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

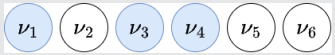
[1]EECS, KTH Royal Institue of Technology, Sweden
[2]Institute of Information Science, Academia Sinica, Taiwan

Combinatorial BAI with fixed confidence

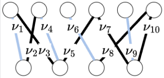
Input: K arms $(\nu_k)_{k\in[K]}$ with mean $\boldsymbol{\mu} \in \mathbb{R}^K$ and $\mathcal{X} \subseteq \{0,1\}^K$

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

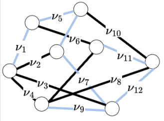
m-sets



matchings



spanning trees



Rule: At each round t, the learner pulls $\boldsymbol{x}(t) \in \mathcal{X}$ and observes $y_k(t) \sim \nu_k$ iff $x_k(t) = 1$, and outputs $\hat{\boldsymbol{\imath}} \in \mathcal{X}$ at her termination round τ.

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 prob. $\geq 1-\delta$ and $\mathbb{P}_{\boldsymbol{\mu}}[\tau < \infty] = 1$ while minimizing $\mathbb{E}_{\boldsymbol{\mu}}[\tau]$.

(Open Question) Is it possible to design a statistically optimal δ-PAC algorithm that runs in polynomial time?

Prior works: a computational-statistical gap

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

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

Solving $F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$ implicitly determines the most confusing parameter (MCP).[1] Below are the existing statistically optimal BAI algorithms:

- **Track-and-Stop**[GK16] requires to repeatedly solve $T^\star(\hat{\boldsymbol{\mu}}(t-1))^{-1}$
- **FWS** [WTP21] has to solve probably $\mathcal{O}(2^K)$ many convex programs
- **CombGame** [JMKK21] is MCP-oracle efficient

> Difficulty in designing an efficient MCP algorithm (to evaluate $F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$) comes from its domain $\mathrm{Alt}(\boldsymbol{\mu}) = \{\boldsymbol{\lambda} \in \Lambda : \boldsymbol{i}^\star(\boldsymbol{\lambda}) \neq \boldsymbol{i}^\star(\boldsymbol{\mu})\}$.

[1] Intuitively speaking, MCP is the closest parameter $\boldsymbol{\lambda}^\star$ to trick a learner with the given allocation $\boldsymbol{\omega}$ into giving an incorrect answer $\boldsymbol{i}^\star(\boldsymbol{\lambda}^\star) \neq \boldsymbol{i}^\star(\boldsymbol{\mu})$.

Our efficient MCP algorithm exploits structural property

Structural properties about $F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$

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

$$\begin{cases} f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \max_{\alpha \geq 0} g_{\boldsymbol{\omega}, \boldsymbol{\mu}}(\boldsymbol{x}, \alpha) & \text{(known by [CGL16])} \\ g_{\boldsymbol{\omega}, \boldsymbol{\mu}}(\boldsymbol{x}, \alpha) \text{ is linear in } \boldsymbol{x} \text{ and concave in } \alpha & \text{(our observation)} \end{cases}$$

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

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

$\Rightarrow$ Rules out many results on average-iterate convergence [DDK11, RS13] and last-iterate convergence [AAS+23, DP19] from applying.

The reason why $\boldsymbol{x}_e$ is required is because we will use gradient-based method to solve $\max_{\boldsymbol{\omega} \in \Sigma} F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$.

Our efficient MCP algorithm exploits structural property

Theorem 1 (MCP) Let $(\boldsymbol{\omega}, \boldsymbol{\mu}) \in \Sigma_+ \times \Lambda$. The output $(\hat{F}, \hat{\boldsymbol{x}})$ returned by (ϵ, θ)-$\texttt{MCP}(\boldsymbol{\omega}, \boldsymbol{\mu})$ satisfies:

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

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

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

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

end

The design of Perturbed Frank-Wolfe Sampling (P-FWS)

By the standard stochastic smoothing [FKM05, DBW12], the smoothed $\bar{F}_{\boldsymbol{\mu},\eta}(\boldsymbol{\omega}) = \mathbb{E}_{\boldsymbol{\mathcal{Z}}\sim\text{Uniform}(B_2)}[F_{\boldsymbol{\mu}}(\boldsymbol{\omega} + \eta\boldsymbol{\mathcal{Z}})]$ objective with noise level $\eta > 0$ has several nice properties:

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

$\Rightarrow$ All `P-FWS` need is the linear maximization $\boldsymbol{i}^\star$-oracle and the gradients (which can be evaluated by the envelope theorem [WTP21])!

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:

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

The design of Perturbed Frank-Wolfe Sampling (P-FWS)

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:

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

Theorem 2 (P-FWS) Let $\boldsymbol{\mu} \in \Lambda$ and $\delta \in (0,1)$. `P-FWS` is δ-PAC and finishes in finite time

- $\mathbb{P}_{\boldsymbol{\mu}}\left[\limsup_{\delta \to 0} \frac{\tau}{\ln \delta^{-1}} \le T^\star(\boldsymbol{\mu})\right] = 1$
- $\mathbb{E}_{\boldsymbol{\mu}}[\tau]$ is bounded by $\mathrm{Poly}(K)$ in moderate-confidence regime and achieves the minimal in high-confidence regime
- the total # of $\boldsymbol{i}^\star$-oracle calls is bounded by $\mathrm{Poly}(K)$.

The design of Perturbed Frank-Wolfe Sampling (P-FWS)

Proof Sketch of Theorem 2 (P-FWS)

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 for $F_{\boldsymbol{\pi}}$ 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 $\Rightarrow$ to show our `P-FWS` 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

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

Preliminary numerical results 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

- 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 linear BAI or best-policy identification.

Reference i

- 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.
- Lijie Chen, Anupam Gupta, and Jian Li, *Pure exploration of multi-armed bandit under matroid constraints*, Proc. of COLT, 2016.
- John C Duchi, Peter L Bartlett, and Martin J Wainwright, *Randomized smoothing for stochastic optimization*, SIAM Journal on Optimization (2012).

Reference ii

- Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim, *Near-optimal no-regret algorithms for zero-sum games*, Proc. of SODA, 2011.
- 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).

- 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.
- Sasha Rakhlin and Karthik Sridharan, *Optimization, learning, and games with predictable sequences*, Proc. of NeurIPS, 2013.
- Po-An Wang, Ruo-Chun Tzeng, and Alexandre Proutiere, *Fast pure exploration via frank-wolfe*, Proc. of NeurIPS, 2021.