UVIC
UNIVERSITY OF VICTORIA CANADA

PANDA
Parallel, Networking and Distributed Applications

Faster Convergence for Unknown-Game Bandits

Zhiming Huang, Jianping Pan (University of Victoria, BC, Canada)

Our Contribution in Brief

PANDA
Parallel, Networking and Distributed Applications

- Improve the **swap-regret** bound in terms of time dependency

$$\text{from } O(T^{\frac{1}{2}}) \text{ to } \tilde{O}(T^{\frac{1}{4}}),$$

if all agents in a game all self-play an **optimistic/adaptive** bandit algorithm

Our Contribution in Brief

PANDA
Parallel, Networking and Distributed Applications

- Improve the **swap-regret** bound in terms of time dependency

$$\text{from } O(T^{\frac{1}{2}}) \text{ to } \tilde{O}(T^{\frac{1}{4}}),$$

if all agents in a game all play an **optimistic/adaptive bandit** algorithm.

- Why minimizing swap regret is important?
- How we improve the bound?

Game Theory

Normal-form Games

PANDA
Parallel, Networking and Distributed Applications

- Many networking problems can be formulated as games
    - A number of agents
    - A set of actions for each agent
    - Reward/Payoff matrix

Game Theory
Normal-form Games

PANDA
Parallel, Networking and Distributed Applications

- Many networking problems can be formulated as games
  - A number of agents
  - A set of actions for each agent
  - Reward/Payoff matrix

		Client 2	
		LTE	WiFi
Client 1	LTE	$(17.5, 17.5)$	$(35, 24)$
	WiFi	$(48, 35)$	$(16, 16)$

E.g., Network selection game

Game Theory

Correlated Equilibrium

- A joint distribution that no one is willing to deviate unilaterally

Game Theory

Correlated Equilibrium

- A joint distribution that no one is willing to deviate unilaterally

		Client 2	
		LTE	WiFi
Client 1	LTE	$(17.5, 17.5)$	$(35, 24)$
	WiFi	$(48, 35)$	$(16, 16)$

E.g., Network selection game

0	0
1	0

Action distribution for the game

Game Theory

Issues in Equilibrium Computation

PANDA
Parallel, Networking and Distributed Applications

- The computation complexity increases exponentially with the scale of the game (e.g., number of players and actions)
- The reward/payoff matrix and number of players may not be known a priori.
- There is no central controller

Game Theory

Issues in Equilibrium Computation

- The computation complexity increases exponentially with the scale of the game (e.g., number of players and actions)
- The reward/payoff matrix and number of players may not be known a priori.
- There is no central controller
- Any distributed solutions?

Game Theory

Issues in Equilibrium Computation

- The computation complexity increases exponentially with the scale of the game (e.g., number of players and actions)
- The reward/payoff matrix and number of players may not be known a priori.
- There is no central controller
- Any distributed solutions?
  - Independ of the game size

Game Theory

Issues in Equilibrium Computation

PANDA
Parallel, Networking and Distributed Applications

- The computation complexity increases exponentially with the scale of the game (e.g., number of players and actions)
- The reward/payoff matrix and number of players may not be known a priori.
- There is no central controller
- Any distributed solutions?
  - Independ of the game size
  - Payoff matrix may not need to be known a priori

Game Theory

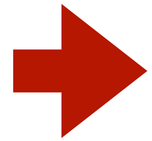
Issues in Equilibrium Computation

- The computation complexity increases exponentially with the scale of the game (e.g., number of players and actions)
- The reward/payoff matrix and number of players may not be known a priori.
- There is no central controller
- Any distributed solutions?
  - Independ of the game size
  - Payoff matrix may not need to be known a priori
  - Be efficient in terms of computational costs and communication costs

Game Theory

Issues in Equilibrium Computation

- The computation complexity increases exponentially with the scale of the game (e.g., number of players and actions)
- The reward/payoff matrix and number of players may not be known a priori.
- There is no central controller
- Any distributed solutions?
  - Independend of the game size
  - Payoff matrix may not need to be known a priori
  - Be efficient in terms of computational costs and communication costs



Online Learning

Online Learning

A Simple Introduction

PANDA
Parallel, Networking and Distributed Applications

- Online Learning/optimization Model:
  - At each round $t = 1, \ldots, T$:
    1. the learner first picks a point $w_t \in \Omega$;
    2. the environment then picks a loss function $f_t : \Omega \to \mathbb{R}$
    3. the learner suffers loss $f_t(w_t)$, and observes some information about f_t.

Online Learning
A Simple Introduction

PANDA
Parallel, Networking and Distributed Applications

- Online Learning/optimization Model:
  - At each round $t = 1, \ldots, T$:
    1. the learner first picks a point $w_t \in \Omega$;
    2. the environment then picks a convex loss function $f_t : \Omega \to \mathbb{R}$
    3. the learner suffers loss $f_t(w_t)$, and observes some information about f_t.

the expert problem	$\Delta(N)$	$f_t(p) = \langle p, \ell_t \rangle$	ℓ_t, thus entire f_t	Full-information Feedback
multi-armed bandits	$\Delta(K)$	$f_t(p) = \langle p, \ell_t \rangle$	only one entry of ℓ_t	Bandit Feedback

- Regret notions
  - External Regret: compare to *N* competitors that always play one fixed acton over time

$$R_n^{\text{ext}}(T) := \max_{w \in [N]} \sum_{t=1}^{T} f_t\left(w_t\right) - \sum_{t=1}^{T} f_t\left(w\right)$$

Learning Alg.

A competitor always a fixed action w

- Regret notions
  - External Regret: compare to N competitors that always play one fixed acton over time

$$R_n^{\text{ext}}(T) := \max_{w \in \Delta(N)} \sum_{t=1}^{T} f_t\left(w_t\right) - \sum_{t=1}^{T} f_t\left(w\right)$$

  - Swap Regret: compare to N^N competitors that play actions defined by $F : [N] \to [N]$

$$R_n^{\text{swa}}(T) := \max_{F \in \mathcal{F}} \sum_{t=1}^{T} f_t\left(w_t\right) - \sum_{t=1}^{T} f_t\left(F(w_t)\right)$$

Learning Alg.

A competitor always actions $F(w_t)$

Online Learning

Connection with Game Theory

PANDA
Parallel, Networking and Distributed Applications

With **full-information feedback**, people prove:

- Every agent plays an **external-regret-minimization** algorithm
  - Time-averaged plays —> NE in two-player zero-sum game

NE: Nash Equilibrium

Online Learning

Connection with Game Theory

PANDA
Parallel, Networking and Distributed Applications

With **full-information feedback**, people prove:

- Every agent plays an **external-regret-minimization** algorithm
  - Time-averaged plays —> NE in two-player zero-sum game
- Every agent plays a **swap-regret-minimization** algorithm
  - Time-averaged plays -> CE in multi-player games

NE: Nash Equilibrium

CE: Correlated Equilibrium

Faster Convergence

Known Results on the Regret Time Dependency

- Full-information Feedback
  - External/Swap regret: $O(\ln T)$ [Anagnostides et al. 2022, Soleymani et al. 2025]

Faster Convergence

Known Results on the Regret Time Dependency

- Full-information Feedback
  - External/Swap regret: $O(\ln T)$ [Anagnostides et al. 2022, Soleymani et al. 2025]
- Bandit feedback
  - External regret: $O(T^{1/4})$ in two-player settings [Wei et al. 2018]
  - Swap regret: $\tilde{O}(T^{1/4})$ in multi-player settings [This Work]

Faster Convergence

High-level Idea on Techniques

PANDA
Parallel, Networking and Distributed Applications

- If each agent n knows all other agents are playing swap-regret-minimizing algorithms:

Faster Convergence

High-level Idea on Techniques

- If each agent n knows all other agents are playing swap-regret-minimizing algorithms:
  - A new piece of information can be utilized

Faster Convergence

High-level Idea on Techniques

- If each agent n knows all other agents are playing swap-regret-minimizing algorithms:
  - A new piece of information can be utilized
  - Optimistic follow-the-regularized-leader (OFTRL) -> **add an estimate of reward/loss in round t** to the regular FTRL
    - $$p_t = \arg\min_{p \in \Delta(\mathcal{A}_n)} \left\langle \sum_{s=1}^{t-1} \hat{l}_s + m_t, p \right\rangle + \frac{1}{\eta}\Psi(p)$$

Faster Convergence

More Details

- To minimize swap-regret:
  - Blum-Mansour Reduction Techniques
    - For A_n arms, calling A_n external-regret-minimizng algorithms as subroutines.
    - Each subroutine outputs a vector $q_a^t, \forall a \in \mathscr{A}_n$.
    - Solving a fixed-point equation to calculate the arm selection prob.

$$\left(p_n^t\right)^\top = \left(p_n^t\right)^\top Q_n^t$$

Each row of Q_n^t is $(q_a^t)^\top$ output by each subroutine

Faster Convergence

More Details - Cont.

PANDA
Parallel, Networking and Distributed Applications

- Each subroutine adopts an optimistic Follow-The-Regularized-Leader (FTRL) algorithm:

$$q_a^t := \arg\max_{q \in \Delta(\mathscr{A}_n)} \left\{ \eta \left\langle q, p_n^{t-1}(a) m_n^t + \sum_{s=1}^{t-1} p_n^s \hat{u}_n^s \right\rangle + \sum_{a' \in \mathscr{A}_n} \ln\left(q\left(a'\right)\right) \right\},$$

- We set m_n^t to be the vector of the latest received reward for each arm
- Set the reward estimator to be (for possibility of analysis)

$$\hat{u}_n^t(a') = m_n^t(a') - \frac{\left(m_n^t(a') - u_n^t(a')\right) \mathbf{1}\left[a_n^t = a'\right]}{p_n^t(a')}.$$

Faster Convergence

Theoretical Results

PANDA
Parallel, Networking and Distributed Applications

- Swap regret for each agent n:

$$R_n^{\mathrm{swa}}(T) \leq \frac{2\left(A_n\right)^2 \ln T}{\eta} + 2\eta \sum_{t=1}^{T} \sum_{m \in [N]} \left\| p_m^t - p_m^{t-1} \right\|_1 - \frac{1}{1024 A_n \eta} \sum_{t=1}^{T} \left\| p_n^t - p_n^{t-1} \right\|_1^2 .$$

Faster Convergence

Theoretical Results

PANDA
Parallel, Networking and Distributed Applications

- Swap regret for each agent n:

$$R_n^{\text{swa}}(T) \leq \frac{2\left(A_n\right)^2 \ln T}{\eta} + 2\eta \sum_{t=1}^{T} \sum_{m\in[N]} \left\| p_m^t - p_m^{t-1} \right\|_1 - \frac{1}{1024 A_n \eta} \sum_{t=1}^{T} \left\| p_n^t - p_n^{t-1} \right\|_1^2 .$$

- Sum for all agents:

$$\sum_{n\in[N]} R_n^{\text{swa}}(T) \leq \frac{2N\left(A_{\max}\right)^2 \ln T}{\eta} + 2\eta N \sum_{t=1}^{T} \sum_{n\in[N]} \left\| p_n^t - p_n^{t-1} \right\|_1 - \frac{1}{1024 A_{\max} \eta} \sum_{t=1}^{T} \sum_{n\in[N]} \left\| p_n^t - p_n^{t-1} \right\|_1^2$$

$$\leq \frac{2N\left(A_{\max}\right)^2 \ln T}{\eta} + 1024\eta^3 A_{\max} N^3 T .$$

Faster Convergence

Theoretical Results

- Swap regret for each agent n:

$$R_n^{\text{swa}}(T) \leq \frac{2\left(A_n\right)^2 \ln T}{\eta} + 2\eta \sum_{t=1}^{T} \sum_{m\in[N]} \left\| p_m^t - p_m^{t-1} \right\|_1 - \frac{1}{1024 A_n \eta} \sum_{t=1}^{T} \left\| p_n^t - p_n^{t-1} \right\|_1^2 .$$

- Sum for all agents:

$$\sum_{n\in[N]} R_n^{\text{swa}}(T) \leq \frac{2N\left(A_{\max}\right)^2 \ln T}{\eta} + 2\eta N \sum_{t=1}^{T} \sum_{n\in[N]} \left\| p_n^t - p_n^{t-1} \right\|_1 - \frac{1}{1024 A_{\max} \eta} \sum_{t=1}^{T} \sum_{n\in[N]} \left\| p_n^t - p_n^{t-1} \right\|_1^2$$

$$\leq \frac{2N\left(A_{\max}\right)^2 \ln T}{\eta} + 1024\eta^3 A_{\max} N^3 T .$$

When $\eta = O\left((\ln T/T)^{\frac{1}{4}} N^{-\frac{1}{2}}\right)$, we have $\sum_{n\in[N]} R_n^{\text{swa}}(T) \leq \tilde{O}(T^{\frac{1}{4}})$

Faster Convergence

Experiment Results

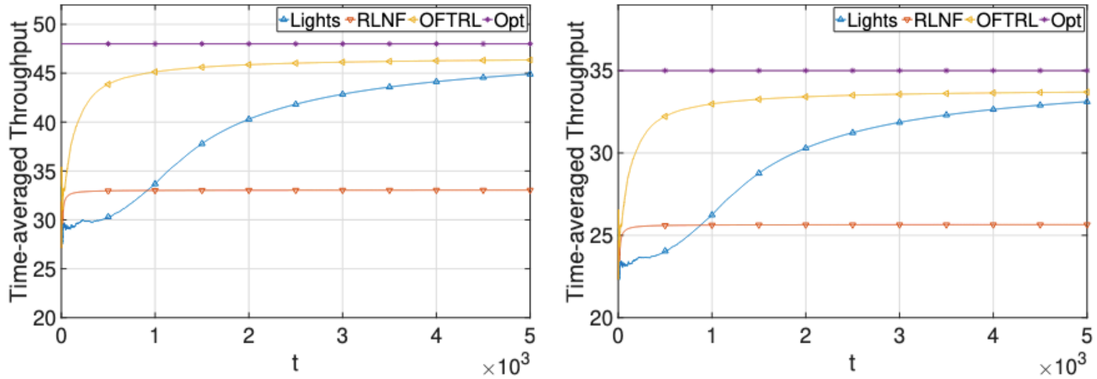
- Take the network selection game as an example

		Client 2	
		LTE	WiFi
Client 1	LTE	$(17.5, 17.5)$	$(35, 24)$
	WiFi	$(48, 35)$	$(16, 16)$

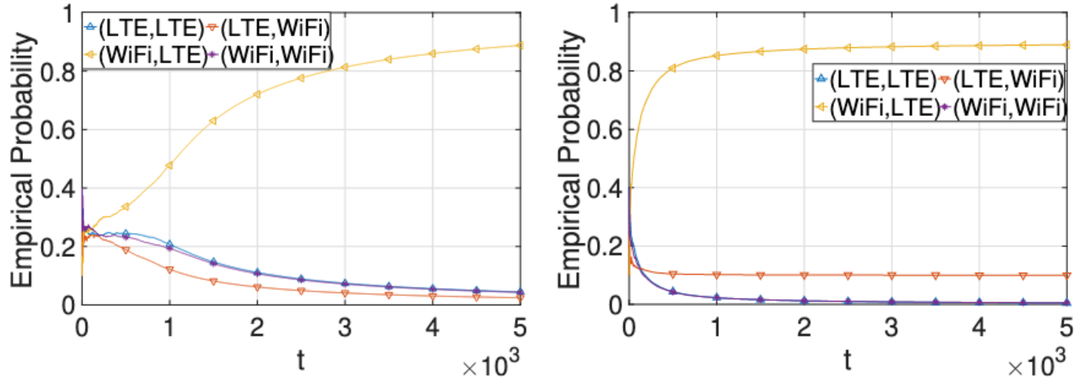
E.g., Network selection game

0	0
1	0

One possible CE



(a) Client 1 (b) Client 2



(a) Lights (b) OFTRL-LogBar-Bandit

Future Research

PANDA
Parallel, Networking and Distributed Applications

- Is it possible to further improve the time dependency?
- Infinite action space?

Future Research

PANDA
Parallel, Networking and Distributed Applications

- Is it possible to further improve the time dependency?
- Infinite action space?

Thank you very much!