Closing the Computational-Statistical Gap in Best Arm Identification for Combinatorial Semi-bandits

Ruo-Chun Tzeng[1], Po-An Wang[1], Alexandre Proutiere[1], and Chi-Jen Lu[2]

Conference on Neural Information Processing Systems, 2023

[1]EECS, KTH Royal Institue of Technology, Sweden
[2]Institute of Information Science, Academia Sinica, Taiwan

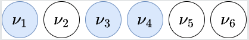
Computational-Statistical Gap in BAI for Combinatorial Semi-bandits

Problem: combinatorial BAI with fixed confidence

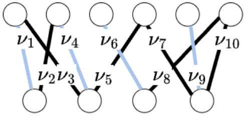
Input: K arms $(\nu_k)_{k\in[K]}$ with mean $\boldsymbol{\mu} \in \mathbb{R}^K$ and $\mathcal{X} \subseteq \{0,1\}^K$

Example: Gaussian reward $\nu_k = \mathcal{N}(\mu_k, 1), \forall k \in [K]$

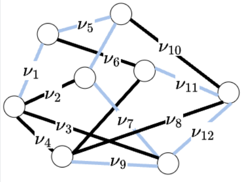
m-sets



matchings



spanning trees



Goal: Identify $\boldsymbol{i}^\star(\boldsymbol{\mu}) \in \text{argmax}_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{\mu} \rangle$ with $\boldsymbol{\mu}$ unknown initially.

Assumptions: (i) $\boldsymbol{i}^\star(\boldsymbol{\mu})$ is unique; (ii) for any $\boldsymbol{v} \in \mathbb{R}^K$, a best action $\boldsymbol{i}^\star(\boldsymbol{v})$ can be found in polynomial time.

Problem: combinatorial BAI with fixed confidence

Input: K arms $(\nu_k)_{k\in[K]}$ with mean $\boldsymbol{\mu} \in \mathbb{R}^K$ and $\mathcal{X} \subseteq \{0,1\}^K$

Rule: At each round t, the learner

- pulls $\boldsymbol{x}(t) \in \mathcal{X}$ and observes $y_k(t) \sim \nu_k$ iff $x_k(t) = 1$
- decides whether to stop and outputs $\hat{\boldsymbol{\imath}} \in \mathcal{X}$
- let τ be the round it stops

Goal: Design a δ-*PAC* algorithm s.t. $\boldsymbol{i}^\star(\boldsymbol{\mu}) \in \operatorname{argmax}_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{\mu} \rangle$ is identified with $\mathbb{P}_{\boldsymbol{\mu}}[\hat{\boldsymbol{\imath}} = \boldsymbol{i}^\star(\boldsymbol{\mu})] \geq 1 - \delta$ and $\mathbb{P}_{\boldsymbol{\mu}}[\tau < \infty] = 1$

(i) *statistically optimal:* information-theoretically *minimal* $\mathbb{E}_{\boldsymbol{\mu}}[\tau]$

(ii) *computationally efficient:* running time *polynomial* in K

Existing δ-PAC algorithms: only (i) or only (ii)

(Open Question) Possible to design a δ-PAC algorithm achieving both?

Challenge in solving the lowerbound problem

Instance-specific sample complexity lower bound [GK16]

For any δ-PAC algorithm[1], $\mathbb{E}_{\boldsymbol{\mu}}[\tau] \geq T^\star(\boldsymbol{\mu})\mathrm{kl}(\delta, 1-\delta)$, where

$$T^\star(\boldsymbol{\mu})^{-1} = \sup_{\boldsymbol{\omega} \in \Sigma} F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \text{ with } F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \inf_{\boldsymbol{\lambda} \in \mathrm{Alt}(\boldsymbol{\mu})} \sum_{k=1}^{K} \frac{\omega_k (\mu_k - \lambda_k)^2}{2}$$

- $\Sigma = \{\sum_{\boldsymbol{x} \in \mathcal{X}} w_{\boldsymbol{x}} \boldsymbol{x} : \boldsymbol{w} \in \Sigma_{|\mathcal{X}|}\}$: all possible arm allocations
- $\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^K : |\boldsymbol{i}^\star(\boldsymbol{\lambda})| = 1\}$: all possible parameters
- $\mathrm{Alt}(\boldsymbol{\mu}) = \{\boldsymbol{\lambda} \in \Lambda : \boldsymbol{i}^\star(\boldsymbol{\lambda}) \neq \boldsymbol{i}^\star(\boldsymbol{\mu})\}$: confusing parameters

[1] Here we assume the arm-k reward distribution is $\nu_k = \mathcal{N}(\mu_k, 1)$

Challenge in solving the lowerbound problem

Instance-specific sample complexity lower bound [GK16]

For any δ-PAC algorithm[1], $\mathbb{E}_{\boldsymbol{\mu}}[\tau] \geq T^\star(\boldsymbol{\mu})\mathrm{kl}(\delta, 1-\delta)$, where

$$T^\star(\boldsymbol{\mu})^{-1} = \sup_{\boldsymbol{\omega}\in\Sigma} F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \text{ with } F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \inf_{\boldsymbol{\lambda}\in\mathrm{Alt}(\boldsymbol{\mu})} \sum_{k=1}^{K} \frac{\omega_k(\mu_k - \lambda_k)^2}{2}$$

- $\Sigma = \{\sum_{\boldsymbol{x}\in\mathcal{X}} w_{\boldsymbol{x}}\boldsymbol{x} : \boldsymbol{w} \in \Sigma_{|\mathcal{X}|}\}$: all possible arm allocations
- $\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^K : |\boldsymbol{i}^\star(\boldsymbol{\lambda})| = 1\}$: all possible parameters
- $\mathrm{Alt}(\boldsymbol{\mu}) = \{\boldsymbol{\lambda} \in \Lambda : \boldsymbol{i}^\star(\boldsymbol{\lambda}) \neq \boldsymbol{i}^\star(\boldsymbol{\mu})\}$: confusing parameters

Each sampling strategy is represented by its arm allocation $\boldsymbol{\omega} \in \Sigma$.

[1] Here we assume the arm-k reward distribution is $\nu_k = \mathcal{N}(\mu_k, 1)$

Challenge in solving the lowerbound problem

Instance-specific sample complexity lower bound [GK16]

For any δ-PAC algorithm[1], $\mathbb{E}_{\boldsymbol{\mu}}[\tau] \geq T^\star(\boldsymbol{\mu})\mathrm{kl}(\delta, 1-\delta)$, where

$$T^\star(\boldsymbol{\mu})^{-1} = \sup_{\boldsymbol{\omega}\in\Sigma} F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \text{ with } F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \inf_{\boldsymbol{\lambda}\in\mathrm{Alt}(\boldsymbol{\mu})} \sum_{k=1}^{K} \frac{\omega_k(\mu_k - \lambda_k)^2}{2}$$

- $\Sigma = \{\sum_{\boldsymbol{x}\in\mathcal{X}} w_{\boldsymbol{x}}\boldsymbol{x} : \boldsymbol{w} \in \Sigma_{|\mathcal{X}|}\}$: all possible arm allocations
- $\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^K : |\boldsymbol{i}^\star(\boldsymbol{\lambda})| = 1\}$: all possible parameters
- $\mathrm{Alt}(\boldsymbol{\mu}) = \{\boldsymbol{\lambda} \in \Lambda : \boldsymbol{i}^\star(\boldsymbol{\lambda}) \neq \boldsymbol{i}^\star(\boldsymbol{\mu})\}$: confusing parameters

The inner optimization measures the distance from $\boldsymbol{\mu}$ to the *most confusing parameter* (MCP) with the best action different from $\boldsymbol{i}^\star(\boldsymbol{\mu})$.
$\Rightarrow$ The best sampling strategy has the largest distance to the MCP.

[1] Here we assume the arm-k reward distribution is $\nu_k = \mathcal{N}(\mu_k, 1)$

Challenge in solving the lowerbound problem

Instance-specific sample complexity lower bound [GK16]

For any δ-PAC algorithm[1], $\mathbb{E}_{\boldsymbol{\mu}}[\tau] \geq T^\star(\boldsymbol{\mu})\mathrm{kl}(\delta, 1-\delta)$, where

$$T^\star(\boldsymbol{\mu})^{-1} = \sup_{\boldsymbol{\omega}\in\Sigma} F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \text{ with } F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \inf_{\boldsymbol{\lambda}\in\mathrm{Alt}(\boldsymbol{\mu})} \sum_{k=1}^{K} \frac{\omega_k(\mu_k - \lambda_k)^2}{2}$$

A standard approach [GK16] achieving asymptotic optimality consists of:

- Chernoff stopping rule: $\tau = \inf\{t : tF_{\hat{\boldsymbol{\mu}}(t)}(\hat{\boldsymbol{\omega}}(t)) > \ln(\frac{t}{\delta}) + o(1)\}$
- Pull arms according to $\boldsymbol{\omega}^\star(\hat{\boldsymbol{\mu}}(t))) = \mathrm{argmax}_{\boldsymbol{\omega}\in\Sigma} F_{\hat{\boldsymbol{\mu}}(t)}(\boldsymbol{\omega})$

Difficulty in determining the most confusing parameter (MCP)

The domain $\mathrm{Alt}(\boldsymbol{\mu}) = \{\boldsymbol{\lambda} \in \Lambda : \boldsymbol{i}^\star(\boldsymbol{\lambda}) \neq \boldsymbol{i}^\star(\boldsymbol{\mu})\}$ of $F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$

$\Rightarrow$ The naive approach as to solve $|\mathcal{X}| - 1$ many convex programs by partitioning $\mathrm{Alt}(\boldsymbol{\mu}) = \cup_{\boldsymbol{x}\neq \boldsymbol{i}^\star(\boldsymbol{\mu})}\{\boldsymbol{\lambda} \in \Lambda : \langle \boldsymbol{i}^\star(\boldsymbol{\mu}) - \boldsymbol{x}, \boldsymbol{\lambda}\rangle < 0\}$.

[1] Here we assume the arm-k reward distribution is $\nu_k = \mathcal{N}(\mu_k, 1)$

Computational inefficiency in prior optimal algorithms

- `Track-and-Stop` [GK16] at each round t has to solve

$$\boldsymbol{\omega}^{\star}(\hat{\boldsymbol{\mu}}(t)) \in \operatorname*{argmax}_{\boldsymbol{\omega}\in\Sigma} F_{\hat{\boldsymbol{\mu}}(t)}(\boldsymbol{\omega}),$$ (computationally expensive)

- `FWS` [WTP21] has to compute $f_{\boldsymbol{x}}(\boldsymbol{w}(t), \hat{\boldsymbol{\mu}}(t))$ of each $\boldsymbol{x} \neq \boldsymbol{i}^{\star}(\hat{\boldsymbol{\mu}}(t))$ to deal with the *nonsmoothness* of $F_{\hat{\boldsymbol{\mu}}(t)}$
- `CombGame` [JMKK21] proposed a `MCP`-oracle efficient algorithm, but no efficient `MCP` oracle exists prior to our work

Our Perturbed Frank-Wolfe Sampling (`P-FWS`)

- `P-FWS` deals with $|\mathcal{X}| \leq 2^K$ actions by *stochastic smoothing*
- All `P-FWS` needs is the linear maximization $\boldsymbol{i}^{\star}$ oracle

Our MCP Algorithm: a no-regret algorithm for solving $F_\mu(\omega)$

A crucial structural observation about $F_\mu(\omega)$

Define $f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \inf_{\boldsymbol{\lambda} \in \mathbb{R}^K : \langle \boldsymbol{i}^\star(\boldsymbol{\mu}) - \boldsymbol{x}, \boldsymbol{\lambda} \rangle < 0} \sum_{k=1}^{K} \frac{\omega_k (\mu_k - \lambda_k)^2}{2}$.

Property of $f_{\boldsymbol{x}}$ and its Lagrangian dual $g_{\omega,\mu}$

$f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \max_{\alpha \geq 0} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha)$ (known by [CGL16])

$g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha)$ is linear in $\boldsymbol{x}$ and concave in α (our observation)

$$F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \min_{\boldsymbol{x} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})} f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \min_{\boldsymbol{x} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})} \max_{\alpha \geq 0} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha) \quad (1)$$

A crucial structural observation about $F_\mu(\omega)$

Define $f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \inf_{\boldsymbol{\lambda} \in \mathbb{R}^K : \langle \boldsymbol{i}^\star(\boldsymbol{\mu}) - \boldsymbol{x}, \boldsymbol{\lambda} \rangle < 0} \sum_{k=1}^K \frac{\omega_k(\mu_k - \lambda_k)^2}{2}$.

Property of $f_{\boldsymbol{x}}$ and its Lagrangian dual $g_{\omega,\mu}$

$f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \max_{\alpha \geq 0} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha)$ (known by [CGL16])

$g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha)$ is linear in $\boldsymbol{x}$ and concave in α (our observation)

$$F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \min_{\boldsymbol{x} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})} f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \min_{\boldsymbol{x} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})} \max_{\alpha \geq 0} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha) \tag{1}$$

Requirement: Not only to estimate $F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$ but also the *equilibrium action* $\boldsymbol{x}_e$ s.t. $F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \max_{\alpha \geq 0} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}_e, \alpha)$.

- $\boldsymbol{x}_e$ is needed to solve $\max_{\boldsymbol{\omega} \in \Sigma} F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$ by the first-order methods
- Existing results [DP19, LNP+21, APFS22, AAS+23] on last-iterate convergence are not applicable as they all consider convex domains

Our proposed MCP algorithm

Algorithm 1: (ϵ, θ)-$\texttt{MCP}(\boldsymbol{\omega}, \boldsymbol{\mu})$

for $n = 1, 2, \cdots$ **do**

(Follow-the-Perturbed-Leader) $\mathcal{Z}_n \sim \exp(1)^K$ and $\eta_n = \frac{c_0}{\sqrt{n}}$

$$\boldsymbol{x}^{(n)} \in \operatorname*{argmin}_{\boldsymbol{x} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})} \left(\sum_{m=1}^{n-1} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha^{(m)}) + \frac{\langle \mathcal{Z}_n, \boldsymbol{x} \rangle}{\eta_n} \right)$$

(Best-Response) $\alpha^{(n)} \in \operatorname*{argmax}_{\alpha \geq 0} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(n)}, \alpha)$

if $\sqrt{n} > \frac{c_\theta(1+\epsilon)}{\epsilon \hat{F}}$, *where* $\begin{cases} \hat{F} = g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(n_\star)}, \alpha^{(n_\star)}) \\ n_\star \in \operatorname{argmin}_{m \leq n} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(m)}, \alpha^{(m)}) \end{cases}$

then return $(\hat{F}, \boldsymbol{x}^{(n_\star)})$;

end

Our proposed MCP algorithm

Algorithm 1: (ϵ, θ)-$\texttt{MCP}(\boldsymbol{\omega}, \boldsymbol{\mu})$

for $n = 1, 2, \cdots$ **do**

(Follow-the-Perturbed-Leader) $\mathcal{Z}_n \sim \exp(1)^K$ and $\eta_n = \frac{c_0}{\sqrt{n}}$

$$\boldsymbol{x}^{(n)} \in \operatorname*{argmin}_{\boldsymbol{x} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})} \left(\sum_{m=1}^{n-1} g_{\boldsymbol{\omega}, \boldsymbol{\mu}}(\boldsymbol{x}, \alpha^{(m)}) + \frac{\langle \mathcal{Z}_n, \boldsymbol{x} \rangle}{\eta_n} \right)$$

(Best-Response) $\alpha^{(n)} \in \operatorname*{argmax}_{\alpha \geq 0} g_{\boldsymbol{\omega}, \boldsymbol{\mu}}(\boldsymbol{x}^{(n)}, \alpha)$

(Computational Cost Per Iteration)

- $\boldsymbol{x}^{(n)}$ can be computed by at most $D = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_0$ calls to $\boldsymbol{i}^\star(\cdot)$
- $\alpha^{(n)}$ is evaluated in $\mathcal{O}(1)$

Our proposed MCP algorithm

The termination condition is designed based on Lemma 1

$$\textbf{if } \sqrt{n} > \frac{c_\theta(1+\epsilon)}{\epsilon \hat{F}}, \textit{ where } \begin{cases} \hat{F} = g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(n_\star)}, \alpha^{(n_\star)}) \\ n_\star \in \operatorname{argmin}_{m \le n} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(m)}, \alpha^{(m)}) \end{cases}$$

such that $\mathbb{P}\Big[F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \le \hat{F} \le (1+\epsilon) F_{\boldsymbol{\mu}}(\boldsymbol{\omega})\Big] \ge 1 - \theta$ holds.

(Lemma 1) If Algorithm 1 runs for N iterations, then

$$\mathbb{P}\left[\underbrace{\frac{1}{N}\sum_{n=1}^{N} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(n)}, \alpha^{(n)})}_{\ge \min_{n=1}^{N} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(n)},\alpha^{(n)}) = \hat{F}} - \underbrace{\frac{1}{N}\min_{\boldsymbol{x} \ne \boldsymbol{i}^\star(\boldsymbol{\mu})} \sum_{n=1}^{N} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha^{(n)})}_{\le \frac{1}{N}\sum_{n=1}^{N} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}_e,\alpha^{(n)}) \le F_{\boldsymbol{\mu}}(\boldsymbol{\omega})} \le \frac{c_\theta}{\sqrt{N}}\right] \ge 1 - \theta.$$

Our proposed MCP algorithm

Theorem 1 (MCP)

Let $(\boldsymbol{\omega}, \boldsymbol{\mu}) \in \Sigma_+ \times \Lambda$. The (ϵ, θ)-$\texttt{MCP}(\boldsymbol{\omega}, \boldsymbol{\mu})$ algorithm outputs $(\hat{F}, \hat{\boldsymbol{x}})$:

- $\mathbb{P}\left[F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \leq \hat{F} \leq (1+\epsilon) F_{\boldsymbol{\mu}}(\boldsymbol{\omega})\right] \geq 1-\theta$
- the number of calls to $\boldsymbol{i}^\star(\cdot)$: $\mathcal{O}\left(\frac{\|\boldsymbol{\mu}\|_\infty^4 \left\|\boldsymbol{\omega}^{-1}\right\|_\infty^2 K^3 D^5 \ln K \ln \theta^{-1}}{\epsilon^2 F_{\boldsymbol{\mu}}(\boldsymbol{\omega})^2}\right)$

By envelop theorem [WTP21], we estimation (sub)gradient of $F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$ by

$$\nabla_{\boldsymbol{\omega}} f_{\hat{\boldsymbol{x}}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \left(\frac{(\mu_k - \lambda_k^\star)^2}{2}\right)_{k \in [K]},$$

where $\boldsymbol{\lambda}^\star$ is the minimizer to the optimization problem of $f_{\hat{\boldsymbol{x}}}(\boldsymbol{\omega}, \boldsymbol{\mu})$.

Our P-FWS: the first poly-time statistically optimal algorithm

Solving $T^\star(\mu)$ with stochastic smoothed objective

- The well-studied stochastic smoothing [FKM05, DBW12] takes the average value in a neighborhood of points:

$$\bar{F}_{\boldsymbol{\mu},\eta}(\boldsymbol{\omega}) = \mathbb{E}_{\mathcal{Z}\sim\text{Uniform}(B_2)}[F_{\boldsymbol{\mu}}(\boldsymbol{\omega} + \eta\mathcal{Z})]$$

- $F_{\boldsymbol{\mu}}$ is ℓ-Lipschitz and its smoothed objective satisfies:
  - $\bar{F}_{\boldsymbol{\mu},\eta}$ is $\frac{\ell K}{\eta}$-smooth and $\bar{F}_{\boldsymbol{\mu},\eta}(\boldsymbol{\omega}) \xrightarrow{\eta\downarrow 0} F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$
  - $\nabla\bar{F}_{\boldsymbol{\mu},\eta}(\boldsymbol{\omega}) = \underset{\mathcal{Z}\sim\text{Uniform}(B_2)}{\mathbb{E}}[\nabla F_{\boldsymbol{\mu}}(\boldsymbol{\omega} + \eta\mathcal{Z})]$

High-level design of P-FWS

Let $\mathcal{X}_0$ be a set s.t. $\forall k \in [K]$, there exists $\boldsymbol{x} \in \mathcal{X}_0$ s.t. $x_k = 1$.

P-FWS alternate between two phases:

$$\begin{cases} \text{pull each } \boldsymbol{x} \in \mathcal{X}_0 \text{ once} & \text{(to avoid high cost and boundary cases)} \\ \text{pull } \boldsymbol{x}(t) \in \text{argmax}_{\boldsymbol{x}\in\mathcal{X}} \left\langle \nabla\bar{F}_{\hat{\boldsymbol{\mu}}(t-1),\eta_t}(\hat{\boldsymbol{\omega}}(t-1)), \boldsymbol{x} \right\rangle & \text{(ideal FW update)} \end{cases}$$

$\texttt{P-FWS}$: the first poly(K)-time optimal algorithm

Theorem 2 (P-FWS)

Let $\boldsymbol{\mu} \in \Lambda$ and $\delta \in (0,1)$. $\texttt{P-FWS}$ with proper parameters is δ-PAC and finishes in finite time;

(i) its $\mathbb{P}_{\boldsymbol{\mu}}\left[\limsup_{\delta \to 0} \frac{\tau}{\ln \delta^{-1}} \leq T^\star(\boldsymbol{\mu})\right] = 1$;

(ii) its $\mathbb{E}_{\boldsymbol{\mu}}[\tau]$ being poly(K) in moderate confidence regime;

(iii) the expected number of $\boldsymbol{i}^\star$ upper bounded by poly(K).

P-FWS: the first poly(K)-time optimal algorithm

Proof Sketch of Theorem 2

Define good events: $\mathcal{E}_t^{(1)}$ when $\hat{\boldsymbol{\mu}}(t)$ is sufficiently close to $\boldsymbol{\mu}$, and $\mathcal{E}_t^{(2)}$ when $\boldsymbol{x}(t)$ is closed to the ideal FW-update.

(Step 1) By maximum theorem [FKV14], we derive uniform continuity of $F_{\boldsymbol{\pi}}$ and $\nabla \bar{F}_{\boldsymbol{\pi},\eta}$ in $\boldsymbol{\pi}$
$\Rightarrow$ to simplify the analysis as if $\hat{\boldsymbol{\mu}}(t) = \boldsymbol{\mu}$ for $t \geq M$

(Step 2) Under $\mathcal{E}_t^{(1)} \cap \mathcal{E}_t^{(2)}$, we derive a recursive formula for the smoothed FW updates $\Rightarrow$ the FW algorithm converges

(Step 3) $\mathbb{E}_{\boldsymbol{\mu}}[\tau] \leq T_0(\delta) + \sum_{t \geq M} \mathbb{P}_{\boldsymbol{\mu}}\left[(\mathcal{E}_t^{(1)} \cap \mathcal{E}_t^{(2)})^c\right]$, where

$$\begin{cases} (\delta\text{-dependent}) \ \frac{T_0(\delta)}{\ln \delta^{-1}} \xrightarrow{\delta \to 0} T^\star(\boldsymbol{\mu}) \\ (\delta\text{-independent}) \ \sum_{t \geq M} \mathbb{P}_{\boldsymbol{\mu}}\left[(\mathcal{E}_t^{(1)} \cap \mathcal{E}_t^{(2)})^c\right] \leq \text{poly}(K) \end{cases}$$

P-FWS: the first poly(K)-time optimal algorithm

Algorithm 1: P-FWS$(\{\epsilon_t, \eta_t, n_t, \rho_t, \theta_t\}_t)$

Initialization: pull each $\boldsymbol{x} \in \mathcal{X}_0$ four times and update estimates

for $t = 4|\mathcal{X}_0| + 1, \cdots$ **do**

  if $\sqrt{\frac{t}{|\mathcal{X}_0|}} \in \mathbb{N}$ ***or*** *costly to estimate* $F_{\hat{\boldsymbol{\mu}}(t-1)}(\hat{\boldsymbol{\omega}}(t-1))$ **then**

    pull each $\boldsymbol{x} \in \mathcal{X}_0$ once;

  else

    pull $\boldsymbol{x}(t) \in \boldsymbol{i}^\star\left(\nabla \tilde{F}_{\hat{\boldsymbol{\mu}}(t-1),\eta_t,n_t}(\hat{\boldsymbol{\omega}}(t-1))\right)$ and update estimates;

  if *not costly to estimate* $F_{\hat{\boldsymbol{\mu}}(t)}(\hat{\boldsymbol{\omega}}(t))$ **then**

    compute $\hat{F}_t$ by $(\epsilon_t, \frac{\delta}{t^2})$-MCP$(\hat{\boldsymbol{\omega}}(t), \hat{\boldsymbol{\mu}}(t))$;

    return $\boldsymbol{i}^\star(\hat{\boldsymbol{\mu}}(t))$ **if** $t\hat{F}_t > (1+\epsilon_t)\beta(t, \frac{4|\mathcal{X}_0|-1}{|\mathcal{X}_0|}\delta)$

end

$\nabla \tilde{F}_{\hat{\boldsymbol{\mu}}(t-1),\eta_t,n_t}(\hat{\boldsymbol{\omega}}(t-1))$ is a n_t-sample estimation to $\nabla \bar{F}_{\hat{\boldsymbol{\mu}}(t-1),\eta_t}(\hat{\boldsymbol{\omega}}(t-1))$

Preliminary Numerical Results

Empirical evaluation on $\mathcal{X}$ as the set of spanning trees

All the experiments[2] are performed on a Macbook Air with 16 GB memory.

Table 1: Averaged sample complexity at $\delta = 0.1$ over 100 independent runs on a graph with $|\mathcal{X}| = 21\,025$ spanning trees.

Algorithm	Sample Complexity
P-FWS (ours)	1 176
CombGame [JMKK21]	1 277

Table 2: Averaged sample complexity at $\delta = 0.1$ over 100 independent runs on a graph with $|\mathcal{X}| = 343\,385$ spanning trees.

Algorithm	Sample Complexity
P-FWS (ours)	1 501
CombGame [JMKK21]	OOM

[2] Our code: https://github.com/rctzeng/NeurIPS2023-PerturbedFWS.

Conclusion and Future Works

Conclusion and open questions

- Our proposed `P-FWS` is the first algorithm to close the statistical-computational gap for combinatorial BAI by exploring the structural properties of the lowerbound problem.
- It remains largely unexplored whether one can close the computational-statistical gap for other tasks, such as
  - combinatorial BAI with semi-bandit feedback (Bernoulli)
  - combinatorial BAI with bandit feedback
  - linear BAI (to have runtime polynomial in the dimension)

- Kenshi Abe, Kaito Ariu, Mitsuki Sakamoto, Kentaro Toyoshima, and Atsushi Iwasaki, *Last-iterate convergence with full-and noisy-information feedback in two-player zero-sum games*, Proc. of AISTATS, 2023.
- Ioannis Anagnostides, Ioannis Panageas, Gabriele Farina, and Tuomas Sandholm, *On last-iterate convergence beyond zero-sum games*, Proc. of ICML, 2022.
- Lijie Chen, Anupam Gupta, and Jian Li, *Pure exploration of multi-armed bandit under matroid constraints*, Proc. of COLT, 2016.

Reference ii

- John C Duchi, Peter L Bartlett, and Martin J Wainwright, *Randomized smoothing for stochastic optimization*, SIAM Journal on Optimization (2012).
- Constantinos Daskalakis and Ioannis Panageas, *Last-iterate convergence: Zero-sum games and constrained min-max optimization*, Proc. of ITCS (2019).
- Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan, *Online convex optimization in the bandit setting: gradient descent without a gradient*, Proc. of SODA, 2005.
- Eugene A Feinberg, Pavlo O Kasyanov, and Mark Voorneveld, *Berge's maximum theorem for noncompact image sets*, Journal of Mathematical Analysis and Applications (2014).

Reference iii

- Aurélien Garivier and Emilie Kaufmann, *Optimal best arm identification with fixed confidence*, Proc. of COLT, 2016.
- Marc Jourdan, Mojmír Mutnỳ, Johannes Kirschner, and Andreas Krause, *Efficient pure exploration for combinatorial bandits with semi-bandit feedback*, Proc. of ALT, 2021.
- Qi Lei, Sai Ganesh Nagarajan, Ioannis Panageas, et al., *Last iterate convergence in no-regret learning: constrained min-max optimization for convex-concave landscapes*, Proc. of AISTATS, 2021.
- Po-An Wang, Ruo-Chun Tzeng, and Alexandre Proutiere, *Fast pure exploration via frank-wolfe*, Proc. of NeurIPS, 2021.