

# **Online Learning for Wireless Communications: Theory, Algorithms, and Applications**

**Cong Shen**

University of Virginia

**Cem Tekin**

Bilkent University

**Mihaela van der Schaar**

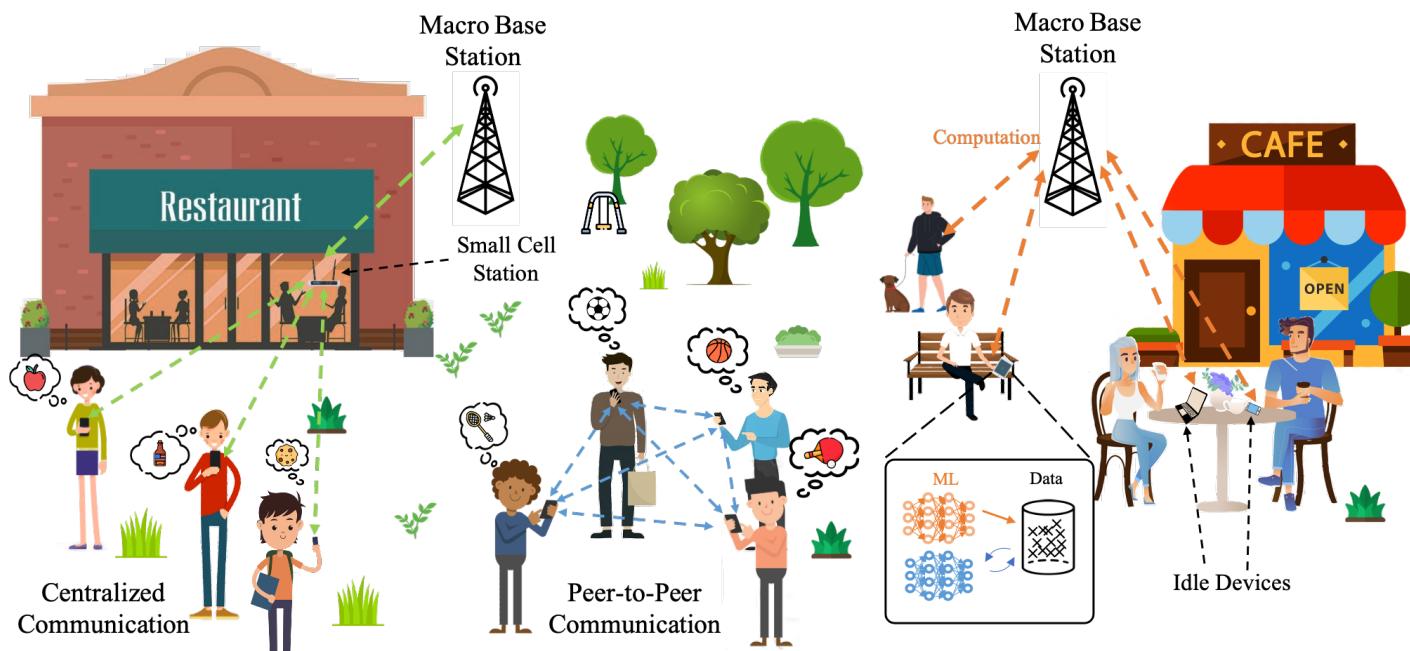
University of Cambridge  
The Alan Turing Institute  
UCLA

**ICC 2021 Tutorial**

June 2021

# Unprecedented Network Complexity

- The million (billion?) dollar question: adaptive, automated and autonomous optimization of multiple heterogeneous network entities that can learn, match and evolve with the system dynamics, while only observing limited information about the environment.



# The Need for Online Learning

- Previous design philosophy is increasingly **insufficient**
  - Optimization
  - Engineering heuristics, domain expert to solve “corner cases”
  - Data-first, model-later supervised learning, including deep learning
- Fundamental issues
  - Extremely **complex** wireless system
  - **Unknown** and **highly dynamic** deployment environment
  - Coupling behavior, **non-convexity**, computational **complexity**, etc

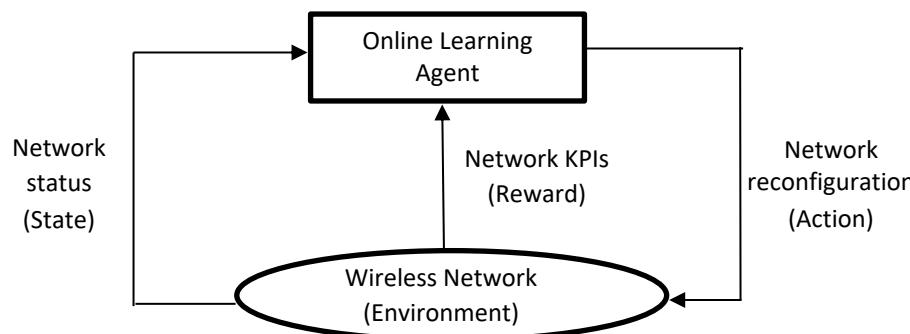
# What is Online Learning?

- From Wikipedia: online learning “[is] a method of machine learning in which data becomes available in a **sequential** order and is used to update the best predictor for future data **at each step**”
- This tutorial focuses on **sequential decision making under uncertainty**, which is intimately related to the applications in wireless communications
- We will mostly talk about two tools in this field
  - Multi-armed bandits
  - Reinforcement learning

# Why Online Learning?

- Characteristics of online learning match very well with wireless communications
  - closed-loop and sequential operation
  - long-term performance criteria
  - existing feedback protocols

Today's wireless systems have much stronger capabilities to enable the adoption of online learning techniques



# Why Online Learning?

- Online learning has matured over the past decade
  - Notable advances in **multi-armed bandits** and **reinforcement learning**, both theory and practice
  - New methods such as **deep reinforcement learning** have seen success in many applications, such as speech recognition, computer vision, natural language processing, and games
- Online learning has flashed promises in solving complex wireless communication problems
  - Spectrum engineering
  - Wireless network management, especially under the O-RAN architecture

# Outline of the Tutorial

- Part 1: Overview of online learning for wireless communications (15min, CS)
- Part 2: Multi-armed bandits (60min, CT)
- Part 3: Reinforcement learning (50min, MvdS)
- Part 4: Design examples in wireless communications (50min, CS)

# Multi-armed Bandits for Wireless Communications

ICC 2021 Tutorial Online Learning for Wireless Communications: Theory, Algorithms, and Applications

Cem Tekin

Associate Professor

Department of Electrical and Electronics Engineering

Cognitive Systems, Bandits, and Optimization Research Group

Bilkent University



[Intro to MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Policies for MAB](#)

[Comparison & Discussion](#)

[Combinatorial MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Combinatorial UCB](#)

[Extensions & Discussion](#)

[Decentralized  
multi-user MAB](#)

[Formulation](#)

[Centralized solution &  
regret](#)

[Decentralized solution](#)

[Extensions & Discussion](#)

[Other models](#)

[References](#)

## 1 Intro to MAB

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## 2 Combinatorial MAB

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## 3 Decentralized multi-user MAB

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## 4 Other models

### Other models

### References

## 5 References

# Part 1: Intro to MAB

Multi-armed Bandits  
for Wireless  
Communications

- 1 **Intro to MAB**
- Formulation & Examples
  - Regret
  - Policies for MAB
  - Comparison & Discussion

## Intro to MAB

- Formulation & Examples
- Regret
- Policies for MAB
- Comparison & Discussion

## Combinatorial MAB

- Formulation & Examples
- Regret
- Combinatorial UCB
- Extensions & Discussion

## Decentralized multi-user MAB

- Formulation
- Centralized solution & regret
- Decentralized solution
- Extensions & Discussion

## Other models

## References

# Multi-armed bandit (MAB) problem



- Gambling in a casino with  $K$  slot machines (arms) sequentially over rounds  $t = 1, 2, \dots$
- When played in round  $t$ , “arm  $a$ ” yields random reward  $R_t$  that comes from unknown  $F_a$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Multi-armed bandit (MAB) problem



- Gambling in a casino with  $K$  slot machines (arms) sequentially over rounds  $t = 1, 2, \dots$
- When played in round  $t$ , “arm  $a$ ” yields random reward  $R_t$  that comes from unknown  $F_a$

In each round  $t$

- Play an arm  $A_t$  based on  $A_1, \dots, A_{t-1}$  &  $R_1, \dots, R_{t-1}$
- Observe & collect its random reward  $R_t \sim F_{A_t}$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Multi-armed bandit (MAB) problem



- Gambling in a casino with  $K$  slot machines (arms) sequentially over rounds  $t = 1, 2, \dots$
- When played in round  $t$ , “arm  $a$ ” yields random reward  $R_t$  that comes from unknown  $F_a$

In each round  $t$

- Play an arm  $A_t$  based on  $A_1, \dots, A_{t-1}$  &  $R_1, \dots, R_{t-1}$
- Observe & collect its random reward  $R_t \sim F_{A_t}$

Goal

- Maximize expected cumulative reward  $\mathbb{E} [\sum_t R_t]$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# MAB in communications - Dynamic channel selection

Multi-armed Bandits  
for Wireless  
Communications

- Time-slotted communication,  $t = 1, 2, \dots$
- $K$  channels  $\{C_1, \dots, C_K\}$  with time-varying qualities
- Channel gains & distributions are unknown

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

- Time-slotted communication,  $t = 1, 2, \dots$
- $K$  channels  $\{C_1, \dots, C_K\}$  with time-varying qualities
- Channel gains & distributions are unknown

In each time slot  $t$

- Select a channel  $A_t$  based on  $A_1, \dots, A_{t-1}$  &  $R_1, \dots, R_{t-1}$
- Transmit on  $A_t$ , then observe  $R_t = \text{ACK/NAK}$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

- Time-slotted communication,  $t = 1, 2, \dots$
- $K$  channels  $\{C_1, \dots, C_K\}$  with time-varying qualities
- Channel gains & distributions are unknown

In each time slot  $t$

- Select a channel  $A_t$  based on  $A_1, \dots, A_{t-1}$  &  $R_1, \dots, R_{t-1}$
- Transmit on  $A_t$ , then observe  $R_t = \text{ACK/NAK}$

Goal

- Maximize expected number of successful transmissions

# MAB in communications - Beam alignment

Multi-armed Bandits  
for Wireless  
Communications

- Time-slotted communication,  $t = 1, 2, \dots$
- $K$  beams  $\{B_1, \dots, B_K\}$  with time varying qualities
- Unknown expected received signal strengths (RSS)

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

- Time-slotted communication,  $t = 1, 2, \dots$
- $K$  beams  $\{B_1, \dots, B_K\}$  with time varying qualities
- Unknown expected received signal strengths (RSS)

In each time slot  $t$

- Select a beam  $A_t$  based on  $A_1, \dots, A_{t-1}$  &  $R_1, \dots, R_{t-1}$
- Transmit with  $A_t$ , then observe  $R_t = \text{noisy RSS}$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

- Time-slotted communication,  $t = 1, 2, \dots$
- $K$  beams  $\{B_1, \dots, B_K\}$  with time varying qualities
- Unknown expected received signal strengths (RSS)

In each time slot  $t$

- Select a beam  $A_t$  based on  $A_1, \dots, A_{t-1}$  &  $R_1, \dots, R_{t-1}$
- Transmit with  $A_t$ , then observe  $R_t = \text{noisy RSS}$

Goal

- Maximize expected cumulative RSS

## Properties of the environment

- Arm set  $[K] = \{1, \dots, K\}$  ([known](#))
- Arms are independent ([known](#))
- Reward distribution  $F = F_1 \times \dots \times F_K$  ([unknown](#))
- Distribution type (e.g., Bernoulli, Gaussian) ([known](#))
- Expected rewards  $\mu = [\mu_1, \dots, \mu_K]$  ([unknown](#))

[Intro to MAB](#)

[Formulation & Examples](#)

Regret

Policies for MAB

Comparison & Discussion

[Combinatorial MAB](#)

[Formulation & Examples](#)

Regret

Combinatorial UCB

Extensions & Discussion

[Decentralized  
multi-user MAB](#)

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

[Other models](#)

[References](#)

## Properties of the environment

- Arm set  $[K] = \{1, \dots, K\}$  ([known](#))
- Arms are independent ([known](#))
- Reward distribution  $F = F_1 \times \dots \times F_K$  ([unknown](#))
- Distribution type (e.g., Bernoulli, Gaussian) ([known](#))
- Expected rewards  $\mu = [\mu_1, \dots, \mu_K]$  ([unknown](#))

## Environment class $\mathcal{E}$

- E.g., (A) all  $K$ -armed bandits with Bernoulli rewards
- E.g., (B) all  $K$ -armed bandits with rewards in  $[0, 1]$  (bounded support)
- This tutorial mostly considers (A) or (B). Others include subgaussian, Gaussian, heavy-tailed, etc.

[Intro to MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Policies for MAB](#)

[Comparison & Discussion](#)

[Combinatorial MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Combinatorial UCB](#)

[Extensions & Discussion](#)

[Decentralized  
multi-user MAB](#)

[Formulation](#)

[Centralized solution &  
regret](#)

[Decentralized solution](#)

[Extensions & Discussion](#)

[Other models](#)

[References](#)

# How to choose $A_t$ ?

Construct a policy for the environment class  $\mathcal{E}$

- History  $\mathcal{H}_t = \{A_1, R_1, \dots, A_{t-1}, R_{t-1}\}$
- Policy  $\pi : \text{histories} \rightarrow \text{distributions over } [K]$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# How to choose $A_t$ ?

Construct a policy for the environment class  $\mathcal{E}$

- History  $\mathcal{H}_t = \{A_1, R_1, \dots, A_{t-1}, R_{t-1}\}$
- Policy  $\pi : \text{histories} \rightarrow \text{distributions over } [K]$

Follow the policy

- Play  $A_t \sim \pi(\cdot | \mathcal{H}_t)$
- Observe  $R_t \sim F_{A_t}$
- Update  $\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{A_t, R_t\}$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Regret of a policy

Goal Maximize

$$\mathbb{E} \left[ \sum_{t=1}^T R_t \right]$$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

Goal Maximize

$$\mathbb{E} \left[ \sum_{t=1}^T R_t \right]$$

Highest cumulative reward over  $T$  rounds?

- Best arm  $a^* = \arg \max_a \mu_a$
- Highest expected reward:  $\mu^* = \max_a \mu_a$
- Highest cumulative expected reward:  $T\mu^*$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Regret of a policy

Goal Maximize

$$\mathbb{E} \left[ \sum_{t=1}^T R_t \right]$$

Highest cumulative reward over  $T$  rounds?

- Best arm  $a^* = \arg \max_a \mu_a$
- Highest expected reward:  $\mu^* = \max_a \mu_a$
- Highest cumulative expected reward:  $T\mu^*$

(Pseudo) Regret

$$\text{Reg}_\pi(T) = T \times \mu^* - \sum_{t=1}^T \mu_{A_t}$$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Regret of a policy

Goal Maximize

$$\mathbb{E} \left[ \sum_{t=1}^T R_t \right]$$

Highest cumulative reward over  $T$  rounds?

- Best arm  $a^* = \arg \max_a \mu_a$
- Highest expected reward:  $\mu^* = \max_a \mu_a$
- Highest cumulative expected reward:  $T\mu^*$

(Pseudo) Regret

$$\text{Reg}_\pi(T) = T \times \mu^* - \sum_{t=1}^T \mu_{A_t}$$

Fact

$$\max_\pi \mathbb{E} \left[ \sum_{t=1}^T R_t \right] = \min_\pi \mathbb{E} [\text{Reg}_\pi(T)]$$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# What is a good policy?

## Definition (Good policy)

For all bandit instances in  $\mathcal{E}$  (e.g., all  $K$ -armed bandits with independent arm rewards in  $[0, 1]$ )

$$\lim_{T \rightarrow \infty} \underbrace{\frac{\mathbb{E} [\text{Reg}_{\pi}(T)]}{T}}_{\text{time avg. regret}} = 0$$

### Intro to MAB

Formulation & Examples

#### Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

#### Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# What is a good policy?

## Definition (Good policy)

For all bandit instances in  $\mathcal{E}$  (e.g., all  $K$ -armed bandits with independent arm rewards in  $[0, 1]$ )

$$\lim_{T \rightarrow \infty} \underbrace{\frac{\mathbb{E} [\text{Reg}_{\pi}(T)]}{T}}_{\text{time avg. regret}} = 0$$

E.g.,  $\mathbb{E} [\text{Reg}_{\pi}(T)] = O(\sqrt{T})$ ,  $\mathbb{E} [\text{Reg}_{\pi}(T)] = O(\log T)$

### Intro to MAB

Formulation & Examples

#### Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

#### Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# What is a good policy?

## Definition (Good policy)

For all bandit instances in  $\mathcal{E}$  (e.g., all  $K$ -armed bandits with independent arm rewards in  $[0, 1]$ )

$$\lim_{T \rightarrow \infty} \underbrace{\frac{\mathbb{E} [\text{Reg}_{\pi}(T)]}{T}}_{\text{time avg. regret}} = 0$$

E.g.,  $\mathbb{E} [\text{Reg}_{\pi}(T)] = O(\sqrt{T})$ ,  $\mathbb{E} [\text{Reg}_{\pi}(T)] = O(\log T)$

A *good policy* should

- Explore arms to discover the best
- Exploit the arm that is believed to be the best
- Be computationally efficient

### Intro to MAB

Formulation & Examples

#### Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

#### Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Regret decomposition

$\Delta_a = \mu^* - \mu_a$  suboptimality gap

$N_{a,t} = \sum_{s=1}^t \mathbb{I}(A_s = a)$  number of plays of “arm  $a$ ” by round  $t$

## Lemma (Regret decomposition)

$$\mathbb{E}[\text{Reg}_\pi(T)] = \sum_{a=1}^K \Delta_a \mathbb{E}[N_{a,T}]$$

### Proof.

$$\mathbb{E}[\text{Reg}_\pi(T)] = T\mu^* - \mathbb{E}\left[\sum_{t=1}^T \mu_{A_t}\right] = \sum_{t=1}^T \mu^* - \mathbb{E}\underbrace{\left[\sum_{t=1}^T \sum_{a=1}^K \mu_a \mathbb{I}(A_t = a)\right]}_{\sum_{a=1}^K \mathbb{I}(A_t = a) = 1} \quad (1)$$

$$= \mathbb{E}\left[\sum_{a=1}^K (\mu^* - \mu_a) \underbrace{\sum_{t=1}^T \mathbb{I}(A_t = a)}_{N_{a,T}}\right] \quad (2)$$

$$= \sum_{a=1}^K \Delta_a \mathbb{E}[N_{a,T}] \quad (3)$$



### Intro to MAB

Formulation & Examples

#### Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

#### Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution & regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Definition (Consistent policy)

$\pi$  is consistent if for all  $F \in \mathcal{E}$  and  $p > 0$

$$\lim_{T \rightarrow \infty} \frac{\mathbb{E}[\text{Reg}_\pi(T)]}{T^p} = 0 \Rightarrow \mathbb{E}[\text{Reg}_\pi(T)] = o(T^p)$$

### Intro to MAB

Formulation & Examples

#### Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

#### Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Regret lower bound

## Definition (Consistent policy)

$\pi$  is consistent if for all  $F \in \mathcal{E}$  and  $p > 0$

$$\lim_{T \rightarrow \infty} \frac{\mathbb{E}[\text{Reg}_\pi(T)]}{T^p} = 0 \Rightarrow \mathbb{E}[\text{Reg}_\pi(T)] = o(T^p)$$

## Theorem (Asymptotic lower bound (Lai and Robbins 1985))

Let  $\mathcal{E}$  be class of bandits with independent single parameter exponential family of reward distributions (e.g.,  $F_a = \text{Ber}(\theta_a)$ ).  
For a consistent policy  $\pi$

$$\liminf_{T \rightarrow \infty} \frac{\mathbb{E}[\text{Reg}_\pi(T)]}{\log T} \geq \sum_{a: \mu_a < \mu^*} \frac{\mu^* - \mu_a}{KL(a, a^*)}$$
$$\Rightarrow \mathbb{E}[\text{Reg}_\pi(T)] = \Omega(\log T)$$

### Intro to MAB

Formulation & Examples

#### Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

#### Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Definition (Asymptotic optimality)

Policies for which

$$\lim_{T \rightarrow \infty} \frac{\mathbb{E}[\text{Reg}_\pi(T)]}{\log T} = \sum_{a: \mu_a < \mu^*} \frac{\mu^* - \mu_a}{\text{KL}(a, a^*)}$$

are called *asymptotically optimal*

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Definition (Asymptotic optimality)

Policies for which

$$\lim_{T \rightarrow \infty} \frac{\mathbb{E}[\text{Reg}_\pi(T)]}{\log T} = \sum_{a: \mu_a < \mu^*} \frac{\mu^* - \mu_a}{\text{KL}(a, a^*)}$$

are called *asymptotically optimal*

## Definition (Order optimality)

Policies for which  $\mathbb{E}[\text{Reg}_\pi(T)] = O(\log T)$  are called *order optimal*

## Greedy policy

- 1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$

[Intro to MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Policies for MAB](#)

[Comparison & Discussion](#)

[Combinatorial MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Combinatorial UCB](#)

[Extensions & Discussion](#)

[Decentralized  
multi-user MAB](#)

[Formulation](#)

[Centralized solution &  
regret](#)

[Decentralized solution](#)

[Extensions & Discussion](#)

[Other models](#)

[References](#)

# Greedy policy

---

## Greedy policy

---

- 1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$
- 2: **Initialization:** sample each arm once

[Intro to MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Policies for MAB](#)

[Comparison & Discussion](#)

[Combinatorial MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Combinatorial UCB](#)

[Extensions & Discussion](#)

[Decentralized  
multi-user MAB](#)

[Formulation](#)

[Centralized solution &  
regret](#)

[Decentralized solution](#)

[Extensions & Discussion](#)

[Other models](#)

[References](#)

# Greedy policy

## Greedy policy

- 1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$
- 2: **Initialization:** sample each arm once
- 3: **At each round  $t > K$ :**
  - Select  $A_t = \arg \max_a \hat{\mu}_a := \frac{V_a}{N_a}$
  - Collect reward  $R_t$
  - Update parameters  $V_{A_t} = V_{A_t} + R_t$ ,  $N_{A_t} = N_{A_t} + 1$

[Intro to MAB](#)[Formulation & Examples](#)[Regret](#)[Policies for MAB](#)[Comparison & Discussion](#)[Combinatorial MAB](#)[Formulation & Examples](#)[Regret](#)[Combinatorial UCB](#)[Extensions & Discussion](#)[Decentralized  
multi-user MAB](#)[Formulation](#)[Centralized solution &  
regret](#)[Decentralized solution](#)[Extensions & Discussion](#)[Other models](#)[References](#)

# Greedy policy

## Greedy policy

1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$

2: **Initialization:** sample each arm once

3: **At each round**  $t > K$ :

- Select  $A_t = \arg \max_a \hat{\mu}_a := \frac{V_a}{N_a}$
- Collect reward  $R_t$
- Update parameters  $V_{A_t} = V_{A_t} + R_t$ ,  $N_{A_t} = N_{A_t} + 1$

- Never explores, always exploits
- Might get stuck selecting a suboptimal arm
- Incurs linear regret in the long run

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution & regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## UCB policy

1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$

[Intro to MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Policies for MAB](#)

[Comparison & Discussion](#)

[Combinatorial MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Combinatorial UCB](#)

[Extensions & Discussion](#)

[Decentralized  
multi-user MAB](#)

[Formulation](#)

[Centralized solution &  
regret](#)

[Decentralized solution](#)

[Extensions & Discussion](#)

[Other models](#)

[References](#)

## UCB policy

- 1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$
- 2: **Initialization:** sample each arm once

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Upper Confidence Bound (UCB) policy

## UCB policy

- 1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$
- 2: **Initialization:** sample each arm once
- 3: **At each round  $t > K$ :**
  - Calculate UCB index of each arm  $a \in [K]$

$$g_a = \underbrace{\frac{V_a}{N_a}}_{\hat{\mu}_a} + \underbrace{\sqrt{\frac{2 \log t}{N_a}}}_{\text{exp. bonus}}$$

- Select the optimistic best arm

$$A_t = \arg \max_a g_a$$

- Collect reward  $R_t \in [0, 1]$
- Update parameters  $V_{A_t} = V_{A_t} + R_t$ ,  $N_{A_t} = N_{A_t} + 1$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Regret of UCB policy

Multi-armed Bandits  
for Wireless  
Communications

[Intro to MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Policies for MAB](#)

[Comparison & Discussion](#)

**Combinatorial MAB**

[Formulation & Examples](#)

[Regret](#)

[Combinatorial UCB](#)

[Extensions & Discussion](#)

**Decentralized  
multi-user MAB**

[Formulation](#)

[Centralized solution &  
regret](#)

[Decentralized solution](#)

[Extensions & Discussion](#)

[Other models](#)

[References](#)

## Lemma ( $g_a$ is a UCB)

$g_a \geq \mu_a$  for all arms in all rounds with high probability

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Lemma ( $g_a$ is a UCB)

$g_a \geq \mu_a$  for all arms in all rounds with high probability

- Assume that  $A_t = a$  is suboptimal, i.e.,  $\Delta_a > 0$ . Then,

$$g_a \geq g_{a^*} \geq \mu^* = \mu_a + \Delta_a > \mu_a$$

- If  $A_t = a$  happens a lot, then,  $g_a \rightarrow \hat{\mu}_a \rightarrow \mu_a$
- $A_t = a$  cannot happen a lot since  $\mu_a < \mu^* \leq g_{a^*}$

# Regret of UCB policy

## Theorem (Regret of UCB (Auer et al. 2002))

$$\begin{aligned}\mathbb{E} [Reg_{UCB}(T)] &\leq 8 \sum_{a: \mu_a < \mu^*} \frac{\log T}{\mu^* - \mu_a} + \left(1 + \frac{\pi^2}{3}\right) \sum_a (\mu^* - \mu_a) \\ &= O\left(\sum_{a: \mu_a < \mu^*} \frac{\log T}{\Delta_a}\right)\end{aligned}$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Theorem (Regret of UCB (Auer et al. 2002))

$$\begin{aligned}\mathbb{E} [Reg_{UCB}(T)] &\leq 8 \sum_{a: \mu_a < \mu^*} \frac{\log T}{\mu^* - \mu_a} + \left(1 + \frac{\pi^2}{3}\right) \sum_a (\mu^* - \mu_a) \\ &= O\left(\sum_{a: \mu_a < \mu^*} \frac{\log T}{\Delta_a}\right)\end{aligned}$$

## Takeaways

- Order optimal  $O(\log T)$  regret
- $\Delta_a$  small  $\Rightarrow$  harder to distinguish  $\Rightarrow$  higher regret
- Exploration achieved by *optimism under uncertainty*
- $A_t$  deterministic given history

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# UCB in action

Multi-armed Bandits  
for Wireless  
Communications

## Intro to MAB

Formulation & Examples

Regret

## Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References

# Thompson (posterior) sampling

Bayesian algorithm (William R. Thompson in 1933)

- 1 Start with prior over bandit instances  $p(F)$
- 2 Compute posterior distribution of the optimal arm  $p(a^*|\mathcal{H}_t)$
- 3  $A_t \sim p(a^*|\mathcal{H}_t)$

[Intro to MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Policies for MAB](#)

[Comparison & Discussion](#)

[Combinatorial MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Combinatorial UCB](#)

[Extensions & Discussion](#)

[Decentralized  
multi-user MAB](#)

[Formulation](#)

[Centralized solution &  
regret](#)

[Decentralized solution](#)

[Extensions & Discussion](#)

[Other models](#)

[References](#)

# Thompson (posterior) sampling

## Intro to MAB

Formulation & Examples

Regret

## Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References

Bayesian algorithm (William R. Thompson in 1933)

- 1 Start with prior over bandit instances  $p(F)$
- 2 Compute posterior distribution of the optimal arm  $p(a^*|\mathcal{H}_t)$
- 3  $A_t \sim p(a^*|\mathcal{H}_t)$

Equivalently

- 1 Start with prior over bandit instances  $p(F)$
- 2 Compute posterior over bandit instances  $p(F|\mathcal{H}_t)$
- 3 Sample a bandit instance  $\hat{F} \sim p(F|\mathcal{H}_t)$
- 4  $A_t = \arg \max_a \mu_a(\hat{F})$

# Thompson sampling for Bernoulli bandits

Multi-armed Bandits  
for Wireless  
Communications

- $F_a = \text{Ber}(\theta_a)$ ,  $F = F_1 \times \cdots \times F_K$
- Prior over  $\theta_a$ : Beta(1, 1)
- Prior over  $(\theta_1, \dots, \theta_K)$ :  $\prod_{a=1}^K \text{Beta}(1, 1)$
- Posterior over  $\theta_a$ : Beta( $1 + \alpha_a$ ,  $1 + \beta_a$ )

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Thompson sampling for Bernoulli bandits

Multi-armed Bandits  
for Wireless  
Communications

- $F_a = \text{Ber}(\theta_a)$ ,  $F = F_1 \times \dots \times F_K$
- Prior over  $\theta_a$ : Beta(1, 1)
- Prior over  $(\theta_1, \dots, \theta_K)$ :  $\prod_{a=1}^K \text{Beta}(1, 1)$
- Posterior over  $\theta_a$ : Beta( $1 + \alpha_a$ ,  $1 + \beta_a$ )

---

## Thompson sampling (Bernoulli arms)

---

1: **Store & update:** success  $\alpha_a$  & failure  $\beta_a$  counts for each  $a \in [K]$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Thompson sampling for Bernoulli bandits

- $F_a = \text{Ber}(\theta_a)$ ,  $F = F_1 \times \cdots \times F_K$
- Prior over  $\theta_a$ : Beta(1, 1)
- Prior over  $(\theta_1, \dots, \theta_K)$ :  $\prod_{a=1}^K \text{Beta}(1, 1)$
- Posterior over  $\theta_a$ : Beta( $1 + \alpha_a$ ,  $1 + \beta_a$ )

---

## Thompson sampling (Bernoulli arms)

---

- 1: **Store & update:** success  $\alpha_a$  & failure  $\beta_a$  counts for each  $a \in [K]$
- 2: **At each round  $t$ :**

- Sample  $\theta_a \sim \text{Beta}(1 + \alpha_a, 1 + \beta_a)$  for all  $a \in [K]$
  - Select  $A_t = \arg \max_a \theta_a$
  - Collect reward  $R_t \in \{0, 1\}$
  - Update parameters  $\alpha_{A_t} = \alpha_{A_t} + R_t$ ,  $\beta_{A_t} = \beta_{A_t} + 1 - R_t$
- 

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Regret bound for Thompson sampling

Multi-armed Bandits  
for Wireless  
Communications

**Theorem (Regret of Thompson sampling (Kaufmann et al. 2012, Agrawal and Goyal 2013))**

*For Bernoulli bandits, for every  $\epsilon > 0$*

$$\begin{aligned}\mathbb{E} [Reg_{TS}(T)] &\leq (1 + \epsilon) \sum_{a: \mu_a < \mu^*} \underbrace{\frac{\log T}{KL(a, a^*)}}_{\Omega(\Delta_a^2)} \Delta_a + O\left(\frac{K}{\epsilon^2}\right) \\ &= O\left(\sum_{a: \mu_a < \mu^*} \frac{\log T}{\Delta_a}\right)\end{aligned}$$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Regret bound for Thompson sampling

**Theorem (Regret of Thompson sampling (Kaufmann et al. 2012, Agrawal and Goyal 2013))**

*For Bernoulli bandits, for every  $\epsilon > 0$*

$$\begin{aligned}\mathbb{E} [Reg_{TS}(T)] &\leq (1 + \epsilon) \sum_{a: \mu_a < \mu^*} \underbrace{\frac{\log T}{KL(a, a^*)}}_{\Omega(\Delta_a^2)} \Delta_a + O\left(\frac{K}{\epsilon^2}\right) \\ &= O\left(\sum_{a: \mu_a < \mu^*} \frac{\log T}{\Delta_a}\right)\end{aligned}$$

## Takeaways

- Asymptotically optimal regret
- Exploration achieved by sampling from posterior
- Randomized policy

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Thompson sampling in action

Multi-armed Bandits  
for Wireless  
Communications

## Intro to MAB

Formulation & Examples

Regret

## Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

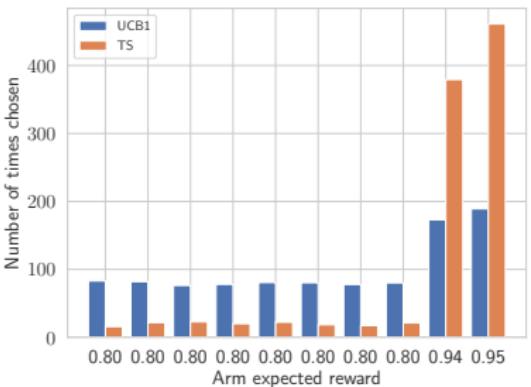
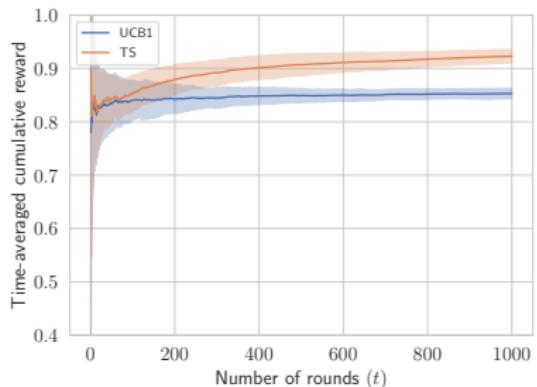
Decentralized solution

Extensions & Discussion

## Other models

## References

# Empirical comparison



- Thompson sampling mostly works very well in practice
- Tighter UCBs  $\Rightarrow$  smaller regret. E.g., KL-UCB (Garivier and Cappé, 2011)

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution & regret

Decentralized solution

Extensions & Discussion

## Other models

References

## Regret bounds obtained so far

- Depend on the bandit instance  $F$  (gap-dependent)
- UCB:  $O\left(\sum_{a: \mu_a < \mu^*} \frac{\log T}{\Delta_a}\right)$
- Thompson sampling:  $O\left(\sum_{a: \mu_a < \mu^*} \frac{\log T}{\Delta_a}\right)$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Regret bounds obtained so far

- Depend on the bandit instance  $F$  (gap-dependent)
- UCB:  $O\left(\sum_{a:\mu_a < \mu^*} \frac{\log T}{\Delta_a}\right)$
- Thompson sampling:  $O\left(\sum_{a:\mu_a < \mu^*} \frac{\log T}{\Delta_a}\right)$

## Gap-free regret bounds

- Hold for any bandit instance
- UCB:  $O(\sqrt{KT \log T})$
- Thompson sampling:  $O(\sqrt{KT \log T})$

## Minimax regret

$$\inf_{\text{all policies}} \sup_{\text{all bandit instances in } \mathcal{E}} \mathbb{E}[\text{Reg}_\pi(T)] = \Omega(\sqrt{KT})$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

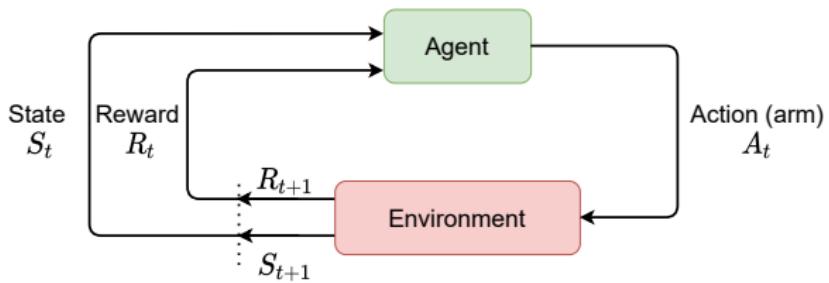
### References

# Multi-armed bandits and reinforcement learning

Multi-armed Bandits  
for Wireless  
Communications

## General RL framework

- Repeated interaction over time  $t = 1, 2, \dots$
- $(S_t, A_t) \rightarrow (S_{t+1}, R_{t+1})$



### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

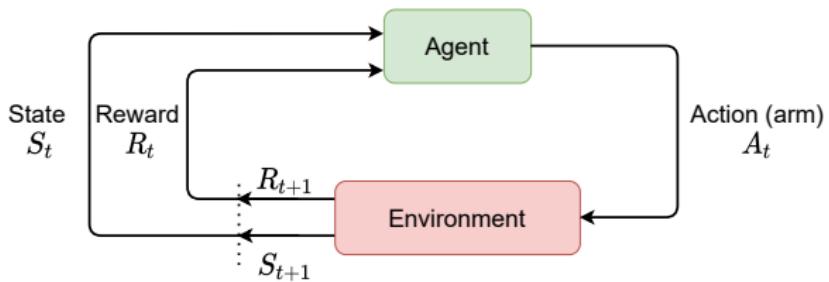
Extensions & Discussion

### Other models

### References

## General RL framework

- Repeated interaction over time  $t = 1, 2, \dots$
- $(S_t, A_t) \rightarrow (S_{t+1}, R_{t+1})$



## Comparison

- General RL:  $S_{t+1}$  depends on past actions and states (e.g., Markov model)
- $K$ -armed stochastic bandit: one state
- More structure  $\Rightarrow$  more specialized algorithms & faster learning/convergence & rigorous optimality guarantees

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Summary of Part 1

- Studied stochastic  $K$ -armed bandit.
  - ▶  $R_{a,t} \sim F_a$  (unknown), indep. of other arms
- Any consistent policy incurs at least  $\Omega(\log T)$  regret
- UCB and TS achieve  $O(\log T)$  regret
- UCB explores by being optimistic
- TS explores by sampling from posterior

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References

## Part 2: Combinatorial MAB

Multi-armed Bandits  
for Wireless  
Communications

- 2 Combinatorial MAB
- Formulation & Examples
  - Regret
  - Combinatorial UCB
  - Extensions & Discussion

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Combinatorial semi-bandits

## Formulation

- $K$  base arms
- Base arm outcome vector:  $\mathbf{X}_t = (X_{1,t}, \dots, X_{K,t}) \sim F$
- $X_{a,t} \in [0, 1]$
- Super arm: collection of base arms, e.g.,  $\{1, 3, 5\}$
- Set of feasible super arms:  $\mathcal{I} \subseteq 2^{\{1, \dots, K\}}$
- Reward of super arm  $S$ :

$$f(S, \mathbf{X}_t) = \sum_{a \in S} X_{a,t} \quad (\text{linear rewards})$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Combinatorial semi-bandits

## Formulation

- $K$  base arms
- Base arm outcome vector:  $\mathbf{X}_t = (X_{1,t}, \dots, X_{K,t}) \sim F$
- $X_{a,t} \in [0, 1]$
- Super arm: collection of base arms, e.g.,  $\{1, 3, 5\}$
- Set of feasible super arms:  $\mathcal{I} \subseteq 2^{\{1, \dots, K\}}$
- Reward of super arm  $S$ :

$$f(S, \mathbf{X}_t) = \sum_{a \in S} X_{a,t} \quad (\text{linear rewards})$$

## In each round $t$

- Select a super arm  $S_t \in \mathcal{I}$
- Collect reward  $f(S_t, \mathbf{X}_t)$  (bandit feedback)
- Observe outcomes  $X_{a,t}$  of  $a \in S_t$  (semi-bandit feedback)

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Combinatorial semi-bandits

## Formulation

- $K$  base arms
- Base arm outcome vector:  $\mathbf{X}_t = (X_{1,t}, \dots, X_{K,t}) \sim F$
- $X_{a,t} \in [0, 1]$
- Super arm: collection of base arms, e.g.,  $\{1, 3, 5\}$
- Set of feasible super arms:  $\mathcal{I} \subseteq 2^{\{1, \dots, K\}}$
- Reward of super arm  $S$ :

$$f(S, \mathbf{X}_t) = \sum_{a \in S} X_{a,t} \quad (\text{linear rewards})$$

## In each round $t$

- Select a super arm  $S_t \in \mathcal{I}$
- Collect reward  $f(S_t, \mathbf{X}_t)$  (bandit feedback)
- Observe outcomes  $X_{a,t}$  of  $a \in S_t$  (semi-bandit feedback)

## Goal

- Maximize  $\mathbb{E} \left[ \sum_{t=1}^T f(S_t, \mathbf{X}_t) \right]$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# CMAB example: multi-user communication

- $L$  users,  $K$  channels ( $L \leq K$ )
- Only one user can transmit on each channel, otherwise collision

Multi-armed Bandits  
for Wireless  
Communications

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## CMAB example: multi-user communication

- $L$  users,  $K$  channels ( $L \leq K$ )
- Only one user can transmit on each channel, otherwise collision

	Ch1	Ch2	Ch3
U1	0.8	0.5	1.0
U2	0.5	0.6	0.9

Optimal matching: (U1,Ch1), (U2,Ch2)

A suboptimal matching: (U1,Ch3), (U2,Ch2)

- Base arms: User-channel pairs  $(u, c)$
- $X_{(u,c),t}$ : throughput of  $u$  on  $c$  as the sole user
- Super arms: One-to-one matchings of users to channels
- Reward: sum throughput  $f(S_t, \mathbf{X}_t) = \sum_{(u,c) \in S_t} X_{(u,c),t}$

### Intro to MAB

- Formulation & Examples
- Regret
- Policies for MAB
- Comparison & Discussion

### Combinatorial MAB

- Formulation & Examples
- Regret
- Combinatorial UCB
- Extensions & Discussion

### Decentralized multi-user MAB

- Formulation
- Centralized solution & regret
- Decentralized solution
- Extensions & Discussion

### Other models

### References

# CMAB example: path selection

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

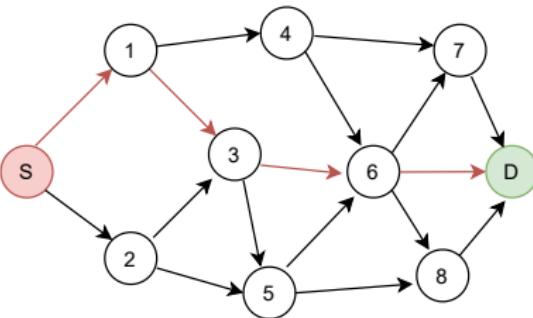
Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References



- Expected outcome of base arm  $a$ :  $\mu_a = \mathbb{E}[X_{a,t}]$
- Expected outcome vector:  $\mu = (\mu_1, \dots, \mu_K)$
- Optimal super arm:  $S^* = \arg \max_{S \in \mathcal{I}} f(S, \mu)$

## Intro to MAB

Formulation & Examples  
Regret  
Policies for MAB  
Comparison & Discussion

## Combinatorial MAB

Formulation & Examples  
Regret

Combinatorial UCB  
Extensions & Discussion

## Decentralized multi-user MAB

Formulation  
Centralized solution &  
regret  
Decentralized solution  
Extensions & Discussion

## Other models

## References

- Expected outcome of base arm  $a$ :  $\mu_a = \mathbb{E}[X_{a,t}]$
- Expected outcome vector:  $\mu = (\mu_1, \dots, \mu_K)$
- Optimal super arm:  $S^* = \arg \max_{S \in \mathcal{I}} f(S, \mu)$

## (Pseudo) regret

$$\text{Reg}_\pi(T) = T \times f(S^*, \mu) - \sum_{t=1}^T f(S_t, \mu)$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

- Expected outcome of base arm  $a$ :  $\mu_a = \mathbb{E}[X_{a,t}]$
- Expected outcome vector:  $\mu = (\mu_1, \dots, \mu_K)$
- Optimal super arm:  $S^* = \arg \max_{S \in \mathcal{I}} f(S, \mu)$

## (Pseudo) regret

$$\text{Reg}_\pi(T) = T \times f(S^*, \mu) - \sum_{t=1}^T f(S_t, \mu)$$

## Fact

$$\max_\pi \mathbb{E} \left[ \sum_{t=1}^T f(S_t, \mathbf{X}_t) \right] = \min_\pi \mathbb{E}[\text{Reg}_\pi(T)]$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Offline optimization oracle

- Knows  $\mathcal{I}$  and  $f(\cdot, \cdot)$
- Given any parameter vector  $\theta$  returns

$$\text{Oracle}(\theta) = \arg \max_{S \in \mathcal{I}} f(S, \theta)$$

[Intro to MAB](#)

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

[Combinatorial MAB](#)

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

[Decentralized  
multi-user MAB](#)

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

[Other models](#)

[References](#)

## Offline optimization oracle

- Knows  $\mathcal{I}$  and  $f(\cdot, \cdot)$
- Given any parameter vector  $\theta$  returns

$$\text{Oracle}(\theta) = \arg \max_{S \in \mathcal{I}} f(S, \theta)$$

Computationally efficient optimization oracles exist for many combinatorial problems of interest

- Maximum weighted bipartite matching: Hungarian algorithm
- Shortest path: Dijkstra's algorithm

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution & regret

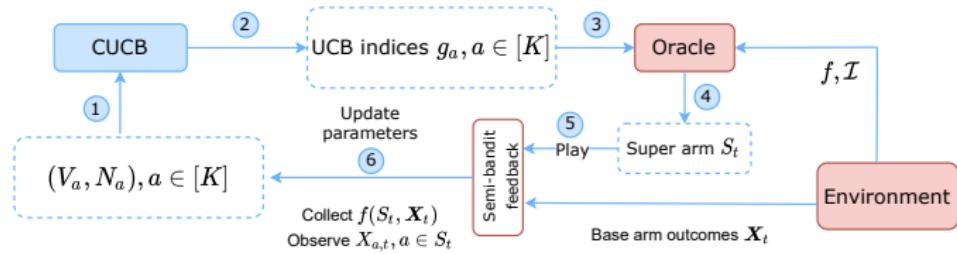
Decentralized solution

Extensions & Discussion

### Other models

### References

# Combinatorial UCB [Gai et al. 2012, Chen et al. 2013, Kveton et al. 2015]



## Combinatorial UCB (CUCB)

1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$

### Intro to MAB

- Formulation & Examples
- Regret
- Policies for MAB
- Comparison & Discussion

### Combinatorial MAB

- Formulation & Examples
- Regret
- Combinatorial UCB**
- Extensions & Discussion

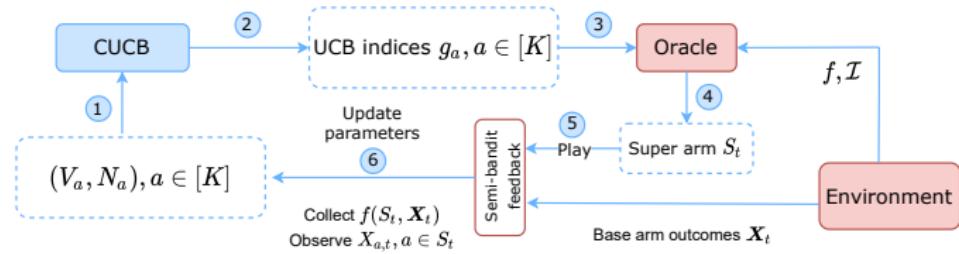
### Decentralized multi-user MAB

- Formulation
- Centralized solution & regret
- Decentralized solution
- Extensions & Discussion

### Other models

### References

# Combinatorial UCB [Gai et al. 2012, Chen et al. 2013, Kveton et al. 2015]



## Combinatorial UCB (CUCB)

- 1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$
- 2: **Initialization:** In first  $t_0$  rounds, initialize each  $a \in [K]$  with one observation

### Intro to MAB

Formulation & Examples  
Regret  
Policies for MAB  
Comparison & Discussion

### Combinatorial MAB

Formulation & Examples  
Regret  
**Combinatorial UCB**  
Extensions & Discussion

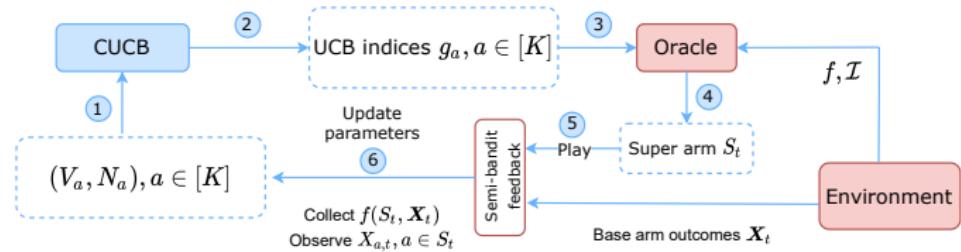
### Decentralized multi-user MAB

Formulation  
Centralized solution & regret  
Decentralized solution  
Extensions & Discussion

### Other models

### References

# Combinatorial UCB [Gai et al. 2012, Chen et al. 2013, Kveton et al. 2015]



## Combinatorial UCB (CUCB)

- 1: **Store & update:** cumulative rewards  $V_a$  & play counts  $N_a$  for each  $a \in [K]$
- 2: **Initialization:** In first  $t_0$  rounds, initialize each  $a \in [K]$  with one observation
- 3: **At each round  $t > t_0$ :**

- Calculate UCB indices of base arms:  $g_a = \hat{\mu}_a + \sqrt{\frac{1.5 \log t}{N_a}}$ ,  $a \in [K]$
- Query optimization oracle with UCB indices

$$S_t = \text{Oracle}(g_1, \dots, g_K) = \arg \max_{S \in \mathcal{I}} f(S, (g_1, \dots, g_K))$$

- Collect reward  $f(S_t, \mathbf{X}_t)$  and observe outcomes  $X_{a,t}$ ,  $a \in S_t$
- Update parameters based on observed outcomes

## Intro to MAB

Formulation & Examples  
 Regret  
 Policies for MAB  
 Comparison & Discussion

## Combinatorial MAB

Formulation & Examples  
 Regret  
**Combinatorial UCB**  
 Extensions & Discussion

## Decentralized multi-user MAB

Formulation  
 Centralized solution & regret  
 Decentralized solution  
 Extensions & Discussion

## Other models

## References

## Theorem (gap-dependent bound (Kveton et al. 2015))

$$\mathbb{E}[Reg_{CUCB}(T)] = O(KL(1/\Delta) \log T)$$

where  $L = \max_{S \in \mathcal{I}} |S|$  and  $\Delta$  is the gap between best and second best super arm

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Regret of combinatorial UCB

## Theorem (gap-dependent bound (Kveton et al. 2015))

$$\mathbb{E}[Reg_{CUCB}(T)] = O(KL(1/\Delta) \log T)$$

where  $L = \max_{S \in \mathcal{I}} |S|$  and  $\Delta$  is the gap between best and second best super arm

## Theorem (gap-free bound (Kveton et al. 2015))

Regret of combinatorial UCB is bounded as

$$\mathbb{E}[Reg_{CUCB}(T)] = O(\sqrt{KLT \log T})$$

where  $L = \max_{S \in \mathcal{I}} |S|$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution  
Extensions & Discussion

### Other models

### References

# Regret of combinatorial UCB

## Theorem (gap-dependent bound (Kveton et al. 2015))

$$\mathbb{E}[Reg_{CUCB}(T)] = O(KL(1/\Delta) \log T)$$

where  $L = \max_{S \in \mathcal{I}} |S|$  and  $\Delta$  is the gap between best and second best super arm

## Theorem (gap-free bound (Kveton et al. 2015))

Regret of combinatorial UCB is bounded as

$$\mathbb{E}[Reg_{CUCB}(T)] = O(\sqrt{KLT \log T})$$

where  $L = \max_{S \in \mathcal{I}} |S|$

## Theorem (lower bounds (Kveton et al. 2015))

$$\mathbb{E}[Reg_{\pi}(T)] = \Omega(KL(1/\Delta) \log T) \text{ gap-dependent}$$

$$\mathbb{E}[Reg_{\pi}(T)] = \Omega(\sqrt{KLT}) \text{ gap-free}$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution & regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Extension to non-linear rewards

Multi-armed Bandits  
for Wireless  
Communications

$r(S, \mu) = \mathbb{E}[f(S, X_t)]$  varies smoothly with  $\mu$

## Assumption (Lipschitz continuity)

$$|r(S, \mu) - r(S, \mu')| \leq B \|\mu_S - \mu'_S\|_1$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Extension to non-linear rewards

$r(S, \mu) = \mathbb{E}[f(S, X_t)]$  varies smoothly with  $\mu$

## Assumption (Lipschitz continuity)

$$|r(S, \mu) - r(S, \mu')| \leq B \|\mu_S - \mu'_S\|_1$$

## Assumption (Monotonicity)

If for all  $a \in [K]$ ,  $\mu_a < \mu'_a$ , then  $r(S, \mu) \leq r(S, \mu')$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Extension to non-linear rewards

$r(S, \mu) = \mathbb{E}[f(S, X_t)]$  varies smoothly with  $\mu$

### Assumption (Lipschitz continuity)

$$|r(S, \mu) - r(S, \mu')| \leq B \|\mu_S - \mu'_S\|_1$$

### Assumption (Monotonicity)

If for all  $a \in [K]$ ,  $\mu_a < \mu'_a$ , then  $r(S, \mu) \leq r(S, \mu')$

Same algorithm, different optimization oracle. Call oracle with UCB indices to get the *estimated optimal* super arm.

### Theorem (Chen et al. 2015)

*Under Lipschitz continuity (or more general bounded smoothness) and monotonicity assumptions*

$$\mathbb{E}[Reg_{CUCB}(T)] = O(\sqrt{T \log T})$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution & regret

Decentralized solution

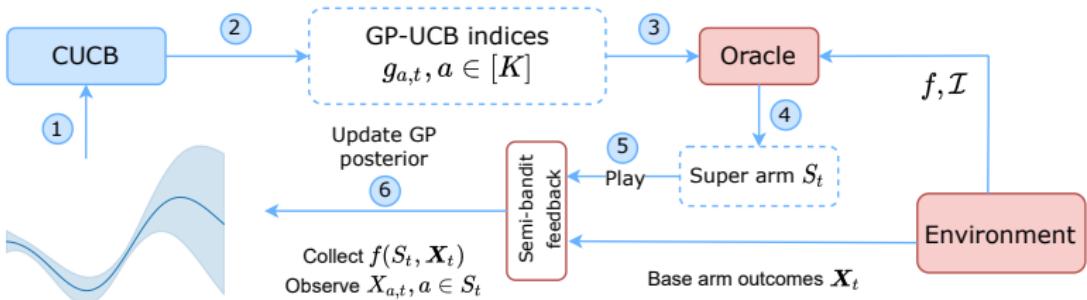
Extensions & Discussion

### Other models

### References

# Modeling base arm outcomes using Gaussian processes (Demirel and Tekin 2021)

- $\mu = (\mu_1, \dots, \mu_K)$  is a sample from a GP with known kernel
- Combinatorial GP-UCB
  - ▶ At each  $t$  compute posterior mean  $\mu_t$  and variance  $\sigma_t^2$  of  $\mu$
  - ▶ Compute GP-UCB indices  $g_{a,t} = \mu_{a,t} + \sqrt{\beta_t} \sigma_{a,t}$  for  $a \in [K]$
  - ▶  $S_t = \text{Oracle}(g_{1,t}, \dots, g_{K,t})$



Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

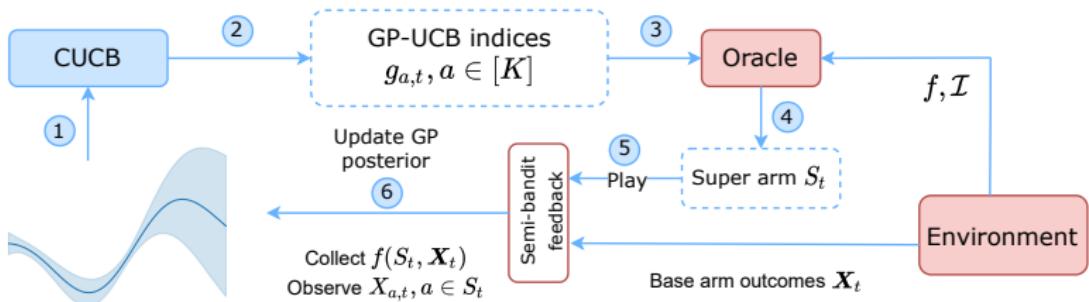
Extensions & Discussion

Other models

References

# Modeling base arm outcomes using Gaussian processes (Demirel and Tekin 2021)

- $\mu = (\mu_1, \dots, \mu_K)$  is a sample from a GP with known kernel
- Combinatorial GP-UCB
  - ▶ At each  $t$  compute posterior mean  $\mu_t$  and variance  $\sigma_t^2$  of  $\mu$
  - ▶ Compute GP-UCB indices  $g_{a,t} = \mu_{a,t} + \sqrt{\beta_t} \sigma_{a,t}$  for  $a \in [K]$
  - ▶  $S_t = \text{Oracle}(g_{1,t}, \dots, g_{K,t})$



- Under GP model expected base arm outcomes are correlated
- Regret bounds depend on maximum information that can be gained about  $\mu$  after  $T$  rounds.

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution & regret

Decentralized solution

Extensions & Discussion

## Other models

## References

# Extension to approximation oracles

- Combinatorial UCB can work with  $(\alpha, \beta)$ -approximation oracles
  - ▶  $\Pr[r(\text{Oracle}(\mu), \mu) \geq \alpha \max_{S \in \mathcal{I}} r(S, \mu)] \geq \beta$

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References

- Combinatorial UCB can work with  $(\alpha, \beta)$ -approximation oracles
  - ▶  $\Pr[r(\text{Oracle}(\mu), \mu) \geq \alpha \max_{S \in \mathcal{I}} r(S, \mu)] \geq \beta$
- Then, regret is measured with respect to  $\alpha\beta$  fraction of the optimal
  - ▶  $\mathbb{E}[\text{Reg}_{\pi}^{\alpha, \beta}(T)] := \alpha\beta T \times \max_{S \in \mathcal{I}} r(S, \mu) - \mathbb{E} \left[ \sum_{t=1}^T f(S_t, \mathbf{X}_t) \right]$

# Combinatorial Thompson sampling (Wang and Chen 2018, Huyuk and Tekin 2019)

## Combinatorial TS (Bernoulli base arms)

- 1: **Store & update:** success  $\alpha_a$  & failure  $\beta_a$  counts for each arm
- 2: **At each round  $t$ :**

- Sample  $\theta_{a,t} \sim \text{Beta}(1 + \alpha_a, 1 + \beta_a)$  for all  $a \in [K]$
- Query optimization oracle with posterior samples

$$S_t = \text{Oracle}(\theta_{1,t}, \dots, \theta_{K,t})$$

- Collect reward  $f(S_t, X_t)$  and observe outcomes  $X_{a,t}$ ,  $a \in S_t$
- Update parameters based on observed outcomes

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Combinatorial Thompson sampling (Wang and Chen 2018, Huyuk and Tekin 2019)

## Combinatorial TS (Bernoulli base arms)

- 1: **Store & update:** success  $\alpha_a$  & failure  $\beta_a$  counts for each arm
- 2: **At each round  $t$ :**

- Sample  $\theta_{a,t} \sim \text{Beta}(1 + \alpha_a, 1 + \beta_a)$  for all  $a \in [K]$
- Query optimization oracle with posterior samples

$$S_t = \text{Oracle}(\theta_{1,t}, \dots, \theta_{K,t})$$

- Collect reward  $f(S_t, X_t)$  and observe outcomes  $X_{a,t}$ ,  $a \in S_t$
- Update parameters based on observed outcomes

- When base arm outcomes are independent, achieves  $O(K \log T/\Delta) + g(K, \Delta)$  gap-dependent regret (Wang and Chen 2018).  
 $\Delta$  = gap between best and second best super arm
- Requires exact computation oracle
- “Usually” works well in practice but  $g(K, \Delta)$  can grow exponentially with  $K$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

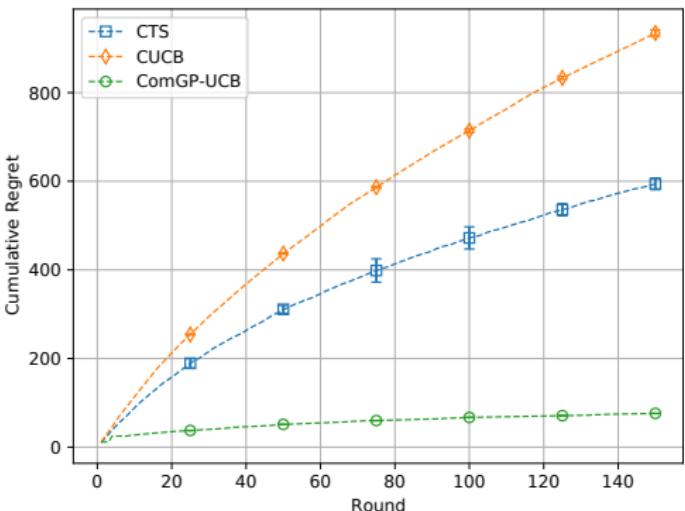
Extensions & Discussion

### Other models

### References

# Empirical comparison

- Linear rewards
- 150 base arms
- Each super arm consists of 10 base arms
- Expected base arm outcomes are sampled from a GP
- CTS samples from independent Gaussian posteriors



## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References

## Summary of Part 2

Multi-armed Bandits  
for Wireless  
Communications

- Introduced combinatorial semi-bandits
  - ▶ Centralized multi-user communication
  - ▶ Path selection

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Summary of Part 2

- Introduced combinatorial semi-bandits
  - ▶ Centralized multi-user communication
  - ▶ Path selection
- Studied CUCB
  - ▶  $O(KL(1/\Delta) \log T)$  gap-dependent regret
  - ▶  $O(\sqrt{KLT} \log T)$  gap-free regret

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Summary of Part 2

- Introduced combinatorial semi-bandits
  - ▶ Centralized multi-user communication
  - ▶ Path selection
- Studied CUCB
  - ▶  $O(KL(1/\Delta) \log T)$  gap-dependent regret
  - ▶  $O(\sqrt{KLT} \log T)$  gap-free regret
- Studied extensions
  - ▶ Non-linear super arm rewards
  - ▶ GP sample base arm outcomes
  - ▶ Approximation oracles
  - ▶ Combinatorial TS

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution  
Extensions & Discussion

### Other models

### References

## Intro to MAB

- Formulation & Examples
- Regret
- Policies for MAB
- Comparison & Discussion

## Combinatorial MAB

- Formulation & Examples
- Regret
- Combinatorial UCB
- Extensions & Discussion

## Decentralized multi-user MAB

- Formulation
- Centralized solution & regret
- Decentralized solution
- Extensions & Discussion

## Other models

## References

3

## Decentralized multi-user MAB

- Formulation
- Centralized solution & regret
- Decentralized solution
- Extensions & Discussion

# Decentralized multi-user MAB

## Formulation

- $L$  players (users),  $K$  arms (channels),  $L \leq K$

Multi-armed Bandits  
for Wireless  
Communications

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Decentralized multi-user MAB

Multi-armed Bandits  
for Wireless  
Communications

## Formulation

- $L$  players (users),  $K$  arms (channels),  $L \leq K$
- Users are synchronized

[Intro to MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Policies for MAB](#)

[Comparison & Discussion](#)

[Combinatorial MAB](#)

[Formulation & Examples](#)

[Regret](#)

[Combinatorial UCB](#)

[Extensions & Discussion](#)

[Decentralized  
multi-user MAB](#)

[Formulation](#)

[Centralized solution &  
regret](#)

[Decentralized solution](#)

[Extensions & Discussion](#)

[Other models](#)

[References](#)

# Decentralized multi-user MAB

## Formulation

- $L$  players (users),  $K$  arms (channels),  $L \leq K$
- Users are synchronized
- In each round  $t = 1, 2, \dots$ 
  - ▶ User  $n$  selects arm  $A_{n,t}$ 
    - More than one user on the same channel  $\Rightarrow$  collision
    - Collision feedback:  $\eta_{n,t} = 1$  if *no collision*, 0 if *collision*

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

#### Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

#### Other models

#### References

# Decentralized multi-user MAB

## Formulation

- $L$  players (users),  $K$  arms (channels),  $L \leq K$
- Users are synchronized
- In each round  $t = 1, 2, \dots$ 
  - ▶ User  $n$  selects arm  $A_{n,t}$ 
    - More than one user on the same channel  $\Rightarrow$  collision
    - Collision feedback:  $\eta_{n,t} = 1$  if *no collision*, 0 if *collision*
  - ▶ User  $n$  observes  $\eta_{n,t}$  on selected arm
    - If *collision*, user gets zero reward
    - If *no collision*, user also observes Bernoulli reward  $X_{n,t} \sim \text{Ber}(\mu_{n,A_{n,t}})$  on selected arm (e.g., packet success/fail due to other channel conditions)

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

#### Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

#### Other models

#### References

# Decentralized multi-user MAB

## Formulation

- $L$  players (users),  $K$  arms (channels),  $L \leq K$
- Users are synchronized
- In each round  $t = 1, 2, \dots$ 
  - ▶ User  $n$  selects arm  $A_{n,t}$ 
    - More than one user on the same channel  $\Rightarrow$  collision
    - Collision feedback:  $\eta_{n,t} = 1$  if *no collision*, 0 if *collision*
  - ▶ User  $n$  observes  $\eta_{n,t}$  on selected arm
    - If *collision*, user gets zero reward
    - If *no collision*, user also observes Bernoulli reward  $X_{n,t} \sim \text{Ber}(\mu_{n,A_{n,t}})$  on selected arm (e.g., packet success/fail due to other channel conditions)
- Parameters  $\{\mu_{n,a}\}_{n \in [L], a \in [K]}$  are positive but unknown

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

#### Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

#### Other models

#### References

# Decentralized multi-user MAB

## Formulation

- $L$  players (users),  $K$  arms (channels),  $L \leq K$
- Users are synchronized
- In each round  $t = 1, 2, \dots$ 
  - ▶ User  $n$  selects arm  $A_{n,t}$ 
    - More than one user on the same channel  $\Rightarrow$  collision
    - Collision feedback:  $\eta_{n,t} = 1$  if *no collision*, 0 if *collision*
  - ▶ User  $n$  observes  $\eta_{n,t}$  on selected arm
    - If *collision*, user gets zero reward
    - If *no collision*, user also observes Bernoulli reward  $X_{n,t} \sim \text{Ber}(\mu_{n,A_{n,t}})$  on selected arm (e.g., packet success/fail due to other channel conditions)
- Parameters  $\{\mu_{n,a}\}_{n \in [L], a \in [K]}$  are positive but unknown
- Users do not observe others' actions and rewards

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

#### Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

#### Other models

#### References

# Decentralized multi-user MAB

## Formulation

- $L$  players (users),  $K$  arms (channels),  $L \leq K$
- Users are synchronized
- In each round  $t = 1, 2, \dots$ 
  - ▶ User  $n$  selects arm  $A_{n,t}$ 
    - More than one user on the same channel  $\Rightarrow$  collision
    - Collision feedback:  $\eta_{n,t} = 1$  if *no collision*, 0 if *collision*
  - ▶ User  $n$  observes  $\eta_{n,t}$  on selected arm
    - If *collision*, user gets zero reward
    - If *no collision*, user also observes Bernoulli reward  $X_{n,t} \sim \text{Ber}(\mu_{n,A_{n,t}})$  on selected arm (e.g., packet success/fail due to other channel conditions)
- Parameters  $\{\mu_{n,a}\}_{n \in [L], a \in [K]}$  are positive but unknown
- Users do not observe others' actions and rewards

## Goal

Maximize expected sum of total rewards

$$\mathbb{E} \left[ \sum_{t=1}^T \sum_{n=1}^L \eta_{n,t} X_{n,t} \right]$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

#### Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

#### Other models

#### References

# Centralized solution & regret

## Centralized version

- Users must select (sit on) different channels.
- Combinatorial bandit with linear rewards
  - ▶ Base arms: user-channel pairs  $(n, a_n)$
  - ▶ Super arms  $S$ : user-channel matchings  $\{(n, a_n)\}_{n=1}^L$
  - ▶ Feasible super arm set  $\mathcal{I}$ : Orthogonal matchings
  - ▶ Optimization oracle: Hungarian algorithm
  - ▶ Unique optimal super arm:  $S^* = \arg \max_{S \in \mathcal{I}} \sum_{(n,a) \in S} \mu_{n,a}$

Multi-armed Bandits  
for Wireless  
Communications

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Centralized solution & regret

## Centralized version

- Users must select (sit on) different channels.
- Combinatorial bandit with linear rewards
  - ▶ Base arms: user-channel pairs  $(n, a_n)$
  - ▶ Super arms  $S$ : user-channel matchings  $\{(n, a_n)\}_{n=1}^L$
  - ▶ Feasible super arm set  $\mathcal{I}$ : Orthogonal matchings
  - ▶ Optimization oracle: Hungarian algorithm
  - ▶ Unique optimal super arm:  $S^* = \arg \max_{S \in \mathcal{I}} \sum_{(n,a) \in S} \mu_{n,a}$

## Regret

$$\mathbb{E}[\text{Reg}_\pi(T)] = T \sum_{(n,a) \in S^*} \mu_{n,a} - \mathbb{E} \left[ \sum_{t=1}^T \sum_{n=1}^L \eta_{n,t} X_{n,t} \right]$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Centralized solution & regret

## Centralized version

- Users must select (sit on) different channels.
- Combinatorial bandit with linear rewards
  - ▶ Base arms: user-channel pairs  $(n, a_n)$
  - ▶ Super arms  $S$ : user-channel matchings  $\{(n, a_n)\}_{n=1}^L$
  - ▶ Feasible super arm set  $\mathcal{I}$ : Orthogonal matchings
  - ▶ Optimization oracle: Hungarian algorithm
  - ▶ Unique optimal super arm:  $S^* = \arg \max_{S \in \mathcal{I}} \sum_{(n,a) \in S} \mu_{n,a}$

## Regret

$$\mathbb{E}[\text{Reg}_\pi(T)] = T \sum_{(n,a) \in S^*} \mu_{n,a} - \mathbb{E} \left[ \sum_{t=1}^T \sum_{n=1}^L \eta_{n,t} X_{n,t} \right]$$

## Theorem (Recall CUCB regret bound)

$$\mathbb{E}[\text{Reg}_{CUCB}(T)] = O(KL(1/\Delta) \log T) \text{ gap-dependent}$$

$$\mathbb{E}[\text{Reg}_{CUCB}(T)] = O(\sqrt{KLT \log T}) \text{ gap-free}$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution & regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Decentralized solution - key ideas

- Assume that time horizon  $T$  is known by all users
- Can we minimize  $\mathbb{E}[\text{Reg}_\pi(T)]$  in the decentralized setting without explicit communication?

Algorithm by Bistritz and Leshem, 2018

- ▶ Phase 1: Each user tries to learn its own expected rewards (exploration)
- ▶ Phase 2: Users try to converge to  $S^*$  distributedly using a utility-based dynamics (agreement)
- ▶ Phase 3: Each user selects the agreed arm (exploitation)

	$T_e$ rounds	$T_g$ rounds	$T - T_e - T_g$ rounds
User 1	explore	agree	exploit
User 2	explore	agree	exploit
⋮	⋮	⋮	⋮
User L	explore	agree	exploit

### Intro to MAB

Formulation & Examples  
Regret  
Policies for MAB  
Comparison & Discussion

### Combinatorial MAB

Formulation & Examples  
Regret  
Combinatorial UCB  
Extensions & Discussion

### Decentralized multi-user MAB

Formulation  
Centralized solution & regret

### Decentralized solution

Extensions & Discussion

### Other models

### References

## Phase 1: Exploration

Each user  $n$  independently runs the following protocol

---

### Exploration

---

- 1: **Store & update:** Cumulative rewards  $V_{n,a}$  & play counts  $N_{n,a}$  for each  $a \in [K]$
  - 2: **for** each  $t \leq T_e$ : **do**
  - 3:     Select  $A_{n,t} \sim \text{Unif}([K])$
  - 4:     Observe no-collision indicator  $\eta_{n,t}$
  - 5:     **if**  $\eta_{n,t} = 1$  **then**
  - 6:         Collect reward  $X_{n,t}$
  - 7:          $V_{n,A_{n,t}} = V_{n,A_{n,t}} + X_{n,t}$ ,  $N_{n,A_{n,t}} = N_{n,A_{n,t}} + 1$
  - 8:     **end if**
  - 9: **end for**
  - 10: **return**  $\hat{\mu}_{n,a} = V_{n,a}/N_{n,a}$  for each arm  $a \in [K]$
- 

#### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

#### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

#### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

#### Other models

#### References

## Phase 1: Exploration

Each user  $n$  independently runs the following protocol

### Exploration

- 1: **Store & update:** Cumulative rewards  $V_{n,a}$  & play counts  $N_{n,a}$  for each  $a \in [K]$
- 2: **for** each  $t \leq T_e$ : **do**
- 3:     Select  $A_{n,t} \sim \text{Unif}([K])$
- 4:     Observe no-collision indicator  $\eta_{n,t}$
- 5:     **if**  $\eta_{n,t} = 1$  **then**
- 6:         Collect reward  $X_{n,t}$
- 7:          $V_{n,A_{n,t}} = V_{n,A_{n,t}} + X_{n,t}$ ,  $N_{n,A_{n,t}} = N_{n,A_{n,t}} + 1$
- 8:     **end if**
- 9: **end for**
- 10: **return**  $\hat{\mu}_{n,a} = V_{n,a}/N_{n,a}$  for each arm  $a \in [K]$

### Lemma (Exploration error probability)

Let  $S_{\text{exp}}^* = \arg \max_{S \in \mathcal{I}} \sum_{(n,a) \in S} \hat{\mu}_{n,a}$  be the estimated optimal allocation. Then, for some positive constants  $c_1$  and  $c_2$ .

$$\Pr(S_{\text{exp}}^* \neq S^*) \leq c_1 \exp(-c_2 T_e)$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Phase 2: Agreement

- Users play an “arm (throne) selection” game using utility-based dynamics aka **Game of Thrones (GoT)**
- Utilities depend on  $\hat{\mu}_{n,a}$  from exploration phase

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Phase 2: Agreement

- Users play an “arm (throne) selection” game using utility-based dynamics aka **Game of Thrones (GoT)**
- Utilities depend on  $\hat{\mu}_{n,a}$  from exploration phase

---

### GoT

---

- 1: **Input:**  $\{\hat{\mu}_{n,a}\}_{a \in [K]}$  from exploration phase
  - 2: **for** each  $t \in \mathcal{G} := (T_e, T_e + T_g]$ : **do**
  - 3:   Select  $A_{n,t}$  based on GoT dynamics
  - 4:   Observe  $\eta_{n,t}$ , collect reward
  - 5:   Record selections
  - 6: **end for**
  - 7: **return** Throne  $a_n^*$  for user  $n$  calculated based on  $\{A_{n,t}\}_{t \in \mathcal{G}}$
- 

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Phase 2: Agreement

### Dynamics for user $n$

- State model

- ▶ User  $n$  keeps a baseline “prefered” arm  $b_n \in [K]$
- ▶ Status: content ( $s_n = C$ ), discontent ( $s_n = D$ )
- ▶ Present state:  $[b_n, s_n]$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Phase 2: Agreement

### Dynamics for user $n$

- State model

- ▶ User  $n$  keeps a baseline “prefered” arm  $b_n \in [K]$
- ▶ Status: content ( $s_n = C$ ), discontent ( $s_n = D$ )
- ▶ Present state:  $[b_n, s_n]$

- Initialization

- ▶  $s_n = C, b_n \sim \text{Unif}([K])$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Phase 2: Agreement

### Dynamics for user $n$

- State model

- ▶ User  $n$  keeps a baseline “prefered” arm  $b_n \in [K]$
- ▶ Status: content ( $s_n = C$ ), discontent ( $s_n = D$ )
- ▶ Present state:  $[b_n, s_n]$

- Initialization

- ▶  $s_n = C$ ,  $b_n \sim \text{Unif}([K])$

- Status based arm selection rule

- ▶ A content user  $n$  “likely sticks” with  $b_n$

$$\Pr(A_n = b_n) = 1 - \epsilon^c,$$

$$\Pr(A_n = a \neq b_n) = \frac{\epsilon^c}{K-1}$$

- ▶ A discontent user  $n$  explores uniformly

$$\Pr(A_n = a) = \frac{1}{K}$$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Phase 2: Agreement

### Dynamics for user $n$ (cont.)

- Utilities

- ▶ Current utility:  $u_n = \underbrace{\hat{\mu}_{n,A_n}}_{\text{from exploration}} \times \underbrace{\eta_n}_{\text{no coll. ind. on } A_n}$
- ▶ Max utility:  $u_{n,\max} = \max_a \hat{\mu}_{n,a}$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Phase 2: Agreement

### Dynamics for user $n$ (cont.)

- Utilities

► Current utility:  $u_n = \underbrace{\hat{\mu}_{n,A_n}}_{\text{from exploration}} \times \underbrace{\eta_n}_{\text{no coll. ind. on } A_n}$

► Max utility:  $u_{n,\max} = \max_a \hat{\mu}_{n,a}$

- State update

►  $s_n = C$  and  $A_n = b_n$  and  $u_n > 0$ :

$$[b_n, C] \rightarrow [b_n, C]$$

►  $s_n = C$  and ( $A_n \neq b_n$  or  $u_n = 0$ ):

$$[b_n, C] \rightarrow \begin{cases} [A_n, C] & \text{w.p. } \frac{u_n}{u_{n,\max}} e^{u_{n,\max} - u_n} \\ [A_n, D] & \text{w.p. } 1 - \frac{u_n}{u_{n,\max}} e^{u_{n,\max} - u_n} \end{cases}$$

►  $s_n = D$ :

$$[b_n, D] \rightarrow \text{same as above}$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Phase 2: Agreement

- Dynamics of users induce a Markov chain on

$$\text{joint state space } \prod_{n \in [L]} ([K] \times \{C, D\})$$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Phase 2: Agreement

- Dynamics of users induce a Markov chain on

$$\text{joint state space } \prod_{n \in [L]} ([K] \times \{C, D\})$$

- Under mild assumptions &  $T_e$  long enough
  - The chain is “ergodic” with stationary distribution  $\pi$
  - (Optimal state in GoT)  $S_{\text{exp}}^* = S^*$  (true optimal state)
  - For sufficiently small  $\epsilon$ ,  $\pi_{S^*} > \frac{1}{2}$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Phase 2: Agreement

- Dynamics of users induce a Markov chain on

$$\text{joint state space } \prod_{n \in [L]} ([K] \times \{C, D\})$$

- Under mild assumptions &  $T_e$  long enough
  - The chain is “ergodic” with stationary distribution  $\pi$
  - (Optimal state in GoT)  $S_{\text{exp}}^* = S^*$  (true optimal state)
  - For sufficiently small  $\epsilon$ ,  $\pi_{S^*} > \frac{1}{2}$

### Succession

- Time spent being content on arm  $a$ :

$$F_{n,a} = \sum_{t \in \mathcal{G}} \mathbb{I}(A_{n,t} = a, s_{n,t} = C)$$

- Throne of user  $n$ :

$$a_n^* = \arg \max_a F_{n,a}$$

### Intro to MAB

Formulation &amp; Examples

Regret

Policies for MAB

Comparison &amp; Discussion

### Combinatorial MAB

Formulation &amp; Examples

Regret

Combinatorial UCB

Extensions &amp; Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &amp; regret

Decentralized solution

Extensions &amp; Discussion

### Other models

### References

## Phase 2: Agreement

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

### Lemma (GoT error probability)

Let  $S_{GoT}^* = \{(n, a_n^*)\}_{n \in [L]}$  be the super arm returned after GoT dynamics. Then,

$$\Pr(S_{GoT}^* \neq S_{exp}^*) \leq c_3 \exp\left(-c_4 \frac{T_g}{T_m}\right)$$

where  $T_g$  is GoT phase length and  $T_m$  is mixing time of the induced Markov chain

## Phase 3: Exploitation

### Exploitation

- 
- 1: **Input:**  $a_n^*$  from GoT phase
  - 2: **for** each  $t \in (T_e + T_g, T]$ : **do**
  - 3:     Select  $a_n^*$
  - 4:     Collect reward  $\eta_{n,t} X_{n,t}$
  - 5: **end for**
- 

Multi-armed Bandits  
for Wireless  
Communications

#### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

#### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

#### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

#### Other models

#### References

## Phase 3: Exploitation

### Exploitation

- 
- 1: **Input:**  $a_n^*$  from GoT phase
  - 2: **for** each  $t \in (T_e + T_g, T]$ : **do**
  - 3:     Select  $a_n^*$
  - 4:     Collect reward  $\eta_{n,t} X_{n,t}$
  - 5: **end for**
- 

### Theorem (Decentralized regret)

For all users set  $T_e = O(\log T)$  and  $T_g = O(\log T)$ . Then,

$$\mathbb{E}[Reg_{GoT}(T)] = O(\log T)$$

#### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

#### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

#### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

#### Other models

#### References

## Phase 3: Exploitation

### Exploitation

- 1: **Input:**  $a_n^*$  from GoT phase
- 2: **for** each  $t \in (T_e + T_g, T]$ : **do**
- 3:     Select  $a_n^*$
- 4:     Collect reward  $\eta_{n,t} X_{n,t}$
- 5: **end for**

### Theorem (Decentralized regret)

For all users set  $T_e = O(\log T)$  and  $T_g = O(\log T)$ . Then,

$$\mathbb{E}[Reg_{GoT}(T)] = O(\log T)$$

### Proof.

- Exploration and GoT error probabilities vanish when  $T_e$  and  $T_g$  are large enough.
- $T_e & T_g = O(\log T) \Rightarrow \Pr(S_{GoT}^* = S_{exp}^* = S^*) \geq 1 - 1/T$ .
  - ▶  $S_{GoT}^* = S^* \Rightarrow$  regret only from Exploration & GoT phases
  - ▶  $S_{GoT}^* \neq S^* \Rightarrow$  regret  $O(T)$  but only w.p. at most  $1/T$

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

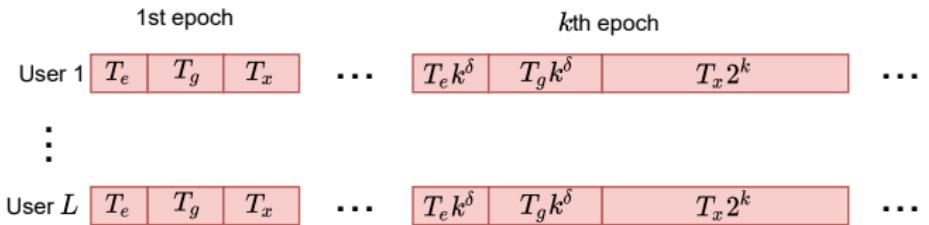
### Other models

### References



# Extensions

- $T$  not known beforehand  $\Rightarrow$  epoch-based structure (Bistritz and Leshem, 2018)



## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

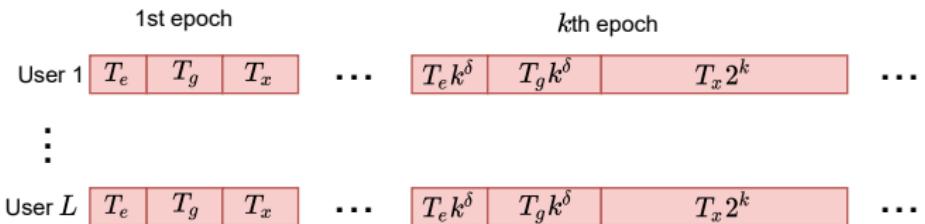
Extensions & Discussion

## Other models

## References

# Extensions

- $T$  not known beforehand  $\Rightarrow$  epoch-based structure (Bistritz and Leshem, 2018)



- Continuous & Markovian rewards, fairness (Bistritz and Leshem, 2021, Bistrizt et al. 2021)

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

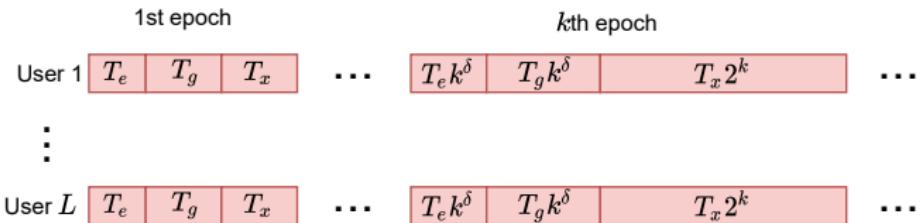
Decentralized solution

Extensions & Discussion

## Other models

## References

- $T$  not known beforehand  $\Rightarrow$  epoch-based structure (Bistritz and Leshem, 2018)



- Continuous & Markovian rewards, fairness (Bistritz and Leshem, 2021, Bistritz et al. 2021)
- Lots of collisions due to uniform exploration  $\Rightarrow$  Orthogonal exploration (Hanawal and Darak, 2018)

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Dynamic rate and channel adaptation (Javanmardi et al. 2021)

- Each user selects a channel and transmission rate

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Extensions - Dynamic rate and channel adaptation

Multi-armed Bandits  
for Wireless  
Communications

## Dynamic rate and channel adaptation (Javanmardi et al. 2021)

- Each user selects a channel and transmission rate
- Expected rewards depend on both channel and rate

### Intro to MAB

Formulation & Examples  
Regret  
Policies for MAB  
Comparison & Discussion

### Combinatorial MAB

Formulation & Examples  
Regret  
Combinatorial UCB  
Extensions & Discussion

### Decentralized multi-user MAB

Formulation  
Centralized solution &  
regret  
Decentralized solution  
Extensions & Discussion

### Other models

### References

### Dynamic rate and channel adaptation (Javanmardi et al. 2021)

- Each user selects a channel and transmission rate
- Expected rewards depend on both channel and rate
- Exploration phase: Sequential halving orthogonal exploration (SHOE)
  - ▶ Sequential halving to identify optimal rate for each channel
  - ▶ Orthogonal exploration to reduce collisions
- Agreement phase: GoT over (channel, channel's estimated best rate) pairs
- Exploitation phase: select (throne, throne's estimated best rate)

#### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

#### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

#### Decentralized multi-user MAB

Formulation

Centralized solution & regret

Decentralized solution

Extensions & Discussion

#### Other models

#### References

# Extensions - No collision sensing for homogeneous case

- Heterogeneous case:  $\mu_{n,c}$  different for each  $n$ .  
Homogeneous case  $\mu_{n,c}$  same for all  $n$

Multi-armed Bandits  
for Wireless  
Communications

[Intro to MAB](#)

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

[Combinatorial MAB](#)

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

[Decentralized  
multi-user MAB](#)

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

[Other models](#)

[References](#)

## Extensions - No collision sensing for homogeneous case

- Heterogeneous case:  $\mu_{n,c}$  different for each  $n$ .  
Homogeneous case  $\mu_{n,c}$  same for all  $n$
- When collision feedback is available:
  - ▶ Implicit communication for homogeneous case (Boursier and Perchet, 2019)
  - ▶ Regret gap to centralized MAB is largely closed

Multi-armed Bandits  
for Wireless  
Communications

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Extensions - No collision sensing for homogeneous case

Multi-armed Bandits  
for Wireless  
Communications

- Heterogeneous case:  $\mu_{n,c}$  different for each  $n$ .  
Homogeneous case  $\mu_{n,c}$  same for all  $n$
- When collision feedback is available:
  - ▶ Implicit communication for homogeneous case (Boursier and Perchet, 2019)
  - ▶ Regret gap to centralized MAB is largely closed
- No-sensing: collision information is unavailable at the players
  - ▶ Zero reward can come indistinguishably from either collision or natural random reward generation
  - ▶ E.g., communication failure can come from either other users transmitting on the same channel or instantaneous channel noise is very large
  - ▶ Regret gap is significant

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

## Extensions - No collision sensing for homogeneous case

- This gap is closed in (Shi et al. 2020). An algorithm based on
  - ▶ Implicit communication + error-correction coding for Z-channel
  - ▶ Result: Decentralized multi-user MAB without collision information can (asymptotically) approach the performance of centralized multi-user MAB
- Information theory and coding theory found a surprising application in MAB

Multi-armed Bandits  
for Wireless  
Communications

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Summary of Part 3

Multi-armed Bandits  
for Wireless  
Communications

- Studied decentralized multi-user MAB
  - ▶ Heterogeneous rewards
  - ▶ Collision feedback
  - ▶ Synchronized users
  - ▶ Distributed algorithm: explore - GoT - exploit
  - ▶  $O(\log T)$  regret

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Contextual bandits

Multi-armed Bandits  
for Wireless  
Communications

In each round  $t$

- 1 Observe context (state)  $x_t$
- 2 Select arm  $A_t \in \{1, \dots, K\}$
- 3 Collect reward  $R_t \sim F_{A_t, x_t}$

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Contextual bandits

Multi-armed Bandits  
for Wireless  
Communications

In each round  $t$

- 1 Observe context (state)  $x_t$
- 2 Select arm  $A_t \in \{1, \dots, K\}$
- 3 Collect reward  $R_t \sim F_{A_t, x_t}$

Examples

- User position, base station position, partial channel state information, etc.

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Contextual bandits

In each round  $t$

- 1 Observe context (state)  $x_t$
- 2 Select arm  $A_t \in \{1, \dots, K\}$
- 3 Collect reward  $R_t \sim F_{A_t, x_t}$

Examples

- User position, base station position, partial channel state information, etc.

Challenge

- How to maximize  $\mathbb{E}[\sum_{t=1}^T R_t]$ ?

Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

Decentralized  
multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

Other models

References

# Contextual bandits

In each round  $t$

- 1 Observe context (state)  $x_t$
- 2 Select arm  $A_t \in \{1, \dots, K\}$
- 3 Collect reward  $R_t \sim F_{A_t, x_t}$

Examples

- User position, base station position, partial channel state information, etc.

Challenge

- How to maximize  $\mathbb{E}[\sum_{t=1}^T R_t]$ ?
- Expected rewards depend on the context.  $\mu_{a,x}$  unknown

[Intro to MAB](#)

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

[Combinatorial MAB](#)

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

[Decentralized  
multi-user MAB](#)

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

[Other models](#)

[References](#)

# Contextual bandits

In each round  $t$

- 1 Observe context (state)  $x_t$
- 2 Select arm  $A_t \in \{1, \dots, K\}$
- 3 Collect reward  $R_t \sim F_{A_t, x_t}$

## Examples

- User position, base station position, partial channel state information, etc.

## Challenge

- How to maximize  $\mathbb{E}[\sum_{t=1}^T R_t]$ ?
- Expected rewards depend on the context.  $\mu_{a,x}$  unknown
- Context set can be very large. Need to explore-exploit efficiently for each context
  - ▶ Need additional structure to learn fast
  - ▶ Contextual linear bandit:  $\mu_{a,x} = \langle \theta_*, \psi(a, x) \rangle$
  - ▶ Lipschitz contextual bandit:  $|\mu_{a,x} - \mu_{a,x'}| \leq B||x - x'||$

### Intro to MAB

- Formulation & Examples
- Regret
- Policies for MAB
- Comparison & Discussion

### Combinatorial MAB

- Formulation & Examples
- Regret
- Combinatorial UCB
- Extensions & Discussion

### Decentralized multi-user MAB

- Formulation
- Centralized solution & regret
- Decentralized solution
- Extensions & Discussion

### Other models

### References

## Contextual bandits in high dimensions [Tekin and van der Schaar 2015a, Turgay et al. 2020]

- Context set can be high dimensional
- Reward for each arm only depends on small subset of (unknown) relevant dimensions (sparsity)
- Traditional contextual bandit algorithms suffer from curse of dimensionality
- Utilizing sparsity in learning result in regret that scales only with relevant dimensions

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

## Contextual bandits in high dimensions [Tekin and van der Schaar

2015a, Turgay et al. 2020]

- Context set can be high dimensional
- Reward for each arm only depends on small subset of (unknown) relevant dimensions (sparsity)
- Traditional contextual bandit algorithms suffer from curse of dimensionality
- Utilizing sparsity in learning result in regret that scales only with relevant dimensions

## Cooperation in contextual bandits [Tekin and van der Schaar 2015b]

- Multi-user setting
- Users have different sets of arms
- A user can select its own arm, or pay other users to select an arm for itself
- Users can accumulate higher rewards with cooperation

### Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

### Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

### Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

### Other models

### References

# Some references

## K-armed bandits

Thompson, William R. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples." *Biometrika* 25.3/4 (1933): 285-294.

Lai, Tze Leung, and Herbert Robbins. "Asymptotically efficient adaptive allocation rules." *Advances in Applied Mathematics* 6.1 (1985): 4-22.

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem." *Machine Learning* 47.2 (2002): 235-256.

Audibert, Jean-Yves, and Sébastien Bubeck. "Minimax policies for adversarial and stochastic bandits." *COLT*. Vol. 7. 2009.

Garivier, Aurelien, and Olivier Cappe. "The KL-UCB algorithm for bounded stochastic bandits and beyond." *Proceedings of the 24th Annual Conference on Learning Theory. JMLR Workshop and Conference Proceedings*, 2011.

Agrawal, Shipra, and Navin Goyal. "Analysis of Thompson sampling for the multi-armed bandit problem." *Conference on learning theory. JMLR Workshop and Conference Proceedings*, 2012.

Kaufmann, Emilie, Nathaniel Korda, and Remi Munos. "Thompson sampling: An asymptotically optimal finite-time analysis." *International Conference on Algorithmic Learning Theory*. 2012.

Kaufmann, Emilie, Olivier Cappe, and Aurelien Garivier. "On Bayesian upper confidence bounds for bandit problems." *Artificial intelligence and statistics*. 2012.

Tekin, Cem, and Mingyan Liu. "Online learning of rested and restless bandits." *IEEE Transactions on Information Theory* 58.8 (2012): 5588-5611.

Agrawal, Shipra, and Navin Goyal. "Further optimal regret bounds for Thompson sampling." *Artificial intelligence and statistics*. 2013.

Russo, Daniel J., et al. "A tutorial on Thompson sampling." *Foundations and Trends in Machine Learning* 11.1 (2018): 1-96.

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References

# Some references

## Combinatorial bandits

Gai, Yi, Bhaskar Krishnamachari, and Rahul Jain. "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations." IEEE/ACM Transactions on Networking 20.5 (2012): 1466-1478.

Cesa-Bianchi, Nicolo, and Gabor Lugosi. "Combinatorial bandits." Journal of Computer and System Sciences 78.5 (2012): 1404-1422.

Chen, Wei, Yajun Wang, and Yang Yuan. "Combinatorial multi-armed bandit: General framework and applications." International Conference on Machine Learning. 2013.

Kveton, Branislav, et al. "Tight regret bounds for stochastic combinatorial semi-bandits." The 18th International Conference on Artificial Intelligence and Statistics. 2015.

Combes, Richard, et al. "Combinatorial bandits revisited." The 29th Conference on Neural Information Processing Systems. 2015.

Chen, Wei, et al. "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms." The Journal of Machine Learning Research 17.1 (2016): 1746-1778.

Wang, Siwei, and Wei Chen. "Thompson sampling for combinatorial semi-bandits." International Conference on Machine Learning. 2018.

Huyuk, Alihan, and Cem Tekin. "Analysis of Thompson sampling for combinatorial multi-armed bandit with probabilistically triggered arms." The 22nd International Conference on Artificial Intelligence and Statistics. 2019.

Huyuk, Alihan, and Cem Tekin. "Thompson sampling for combinatorial network optimization in unknown environments." IEEE/ACM Transactions on Networking 28.6 (2020): 2836-2849.

Perrault, Pierre, et al. "Statistical Efficiency of Thompson Sampling for Combinatorial Semi-Bandits." Advances in Neural Information Processing Systems. 2020.

Demirel, Ilker, and Cem Tekin. "Combinatorial Gaussian process bandits with probabilistically triggered arms." International Conference on Artificial Intelligence and Statistics. 2021.

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References

# Some references

## Decentralized multi-user bandits

Liu, Keqin, and Qing Zhao. "Distributed learning in multi-armed bandit with multiple players." *IEEE Transactions on Signal Processing* 58.11 (2010): 5667-5681.

Kalathil, Dileep, Naumaan Nayyar, and Rahul Jain. "Decentralized learning for multiplayer multiarmed bandits." *IEEE Transactions on Information Theory* 60.4 (2014): 2331-2345.

Tekin, Cem, and Mingyan Liu. "Online learning in decentralized multi-user spectrum access with synchronized explorations." *IEEE Military Communications Conference*. IEEE, 2012.

Rosenski, Jonathan, Ohad Shamir, and Liran Szlak. "Multi-player bandits - a musical chairs approach." *International Conference on Machine Learning*. PMLR, 2016.

Bistritz, Ilai, and Amir Leshem. "Distributed multi-player bandits-a game of thrones approach." *Advances in Neural Information Processing Systems (NeurIPS)* (2018).

Boursier, Etienne, and Vianney Perchet. "SIC-MMAB: synchronisation involves communication in multiplayer multi-armed bandits." *The 33rd Conference on Neural Information Processing Systems*. 2019.

C. Shi, W. Xiong, C. Shen, and J. Yang, "Decentralized multi-player multi-armed bandits with no collision information," *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, Aug. 2020

Bistritz, Ilai, and Amir Leshem. "Game of thrones: Fully distributed learning for multiplayer bandits." *Mathematics of Operations Research* 46.1 (2021): 159-178.

Bistritz, Ilai, et al. "One for all and all for one: Distributed learning of fair allocations with multi-player bandits." *IEEE Journal on Selected Areas in Information Theory* (2021).

Javammardi, Alireza, Muhammad Anjum Qureshi, and Cem Tekin. "Decentralized Dynamic Rate and Channel Selection over a Shared Spectrum." *IEEE Transactions on Communications* (2021).

Hanawal, Manjesh Kumar, and Sumit Darak. "Multi-player bandits: A trekking approach." *IEEE Transactions on Automatic Control* (2021).

C. Shi and C. Shen, "On no-sensing adversarial multi-player multi-armed bandits with collision communications," *IEEE Journal on Selected Areas in Information Theory, Special Issue on Sequential, Active, and Reinforcement Learning* (2021).

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution & regret

Decentralized solution

Extensions & Discussion

## Other models

## References

# Some references

## Contextual bandits

Langford, John, and Tong Zhang. "The epoch-greedy algorithm for contextual multi-armed bandits." *Advances in Neural Information Processing Systems* 20.1 (2007): 96-1.

Chu, Wei, et al. "Contextual bandits with linear payoff functions." *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011.

Slivkins, Aleksandrs. "Contextual bandits with similarity information." *Proceedings of the 24th annual Conference On Learning Theory*. 2011.

Krause, Andreas, and Cheng Soon Ong. "Contextual Gaussian Process Bandit Optimization." *NIPS*. 2011.

May, Benedict C., et al. "Optimistic Bayesian sampling in contextual-bandit problems." *Journal of Machine Learning Research* 13 (2012): 2069-2106.

Tekin, Cem, and Mihaela van der Schaar. "RELEAF: An algorithm for learning and exploiting relevance." *IEEE Journal of Selected Topics in Signal Processing* 9.4 (2015): 716-727.

Tekin, Cem, and Mihaela van der Schaar. "Distributed online learning via cooperative contextual bandits." *IEEE transactions on signal processing* 63.14 (2015): 3700-3714.

Turgay, Eralp, Doruk Oner, and Cem Tekin. "Multi-objective contextual bandit problem with similarity information." *International Conference on Artificial Intelligence and Statistics*.

Turgay, Eralp, Cem Bulucu, and Cem Tekin. "Exploiting Relevance for Online Decision-Making in High-Dimensions." *IEEE Transactions on Signal Processing* (2020).

Qureshi, Muhammad Anjum, and Cem Tekin. "Fast learning for dynamic resource allocation in AI-enabled radio networks." *IEEE Transactions on Cognitive Communications and Networking* 6.1 (2019): 95-110.

## Books/surveys

Lattimore, Tor, and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Slivkins, Aleksandrs. "Introduction to multi-armed bandits." *Foundations and Trends in Machine Learning* 12.1-2 (2019): 1-286.

Bubeck, Sébastien, and Nicolo Cesa-Bianchi. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems." *Foundations and Trends in Machine Learning* 5.1 (2012): 1-122.

## Intro to MAB

Formulation & Examples

Regret

Policies for MAB

Comparison & Discussion

## Combinatorial MAB

Formulation & Examples

Regret

Combinatorial UCB

Extensions & Discussion

## Decentralized multi-user MAB

Formulation

Centralized solution &  
regret

Decentralized solution

Extensions & Discussion

## Other models

## References

**Intro to MAB**

- Formulation & Examples
- Regret
- Policies for MAB
- Comparison & Discussion

**Combinatorial MAB**

- Formulation & Examples
- Regret
- Combinatorial UCB
- Extensions & Discussion

**Decentralized  
multi-user MAB**

- Formulation
- Centralized solution & regret
- Decentralized solution
- Extensions & Discussion

**Other models**

**References**

Thank you

End of Part “Multi-armed Bandits for  
Wireless Communications”

# Wireless communications – Reinforcement Learning Problems (From single-agent learning to multi-agent learning)

## Mihaela van der Schaar

John Humphrey Plummer Professor of Machine Learning, Artificial Intelligence and Medicine, University of Cambridge  
Chancellor's Professor, University of California Los Angeles



van\_der\_Schaar  
\LAB  
[vanderschaar-lab.com](http://vanderschaar-lab.com)



UNIVERSITY OF  
CAMBRIDGE



[mv472@cam.ac.uk](mailto:mv472@cam.ac.uk)



@MihaelaVDS



[linkedin.com/in/  
mihaela-van-der-schaar/](https://linkedin.com/in/mihaela-van-der-schaar/)

# Wireless communications as an ML problem

- Single-agent wireless communication
  - Cross-layer optimization
    - Static settings: multi-objective optimization
    - Dynamic settings:
      - Solution 1: solve the same multi-objective optimization repeatedly -> Use supervised learning -> myopic, sub-optimal solution
      - Solution 2: centralized reinforcement learning -> complex, often sub-optimal solution
      - Solution 3: decentralized reinforcement learning -> optimal solution
  - Multi-agent wireless communication
    - Compliant users
      - Power control as a learning game
      - Slotted MAC protocols – going beyond slotted CSMA/CA – learning without communication
    - Strategic users
      - Resource competition – mechanism design + multi-agent reinforcement learning
  - Beyond wireless communications: social networks, content caching etc.

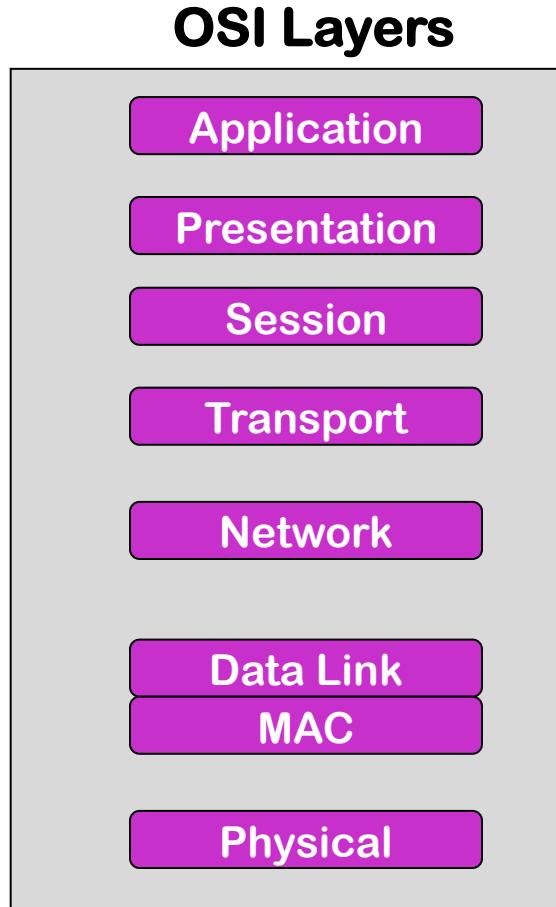


# Wireless communications as an ML problem

- Single-agent wireless communication
  - Cross-layer optimization
    - Static settings: multi-objective optimization
    - Dynamic settings:
    - Solution 1: solve the same multi-objective optimization repeatedly -> Use supervised learning -> myopic, sub-optimal solution



# Wireless communications: An ML problem?



- RF
  - Transmit power
  - Antenna direction
- Baseband
  - Modulation
  - Equalization
- Link/MAC
  - Frame length
  - Error correction coding
  - ARQ
  - Admission Control and Scheduling
  - Packetization
- Transport/Network
  - Signaling
  - TCP/UDP
  - Packetization
- Application
  - Compression strategies
  - Concealment, Post-processing etc.
  - Rate/Format adaptation
  - Channel coding/ARQ
  - Number of streams/flow
  - Scheduling
  - Packetization

**Selection of algorithms  
and hyper-parameters  
is needed!**



# Can protocols alone provide optimal solutions for wireless communications?

- NO!!
- Remember, a protocol = a set of standards defining “message” formats & exchange rules, but not how to select the algorithms, parameters, optimizations of the protocol!

We need cross-layer design and optimization



# Why Cross-Layer Design and Optimization?

Cross-layer design and optimization is essential because:

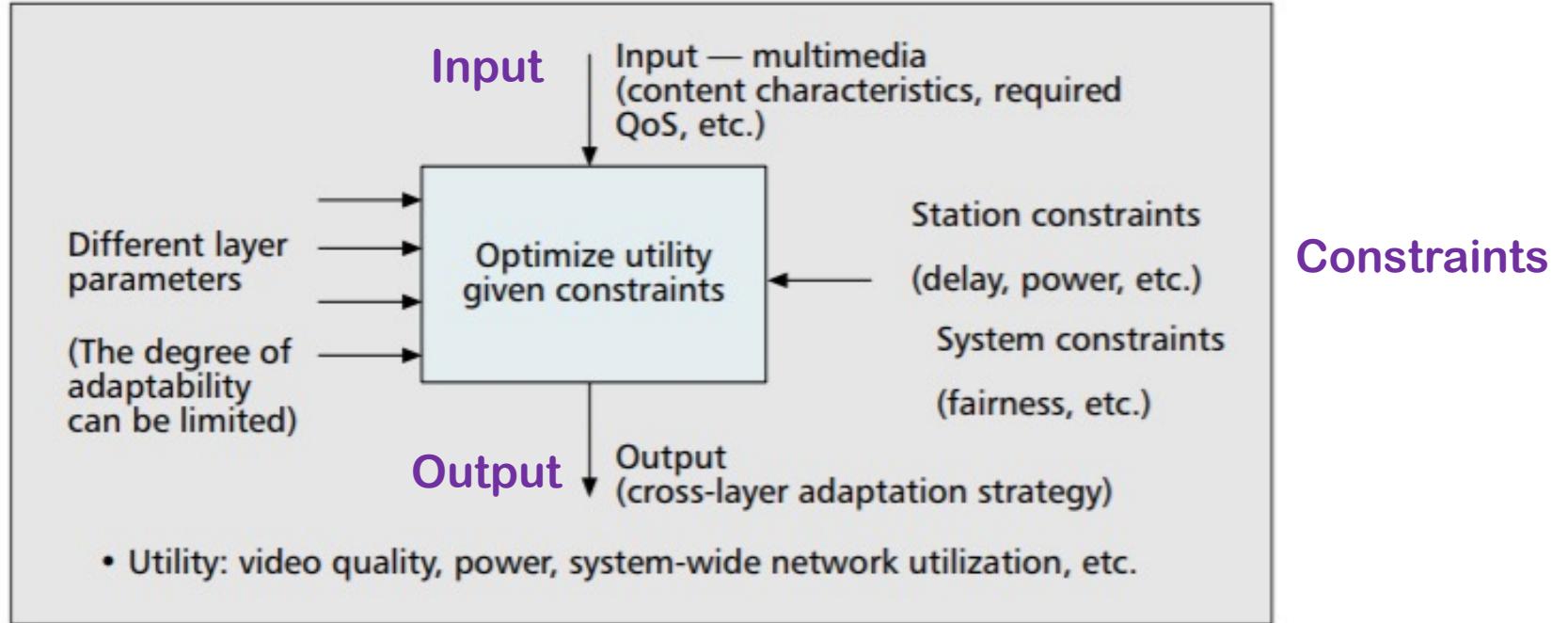
- it leads to improved performance over existing wireless networks;
- It provides guidelines for designing and optimizing the inter-layer message exchanges (middleware);
- it provides valuable insights on how to design the next generation networking algorithms and protocols.



# Wireless transmission – A cross-layer optimization problem

Algorithms/parameters  
to optimize

Utility



■ **Figure 1.** The conceptual framework of cross-layer optimization.

M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, “Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs,” *IEEE J. Sel. Areas Commun.*, Dec. 2003.

M. van der Schaar and S. Shankar, “Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms,” *IEEE Wireless Commun. Mag.*, Aug. 2005.



van\_der\_Schaar  
\ LAB

vanderschaar-lab.com

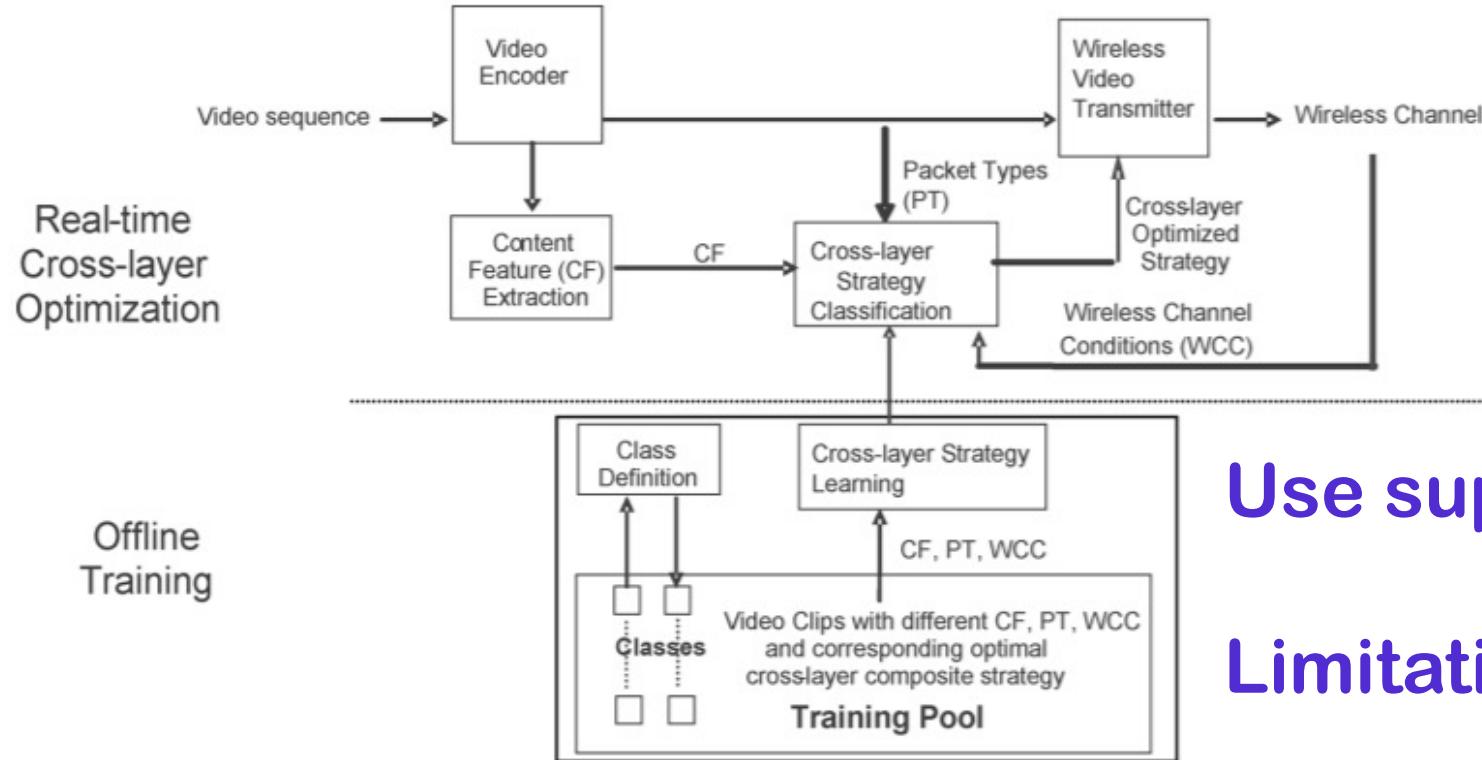
# Cross-layer optimization problem: Challenges

- Time-varying inputs: Application requirements, wireless channels
- Time-varying constraints
- Need to solve the cross-layer problem repeatedly, each time inputs or constraints change!
- Very expensive!
- Solution?

M. van der Schaar and S. Shankar, “Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms,” *IEEE Wireless Commun. Mag.*, Aug. 2005.



# How to adapt to changing demands and resources? A first machine learning approach



Use supervised learning!

Limitation: myopic learning

M. van der Schaar, D. Turaga, and R. Wong, "Classification-Based System For Cross-Layer Optimized Wireless Video Transmission," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 1082-1095, Oct. 2006.



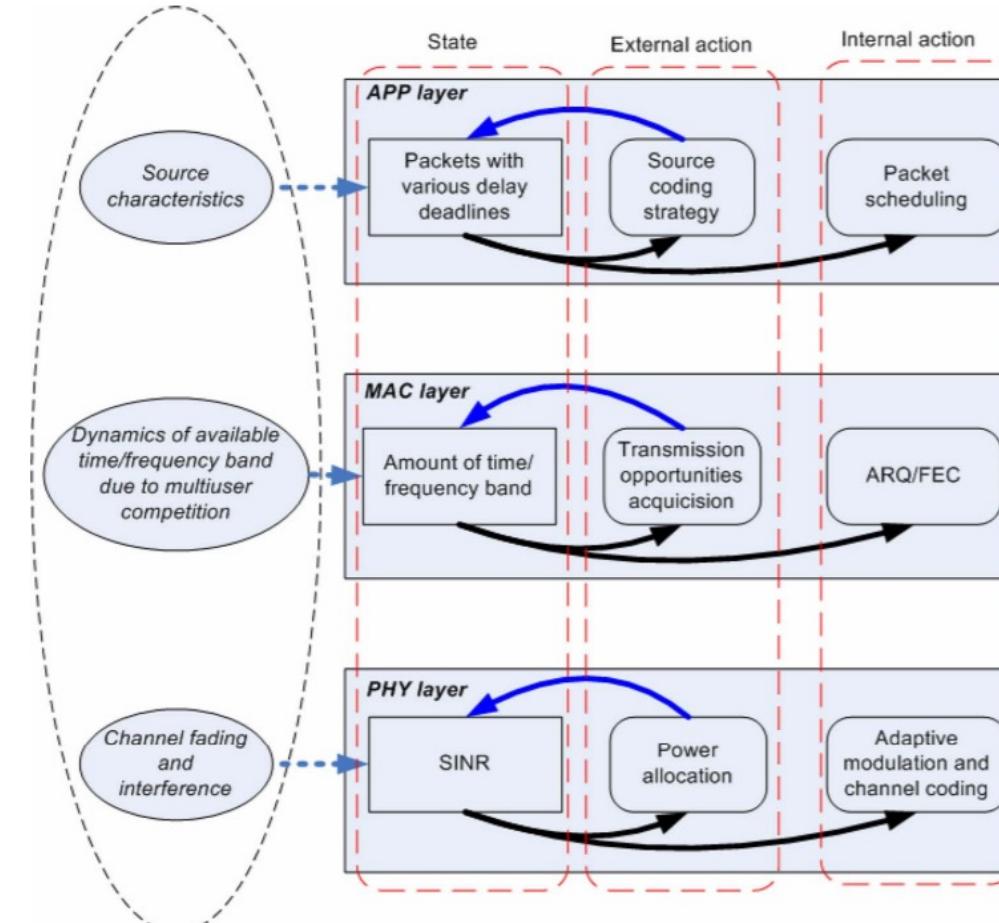
van\_der\_Schaar  
\ LAB

vanderschaar-lab.com

# How to optimally adapt to changing environments? Learning and decision making under uncertainty

## Reinforcement learning

F. Fu and M. van der Schaar, *IEEE Trans. Multimedia*, Jun. 2007.



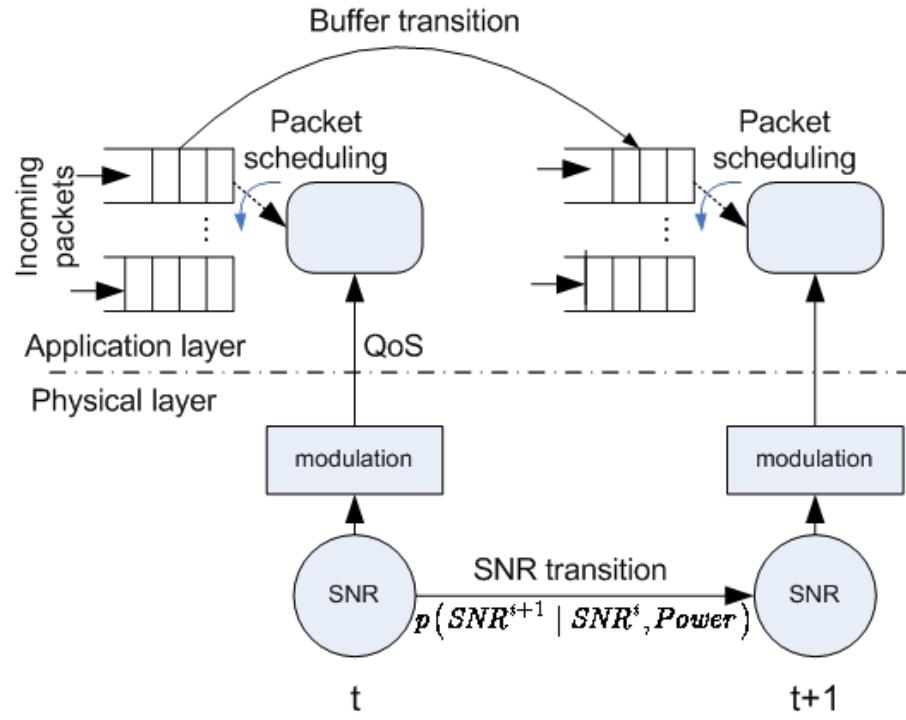
# Reinforcement-learning basics

**Discrete-time Markov Decision Process (MDP):**  $\mathcal{M} = (S, A, c, p, \gamma)$

- $S$  is a finite set of states
  - $s \in S$  is a specific state (possibly a vector)
- $A$  is a finite set of actions
  - $a \in A$  is a specific action (possibly a vector)
- $c: S \times A \rightarrow \mathbb{R}$  is a cost function
- $p: S \times A \times S \rightarrow [0,1]$  is a transition probability function
- $\gamma \in [0,1)$  is a discount factor



# Wireless comms as a reinforcement-learning problem



## Application Layer

- State: **buffer fullness**
- Actions: **packet scheduling, source coding**
- State transition: **buffer transition**
- Reward: **distortion reduction**

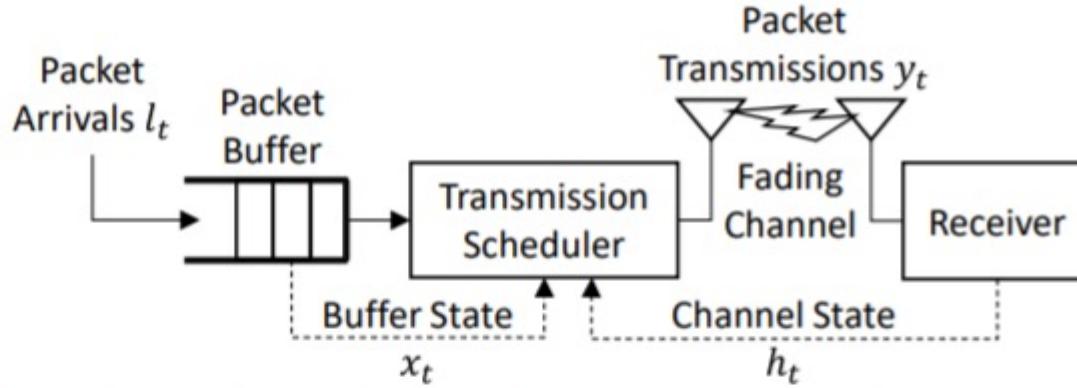
## Physical Layer

- State: **SNR**
- Actions: **modulation, power**
- State transition: **SNR transition**
- Cost: **consumed power**

Current action impacts both immediate as well as future reward



# Wireless comms as a reinforcement-learning problem



Point-to-point time-slotted wireless communication system

- **Time step index:**  $t$
- **Packet buffer state:**  $x_t \in S_x = \{0, 1, \dots, N_x\}$
- **Channel state:**  $h_t \in S_h$ 
  - Finite-state Markov chain  $p_h(h'|h)$
- **System state:**  $s_t = (x_t, h_t) \in S_x \times S_h = S$
- **Scheduling action:**  $y_t \in \{0, 1, \dots, x_t\}$
- **Packet arrivals:**  $l_t \in \{0, 1, \dots\}$ 
  - Arrivals are i.i.d.  $p_l(l)$

Unknown



# Wireless comms as a reinforcement-learning problem

## The **buffer state**

$$x_{t+1} = \min(x_t - y_t + l_t, N_x)$$

- $\{x_t : t = 0, 1, \dots\}$  can be modeled as a controlled Markov chain:

$$p_x(x'|x, y) = \sum_{l=0}^{\infty} p_l(l) I_{\{x' = \min(x-y+l, N_x)\}}$$

  
Unknown



# Wireless comms as a reinforcement-learning problem

We define the **cost** as a weighted sum of two terms

$$\rho(h, y) + \lambda g(x, y)$$

- $\rho(h, y)$  is the **transmission power** to transmit  $y$  packets in channel state  $h$
- $g(x, y)$  is a **buffer cost** to penalize large queue backlogs and overflows

$$g(x, y) = \underbrace{x}_{\text{Holding Cost}} + \underbrace{\eta \sum_{l=0}^{\infty} p_l(l) \max(x - y + l - N_x, 0)}_{\substack{\text{Expected Overflow Cost} \\ \text{Unknown}}}$$

- $\lambda \geq 0$  trades off the transmission costs and buffer costs



# Wireless comms as a reinforcement-learning problem

- **State:**  $s \triangleq (x, h)$  (buffer, channel)
- **Action:**  $a = y$  (number of packet transmissions)
- **Cost:**  $c(s, a) = \underbrace{\rho(h, y)}_{\text{Power cost}} + \lambda \underbrace{g(x, y)}_{\text{Buffer cost}}$
- **Transition probability:**  $p(s'|s, a) = p_x(x'|x, y)p_h(h'|h)$
- **Discount factor:**  $\gamma \in [0, 1)$



# Reinforcement-learning basics: Policy and Value

- **Deterministic policy:**  $\pi: S \rightarrow A$ 
  - maps states to actions
- **Value:**  $V^\pi(s)$ 
  - indicates how good or bad it is to be in state  $s$  under policy  $\pi$ :

$$V^\pi(s) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t c(s_t, \pi(s_t)) | s_0 = s \right], \quad \forall s \in S$$

- We can express  $V^\pi(s)$  recursively using the transition probability function

$$V^\pi(s) = c(s, \pi(s)) + \gamma \sum_{s' \in S} p(s'|s, \pi(s)) V^\pi(s'), \quad \forall s \in S$$



# Reinforcement-learning basics: Objective

- **Objective:** Find a policy that optimizes the value function

$$V^*(s) = \min_{\pi} V^{\pi}(s), \forall s \in S$$

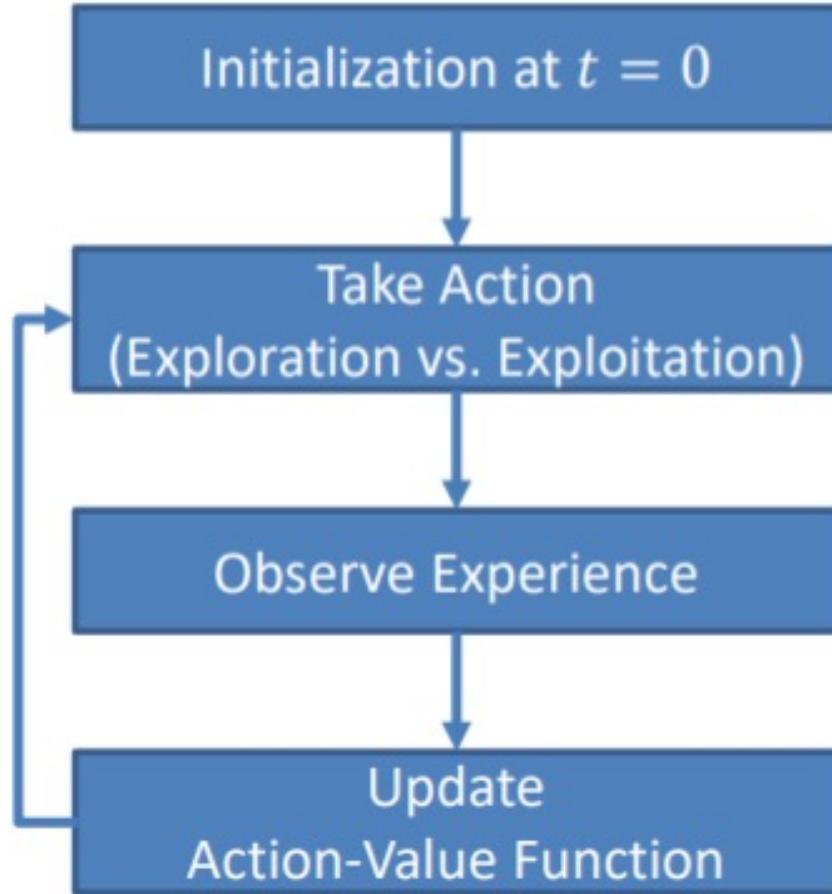
- $V^*(s)$  satisfies the following Bellman equation

$$V^*(s) = \underbrace{\min_{a \in A} \left\{ c(s, a) + \gamma \sum_{s' \in S} p(s'|s, a)V^*(s') \right\}}_{Q^*(s, a)}, \quad \forall s \in S$$

- $V^*$  is the **optimal value function**
- $Q^*$  is the **optimal action-value function**
- $\pi^*$  is the **optimal policy**



# Conventional model-free RL:Q-learning



$$Q_0(s, a), \forall s \in S, \forall a \in A$$

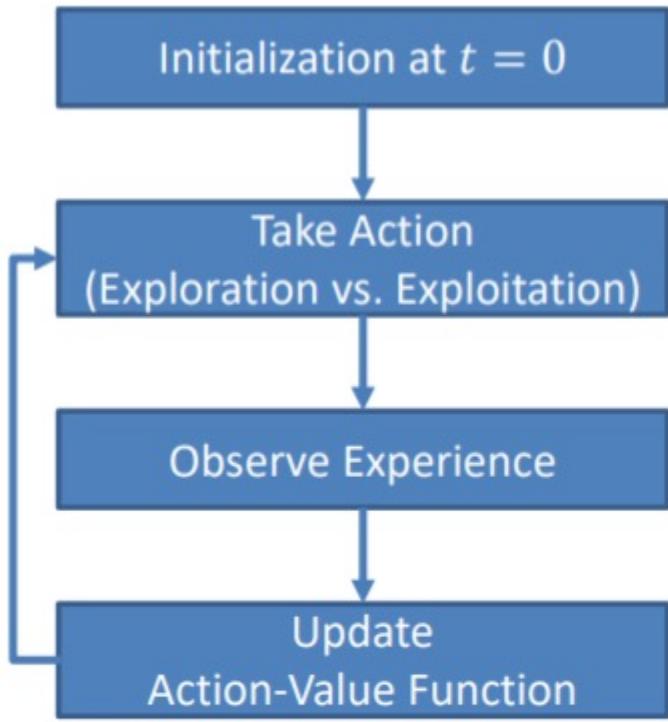
$$a_t = \begin{cases} \arg \min_a Q_t(s_t, a), & \text{with probability } \epsilon \\ \text{rand}(A), & \text{with probability } 1 - \epsilon \end{cases}$$

$$\sigma_t = (s_t, a_t, c_t, s_{t+1}) \begin{cases} E[c_t] = c(s_t, a_t) \\ s_{t+1} \sim p(s' | s_t, a_t) \end{cases}$$

$$Q_{t+1}(s_t, a_t) \leftarrow (1 - \alpha_t)Q_t(s_t, a_t) + \alpha_t [c_t + \gamma \max_{a \in A} Q_t(s_{t+1}, a)]$$



# Conventional model-free RL:Q-learning



$$Q_0(s, a), \forall s \in S, \forall a \in A$$

$$a_t = \begin{cases} \arg \min_a Q_t(s_t, a), & \text{with probability } \epsilon \\ \text{rand}(A), & \text{with probability } 1 - \epsilon \end{cases}$$

3

$$\sigma_t = (s_t, a_t, c_t, s_{t+1}) \begin{cases} E[c_t] = c(s_t, a_t) \\ s_{t+1} \sim p(s'|s_t, a_t) \end{cases}$$

2

$$Q_{t+1}(s_t, a_t) \leftarrow (1 - \alpha_t)Q_t(s_t, a_t) + \alpha_t [c_t + \gamma \max_{a \in A} Q_t(s_{t+1}, a)]$$

1. Updates only one state-action pair in each step
2. Assumes no a priori information
3. Requires exploration

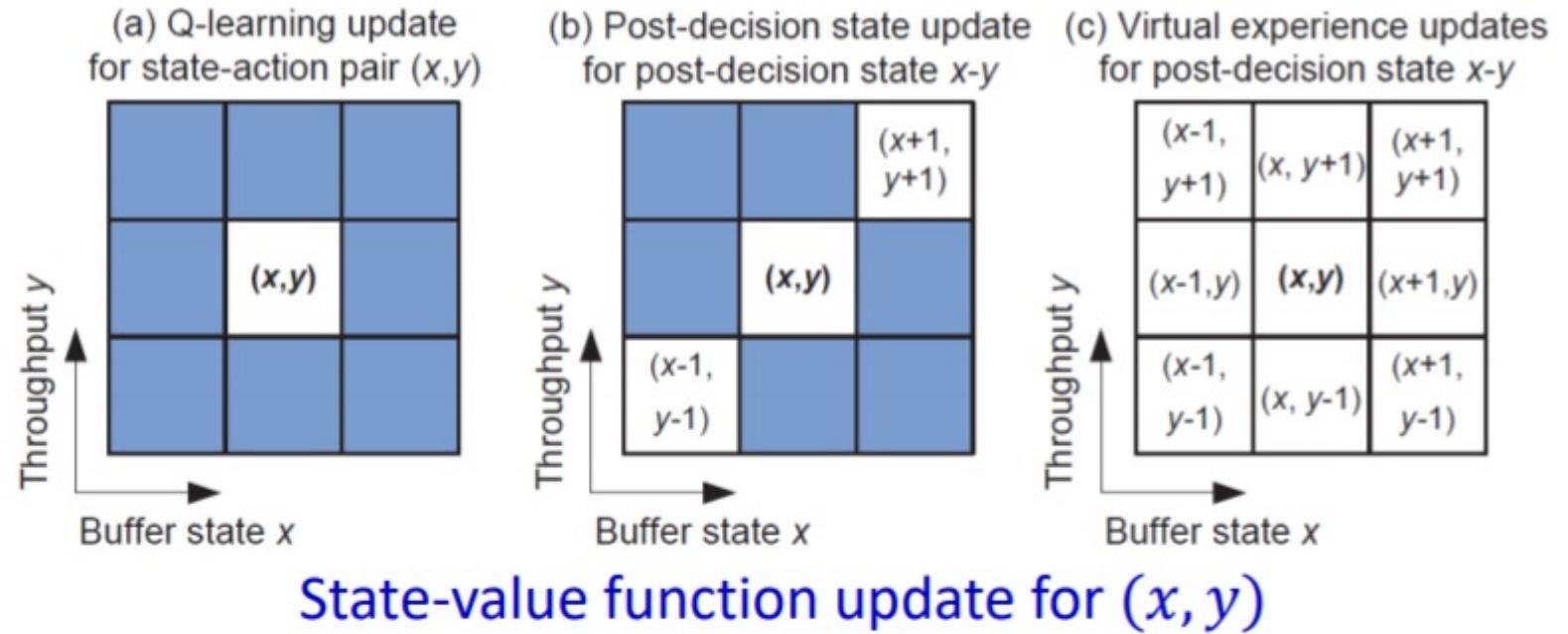


# Reinforcement-learning basics: Model-free vs Model-based

- If  $c(s, a)$  and  $p(s'|s, a)$  are known, then  $V^*$ ,  $Q^*$ , and  $\pi^*$  can be computed using **dynamic programming**
- Otherwise, they can be learned using **reinforcement learning**
  - **Model-free:** Learn a value function or policy w/o learning a model
    - e.g., Q-learning, SARSA, policy-gradient and actor-critic methods
  - **Model-based:** Learn a model for  $c(s, a)$  and  $p(s'|s, a)$ , and then derive a value function or policy
    - e.g., ARTDP, Dyna, Dyna-Q
  - Both approaches are purely *data-driven*



# Improvements in wireless comm RL



Algorithm	Action Selection Complexity	Learning Update Complexity	$A$ : Action set $S$ : State set $E$ : Number of virtual experience updates
Q-learning	$O(A)$	$O(A)$	
PDS learning	$O(SA)$	$O(SA)$	
Virtual experience	$O(SA)$	$O(ESA)$	

With F. Fu  
With N. Mastronarde



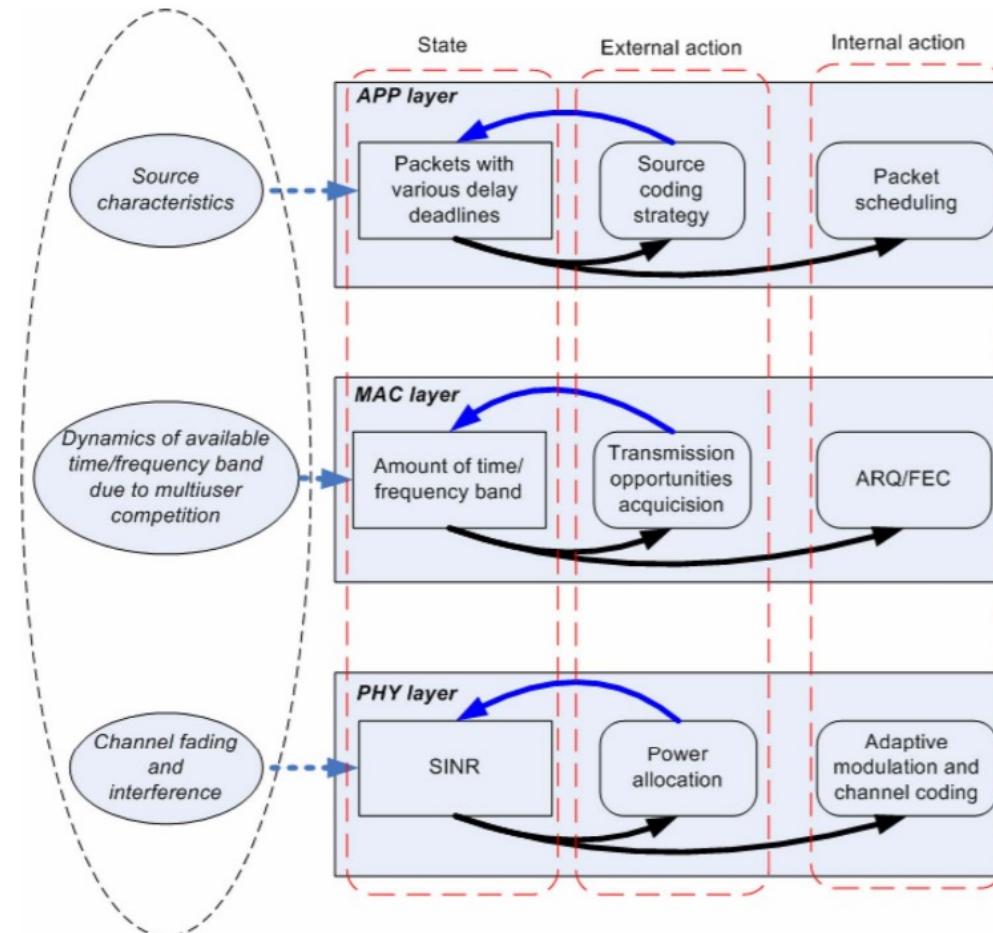
van\_der\_Schaar  
\ LAB

vanderschaar-lab.com

# How to adapt to changing demands and resources? Learning and decision making under uncertainty

## Reinforcement learning

F. Fu and M. van der Schaar, *IEEE Trans. Multimedia*, Jun. 2007.



van\_der\_Schaar  
\ LAB

vanderschaar-lab.com

# Wireless comms as a reinforcement-learning problem

- **Packet losses and retransmissions**
  - Add packet loss distribution  $p(\cdot|h_t, y_t)$
  - Lost packets remain in the buffer for retransmission
- **Dynamic power management (DPM)** With N. Mastronarde
  - Add power management state (ON/OFF) and action (Switch ON/OFF)
  - Add switching cost and delay
- **Energy harvesting** With N. Mastronarde
  - Add battery state and stochastic energy arrivals



# Wireless communications as an ML problem

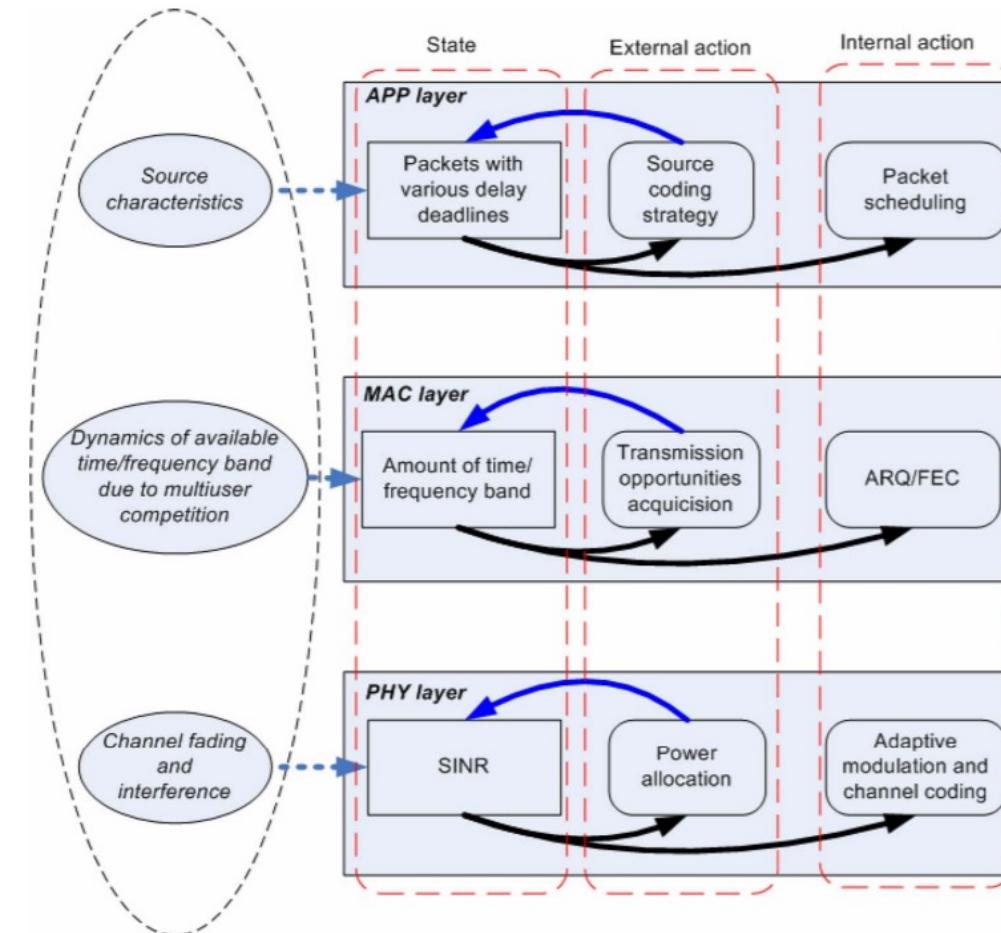
- Single-agent wireless communication
  - Cross-layer optimization
    - Static settings: multi-objective optimization
    - Dynamic settings:
      - Solution 1: solve the same multi-objective optimization repeatedly -> Use supervised learning -> myopic, sub-optimal solution
      - Solution 2: centralized reinforcement learning -> complex, often sub-optimal solution
      - Solution 3: decentralized reinforcement learning -> optimal solution



# How to adapt to changing demands and resources? Different Layers – Different dynamics!

## Reinforcement learning

F. Fu and M. van der Schaar, "A New Systematic Framework for Autonomous Cross-Layer Optimization," *IEEE Trans. Veh. Tech.*, vol. 58, no. 4, pp. 1887-1903, May, 2009.

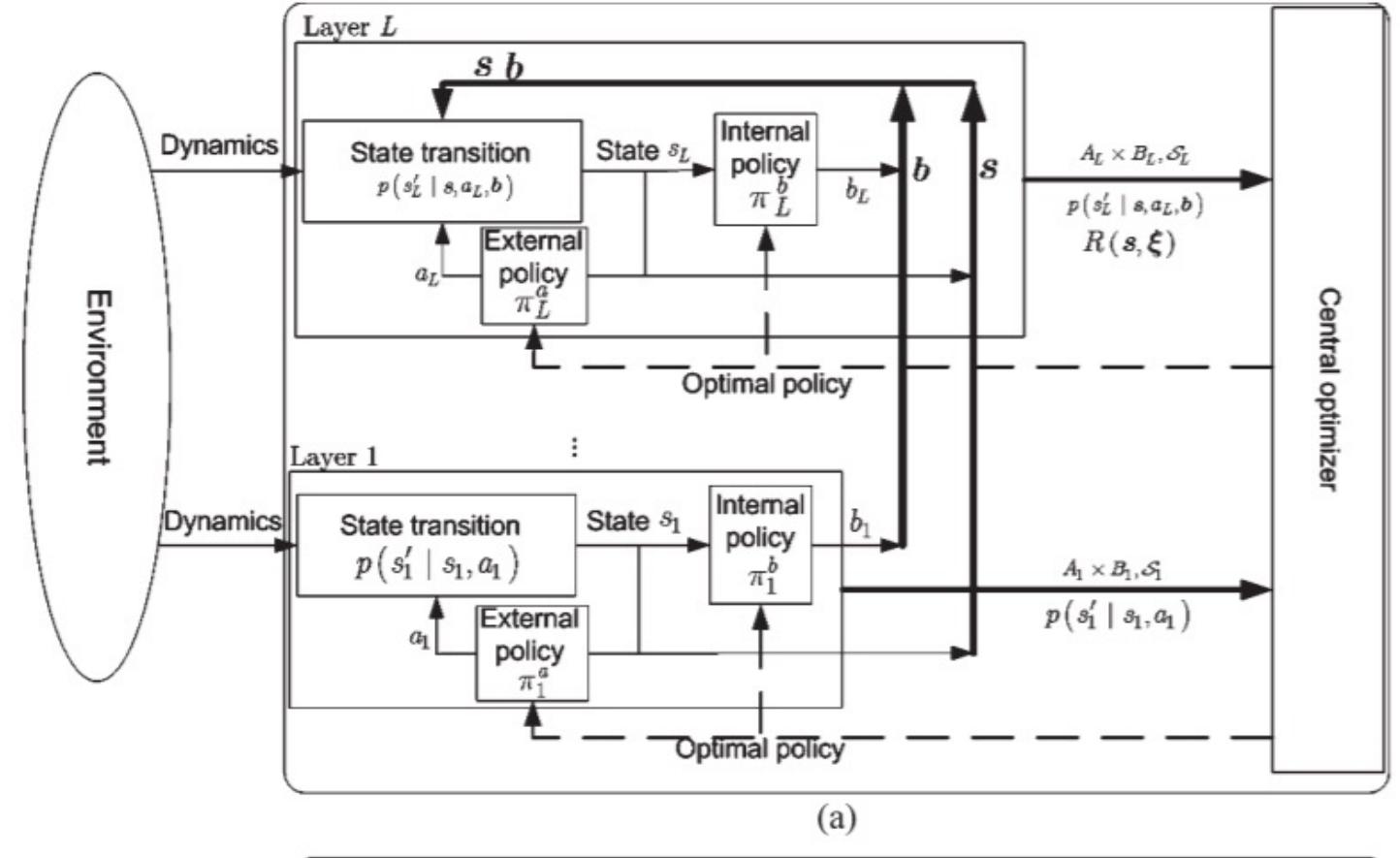


van\_der\_Schaar  
\ LAB

vanderschaar-lab.com

# Reinforcement learning – A centralized approach?

F. Fu and M. van der Schaar, "A New Systematic Framework for Autonomous Cross-Layer Optimization," *IEEE Trans. Veh. Tech.*, vol. 58, no. 4, pp. 1887-1903, May, 2009.



# Challenges for cross-layer design and optimization

- Decision making - coupled among layers
- Environmental dynamics at various layers (different time scales)
- Adaptation granularity (multiple time scales/granularities at different layers)
- How to exchange information among layers? What information should be exchanged?
- Which layer/layers should perform the optimization?
- Protocol compliant? Protocols are determined and controlled by different entities/companies
- Violate OSI stack?



# Other Challenges

- Current cross-layer optimization violates layered architecture
  - Centralized
  - Lead to dependent layer design
  - Reduce network stability and extensibility
- Decision maker requires to know
  - All possible strategy combination
  - Dynamics from different layers
- Objective
  - Myopic performance (maximizing current utility) or (also) impact on the future performance?

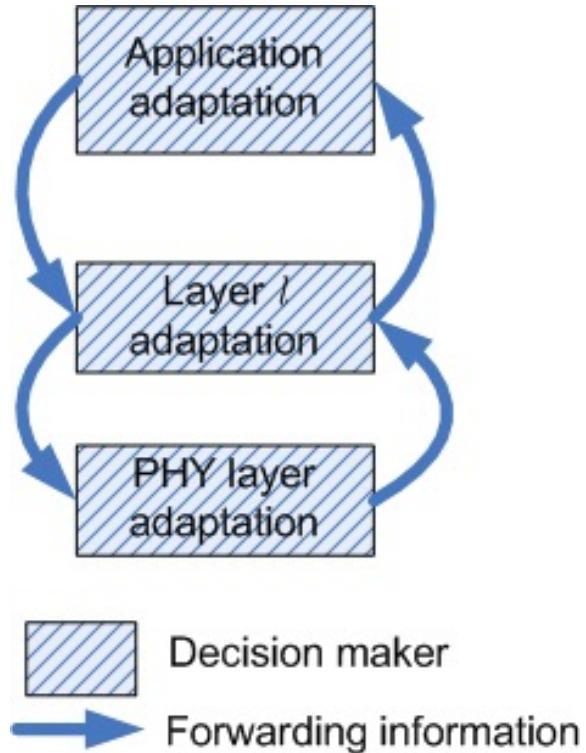
**Why do we care about future performance at the current time?**

Current decisions impact immediate *and* future performance

- **Video Rate & Distortion:** Coding decisions for current video data unit impact bit-budget, rate, and distortion of future data units
- **Power & Delay:** Time to transmit current video packets impacts time available (and power required) to transmit future video packets



# Systematic layered optimization with information exchange



Key questions to be answered:

- 1. How can each layer optimally configure its own parameters?**
- 2. What information should be exchanged between layers and how?**

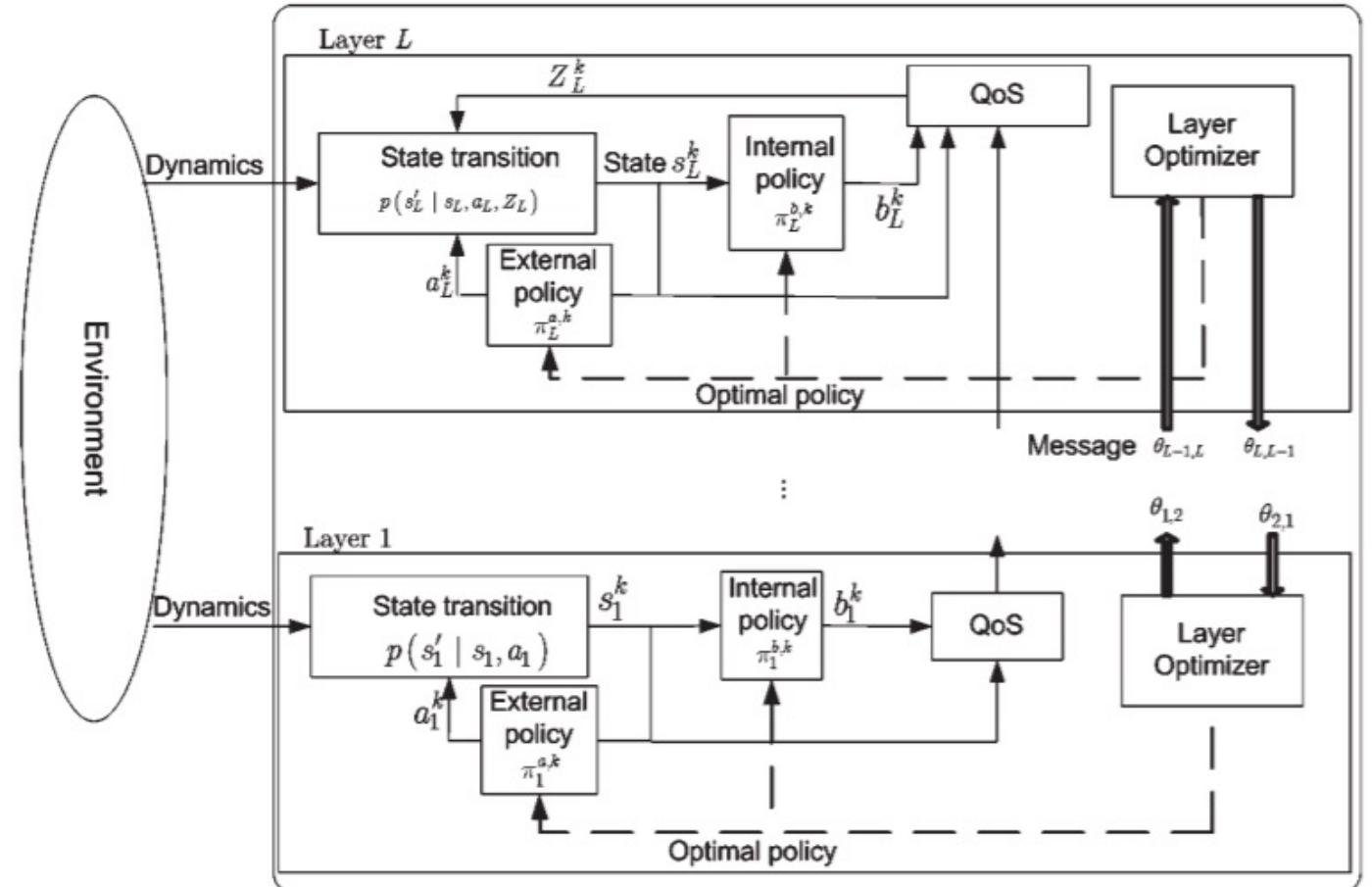


# Reinforcement learning – A distributed reinforcement learning approach

Multiple agents acting to optimize  
common goal facing different dynamics

New methods for reinforcement  
learning – needed!

F. Fu and M. van der Schaar, "A New Systematic Framework for Autonomous Cross-Layer Optimization," *IEEE Trans. Veh. Tech.*, vol. 58, no. 4, pp. 1887-1903, May, 2009.



# Centralized Markov Decision Process (MDP) formulation

- Tuple  $(S, \mathcal{X}, p, R)$ 
  - state space  
 $s = (\text{buffer length}, \text{SNR}) \in S$
  - action space  
 $\xi = (\text{power}, \text{modulation}, \text{packet scheduling}, \text{source coding}) \in \mathcal{X}$
  - transition probability  
 $p(s|s, \xi) = p(s^{k+1} = s | s^k = s, \xi^k = \xi)$
  - $R(s, \xi)$  immediate reward at state  $s$  (e.g. application quality) when performing action  $\xi$
- Goal is to maximize some cumulative function of the rewards

$$\max_{\xi^t \in \mathcal{X}} \left\{ \sum_{t=1}^{\infty} \alpha^t R(s^t, \xi^t) \right\}$$



# Centralized DP operator

- Value iteration algorithm

$$V_n(s) = \max_{\xi \in X} \left\{ R(s, \xi) + \alpha \sum_{s' \in S} p(s' | s, \xi) V_{n-1}(s') \right\}$$

- Policy iteration algorithm

- Policy evaluation

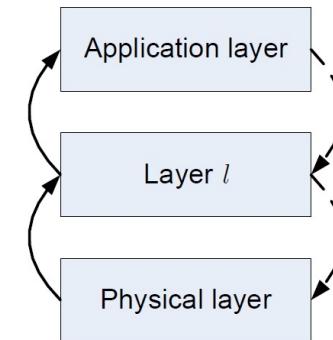
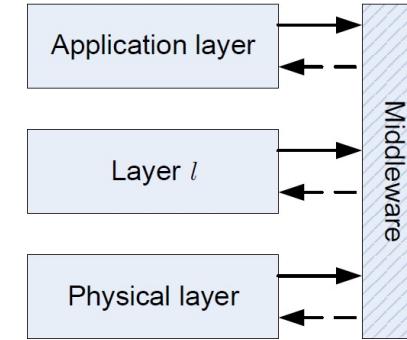
$$V^{\pi_n} = R(s, \pi_n(s)) + \alpha \sum_{s' \in S} p(s' | s, \pi_n(s)) V^{\pi_n}(s')$$

- Policy improvement

$$\pi_{n+1}(s) = \arg \max_{\xi \in \mathcal{X}} \left\{ R(s, \xi) + \alpha \sum_{s' \in S} p(s' | s, \xi) V^{\pi_n}(s') \right\}$$

- Key step: **Dynamic programming (DP) operator**

$$\max_{\xi \in \mathcal{X}} \left\{ R(s, \xi) + \alpha \sum_{s' \in S} p(s' | s, \xi) V(s') \right\}$$



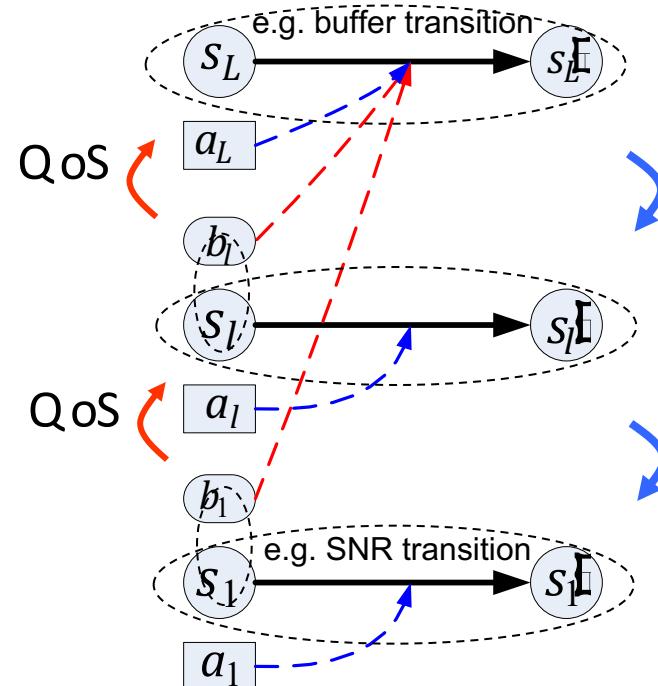
# Decentralized learning solutions: Key insights

- Functionality of each layer is specified in terms of QoS services that received from layer(s) below and that required to provide to layer(s) above.
- The advantage of layered architectures is that the designer or implementer of the protocol or algorithm at a particular layer can focus on the design of that layer, without being required to consider all the parameters and algorithms of the rest of the stack.
- Key insight: decentralization in terms of QoS – provision (feasibility) and requirements (selection)



# Key ideas of layered DP operator

- Define instantaneous QoS exchange to determine internal actions – upward message exchange
- Design layered DP operator to determine external actions – downward message exchange



# Structure of cross-layer optimization

- State:

$$s = (s_1, \dots, s_L)$$

$$\xi = (a, b)$$

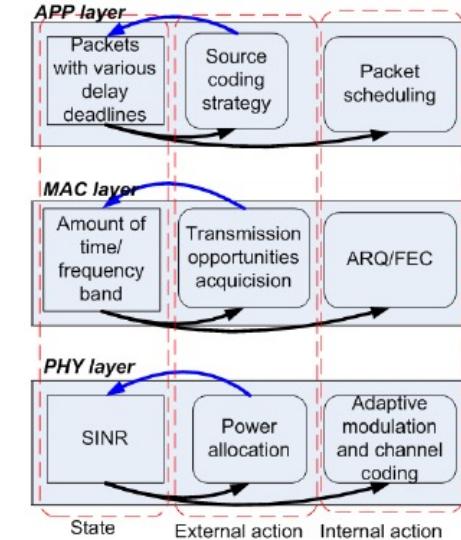
- Action

- External action  $a = (a_1, \dots, a_L)$

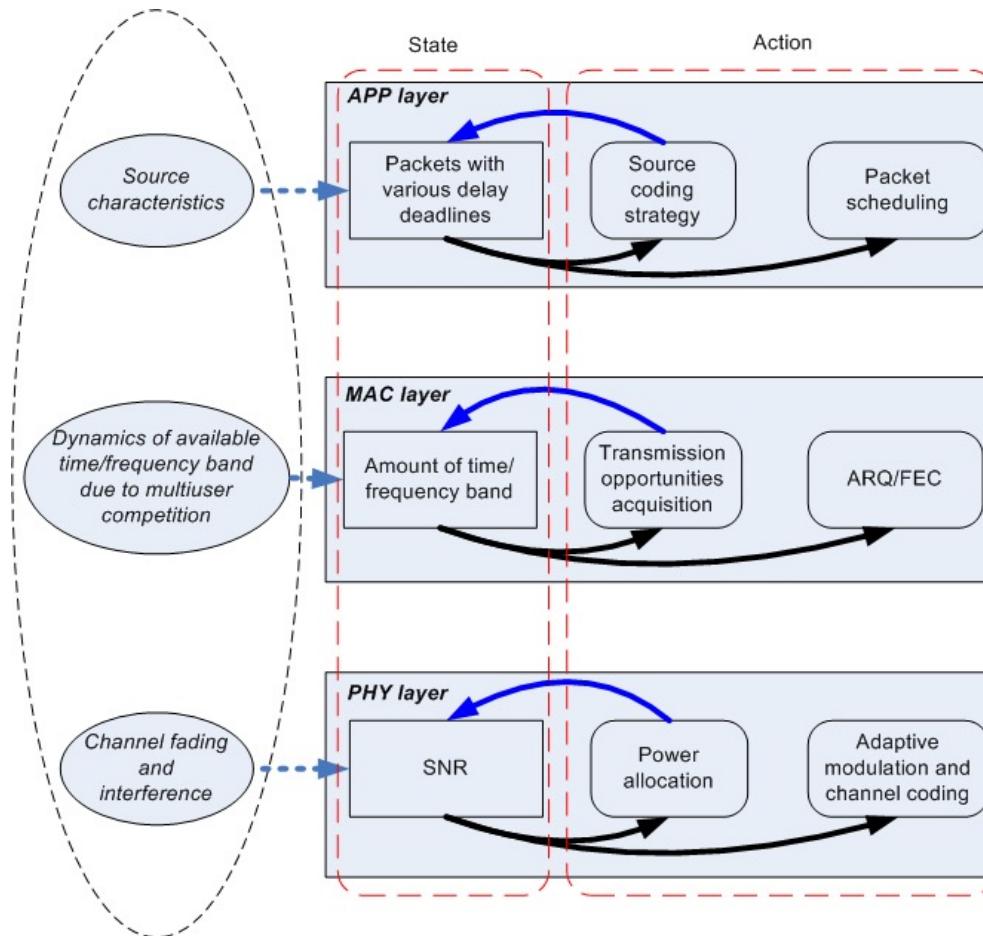
- The actions performed to determine the state transition in order to obtain higher future reward, e.g. power allocation, source coding strategy, etc.

- Internal action  $b = (b_1, \dots, b_L)$

- The actions performed to jointly determine how many packets can be successfully delivered to the destination (i.e. determine current utility, given current state), e.g. Adaptive modulation, ARQ/FEC, etc.



# Example for cross-layer optimization



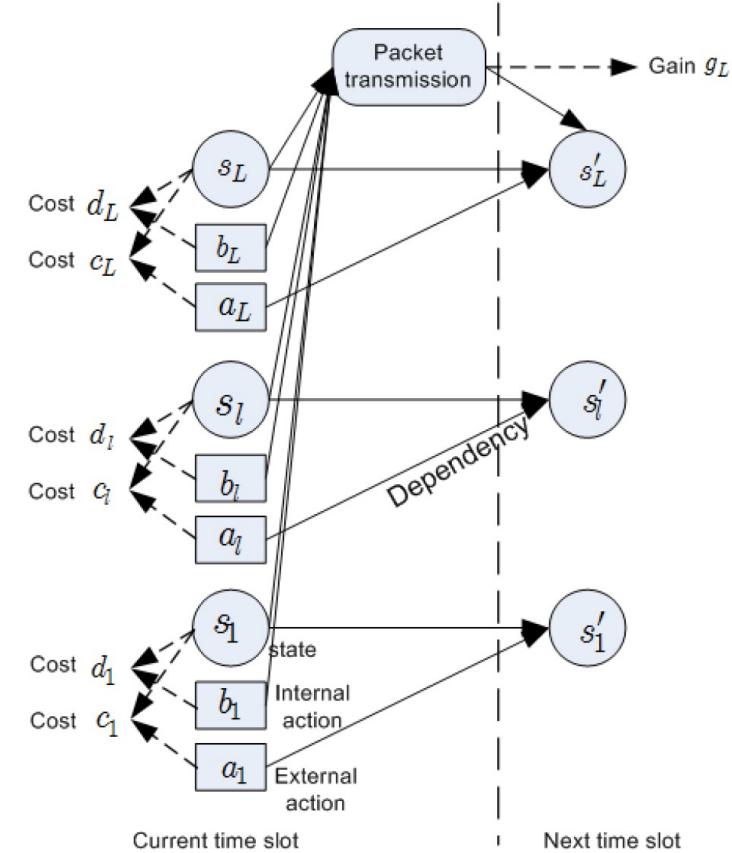
# Structure of cross-layer optimization (cont.)

- Transition probability

$$p(s' | s, \xi) = \prod_{l=1}^{L-1} p(s'_l | s_l, a_l) p(s'_L | s, a_L, b_{1,\dots,L-1})$$

- Utility function

$$R(s, \xi) = \underbrace{g_L(s, a_L, b_{1,\dots,L-1})}_{\text{Application utility}} - \lambda^b \underbrace{d(s, a_L, b_{1,\dots,L-1})}_{\text{internal cost}} - \sum_{l=1}^{L-1} \lambda_l^a \underbrace{c_l(s_l, a_l)}_{\text{external cost}}$$

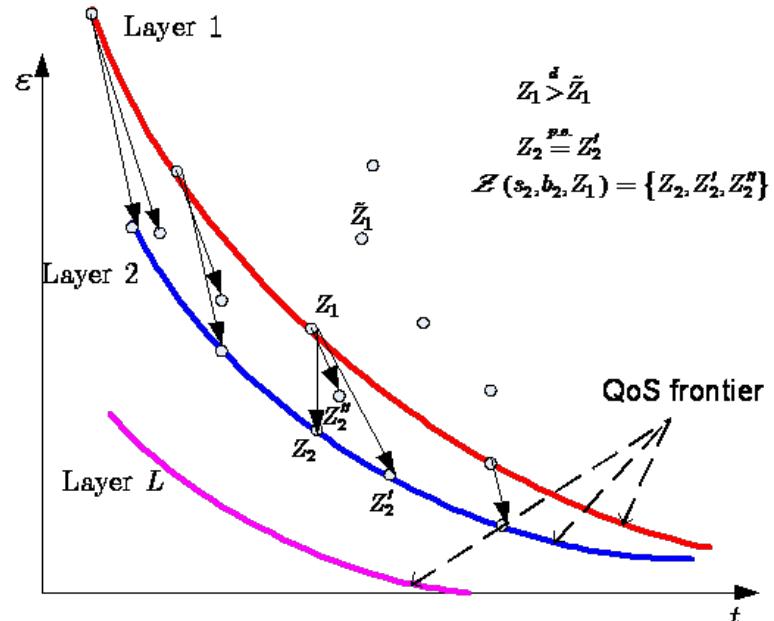


# QoS and internal reward computation

$$Z_l = (\varepsilon_l, \tau_l, v_l)$$

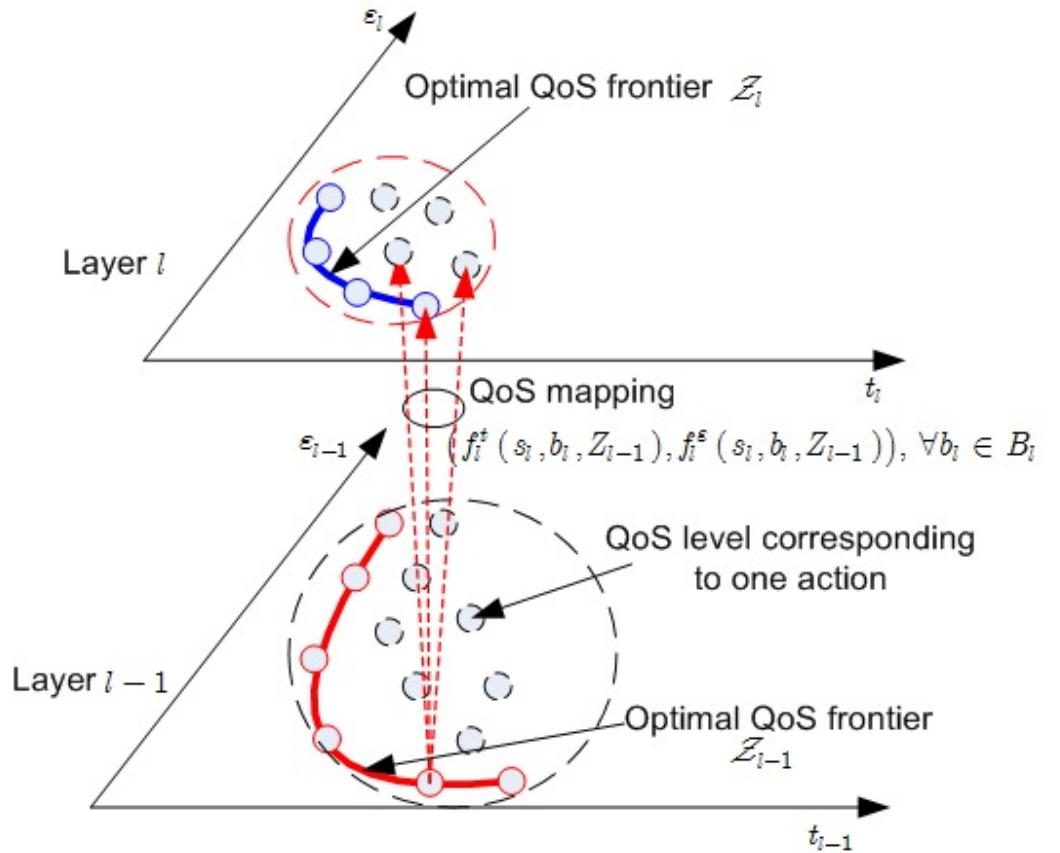
- packet loss probability  $\varepsilon_l$
- transmission time per packet  $\tau_l$
- transmission cost per packet  $v_l$

$$Z_l = (\varepsilon_l, \tau_l, v_l) = (f_l^\varepsilon(s_l, b_l, Z_{l-1}), f_l^\tau(s_l, b_l, Z_{l-1}), f_l^v(s_l, b_l, Z_{l-1}))$$



# Instantaneous QoS

- **Instantaneous QoS definition:**  $Z_l = (\varepsilon_l, \tau_l, v_l)$ 
  - packet loss probability  $\varepsilon_l$
  - transmission time per packet  $\tau_l$
  - transmission cost per packet  $v_l$



# Proposed layered DP operator

Centralized DP operator (existing):

$$V(s) = \max_{\xi} \left\{ R(s, \xi) + \alpha \sum_{s' \in S} p(s' | s, \xi) V(s') \right\}$$
$$V(s_1, \dots, s_L) = \max_{\substack{a_1, \dots, a_L \\ b_1, \dots, b_{L-1}}} \left\{ g_L(s, a_L, b_1, \dots, b_{L-1}) - \lambda^b d(s, a_L, b_1, \dots, b_{L-1}) - \sum_{l=1}^{L-1} \lambda^x c_l(s_l, a_l) + R_{in}(s_L, a_L, Z_{L-1}) \right.$$

Computed based on *QoS*

$$\left. \alpha \sum_{s' \in S} \prod_{l=1}^{L-1} p(s'_l | s_l, a_l) p(s'_L | s, a_L, b_1, \dots, b_{L-1}) V(s'_1, \dots, s'_L) \right\}$$



# Proposed layered DP operator (Cont'd)

Layered DP operator (proposed):

Layer  $L$

$$V_{L-1}(s'_1, \dots, s'_{L-1}, s_1, \dots, s_L) = \max_{a_L, Z_{L-1}} \left[ R_{in}(s_L, a_L, Z_{L-1}) - \lambda_L c_L(s_L, a_L) + \gamma \sum_{s_L \in \mathcal{S}_L} p(s'_L | s_L, a_L, Z_{L-1}) V_L(s'_1, \dots, s'_L) \right]$$

Layer  $l$

$$V_{l-1}(s_1, \dots, s_L, s'_1, \dots, s'_{l-1}) = \max_{a_l \in A_l} \left[ -\lambda_l c_l(s_l, a_l) + \sum_{s'_l \in \mathcal{S}_l} p(s'_l | s_l, a_l) V_l(s_1, \dots, s_L, s'_1, \dots, s'_l) \right]$$

Layer 1

$$V(s_1, \dots, s_L) = \max_{a_1 \in A_1} \left[ -\lambda_1 c_1(s_1, a_1) + \sum_{s'_1 \in \mathcal{S}_1} p(s'_1 | s_1, a_1) V_1(s_1, \dots, s_L, s'_1) \right]$$



# Proposed layered DP operator (Cont'd)

Layered DP operator (proposed):

Layer  $L$

$$V_{L-1}(s'_1, \dots, s'_{L-1}, s_1, \dots, s_L) = \max_{a_L, Z_{L-1}}$$
$$\left[ R_{in}(s_L, a_L, Z_{L-1}) - \lambda_L c_L(s_L, a_L) + \gamma \sum_{s'_L \in \mathcal{S}_L} p(s'_L | s_L, a_L, Z_{L-1}) V_L(s'_1, \dots, s'_L) \right]$$

Layer  $l$

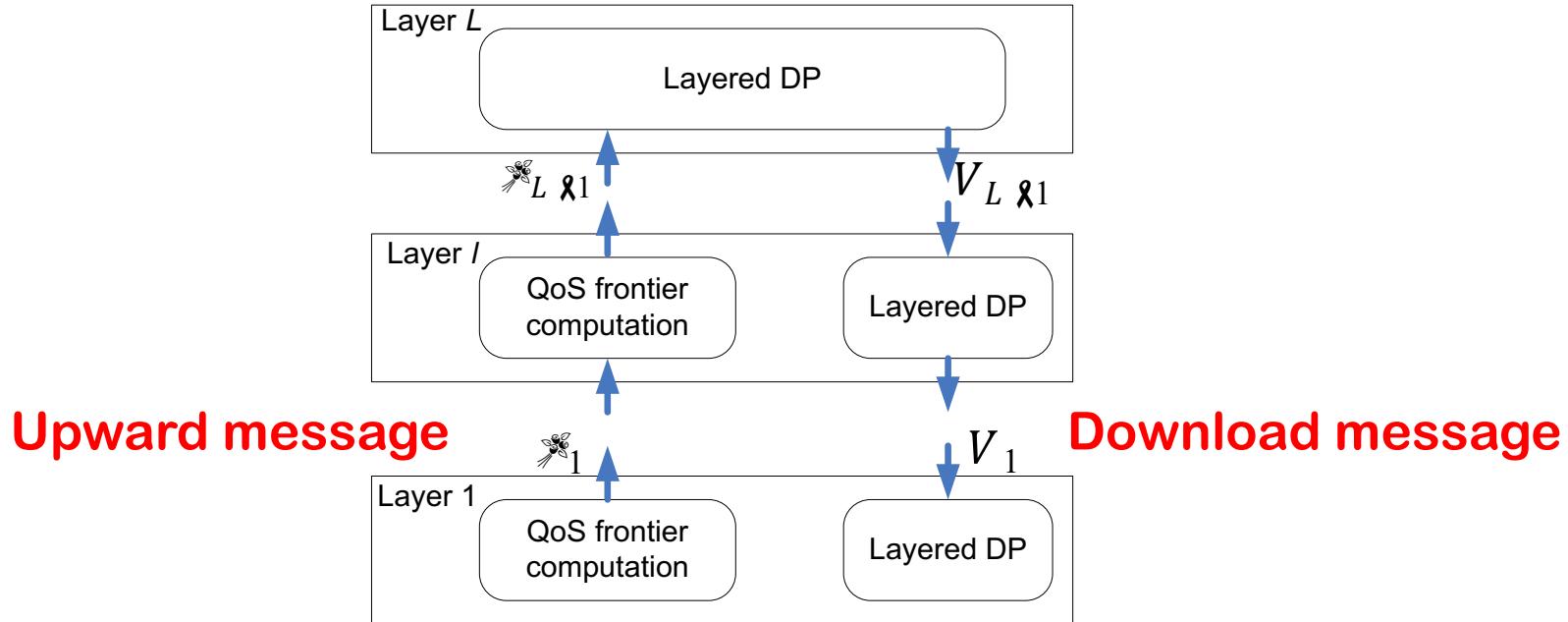
$$V_{l-1}(s_1, \dots, s_L, s'_1, \dots, s'_{l-1}) =$$
$$\max_{a_l \in A_l} \left[ -\lambda_l c_l(s_l, a_l) + \sum_{s'_l \in \mathcal{S}_l} p(s'_l | s_l, a_l) V_l(s_1, \dots, s_L, s'_1, \dots, s'_l) \right]$$

Layer 1

$$V(s_1, \dots, s_L) =$$
$$\max_{a_1 \in A_1} \left[ -\lambda_1 c_1(s_1, a_1) + \sum_{s'_1 \in \mathcal{S}_1} p(s'_1 | s_1, a_1) V_1(s_1, \dots, s_L, s'_1) \right]$$

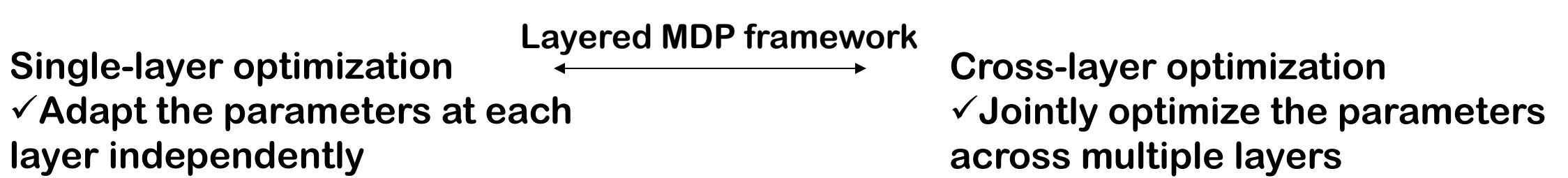


# Message exchange & layer optimizer



# Advantages of layered MDP framework

- This layered framework allows us to
  - Keep the layered architecture unaffected.
  - Independently upgrade protocols
  - Design minimal message exchange between layers
  - Provide performance bound for current ad-hoc layer-centric optimization



# Cross-layer optimization for delay-sensitive multimedia (Learning with real-time constraints)

Packet dependencies, delays, etc.

More sophisticated state definitions

More sophisticated environmental dynamics

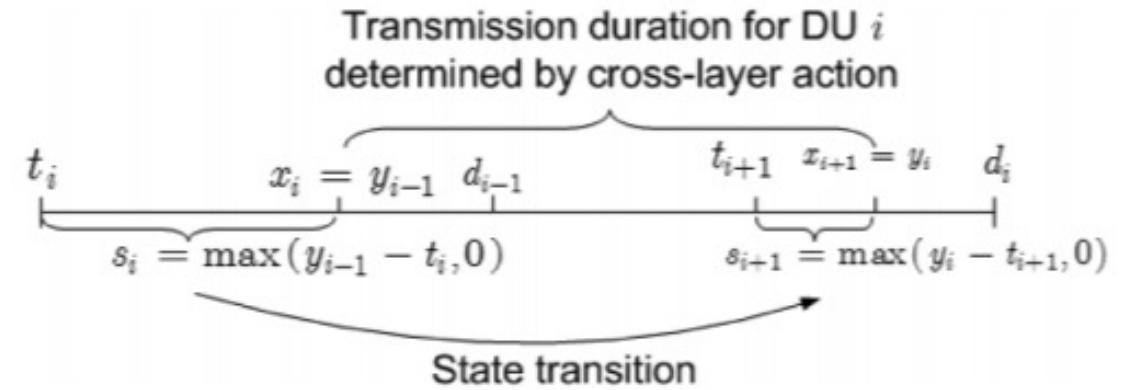


Fig. 2. State of DU  $i$  and state transition from DU  $i$  to DU  $i + 1$ .

F. Fu and M. van der Schaar, "Decomposition Principles and Online Learning in Cross-Layer Optimization for Delay-Sensitive Applications", *IEEE Trans. Signal Process.*, vol 58, no. 3, pp. 1401-1415, Feb. 2010

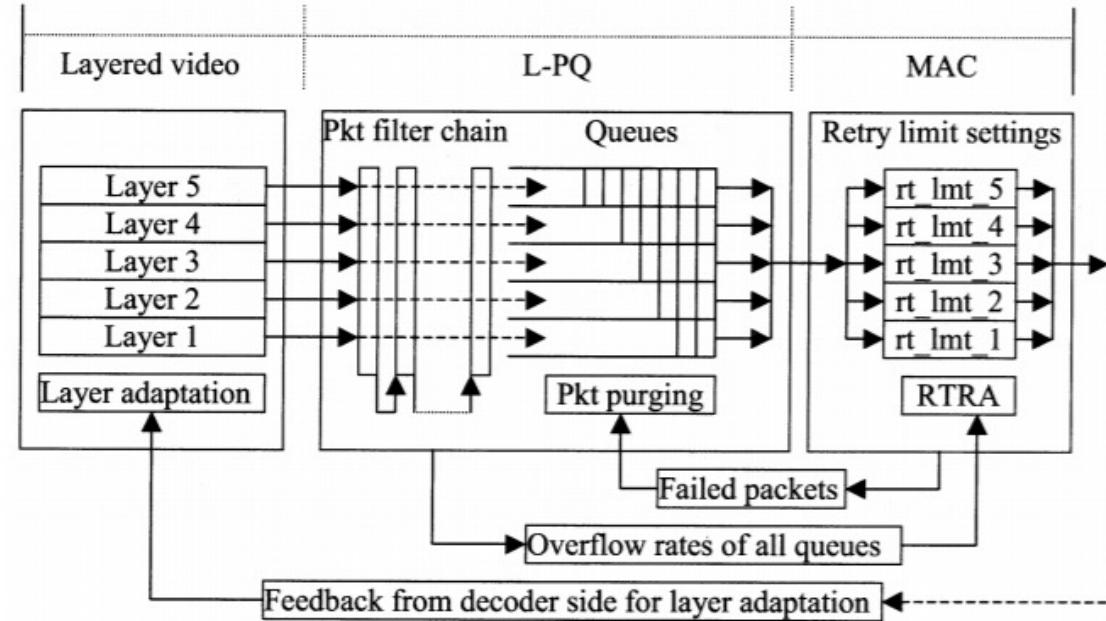


van\_der\_Schaar  
\ LAB

vanderschaar-lab.com

# How to adapt to changing demands and resources? Model-based adaptation using queuing theory

More sophisticated state definitions  
More sophisticated environmental dynamics



Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 278-290, Apr. 2004.

H. P. Shiang and M. van der Schaar, "Multi-user video streaming over multi-hop wireless networks: A distributed, cross-layer approach based on priority queuing," *IEEE J. Sel. Areas Commun.*, May 2007.

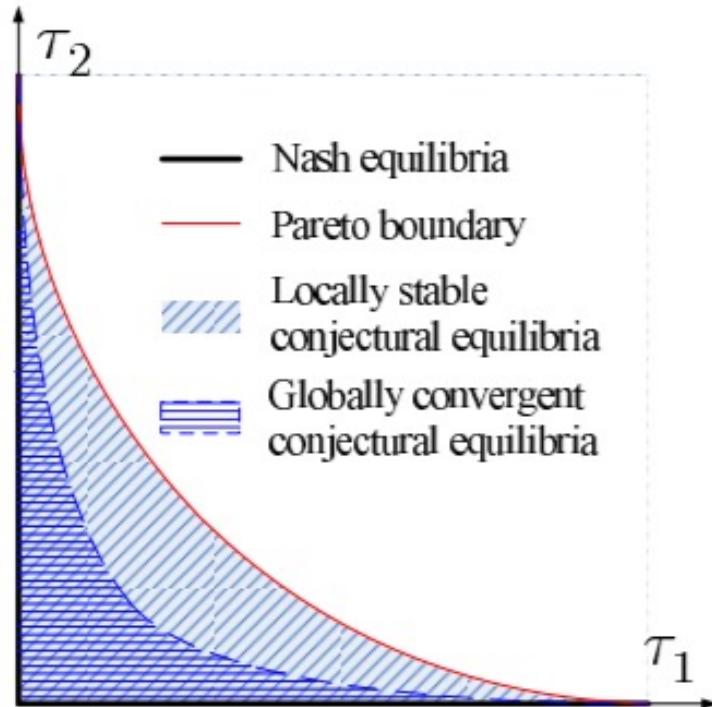


# Wireless communications as an ML problem

- Single-agent wireless communication
- Multi-agent wireless communication
  - Compliant users
    - Power control as a learning game
    - Slotted MAC protocols – going beyond slotted CSMA/CA – learning without communication
  - Strategic users
    - Resource competition – mechanism design + multi-agent reinforcement learning



# From single to multiple users: Interference Management using Multi-user Learning



Y. Su and M. van der Schaar, "Conjectural Equilibrium in Multi-user Power Control Games", *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3638-3650, Sep. 2009.

Y. Su and M. van der Schaar, "Dynamic Conjectures in Random Access Networks Using Bio-inspired Learning," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 4, pp. 587-601, May 2010.



# From single to multiple users: Better congestion control by learning to coordinate without communication

	<i>Lee+Walrand (ZC)</i>	<i>Fang et al (L-ZC)</i>	<i>Park + van der Schaar</i>	<i>He et al (SRB)</i>	<i>This paper (PC)</i>
<i>Channel Sensing When Not Transmitting</i>	Yes	Yes	Yes	Yes	No
<i>Fully Coordinated</i>	No	No	No	No	Yes
<i>Learn # of Stations</i>	No	No	No	No	Yes
<i>Number of Stations</i>	Unknown	Unknown	Known/ Unknown	Unknown	Known/ Unknown
<i>Convergence* (Small Networks)**</i>	Very Fast	Very Fast	Fast	Medium	Very Fast
<i>Convergence* (Large Networks)**</i>	Fast	Fast	Medium	Slow	Fast

\* Convergence speed is the number of slots required such that the system enters the steady state (perfect coordination or zero collision) with probability 0.999. (Very Fast: < 100 slots; Fast : 100 - 500 slots; Medium: 500 – 5000 slots; Slow: > 5000 slots.)

\*\* Small Networks: number of stations from 1 to 10; Large Networks: number of stations from 20 to 50.

TABLE I  
COMPARISON OF ASSUMPTIONS, RESULTS



van\_der\_Schaar  
\ LAB

vanderschaar-lab.com

**W. Zame, J. Xu and M. van der Schaar,  
"Winning the Lottery: Learning Perfect  
Coordination with Minimal Feedback," in *IEEE  
J. Sel. Topics in Signal Process.*, Oct. 2013**

**W. Zame, J. Xu and M. van der Schaar,  
"Cooperative Multi-Agent Learning and  
Coordination for Cognitive Radio  
Networks," *IEEE J. Sel. Areas Commun.*, Mar.  
2014.**

# From single to multiple users: Energy-Efficient Nonstationary Power Control in Wireless Networks

With Yuanzhang Xiao

Optimal foresighted multi-user wireless video  
IEEE Journal of Selected Topics in Signal Processing 9 (1), 89-101

Dynamic spectrum sharing among repeatedly interacting selfish users with imperfect monitoring  
IEEE Journal on Selected Areas in Communications 30 (10), 1890-1899

Distributed interference management policies for heterogeneous small cell networks  
IEEE Journal on Selected Areas in Communications 33 (6), 1112-1126

Efficient interference management policies for femtocell networks  
IEEE Transactions on Wireless Communications 14 (9), 4879-4893

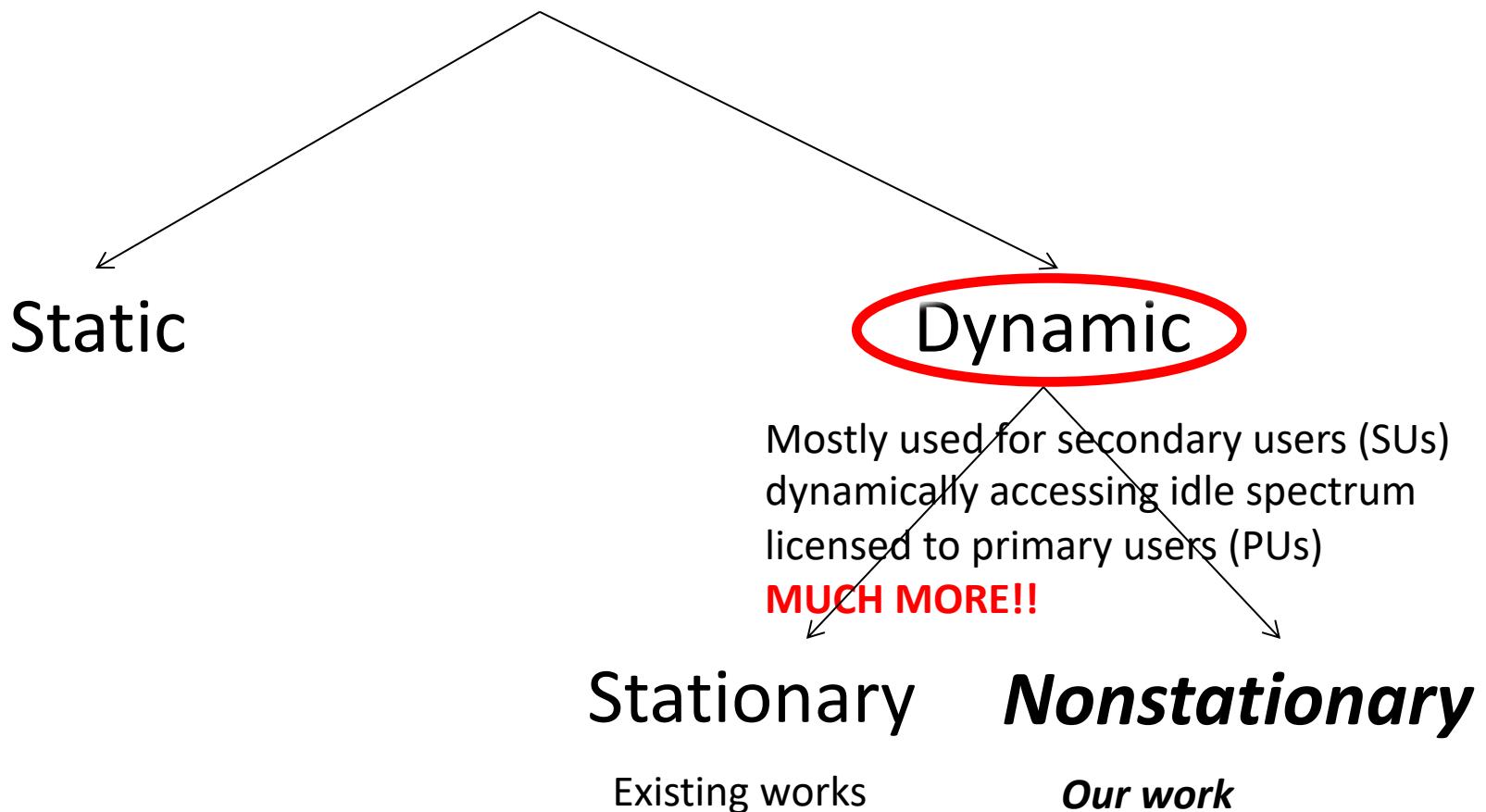


van\_der\_Schaar  
\ LAB

vanderschaar-lab.com

# Taxonomy

## (Spectrum) Sharing Policies

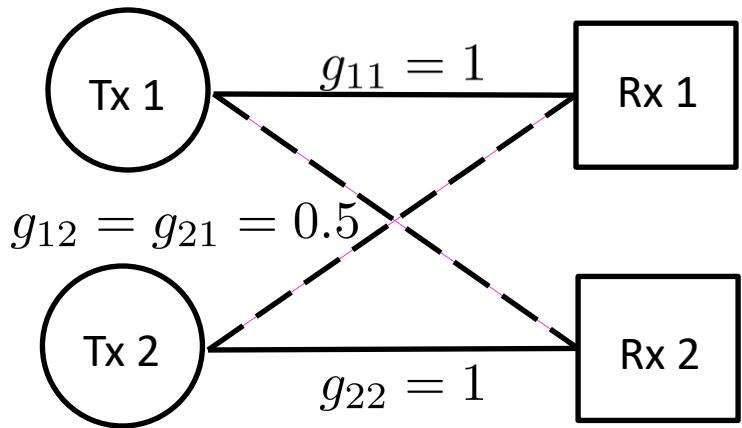


# Stationary vs. nonstationary policies

- Stationary policies: action depends on the *current state only*
- Nonstationary policies: action depends on the history as well as the current state
- Most existing works are *stationary*
  - Model the system as a *single*-user MDP: stationary policy is optimal
  - Model the system as a multi-user MDP, but restrict attention to stationary policies: *suboptimal*
- Well-known that the optimal resource sharing policies should be *nonstationary* [Altman 1999][Shimkin 1994] when
  - decentralized users have conflicting objectives
  - decentralized users have coupled constraints

# Illustration – Stationary policies

A simple two (secondary) user network:



Noise power at both users' receivers: 5 mW

Both users discount throughput and energy consumption by  $\delta = 0.8$

Minimum average throughput requirements

User 1 - 2 bits/s/Hz

User 2 - 1 bits/s/Hz

Channel gains are fixed

**Goal: minimize energy**

State: channel conditions (fixed)

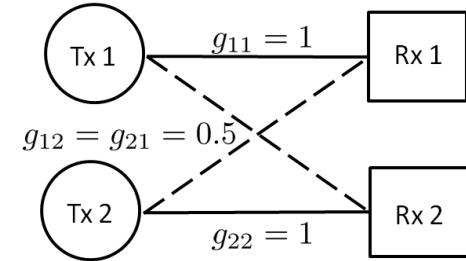
Action: transmit power levels

**Stationary policy:** Both users transmit at fixed power levels simultaneously  
(achieved by the power control algorithm by Yates *et. al.*)

Avg. energy consumption for user 1: 90 mW

Avg. energy consumption for user 2: 50 mW

# Illustration – Simple nonstationary policies



## A simple nonstationary policy: alternating round-robin TDMA

Transmit schedule: 1212...

(Actions are time dependent)

Avg. energy consumption for user 1: 31 mW

Avg. energy consumption for user 2: 8 mW

**Better....**

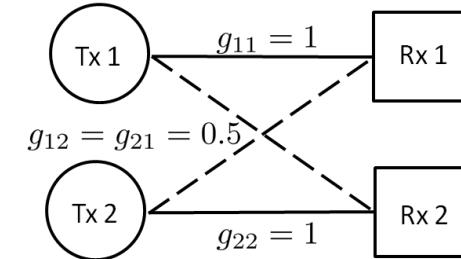
# Illustration – Simple nonstationary policies

**A modified round-robin TDMA policy:**

Transmit schedule: 112112...

Avg. energy consumption for user 1: 22 mW

Avg. energy consumption for user 2: 17 mW



Still better....

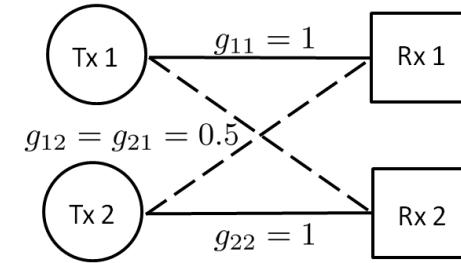
More complicated cycles?

Finding the “best” cycle is complicated – combinatorial

# Illustration – Optimal nonstationary policies

The optimal policy is NOT cyclic

Transmit schedule: 1111212112211221...



Moral:

- Optimal policy is not cyclic!

Good news:

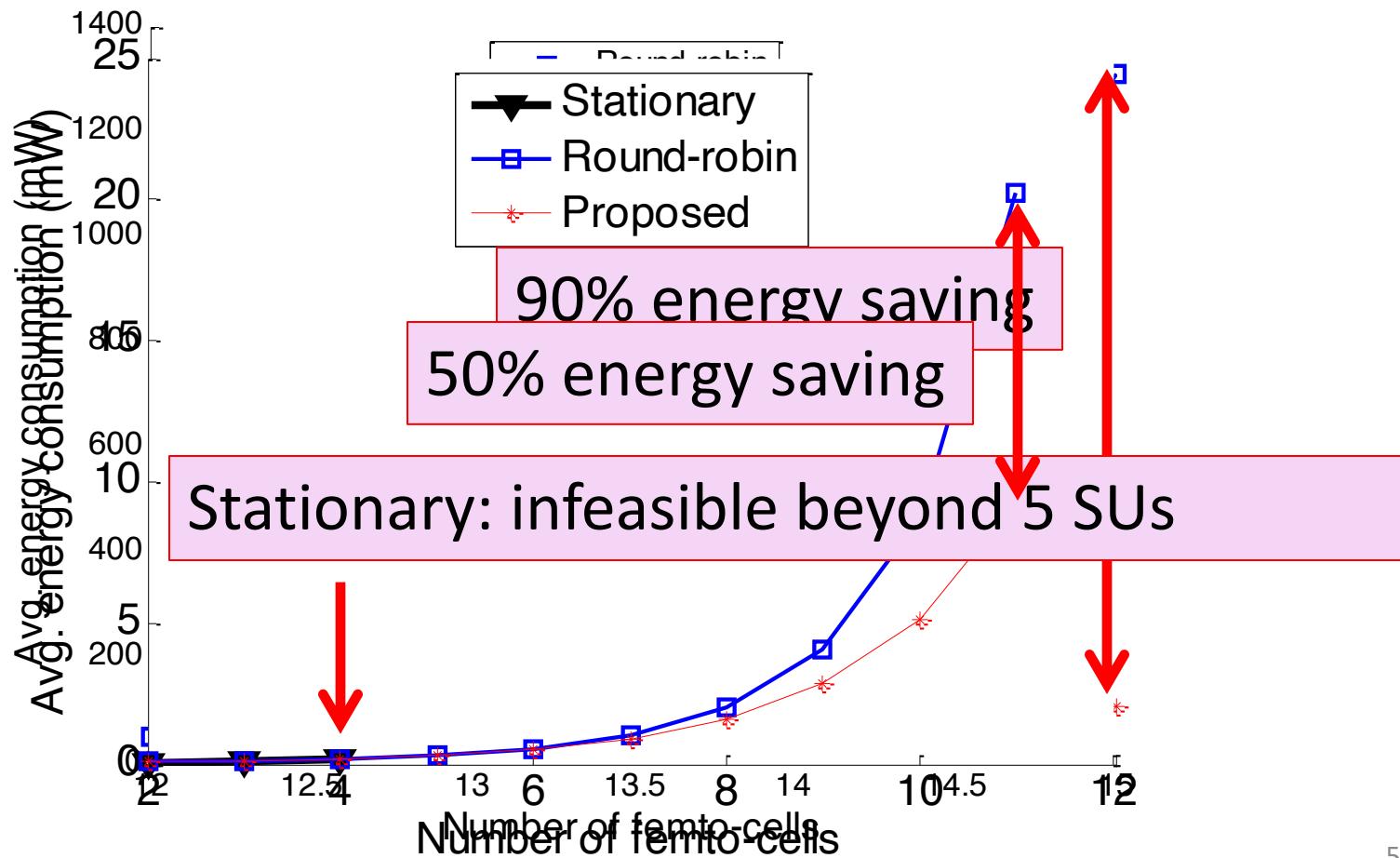
- We construct a simple, intuitive and general algorithm to build such policies
- Significant performance improvements

# Energy efficiency

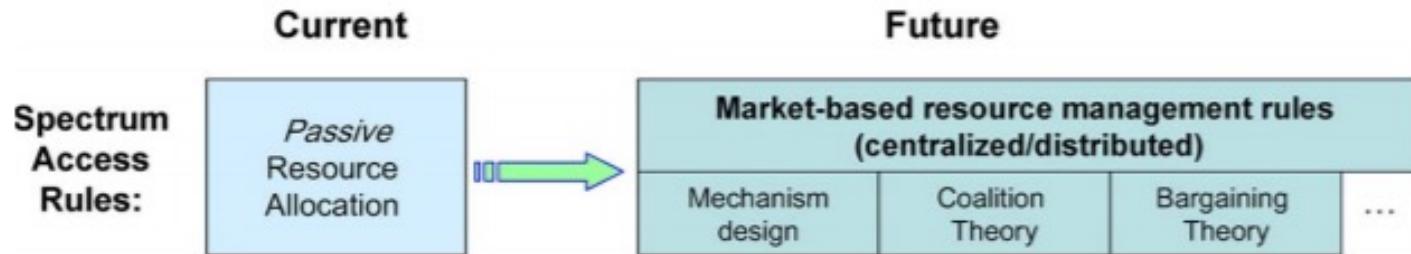
1 PU with minimum throughput requirement of 1 bit/s/Hz

2-15 SUs with minimum throughput requirement of 0.5 bit/s/Hz

Same number of SUs:



# From benevolent to strategic users: Multi-user wireless games, learning and decisions



**Fig. 2.** Evolution of spectrum access rules to create a dynamic wireless resource market.

F. Fu and M. van der Schaar, "Noncollaborative Resource Management for Wireless Multimedia Applications Using Mechanism Design," *IEEE Trans. Multimedia*, Jun. 2007

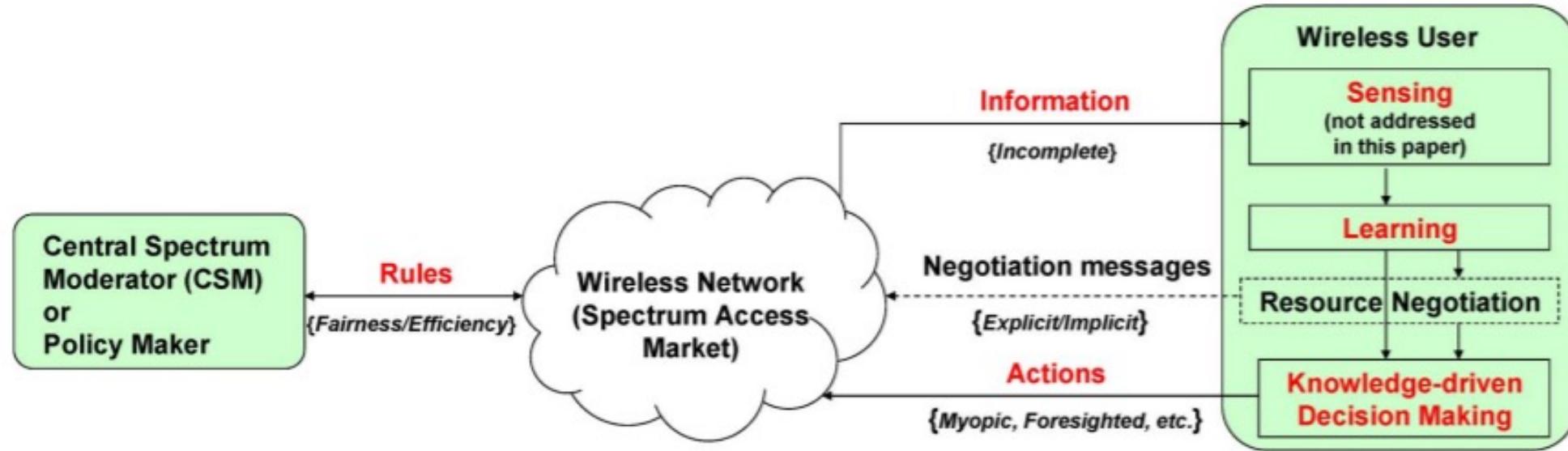
F. Fu, T. M. Stoenescu, and M. van der Schaar, "A Pricing Mechanism for Resource Allocation in Wireless Multimedia Applications," *IEEE Journal of Selected Topics in Signal Process.*, Aug. 2007

H. Park and M. van der Schaar, "Coalition-based Resource Reciprocity Strategies for P2P Multimedia Broadcasting," *IEEE Trans. Broadcast*. Sep. 2008

H. Park and M. van der Schaar, "Bargaining Strategies for Networked Multimedia Resource Management," *IEEE Trans. Signal Process.*, Jul. 2007.



# Reinforcement learning for multiple strategic users

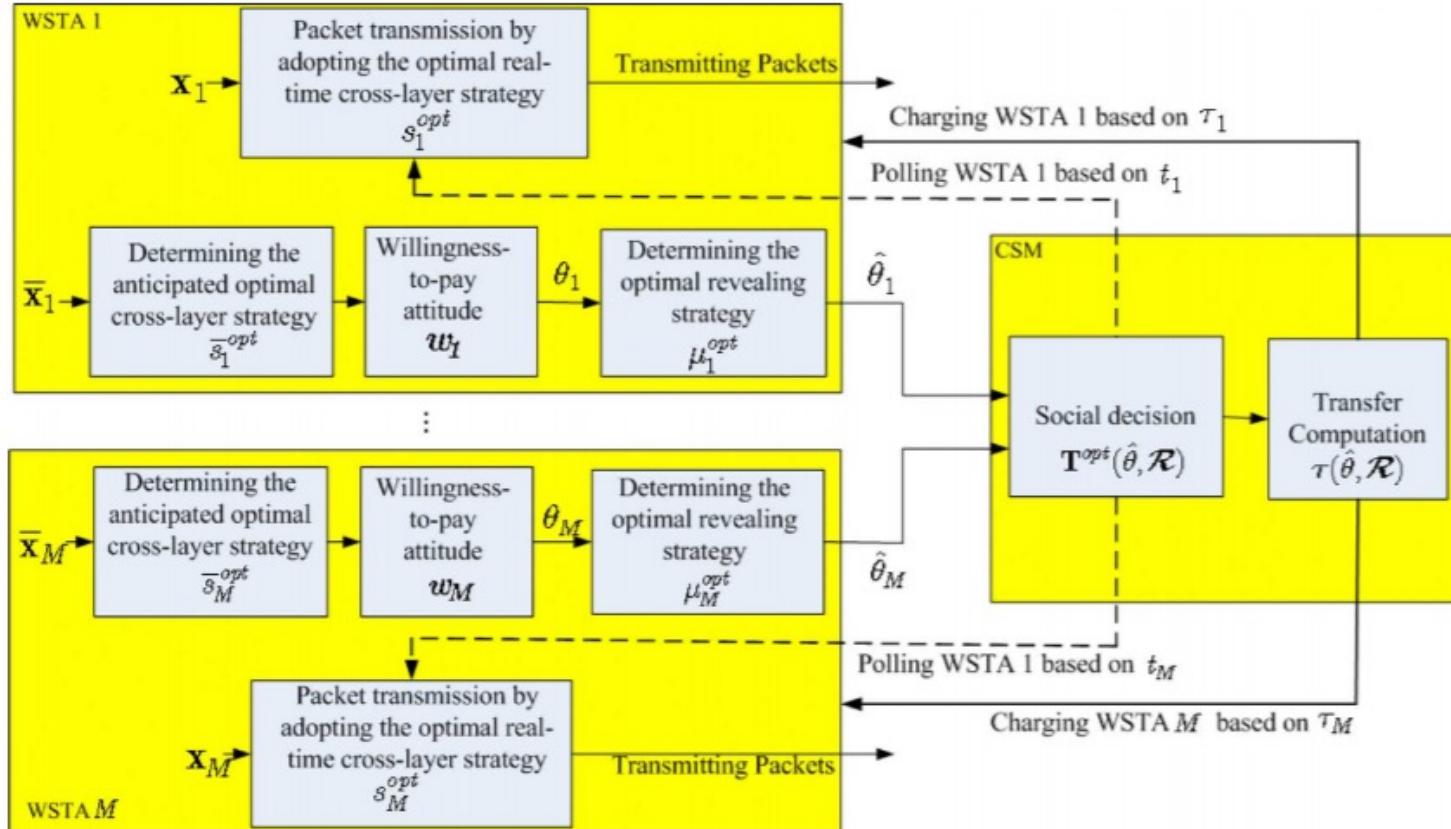


**Fig. 3. Knowledge-driven wireless networking.**

M. van der Schaar and F. Fu, "Spectrum Access Games and Strategic Learning in Cognitive Radio Networks for Delay-Critical Applications," *Proc. of IEEE, Special issue on Cognitive Radio*, Apr. 2009.



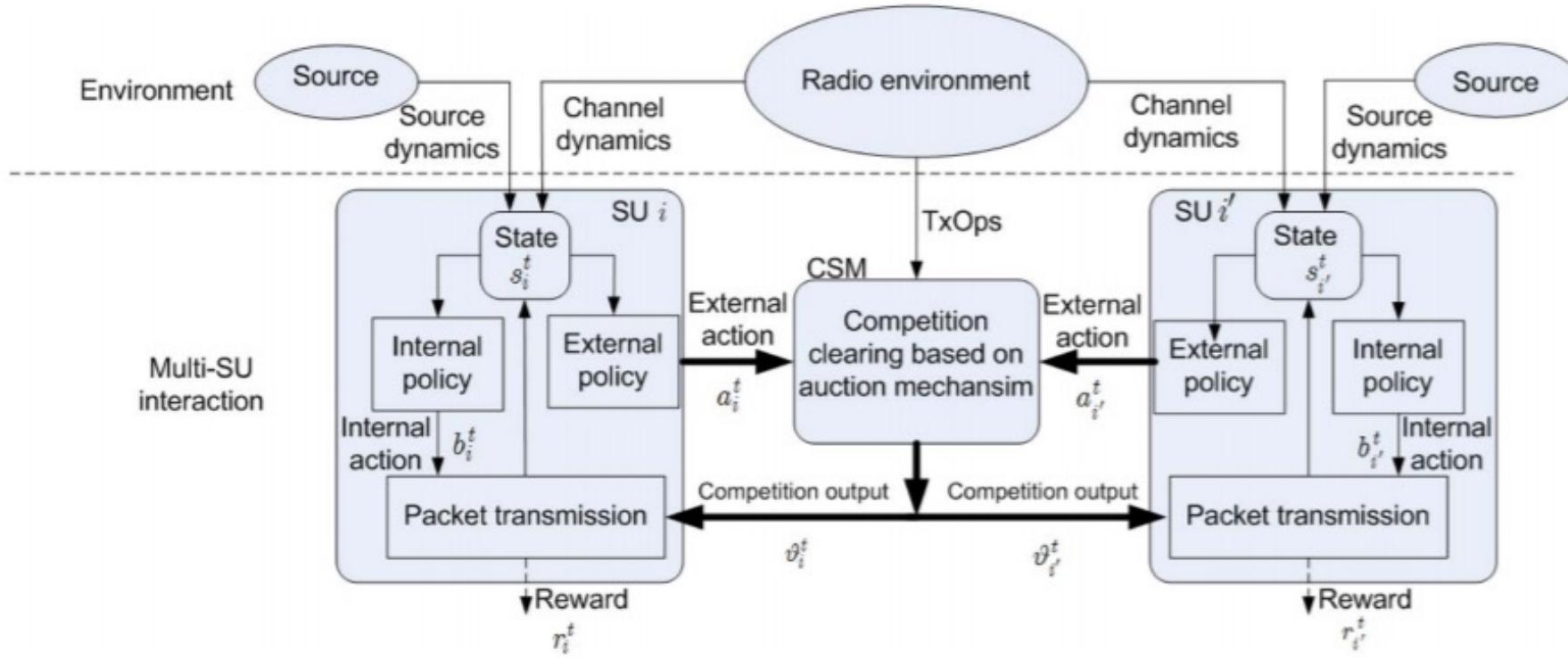
# Mechanism design based resource allocation and Multi-agent Reinforcement Learning



F. Fu and M. van der Schaar,  
"Noncollaborative Resource Management for Wireless Multimedia Applications Using Mechanism Design," *IEEE Trans. Multimedia*, Jun. 2007.



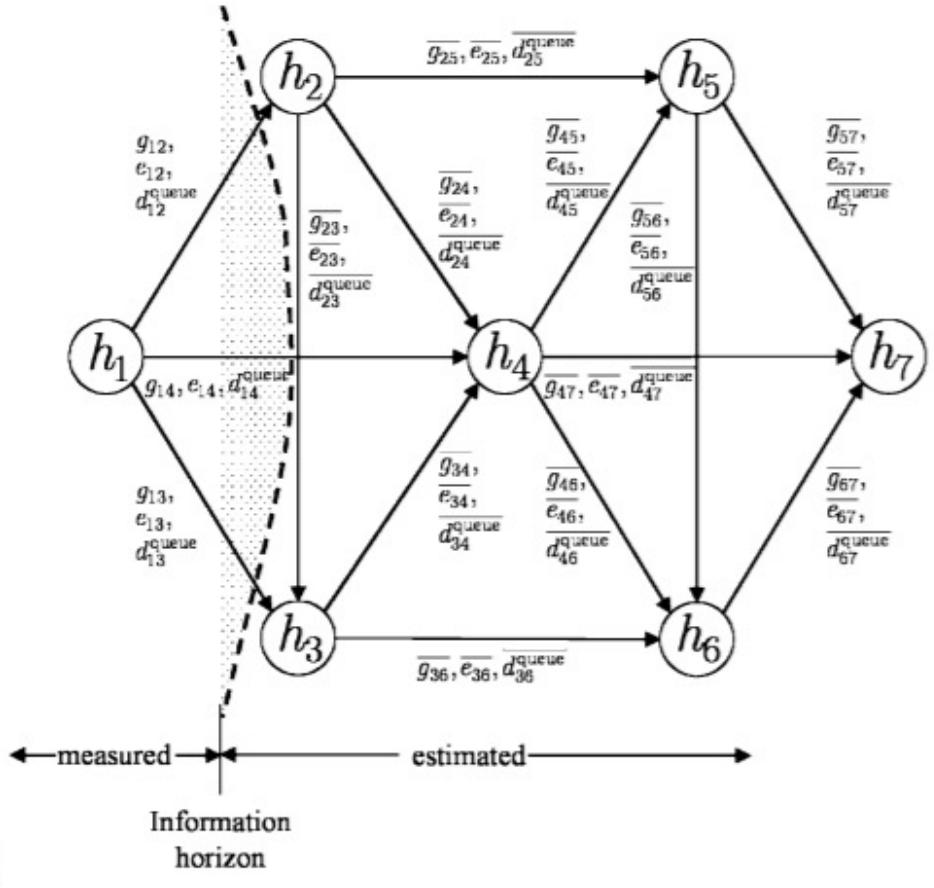
# Multi-user wireless games and reinforcement learning



F. Fu and M. van der Schaar, "Learning to Compete for Resources in Wireless Stochastic Games," *IEEE Trans. Veh. Tech.*, vol. 58, no. 4, pp. 1904-1919, May 2009.



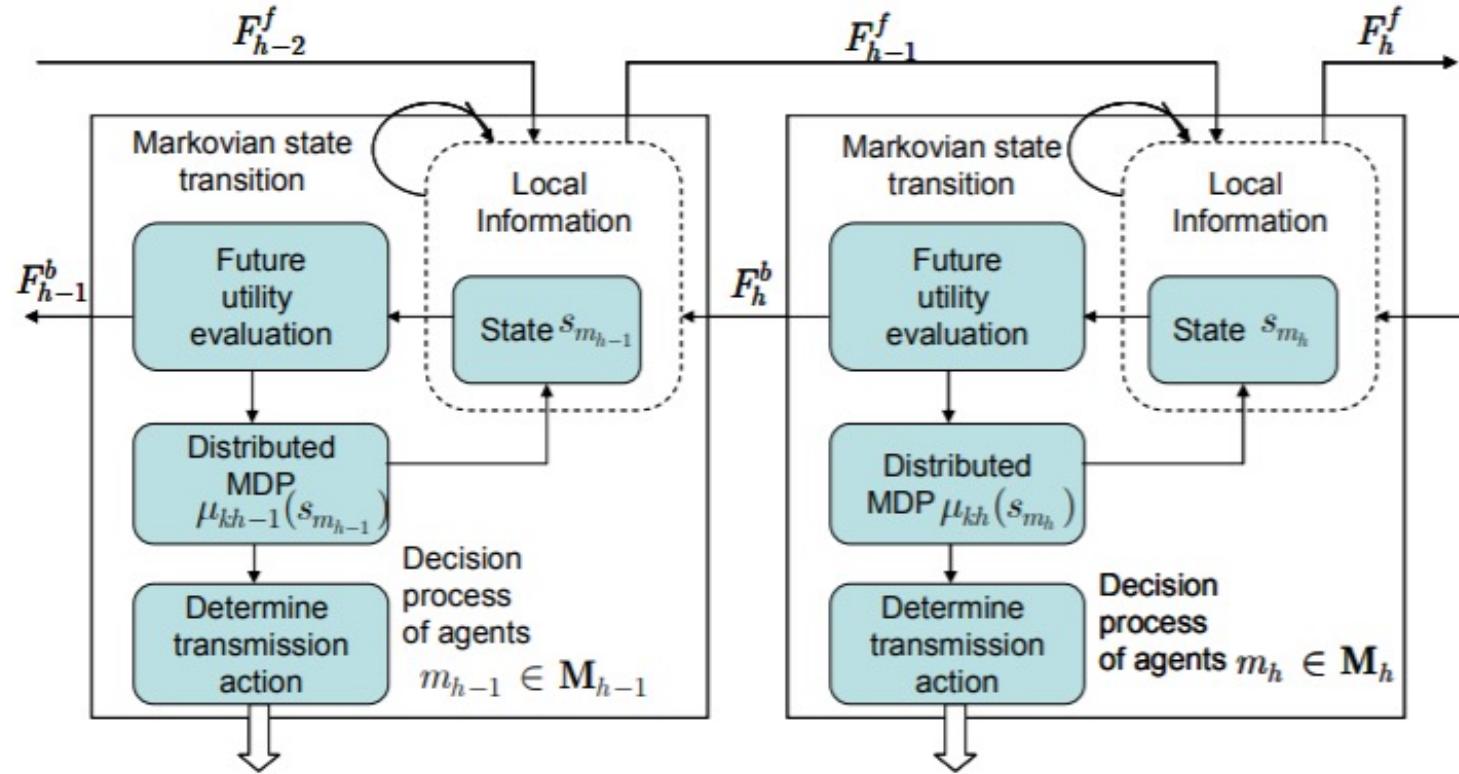
# Building BIG networks! Distributed optimization in multi-hop networks



Y. Andreopoulos, N. Mastronarde, and M. van der Schaar, "Cross-layer Optimized Video Streaming Over Wireless Multihop Mesh Networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 2104-2115, Nov. 2006.



# Building BIG networks! Multi-agent reinforcement learning in multi-hop nets



H. P. Shiang and M. van der Schaar,  
"Online Learning in Autonomic Multi-Hop Wireless Networks for Transmitting Mission-Critical Applications," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 5, pp. 728-741, June 2010.



# Wireless communications as an ML problem

- Single-agent wireless communication
- Multi-agent wireless communication
  - Compliant users
    - Power control as a learning game
    - Slotted MAC protocols – going beyond slotted CSMA/CA – learning without communication
  - Strategic users
    - Resource competition – mechanism design + multi-agent reinforcement learning
- Beyond wireless communications: social networks, content caching etc.



# Internet of Things: Communication among smart devices

TABLE II  
COMPARISON OF DIFFERENT POLICIES.

	[13]	[15]	[19]	[20]	This paper
Required Knowledge	Packet-level	Packet-level	Statistics	Statistics	Statistics
Delay Constraint	Yes	Yes	No	Yes	Yes
Allocation Method	Earliest deadline first	State-dependent	Minimal lead first	Weighted round robin	Non-stationary
Complexity	High	High	Low	Low	Low

J. Xu, Y. Andreopoulos, Y. Xiao and M. van der Schaar, "Non-stationary Resource Allocation Policies for Delay-constrained Video Streaming: Application to Video over Internet-of-Things-enabled Networks," *J. Sel. Areas in Commun.*, Apr. 2014.



# Incentivizing distributed online exchanges: Fiat money – a first theory

M. van der Schaar, J. Xu and W. Zame, "Efficient Online Exchange via Fiat Money," in *Economic Theory*, vol. 54, no. 2, pp. 211-248, Oct. 2013

**Theorem 4** *For each benefit/cost ratio  $r > 1$  and discount factor  $\beta < 1$  the set  $EQ(r, \beta)$  is either empty or consists of protocols that involve only (possibly degenerate) mixtures of two threshold strategies with adjacent thresholds.*

**Theorem 5** *If  $\Pi = (\alpha, \sigma)$  is a robust equilibrium then  $\sigma$  is a pure threshold strategy.*

**Theorem 6** *Fix a protocol  $\Pi = (\alpha, \sigma_K)$ .*

- (i) *For each benefit/cost ratio  $r > 1$ , the set  $\{\beta : \Pi \in EQ(r, \beta)\}$  is a non-degenerate closed interval  $[\beta_L(\Pi), \beta_H(\Pi)]$  whose endpoints are continuous functions of  $r$ .*
- (ii) *For each discount factor  $\beta < 1$ , the set  $\{r : \Pi \in EQ(r, \beta)\}$  is a non-degenerate closed interval  $[r_l(\Pi), r_H(\Pi)]$  whose endpoints are continuous functions of  $\beta$ .*



# Fiat-Money: Applications in wireless networks

## Reinforcement learning to learn how to trade tokens

### Multi-hop networks

J. Xu and M. van der Schaar, "Token System Design for Autonomic Wireless Relay Networks," in *IEEE Trans. on Commun.*, July 2013

### Interference management

C. Shen, J. Xu and M. van der Schaar, "Silence is Gold: Strategic Interference Mitigation Using Tokens in Heterogeneous Small Cell Networks," *IEEE J. Sel. Areas Commun.*, June 2015

### Device-to-Device Communication

N. Mastronarde, V. Patel, J. Xu, L. Liu, and M. van der Schaar , "To Relay or Not to Relay: Learning Device-to-Device Relaying Strategies in Cellular Networks," *IEEE Transactions on Mobile Computing*, June 2016.



van\_der\_Schaar  
\ LAB

vanderschaar-lab.com

# Context-Aware Caching – a contextual learning approach

TABLE I

COMPARISON WITH RELATED WORK ON LEARNING-BASED CACHING WITH PLACEMENT AND DELIVERY PHASE.

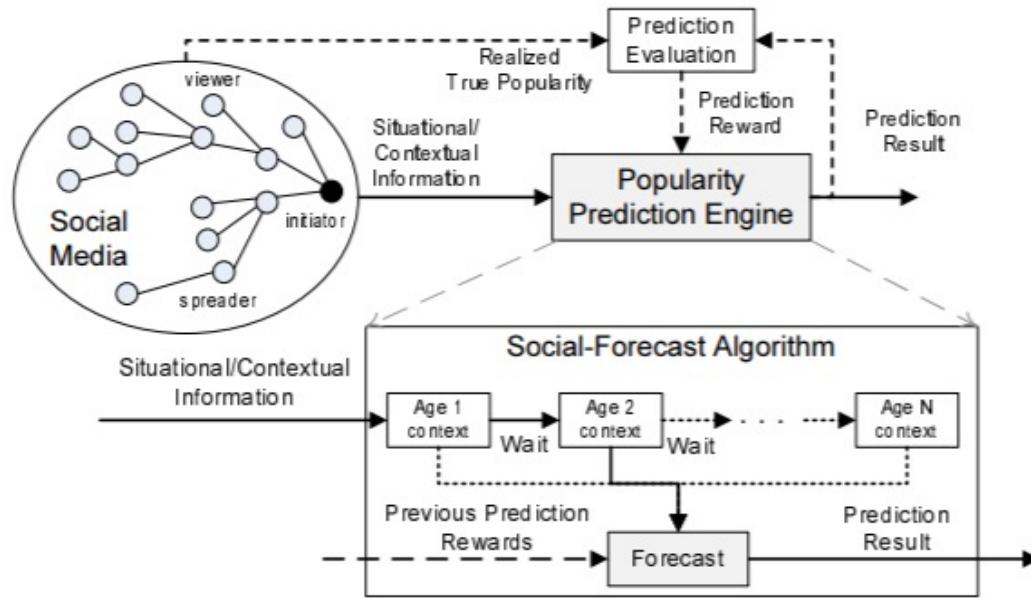
	[13], [14]	[15]–[17]	[18]	[19]	This work
Model-Free	Yes	Yes	No	Yes	Yes
Online/Offline-Learning	Offline	Online	Online	Online	Online
Free of Training Phase	No	Yes	Yes	No	Yes
Performance Guarantees	No	Yes	No	No	Yes
Diversity in Content Popularity	No	No	No	Yes	Yes
User Context-Aware	No	No	No	No	Yes
Service Differentiation	No	No	No	No	Yes

S. Li, J. Xu, M. van der Schaar, and W. Li, "Trend-Aware Video Caching through Online Learning," *IEEE Transactions on Multimedia*, 2016.

S. Muller, O. Atan, M. van der Schaar and A. Klein, "Context-Aware Proactive Content Caching With Service Differentiation in Wireless Networks," in *IEEE Transactions on Wireless Communications*, Feb. 2017.



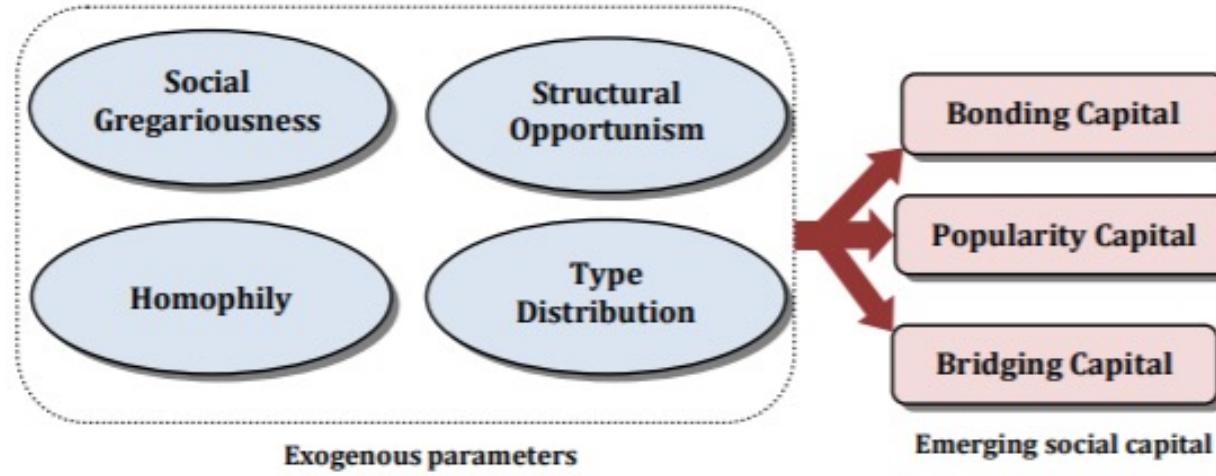
# Forecasting popularity in networks – a contextual learning approach



J. Xu, M. van der Schaar, J. Liu and H. Li, "Forecasting Popularity of Videos using Social Media," *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, Nov. 2014.



# Network science: Learning in social networks



A. Alaa, K. Ahuja, and M. van der Schaar, "A Micro-foundation of Social Capital in Evolving Social Networks," *IEEE Transactions on Network Science and Engineering*, 2017.

S. Zhang and M. van der Schaar, "From Acquaintances to Friends: Homophily and Learning in Networks," *the 2017 JSAC Game Theory for Networks special issue.*, 2017



# Wireless communications as an ML problem

- Single-agent wireless communication
  - Multi-agent wireless communication
  - Beyond wireless communications: social networks, content caching etc.
- 
- The journey continued
- [www.vanderschaar-lab.com/publications/communications-and-networks](http://www.vanderschaar-lab.com/publications/communications-and-networks)



## Part 4: Design Examples in Wireless Communications

Cong Shen

Charles L. Brown Department of Electrical and Computer Engineering  
University of Virginia

Email: [cong@virginia.edu](mailto:cong@virginia.edu)

ICC 2021 Tutorial: Online Learning for Wireless Communications:  
Theory, Algorithms, and Applications

## Example Applications

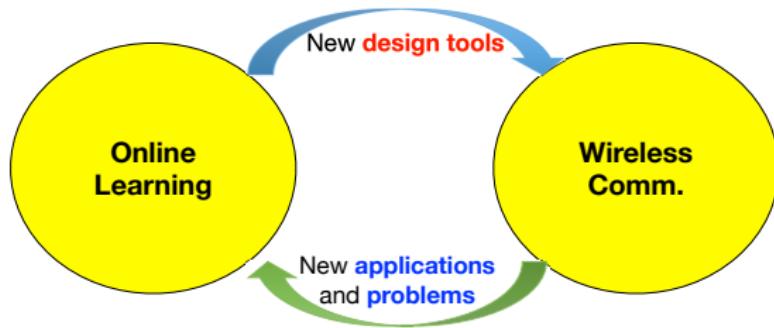
- For the remainder of this tutorial, we will focus on the applications of online learning, in particular **multi-armed bandits**, in wireless communications.

## Example Applications

- For the remainder of this tutorial, we will focus on the applications of online learning, in particular **multi-armed bandits**, in wireless communications.
- These examples demonstrate a “**coupled**” relationship between online learning and wireless comm.

## Example Applications

- For the remainder of this tutorial, we will focus on the applications of online learning, in particular **multi-armed bandits**, in wireless communications.
- These examples demonstrate a “**coupled**” relationship between online learning and wireless comm.
  - Advances in MAB can directly impact the algorithm and protocol design of wireless
  - Unique challenges in wireless also provide new and meaningful challenges for MAB research

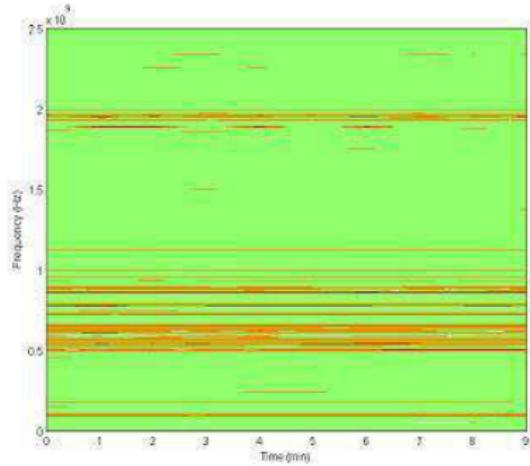
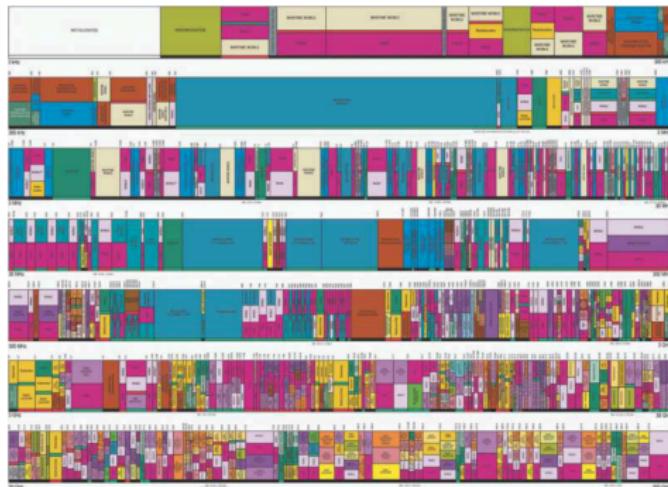


# Example Applications

We will focus on two wireless applications, to demonstrate the intimate relationship between MAB and wireless comm.

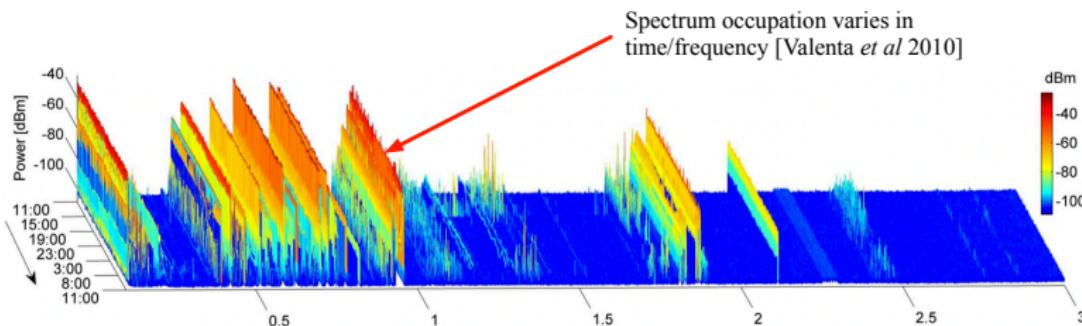
- Spectrum Engineering.
- Mobility Management.

# Application: Opportunistic Spectrum Access (OSA)

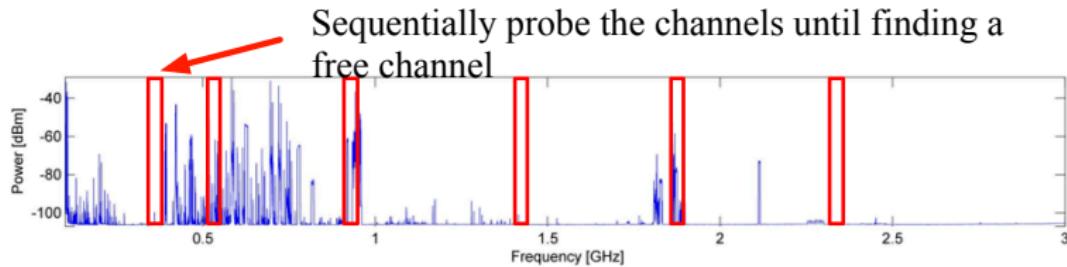


Opportunistic Spectrum Access (OSA): dynamically probe and access channels that are not used

\* Image credit: Wikipedia & D. Cabric's PhD thesis



Each time frame, the transmitter dynamically identifies a free channel to use by **channel sensing/probing**



# MAB for OSA

Standard MAB is a natural tool for OSA...

MAB	OSA
arms	channels
pull an arm	measure the channel
rewards	channel quality (e.g., SINR, throughput)
regret minimization	maximizing long-term system utility
best arm identification	selecting channel with highest quality

# MAB for OSA

... But standard MAB is far from perfect for OSA. We show two examples.

- Good arm identification (GAI), instead of best arm identification (BAI), for OSA [1]
- Federated multi-armed bandits (FMAB) to solve the network listen problem [2][3]

[1] Z. Wang, Z. Ying, CS, "Opportunistic Spectrum Access via Good Arm Identification," IEEE GlobalSIP 2018, Nov. 2018

[2] C. Shi, CS, "Federated Multi-armed Bandits," The 35th AAAI Conference on Artificial Intelligence (AAAI), Feb. 2021

[3] C. Shi, CS, J. Yang, "Federated Multi-armed Bandits with Personalization," The 24rd International Conference on Artificial Intelligence and Statistics (AISTATS), Apr. 2021

# MAB for wireless network optimization

- Mobility Management

- ▶ Relationship of 3GPP mobility protocol and MAB [4]
- ▶ Cascading-bandit to construct efficient Neighbor Cell List (NCL) for mobility management [5,6]

[4] CS, M. van der Schaar, "A Learning Approach to Frequent Handover Mitigations in 3GPP Mobility Protocols," IEEE WCNC, March 2017.

[5] R. Zhou, C. Gan, J. Yang, CS, "Cost-aware Cascading Bandits," in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), July 2018

[6] C. Wang, R. Zhou, J. Yang, CS, "A Cascading Bandit Approach to Efficient Mobility Management in Ultra-Dense Networks," IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Oct. 2019

# Good arm identification for OSA

## Arm selection for OSA

- BAI can be directly applied to OSA, but ...
- Even **optimal** algorithms can have **slow convergence**, **when the game is hard** (suboptimal arms are very close to the best arm)

## Arm selection for OSA

- BAI can be directly applied to OSA, but ...
- Even **optimal** algorithms can have **slow convergence**, **when the game is hard** (suboptimal arms are very close to the best arm)

**Is it really necessary to spend resources distinguishing the best channel and the next best channel that is 99%?**

## Arm selection for OSA

- BAI can be directly applied to OSA, but ...
- Even **optimal** algorithms can have **slow convergence**, **when the game is hard** (suboptimal arms are very close to the best arm)

**Is it really necessary to spend resources distinguishing the best channel and the next best channel that is 99%?**

- For OSA, the goal should be to find a good-enough channel **efficiently**

## Notations

- A set of candidate radio channels  $\mathcal{K} = \{1, 2, \dots, K\}$
- Channel quality feedback  $r_i(t), \forall i \in \mathcal{K}$ 
  - ▶  $r_i(t)$  is random variable with mean  $\mu_i = \mathbb{E}[r_i(t)]$
  - ▶ Examples include SINR, capacity, etc
- Minimum acceptable performance threshold for the user:  $\tau$ 
  - e.g., minimum rate requirement for a given QoS class
- Output a channel  $\Omega(T)$  and make sure it is “**good enough**”  
 $\mu_{\Omega(T)} \geq \tau$  with high confidence

# The Bandit Model

- **arms:**  $\mathcal{K} = \{1, 2, \dots, K\}$ ;
- **action:** Choosing a channel  $I(t) \in \mathcal{K}$  to sense;
- **reward:** Channel quality feedback  $r_{I(t)}(t)$ ;
- **goal:** Minimize error probability  $e_T = \mathbb{P}(\mu_{\Omega(T)} < \tau)$ .

# Good Channel Identification Algorithm (GCI)

---

**Algorithm 1** Good Channel Identification

---

**Input:**  $\mathcal{K}, \tau, T$ .

**Initialize:** Sample each channel once with feedback  $\hat{\mu}_i(K)$ ; Set  $T_i(K) = 1$ ,  $t = K + 1$ ; Calculate  $L_i(K)$  and  $U_i(K)$ .

1: **while**  $t \leq T$  **do**

2:   Compute  $B_i(t) = \max_{k \neq i} U_k(t) - L_i(t)$ .

3:    $l_t = \arg \min_{i \in \mathcal{K}} B_i(t)$ ,  $u_t = \arg \max_{j \neq l_t} U_j(t)$ .

4:   Select channel  $I(t) = \arg \max_{k \in \{u_t, l_t\}} \alpha_k(t)$ .

5:   Observe feedback  $r_{I(t)}(t)$  and update  $\hat{\mu}_{I(t)}(t) = \frac{\hat{\mu}_{I(t)}(t-1)T_{I(t)}(t-1) + r_{I(t)}(t)}{T_{I(t)}(t-1)+1}$ ,  $T_{I(t)}(t) = T_{I(t)}(t-1)+1$ ,

6:    $t = t + 1$ .

7: **end while**

**Return:**  $\Omega(T) = \arg \min_{l_t} B_{l_t}(t)$ .

---

Figure: GCI algorithm

- Construct confidence intervals  
 $U_i(t) = \hat{\mu}_i(t) + \alpha_i(t)$ ,  
 $L_i(t) = \hat{\mu}_i(t) - \alpha_i(t)$
- $l_t$  - minimum upper bound of the gap compared to the empirically optimal channel  
 $u_t$  - the estimated optimal channel other than  $l_t$
- Select the most uncertain channel between  $l_t$  and  $u_t$

# Error probability of GCI

## Theorem

With  $\hat{H}_1(t) = \sum_{i=1}^K \frac{1}{(\hat{\mu}^*(t)-\tau)^2} \mathbb{1}_{\hat{\mu}_i(t) \geq \tau} + \frac{4}{(\hat{\mu}^*(t)+\tau-2\hat{\mu}_i(t))^2} \mathbb{1}_{\hat{\mu}_i(t) < \tau}$  and  $\alpha_i(t) = \sqrt{\frac{T-K}{4\hat{H}_1(t)T_i(t)}}$ , the error probability  $e_T$  of Algorithm 1 satisfies:

$$\begin{aligned} e_T &= \mathbb{P}(\mu_{\Omega(T)} < \tau) = \mathbb{P}(\mu_1 - \mu_{\Omega(T)} > \mu_1 - \tau) \\ &\leq 2KT \exp\left(-\frac{T-K}{2H_1}\right), \end{aligned}$$

where  $H_1 = \sum_{i=1}^K \frac{1}{\max(\frac{\delta_i + \Delta_1}{2}, \Delta_1)^2}$  and  $\Delta_1 = \mu_1 - \tau$ .

Exponentially decreasing with  $T$ , faster compare with Best Arm Identification due to the smaller hardness quantity  $H_1$

# Simulation

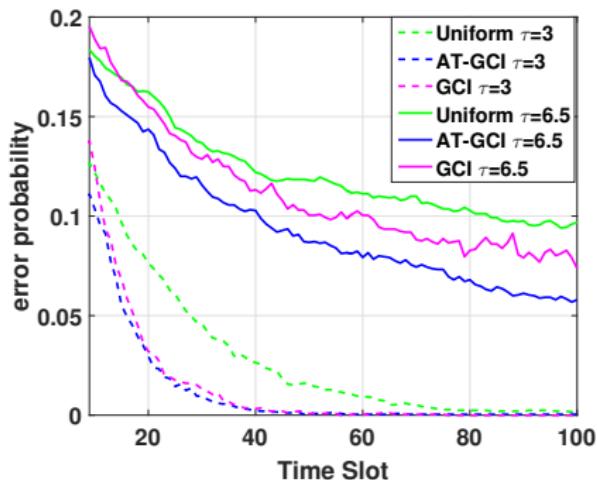


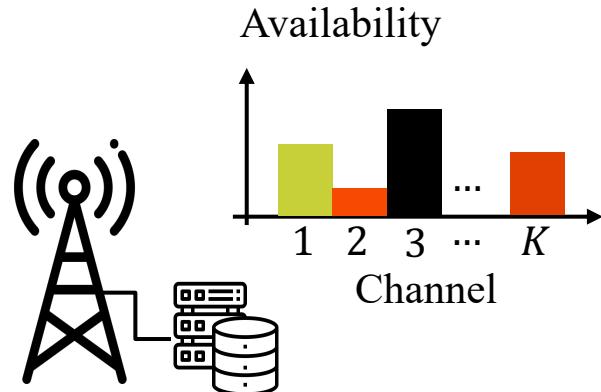
Figure: Error probability when  $\tau = 6.5$  and  $\tau = 3$ .

- Users arrive randomly  $\sim$  homogenous Poisson Point Process (PPP) with density  $\lambda_i$
- Users are uniformly distributed on all channels
- Interference are computed based on pathloss model
- Feedback:  $r_i(t) = \log \left( 1 + \frac{P}{N_0 + \sum_{k \in \mathcal{N}_i(t)} P_r^k} \right)$

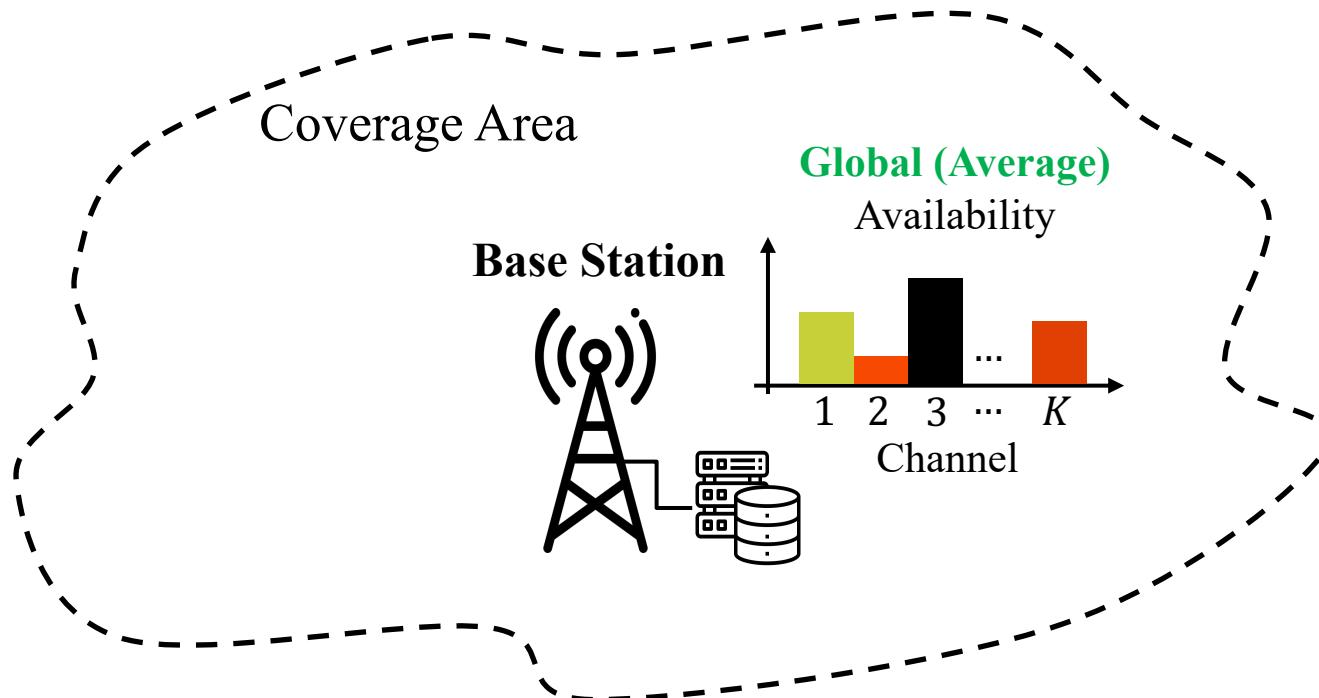
# Federated Multi-armed Bandits

# Motivating Wireless Application

- Cognitive radio (CR) and dynamic spectrum access is often used as a motivating example for MAB research
  - BS wants to use the statistically “best” channel for communication
  - A good match to (stochastic) MAB problems



# Network View

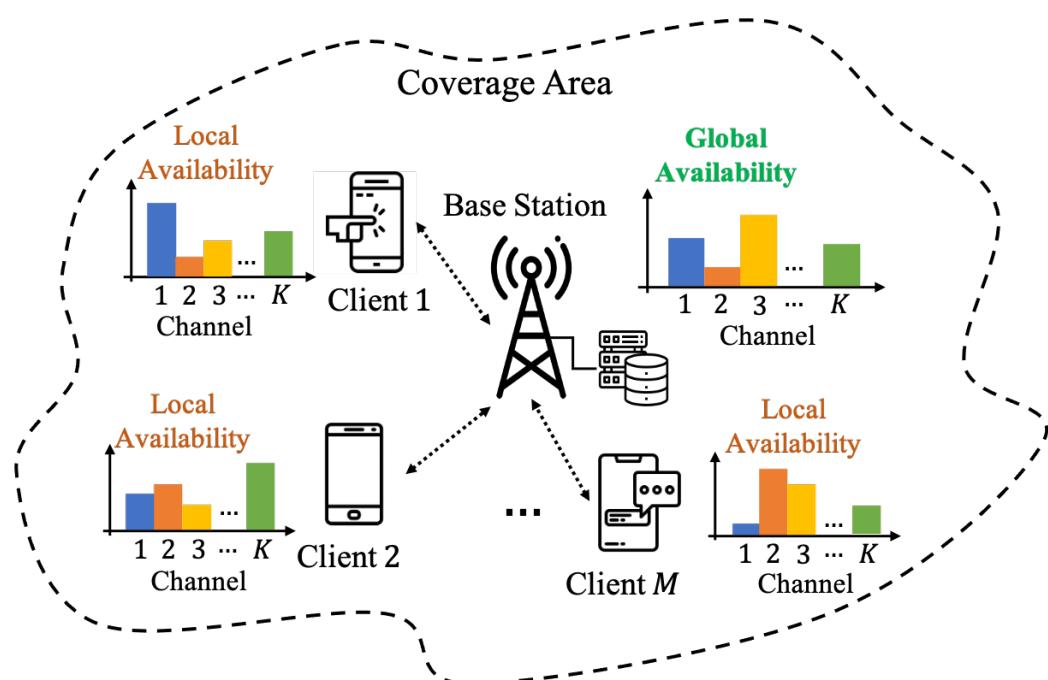


- BS wants to use a single “best” channel for the entire area
- Ground truth: a channel’s global availability is the **average over the intended coverage area**

$$p_k = \mathbb{E}_{x \sim u(\mathcal{D})} [p_k(x)] = \iint_{\mathcal{D}} \frac{1}{D} p_k(x) dx$$

# Network View

- Challenges for BS to select a globally optimal channel
  - Fixed location: network listen mode cannot solve the problem
  - Requires continuous sampling of the coverage area
  - Learning cost must be controlled (convergence speed)
- Solution: leverage UEs that are randomly distributed in the coverage area to sample  $p_k(x)$  at
  - Location  $x$
  - Channel  $k$

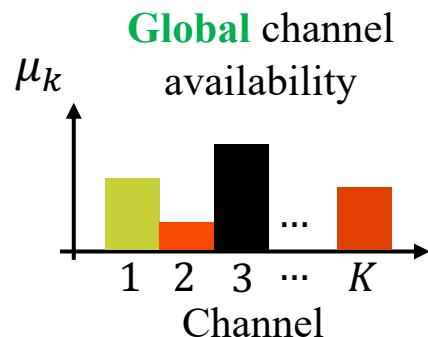


# Federated Bandits

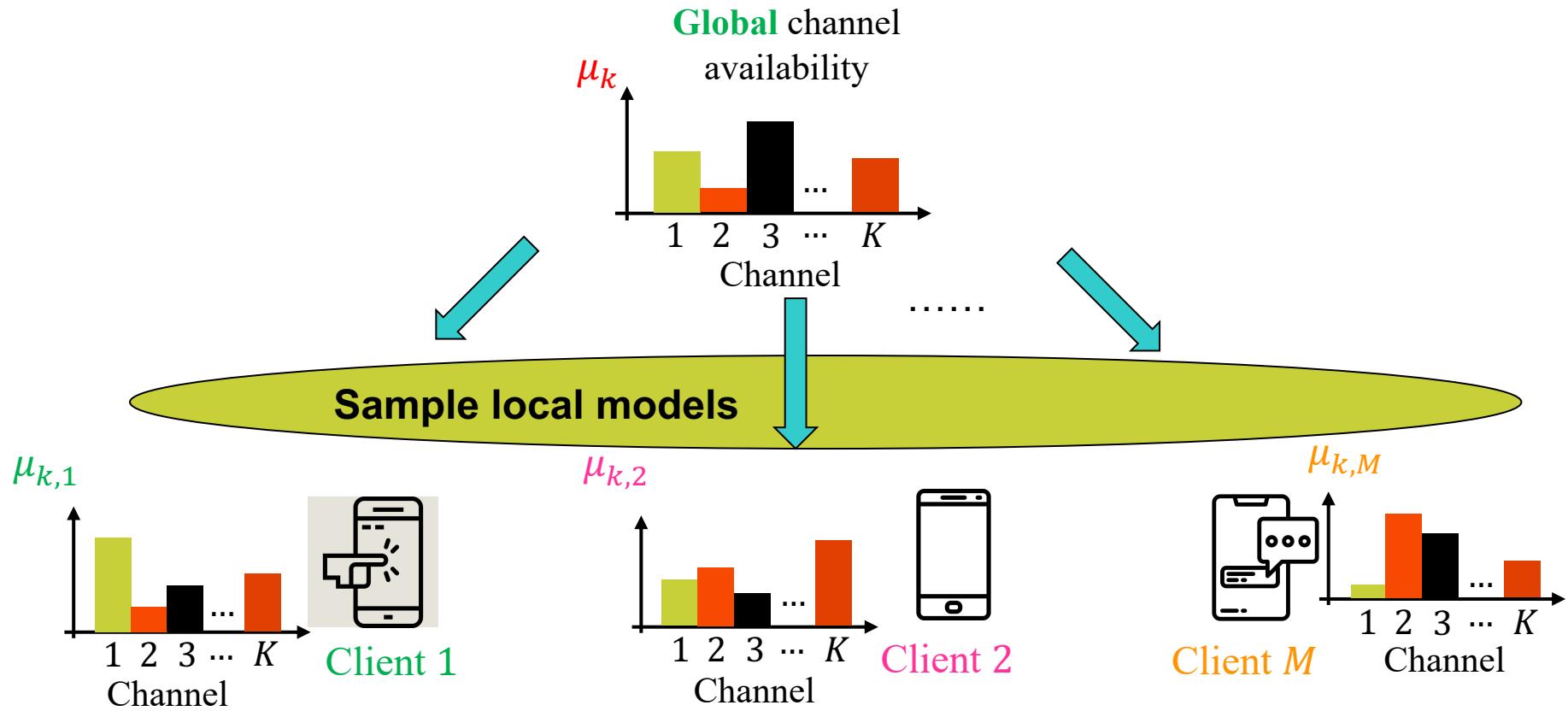
- Key observations
  - Server (BS) wants to learn the global model, but **lacks direct access**
  - Client (UE) can play the (local) bandit game, but only has local observation that is **a (noisy) part of the overall picture**
  - **Heterogeneous local models** for clients
  - No one can solve the problem by itself; **coordination** between server and clients becomes crucial (just like FL)
- Challenges
  - **Discrete** sampling to approach the continuous ground-truth
  - Often there are **misaligned objectives** between local bandit model and global bandit model (see previous figure)
- The general principles of FL still apply, but the underlying model is a bandit one, leading to **Federated MAB**

# General FMAB Framework

- How do we model the global-local relationship?
- Our first attempt views the local models as random variables, which are drawn IID from a latent global distribution
- This leads to a difficult approximate model
  
- Consider a global MAB model with  $K$  arms
- Arm  $k$  has mean reward  $\mu_k$ , for  $k = 1, \dots, K$



# Generating Approximate Model



- Global model is a **fixed** ground truth, but is **unknown**
- Each local model is a random “sampling” of this unknown global model, based on a latent distribution
  - $\mu_{k,m}$  is an IID sample from the latent distribution with mean  $\mu_k$

# Solving the Approximate Model

- Challenge 1: **average local models  $\neq$  global model**
  - Assuming **perfect knowledge** of local models, we have an average local model of

$$\hat{\mu}_k = \frac{1}{M} \sum_{m=1}^M \mu_{k,m}$$

- Such  $\hat{\mu}_k$  is not the same as the true  $\mu_k$
- Relax: the **optimal arms** in both models may not be the same

$$k_* = \operatorname{argmax} \mu_k$$
$$k'_* = \operatorname{argmax} \hat{\mu}_k$$

$$k_* \xrightarrow{?} k'_*$$

- How to ensure the optimal arms are the same?
  - Natural idea: involving more clients, i.e., **increase  $M$**

# Solving the Approximate Model

- Question: how many clients ( $M$ ) are needed?
  - Under-sampling: cannot guarantee matching optimal arms
  - Over-sampling: unnecessarily high communication cost
  - Need sufficient-but-not-excessive amount of clients
- Concentration inequalities tell us the exact order:

$$M_t = \Theta(\Delta^{-2} \log(KT))$$

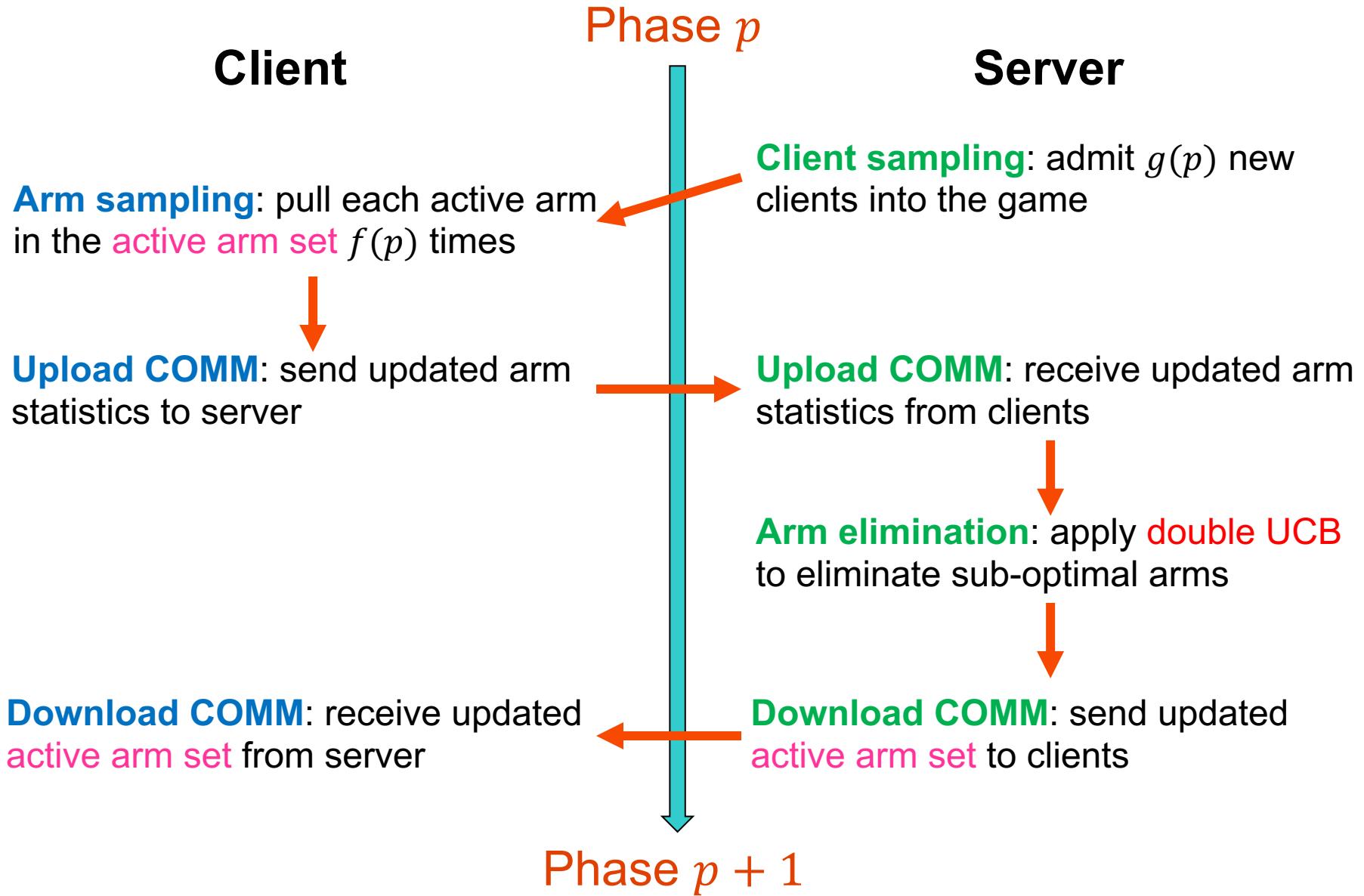
where  $\Delta$  is the suboptimality gap

- But we do not know  $\Delta$
- Challenge 2: balancing costly communication and necessary server-clients coordination
  - This is similar to FL where communication load should be carefully controlled

# Key Ideas

- Periodic communication, with period determined by the regret analysis to control its impact
- Gradually increase the number of clients after each communication round
  - Learning the suboptimality gap  $\Delta$  along the way
  - Ensures sufficient-but-not-excessive amount of clients
- Simultaneously handle client sampling and arm sampling, by developing a “double UCB” technique

# The Fed2-UCB Algorithm



# The Double UCB Principle

- Confidence bound at phase  $p$ :

$$B_{p,2} = \underbrace{\sqrt{6\sigma^2\eta_p \log(T)}}_{\text{arm sampling}} + \underbrace{\sqrt{6\sigma_c^2 \log(T)/M(p)}}_{\text{client sampling}}$$

- Regret analysis:

$$R(T) = O \left( \sum_{k \neq k_*} \frac{\kappa \log(T)}{\mu_* - \mu_k} + C \frac{\log(T)}{\kappa \Delta^2} \right)$$

Fed2-UCB achieves an **order-optimal** regret for the approximate model

# From Approximate to Exact Models

- A simplified version is the exact model: global model is equal to the exact average of local models
  - Fixed number of clients
- Fed2-UCB degenerates to Fed1-UCB
  - No client sampling issues; a much easier problem
  - Only consider uncertainty from arm sampling

$$B_{p,1} = \sqrt{\frac{6\sigma^2 \log(T)}{MF(p)}}$$

# **Equivalence of 3GPP Mobility Protocol and MAB, and Cascading-bandit-based Mobility Design**

# Mobility

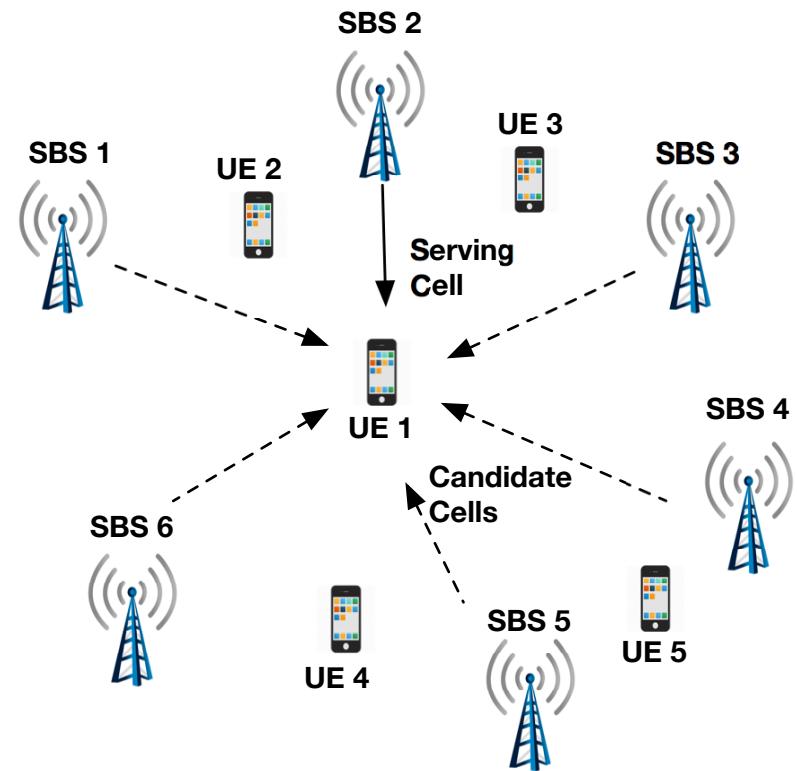
- Determines whether UE needs to switch serving BS, and if so which BS
- A long line of literature...
  - Game Theory
  - Optimization Theory
- Problem:
  - Myopic (or short-term) view of mobility
  - Assume known deployment information and static system dynamics

# Contributions

- Switch the view of mobility from one-shot to long-term
- Two main contributions:
  1. Establish an equivalence between 3GPP mobility protocols and bandit learning algorithms
    - Original protocol =  $\epsilon$ -greedy bandit algorithm
    - Hence we can leverage the theoretical results of MAB to analyze the performance of 3GPP protocol
  2. Inspired by this equivalence, we propose a learning-based approach to address FHO
    - Take into account *handover cost*, which forces the protocol to “slow down” handover rate

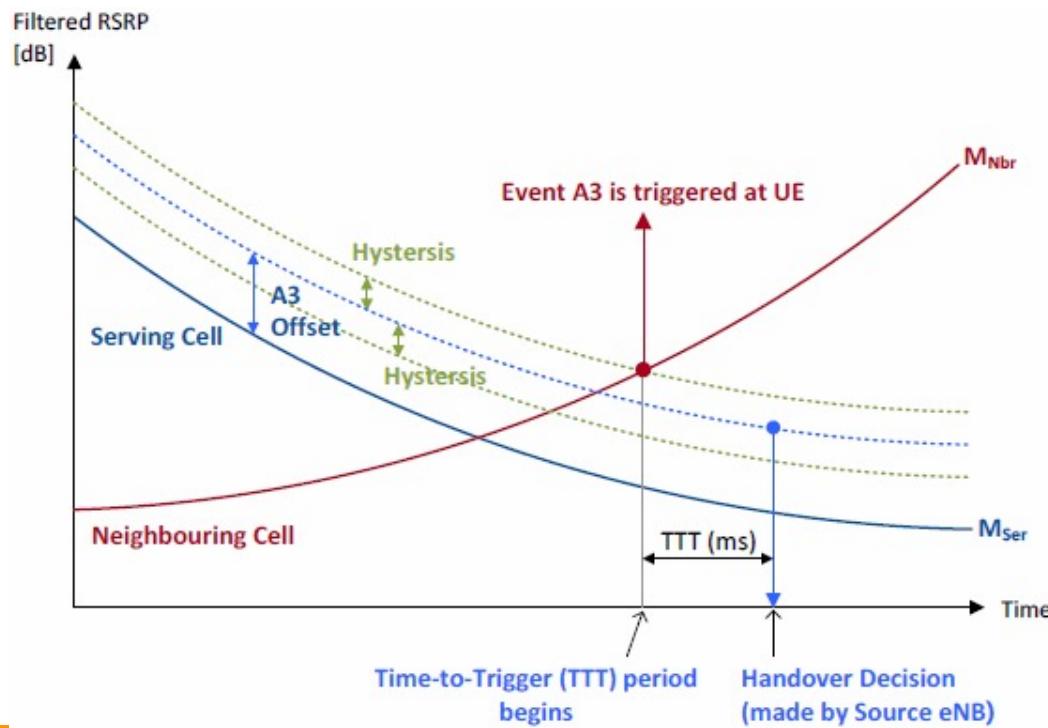
# System Model

- $N$  SBSs  $\mathcal{N}_{\text{SBS}} = \{1, \dots, N\}$
- $M$  UEs  $\mathcal{M}_{\text{UE}} = \{1, \dots, M\}$
- Indoor UDN scenario
- Open access
- Consider static or slow-moving UEs



# Handover Protocol

- Main Issue: FHO between multiple SBSs
  - 3GPP handover is mostly based on RF
$$\text{RSRP}_t + \xi_t > \text{RSRP}_s + \xi_s + \kappa_{\text{hist}}$$
  - Enhancement: using CRE to bias UE towards SBS
  - Such principle is myopic in nature, and may lead to FHO problems



# Statistical Modeling

- Mobility: lack of accurate information
  - Had the UE known which SBS is the best in the near future, it would have selected this SBS and stay on it
- Modeling :
  - Performance of interest for a SBS following some statistical distribution
  - Each "use" of the SBS is a sampling from this distribution
- UE is trying to get good "performance" while also "learn" the information of the  $N$  SBSs
  - Fast convergence is another important criterion

# Statistical Bandit

- Each SBS = an arm in bandit
- UE's performance = reward distribution
- UE is handed over to an SBS = sample the reward distribution of the corresponding arm
- Tradeoff between **exploitation** and **exploration**
  - UE is already on a "decent" SBS -> take advantage
  - Other SBS may be even better (long-term) -> user association decision

# Stochastic MAB and 3GPP Mobility

- Stochastic MAB algorithms :
  - greedy,  $\epsilon$ -greedy
  - Softmax (Boltzmann Exploration)
  - UCB series
- A natural idea: apply these methods to mobility...
- ...3GPP actually already does so, **unconsciously**

# Stochastic MAB and 3GPP Mobility

- **Result 1** :  $\epsilon$ -greedy algorithm = 3GPP protocol

---

**Algorithm 1:** The eGA handover algorithm.

---

**Input:** initial estimate  $\{\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_N\}$

**for**  $i = 1$  **to**  $T$  **do**

$a = \arg \max_n \hat{\mu}_n;$

Assign BS  $a$  with probability  $1 - \epsilon + \epsilon/N$ , and all other BSs  $\epsilon/N$ ;

Randomly select a BS  $a_t$  according to their distributions;

UE associates with BS  $a_t$ ;

UE receives reward  $R_{a_t, t}$ ;

$\hat{\mu}_{a_t} \leftarrow \hat{\mu}_{a_t} + \alpha (R_{a_t, t} - \hat{\mu}_{a_t});$

Store  $\{a_t, R_{a_t, t}\}$

**end**

**Output:**  $\{a_t, R_{a_t, t}\}_{t=1}^T$

---

# Stochastic MAB and 3GPP Mobility

- **Result 1** :  $\epsilon$ -greedy algorithm = 3GPP protocol

$\epsilon$ -greedy	3GPP
Initial prob. of each arm	UE RF measurement
$\epsilon$ -based prob. selec. of arm	Random RF variation s.t. serving cell's RSRP1 drops below th (A2), or RSRP2-RSRP1>th (A3)
Random reward	Throughput or other QoS/QoE

- Importance of this result :
  - MAB theory has rigorous theoretical proof of the sub-optimality and non-convergence of  $\epsilon$ -greedy
  - The equivalence can leverage such theoretical results to explain the problem of current 3GPP protocol

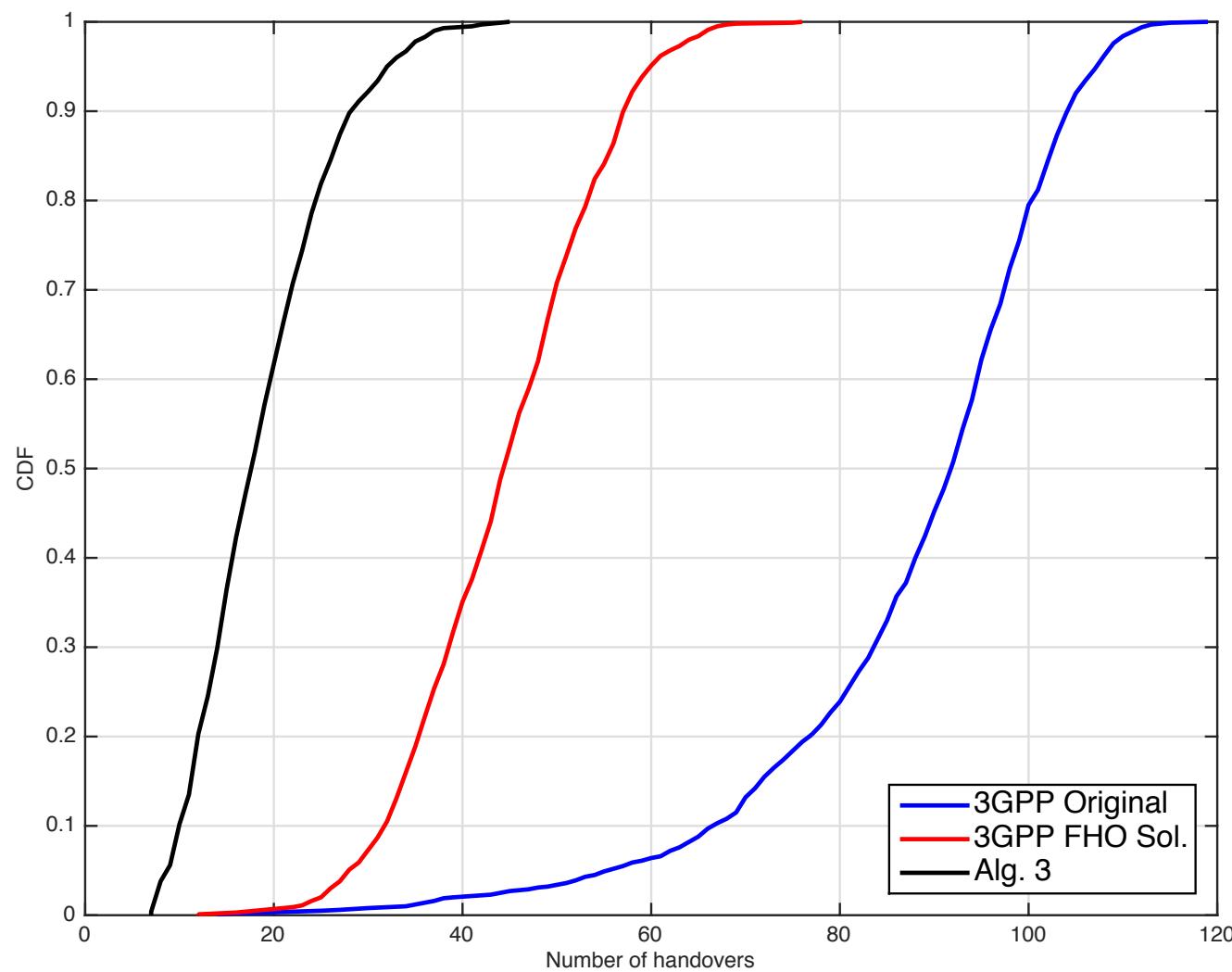
# How to Improve ?

- Idea 1 : Resort to better stochastic MAB alg.
  - UCB
- Idea 2 : To deal with FHO, the alg. needs to explicitly punish switching
  - Stochastic MAB + switching cost
  - New alg. introduces **blocking**, and **the size of block increases over time**
  - Intuitively, these ideas have been used in current FHO solutions in 3GPP, but there are important design differences

# Understanding Existing FHO Solutions

- Heuristic algorithm
  - Identification : declare FHO if an UE switches more than Y times over the past X minutes
  - Solution: adjust the UE handover parameters so that it is harder to switch (3GPP sticky biasing solution)
- This solution is very similar to the MAB-based method
  - Frequent switching at the beginning
  - Then slow down, until convergence
- MAB alg. with switching cost can be rigorously analyzed in theory

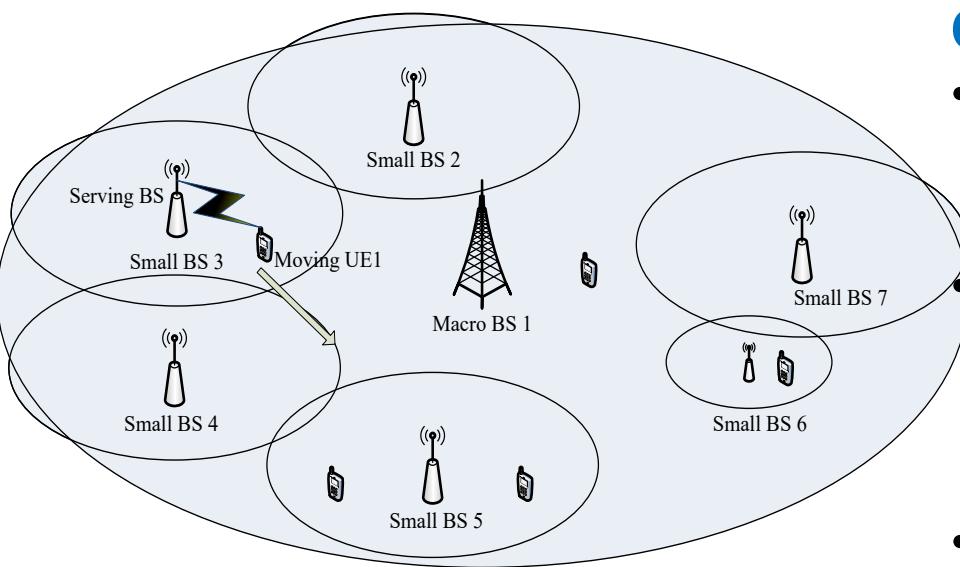
# Simulations



# **A Cascading Bandit Approach to Efficient Mobility Management**

# Introduction

- Dense deployment of low-power base stations (BS) is considered as the key solution to the dramatic growth of traffic demand in cellular networks.
- A bottleneck limits the performance of ultra-dense networks (UDN): user association and mobility management



## Challenges:

- Each user equipment (UE) has the possibility to connect to many BSs
- Channel state between each UE-BS pair is time-varying and even unpredictable.
- A small physical movement may lead to abrupt channel condition changes.

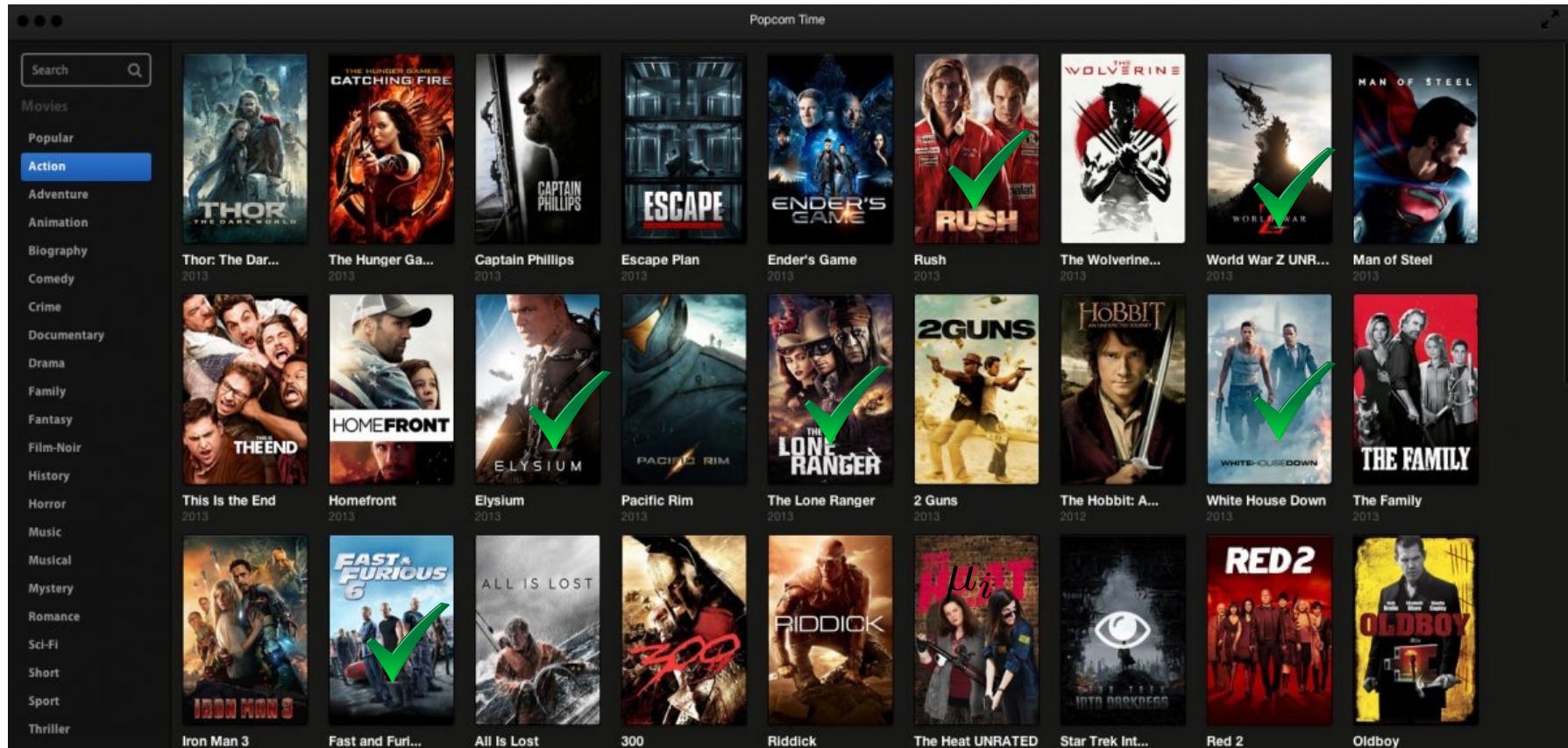
# Existing Approaches

- 3GPP: UE measures all candidate BSs and hands over to the best BS
  - Myopic, causes issues such as frequent handover and Ping-Pong
- Optimization-based: given various UE and BS information, maximize certain system utility
  - It relies on accurate system information, such as the channel statistics and utility function, which may not be practical
  - System dynamics can quickly render the solution sub-optimal, which requires to run the optimization frequently
- Our approach: cost-award cascading bandits based online learning
  - Learn system statistics based on historical measurements
  - Adjust decisions on-the-fly
  - Balance exploration-exploitation tradeoff to achieve low performance loss during learning

# Classical Cascading Bandit Model [Kveton et al., 2015]

Set of items (arms):  $[K] = \{1, 2, \dots, K\}$

Each item will be *clicked* with **unknown** probability



Learner's objective: recommend **a list of items** each time to  
*maximize the click-through rate + learn all arm stats*

# Classical Cascading Bandit Model [Kveton et al., 2015]

Recommend list:  
 $L$  out  $K$  items



Cascading Feedback:



$t = 2$



$t = 3$



⋮

# Classical Cascading Bandit Model [Kveton et al., 2015]

Non-additive reward:

Get reward one if **one** of the items on the list is *clicked*.

**Hard constraint on the size of the list:**

The **rank** of the items on the list **does NOT** affect the reward.

Impose cost for pulling arms

**Soft constraint on the size of the list:**

The **rank** of the items on the list **does** affect the **net reward**, i.e., **reward minus cost**.

# Optimal Offline Policy: $\theta_i, c_i$ are known beforehand

Ranking the arms according to  $\theta_i/c_i$

$$\frac{\theta_{1^*}}{c_{1^*}} \geq \frac{\theta_{2^*}}{c_{2^*}} \geq \dots \geq \frac{\theta_{L^*}}{c_{L^*}} > 1 > \frac{\theta_{(L+1)^*}}{c_{(L+1)^*}} \geq \dots \geq \frac{\theta_{K^*}}{c_{K^*}}.$$

“Good” arms

“Bad” arms

Optimal list of arms to pull:  $I^* = \{1^*, 2^*, \dots, L^*\}$   
The user *stops* once it observes an arm with  $X_{i,t} = 1$

## Remark

- The size of the optimal list is not fixed but determined by  $\theta_i/c_i$
- The actual list of pulled arms depends on the realization of  $X_{i,t}$
- Ranking of the arms affects the total cost incurred during examination
- The optimal policy achieves a balanced tradeoff between reward and cost

# Online Algorithm: $\theta_i, c_i$ are unknown beforehand

Objective: maximize the cumulative expected net reward

## Cost-aware Cascading UCB (CC-UCB)

- Estimate the UCB of  $\theta_i$  and LCB of  $c_i$  each time
- Perform the optimal offline policy using the UCB/LCB estimates

Upper bound on the cumulative regret

$$R(T) \leq \sum_{i \in [K] \setminus I^*} c_i \frac{16\alpha \log T}{\Delta_i^2} + O(1) \quad \text{where } \Delta_i := c_i - \theta_i$$

## Remark

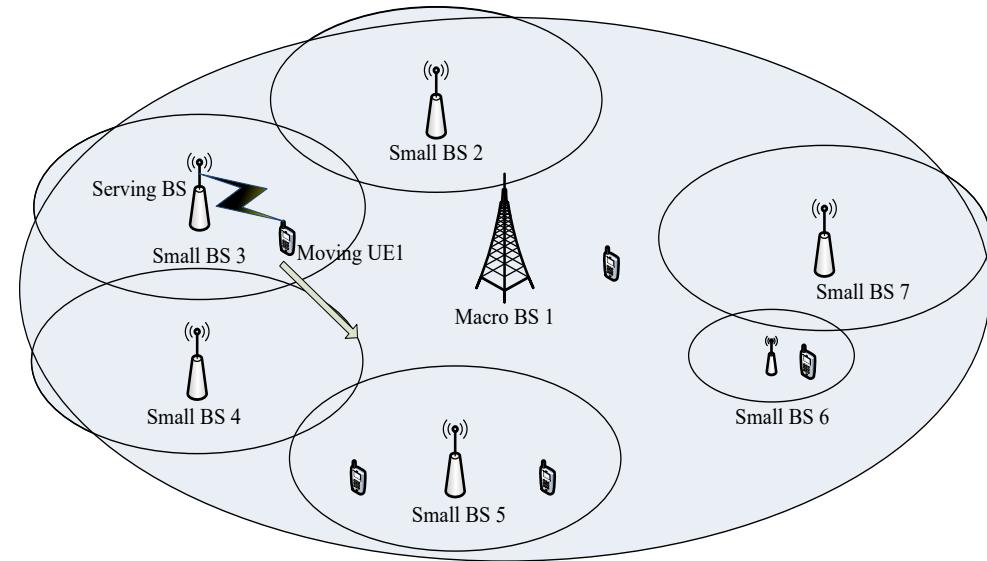
- The upper bound depends on  $\Delta_i$  instead of the gap between  $\theta_i/c_i$
- The regret caused by pulling good arms in a wrong order is bounded
- The regret is mainly due to pulling bad arms, which is determined by  $\theta_i, c_i$
- When  $c_i$  are known a priori, the bound can be reduced by a factor of 4

# Applying CCB to Mobility Management

- Mobility Management Entity (MME) maintains an **active neighbor Cell List (NCL)** for each BS
- When an HO is triggered, MME sends the list to the UE
- The UE examines these BSs **sequentially** until it chooses **the first suitable candidate BS** based on measurements.
- If none of the candidate BSs satisfies the HO condition, UE stays on the current serving BS.
- **Feedback** from UE to MME:
  - HO result
  - **Probing cost** (delay, energy, etc)

## Advantages:

- **Size of active NCL is small**
  - Measuring delay is limited
- **BSs on the list are ordered**
  - Find a suitable BS before checking many candidates



# Theoretical Performance of CCB-NCL

- With **perfect feedback**:
  - CCB-NCL inherits the **order-optimality** of CC-UCB
- With **delayed feedback**: if the delay is bounded by  $d$  steps

$$R_d(T) \leq \sum_{i \in [K] \setminus I^*} c_i \frac{16K\alpha \log(T - d)}{\Delta_i^2} + O(d).$$

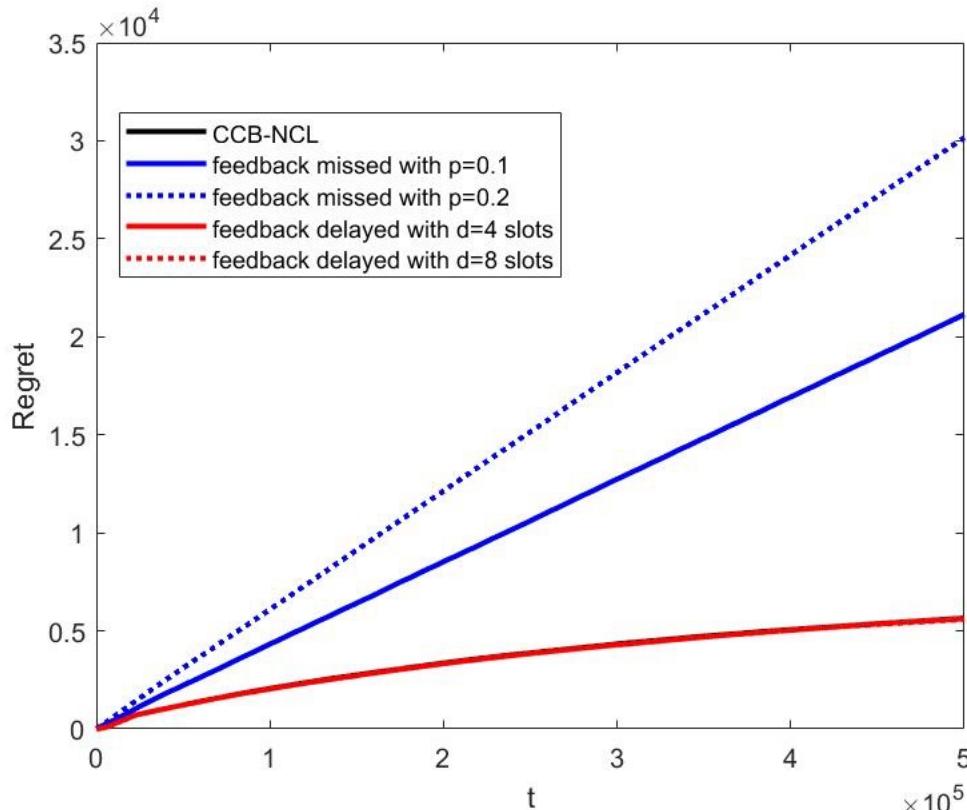


Feedback from  $[1, T-d]$



Missing feedback over  $[1, d]$

# Experiment Results: Regret



## Remark

- Simulation is based on the configuration #4b HetNet scenario in 3GPP spec.
- The regret under CCB-NCL grows sublinearly in time, which is consistent with the upper bound.
- Delayed feedback has very little impact to the overall regret.
- Missing feedback leads to much larger performance degradation.