players

## Detect malicious behavior

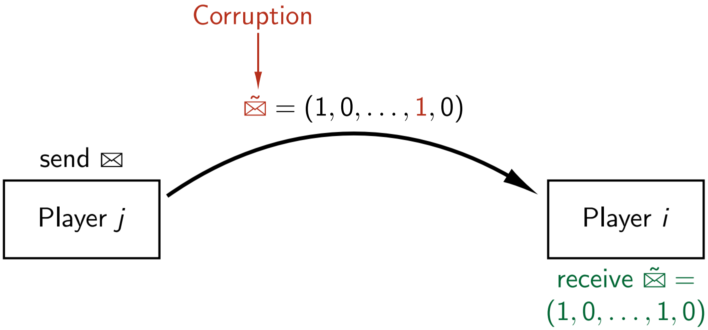
Only way to corrupt communication: transform $0 \to 1$ (create collision)



# Selfish Players

## Detect malicious behavior

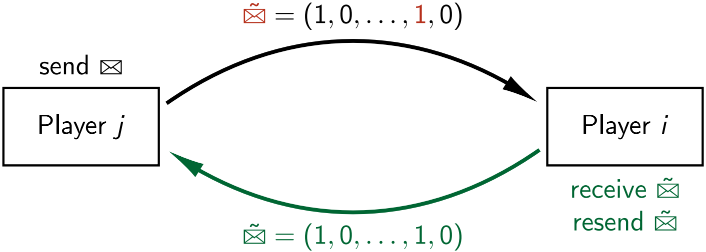
Only way to corrupt communication: transform $0 \to 1$ (create collision)



# Selfish Players

## Detect malicious behavior

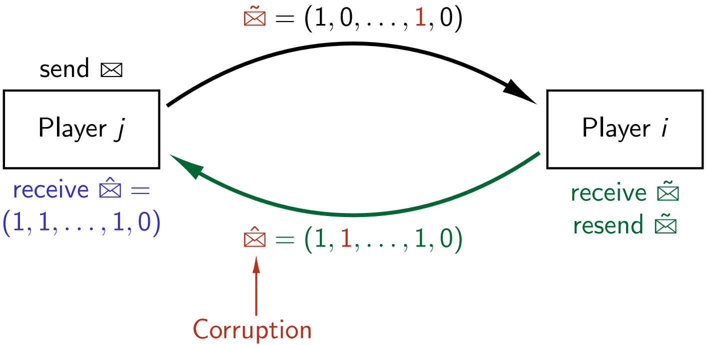
Only way to corrupt communication: transform $0 \to 1$ (create collision)



# Selfish Players

## Detect malicious behavior

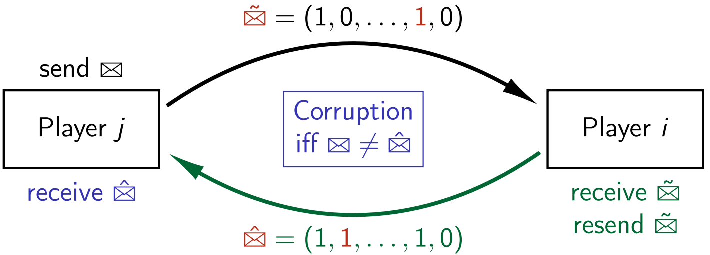
Only way to corrupt communication: transform $0 \to 1$ (create collision)



# Selfish Players

## Detect malicious behavior

Only way to corrupt communication: transform $0 \rightarrow 1$ (create collision)




**detect** corruption in sent messages

# Selfish Players

## Collective punishment

**Grim Trigger:** malicious player detected $\rightarrow$ collective punishment until $T$. **How?**

**1st idea:** sample any arm with probability $\frac{1}{K}$.
Selfish player can earn $\mu_{(1)}(1-1/K)^{M-1} \rightarrow$ not enough.

# Selfish Players

## Collective punishment

**Grim Trigger:** malicious player detected $\rightarrow$ collective punishment until $T$. **How?**

**1st idea:** sample any arm with probability $\frac{1}{K}$.
Selfish player can earn $\mu_{(1)}(1-1/K)^{M-1} \rightarrow$ not enough.

**2nd idea:** sample arm $k$ with proba $\approx 1 - \left(\gamma \frac{\sum_{j=1}^{M} \mu_{(j)}}{M\mu_k}\right)^{\frac{1}{M-1}}$.

Selfish player earns $\approx \gamma \frac{\sum_{j=1}^{M} \mu_{(j)}}{M}$ on $k$. Relative loss $1-\gamma \rightarrow$ great!

# Selfish Players

## Collective punishment

**Grim Trigger:** malicious player detected $\rightarrow$ collective punishment until $T$. **How?**

**1st idea:** sample any arm with probability $\frac{1}{K}$.
Selfish player can earn $\mu_{(1)}(1-1/K)^{M-1} \rightarrow$ not enough.

**2nd idea:** sample arm $k$ with proba $\approx 1 - \left(\gamma \frac{\sum_{j=1}^{M} \mu_{(j)}}{M\mu_k}\right)^{\frac{1}{M-1}}$.

Selfish player earns $\approx \gamma \frac{\sum_{j=1}^{M} \mu_{(j)}}{M}$ on $k$. Relative loss $1-\gamma \rightarrow$ great!

### Theorem

Playing SIC-GT for all players:

1. $\mathbb{E}[R_T] \lesssim \sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + MK^2 \log(T)$
2. $\varepsilon$-Nash equilibrium with: $\varepsilon \lesssim \sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_{(k)}} + \frac{K^3 \log(T)}{\mu_{(K)}}$.

# Towards a new formulation

Hard communication **undesirable** in practice, but **best** in theory

Weakness in the current formulation?

- Focus too much on dependence in $T$?
- Players should not be cooperative? SIC-MMAB still possible
  $\rightarrow$ what about stronger notions of equilibria? (e.g., subgame perfect eq.)

# Towards a new formulation

Hard communication **undesirable** in practice, but **best** in theory

Weakness in the current formulation?

- Focus too much on dependence in $T$?
- Players should not be cooperative? SIC-MMAB still possible
  $\rightarrow$ what about stronger notions of equilibria? (e.g., subgame perfect eq.)
- Players should not be synchronized
  - enter/leave the game at different times
    $\rightarrow$ non communicating algorithm possible, but for a weak dynamic model

# Towards a new formulation

Hard communication **undesirable** in practice, but **best** in theory

Weakness in the current formulation?

- Focus too much on dependence in $T$?
- Players should not be cooperative? SIC-MMAB still possible
  $\rightarrow$ what about stronger notions of equilibria? (e.g., subgame perfect eq.)
- Players should not be synchronized
  - enter/leave the game at different times
    $\rightarrow$ non communicating algorithm possible, but for a weak dynamic model
  - no shared time discretization (asynchronous)
    $\rightarrow$ see Hugo's talk for a first solution in multiplayer bandits
    $\rightarrow$ weaker asynchronicity for queuing systems

# Decentralized queuing systems

# Motivation

Classical repeated games $\longleftrightarrow$ repetition of the same single round game
no dependence on the past, except in learning

Road traffic
independence of rounds

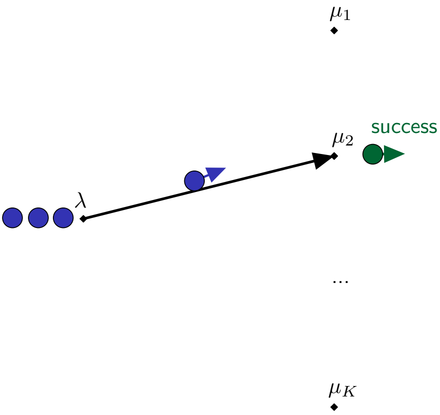
Second-by-second packet routing
Dropped packets have to be resent in next rounds

$\rightarrow$ Learning in repeated games **with carryover**?

# Model: single queue

At each $t = 1, \ldots, \infty$

- packet arrives with proba $\lambda$
- sends a packet to server $k \in [K]$
- server $k$ clears with proba $\mu_k$
- if fails $\to$ packet back in queue

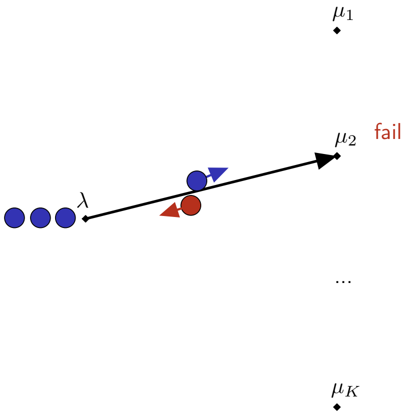



$\to$ **multi-armed bandits approach**

# Model: single queue

At each $t = 1, \ldots, \infty$

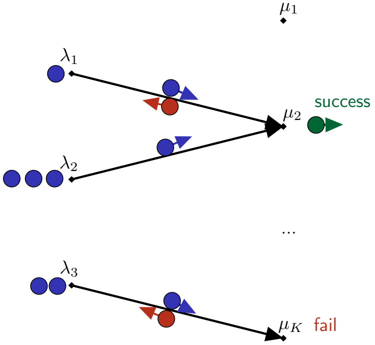
- packet arrives with proba $\lambda$
- sends a packet to server $k \in [K]$
- server $k$ clears with proba $\mu_k$
- if fails $\rightarrow$ packet back in queue




$\rightarrow$ **multi-armed bandits approach**

# Model: multiple queues

- $M$ queues ($M \leq K$)
- Heterogeneous arrival rates $\lambda_i$
- each queue chooses $\pi^m(t) \in [K]$
- Server treats one packet at a time
  - chooses oldest packet




→ outcome depends on the packets' age (carryover)
→ multiplayer bandits approach?

# Stability

$Q_t^i$ number of packets in queue $i$ at time $t$

A queue $i$ is **stable** if for any $r$, there is a constant $C_r > 0$ such that

$$\mathbb{E}[(Q_t^i)^r] \leq C_r \qquad \forall t \in \mathbb{N}$$

Define **slack**

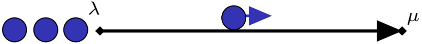
$$\eta = \max\left\{\eta' \in \mathbb{R}_+ \mid \forall m \in [M], \eta' \sum_{i=1}^{m} \lambda_{(i)} \leq \sum_{i=1}^{m} \mu_{(i)}\right\}$$

**Centralized case:** there is a stable strategy iff $\eta > 1$

**Goal:** decentralized stable strategies for small $\eta$

# Centralized case

Single queue, single server



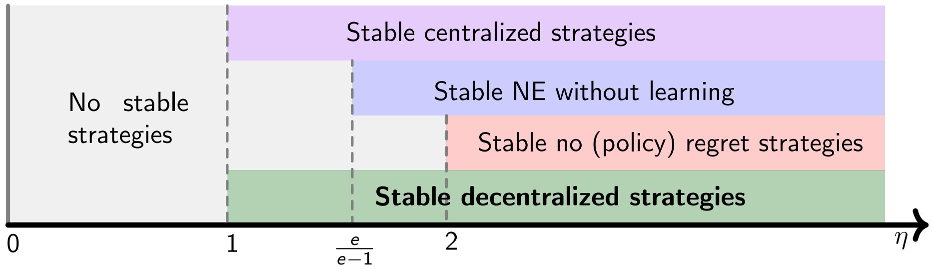

Random walk (with frontier at 0)

- $\lambda < \mu$ $\rightarrow$ negative bias, stable
- $\lambda = \mu$ $\rightarrow$ no bias, queue size grows in $\sqrt{t}$
- $\lambda > \mu$ $\rightarrow$ positive bias, queue size in $(\lambda - \mu)t$

$\implies$ centralized strategy stable iff $\eta > 1$

# Frameworks comparison

| Multiplayer Bandits | Decentralized Queuing Systems |
|---|---|
| symmetric collision | asymmetric collision |
| synchronous | idle if no packet left |
| minimize regret | stability |




**patience is not enough** to go below $\eta = 2$
$\rightarrow$ need for **coordination/cooperation** between players

# A stable learning strategy

**Assumptions:**

- queues know $M$ and pre-assigned ranks $i \in [M]$
- shared randomness between queues
- no collision sensing

**Theorem[3]**

If $\eta > 1$ and all queues follow ADeQuA, then the system is stable.

**ADeQuA**: at each $t$, using *shared randomness* $\begin{cases} \text{explore with proba } \varepsilon_t \\ \text{exploit with proba } 1 - \varepsilon_t \end{cases}$

Exploration: estimate $\mu$ + use collisions to estimate $\lambda$
Exploitation: joint distribution over servers

[3] Sentenac F., Boursier E. and Perchet V. Decentralized Learning in Online Queuing Systems. *NeurIPS 2021.*

# Exploration

All queues explore simultaneously and explore either $\boldsymbol{\mu}$ or $\boldsymbol{\lambda}$ with proba $\varepsilon_t$

Explore $\boldsymbol{\mu}$: queues choose servers without colliding
$\rightarrow$ accurate estimations of all $\mu_k$

**Assumption:** servers break ties in packets' age uniformly at random

Explore $\boldsymbol{\lambda}$: when queue $i$ explores queue $j$, both choose same server $k$ **with packet generated at $t$ (if it exists)**
$i$ clears with probability $(1 - \frac{\lambda_j}{2})\mu_k$ $\rightarrow$ estimate $\lambda_j$

# Exploitation: centralized

When centralized:

- $\phi : (\hat{\lambda}, \hat{\mu}) \mapsto P$, marginals ensuring stability (dominant mapping)
- $\psi : P \mapsto A$, coupling without collision (Birkhoff von Neumann decomposition)

| Centralized exploitation | |
|---|---|
| Draw $\omega \sim \mathcal{U}(0,1)$ | // shared randomness |
| Play $\psi(\phi(\hat{\lambda}, \hat{\mu}))(\omega)$ | |

When decentralized:

- compute mapping $\hat{A}^i = \psi(\phi(\hat{\lambda}^i, \hat{\mu}^i)) : [0,1] \to \mathbb{R}^M$
- play $\hat{A}^i(\omega)(i)$

# Exploitation: decentralized

Compute mapping $\hat{A}^i = \psi(\phi(\hat{\lambda}^i, \hat{\mu}^i))$

**Problem:** estimates $(\hat{\lambda}^i, \hat{\mu}^i)$ differ (but are close)
General dominant mappings and BvN decompositions are non-continuous

$\|\hat{A}^i - \hat{A}^j\|$ arbitrarily large $\implies$ too many collisions

If $\phi$ and $\psi$ **regular** $\rightarrow \|\hat{A}^i - \hat{A}^j\|$ small
$\implies$ small amount of collisions

**Challenge:** design **regular** dominant mapping and BvN decomposition

# Dominant mapping

**Goal** $\phi : \mathbb{R}^N \times \mathbb{R}^K \to \mathrm{Bisto}(N, K)$ such that for any $(\lambda, \mu)$:

$$\lambda < P\mu \qquad \text{if possible}$$

Usual dominant mappings sort $\lambda$ and $\mu$ $\to$ discontinuity

$$\phi(\lambda, \mu) = \operatorname*{arg\,min}_{P \in \mathrm{Bisto}(N,K)} \max_{i \in [N]} - \ln \left( \sum_{j=1}^{K} P_{i,j}\mu_j - \lambda_i \right) + \frac{1}{2K} \|P\|_2^2.$$

- locally Lipschitz objective
- strong convexity $\implies$ regularity of arg min
- optimization methods to approximate $\phi$

# Birkoff von Neumann decomposition

**Goal** $\psi : \mathrm{Bisto}(N, K) \to \mathcal{P}(\mathfrak{S}_{N,K})$ such that for any matrix $P$:

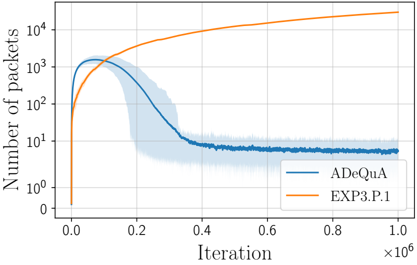
$$\mathbb{E}[\psi(P)] = P$$

**Birkoff algorithm:** computation of successive perfect matchings
$\to$ not necessarily continuous
$\to$ can be made continuous by computing minimal cost matchings wrt to some (arbitrary) cost

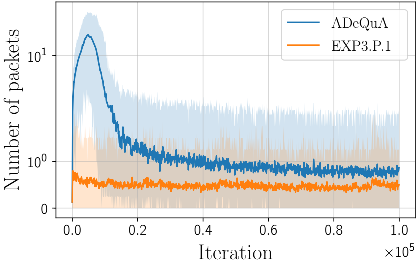
$$\underbrace{\mathbb{P}_{\omega\sim\mathcal{U}(0,1)}(\psi(\hat{P}^i)(\omega) \neq \psi(\hat{P}^j)(\omega))}_{\geq \text{ probability of collision}} \leq 2^{2K^2} \|\hat{P}^i - \hat{P}^j\|_\infty .$$

$\to$ exponential dependency yields large number of packets at intermediate times

# Simulations




Hard instance, $\eta < 2$.




Easy instance, $\eta > 2$.

- No regret strategies: unstable
- ADeQuA: stable & number of packets decreases after learning

- both strategies stable
- No regret better suited to easy instances?

# Recap

**Decentralized sequential learning**

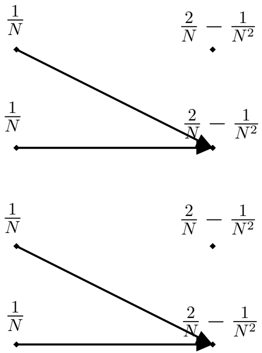
- centralized performance possible in multiplayer bandits, queuing systems...
- still holds for competitive players
- synchronicity of players is oversimplifying?
- first (weak) solutions for both dynamic and asynchronous models

**Perspectives**

- design learning strategies wrt stronger equilibria
- general dynamic/asynchronous model
- relation to other problems (decentralized queuing, competing bandits . . . )

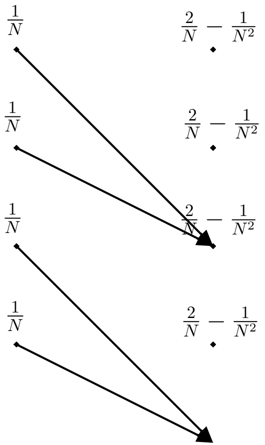
# Thank you!

# Counter Example (first phase)




First phase of length $\alpha T$
Pairwise actions

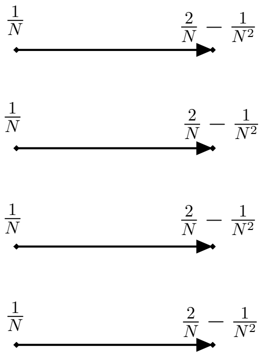
# Counter Example (first phase)




First phase of length $\alpha T$
Pairwise actions

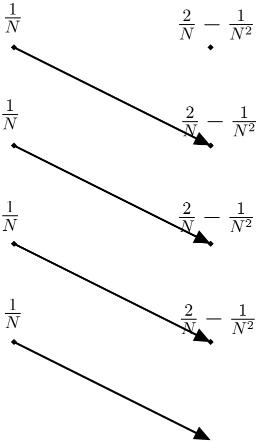
$\rightarrow$ accumulate packets during this phase

# Counter Example (second phase)



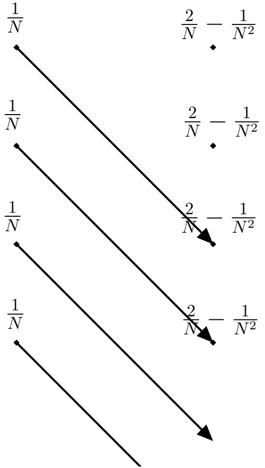

Second phase of length $(1 - \alpha)T$
No collision

# Counter Example (second phase)




Second phase of length $(1 - \alpha)T$
No collision

# Counter Example (second phase)

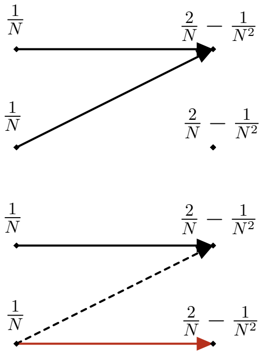



Second phase of length $(1-\alpha)T$
No collision

$\rightarrow$ clear packets during this phase
if $\alpha$ large enough, still accumulate overall $\Omega(T)$ packets
$\rightarrow$ unstable

# Counter example (no policy regret)

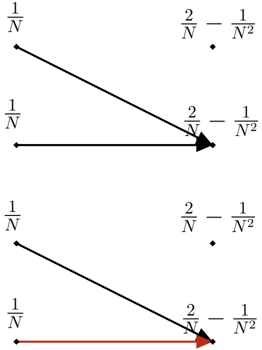
What if queue $i$ deviates and plays $p \in \mathcal{P}([K])$ at each round?




**First phase**
$\rightarrow$ clear all packets

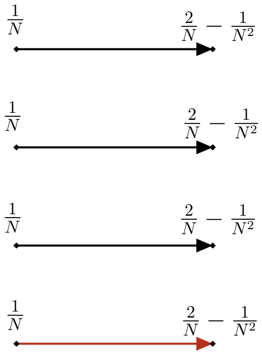
# Counter example (no policy regret)

What if queue $i$ deviates and plays $p \in \mathcal{P}([K])$ at each round?




**First phase**
$\rightarrow$ clear all packets

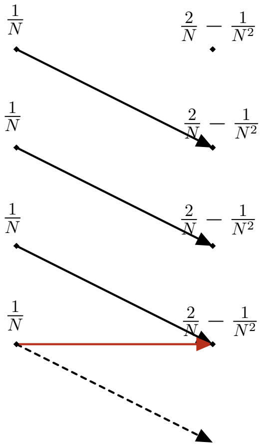
# Counter example (no policy regret)




**Second phase**
many collisions
other queues have priority
accumulate $\Omega(T)$ packets

for $\alpha$ small enough, accumulate more packets when deviating
$\rightarrow$ No policy regret strategies!

# Counter example (no policy regret)



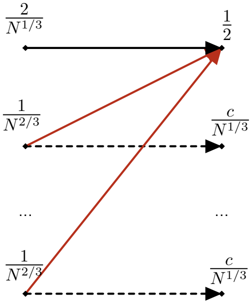

**Second phase**
many collisions
other queues have priority
accumulate $\Omega(T)$ packets

for $\alpha$ small enough, accumulate more packets when deviating
$\rightarrow$ No policy regret strategies!

# Priority choice

A server can treat only one packet at a time.
Which packet to choose?

**At random?**
$\rightarrow$ unstable Nash equilibria with large $\eta$ ($\gtrsim N^{1/3}$)




$N = K$
$\lambda_1 = \frac{2}{N^{1/3}}$ and $\lambda_i = \frac{1}{N^{2/3}}$ for all $i \geq 2$
$\mu_1 = \frac{1}{2}$ and $\lambda_i = \frac{c}{N^{1/3}}$ for all $i \geq 2$

queue 1 cannot clear

# Priority choice

A server can treat only one packet at a time.
Which packet to choose?

**Treat oldest packet**
$\rightarrow$ force better Nash equilibria
$\rightarrow$ carryover effect

if some queue accumulates packets $\rightarrow$ gets priority
bad performance for other queues on the long run $\rightarrow$ incites to cooperation

# Patient game

Define game $\mathcal{G} = ([N], (c_i)_{i=1}^n, \boldsymbol{\mu}, \boldsymbol{\lambda})$ with

**Action Space:** $p_i \in \mathcal{P}([K])$

**Cost Function:** All queues choose their server $a_t^i \sim p_i$ at each time step and

$$c_i(p_i, \boldsymbol{p_{-i}}) = \lim_{t \to +\infty} \frac{T_t^i}{t}$$

- $T_t^i$ is the age of the oldest packet in queue $i$ at time $t$
- this limit exists (deterministically)
- queue $i$ is stable $\implies c_i(p_i, \boldsymbol{p_{-i}}) = 0$