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]

May 2, 2026

[1]EECS, KTH, Sweden
[2]IIS, AS, Taiwan

Introduction

1-page Summary

In combinatorial semi-bandits,...

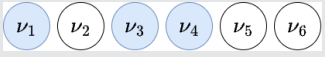
- most statistically efficient algorithms suffer computational inefficiency [Garivier and Kaufmann, 2016, Wang et al., 2021, Jourdan et al., 2021].
- most computationally efficient algorithms suffer statistical inefficiency [Katz-Samuels et al., 2020, Du et al., 2021].
- we fulfill the computational-statistical gap.

Combinatorial semi-bandits

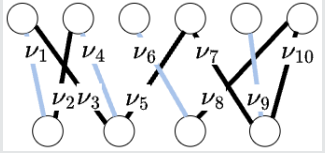
- K arms (K probability distributions), say $\nu_1 = \mathcal{N}(\mu_1, 1), \nu_2 = \mathcal{N}(\mu_2, 1) \ldots, \nu_K = \mathcal{N}(\mu_K, 1)$.
- Set of actions, $\mathcal{X} \subseteq \{0,1\}^K$, with a combinatorial structure.

Examples for combinatorial structures

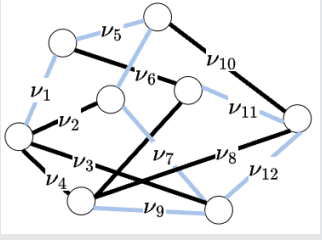
m-sets



matchings



spanning trees



In round t, an agent

1. pulls an action $\boldsymbol{x}(t) \in \mathcal{X}$
2. receives the reward $y_k(t) \sim \mathcal{N}(\mu_k, 1)$ if $x_k(t) = 1$

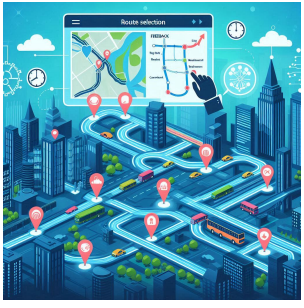
Best Action Identification with Fixed Confidence

Goal: identify the best action $i^\star(\boldsymbol{\mu}) \in \arg\max_{x \in \mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{\mu} \rangle$.

A strategy consist of

- (sampling rule) $\boldsymbol{x}(t) \in \mathcal{F}_t := \sigma(\boldsymbol{x}(1), \boldsymbol{y}(1), \ldots, \boldsymbol{x}(t-1), \boldsymbol{y}(t-1))$ (arm to explore)
- (stopping rule) τ (round to stop)
- (decision rule) $\mathcal{F}_\tau$-measurable $\hat{\boldsymbol{\imath}} \in \mathcal{X}$ (guess of best action to return)

Wish to minimize $\mathbb{E}_{\boldsymbol{\mu}}[\tau]$ subject to $\mathbb{P}_{\boldsymbol{\mu}}[\hat{\boldsymbol{\imath}} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})] \leq \delta$.



In traffic management, authorities might select a set of routes for traffic flow optimization. Feedback on each route (e.g., congestion levels, travel times) helps improve future route selections.

Information-theoretic Lower Bound [Garivier and Kaufmann, 2016]

For any δ-PAC algorithm (stop in finite time a.e. and $\mathbb{P}_{\boldsymbol{\mu}}[\hat{\imath} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})] \leq \delta$),

$$\liminf_{\delta \to 0} \frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau]}{\ln(1/\delta)} \geqslant \left(\sup_{\boldsymbol{\omega} \in \Sigma} \inf_{\boldsymbol{\lambda} \in \mathsf{Alt}(\boldsymbol{\mu})} \sum_{k=1}^{K} \frac{\omega_k (\mu_k - \lambda_k)^2}{2} \right)^{-1}. \tag{1}$$

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

(1)$\Rightarrow$ An optimal algorithm has a sampling strategy described by

$$\hat{\boldsymbol{\omega}}(t) \xrightarrow{t \to \infty} \boldsymbol{\omega}^\star(\boldsymbol{\mu}) := \arg\max_{\boldsymbol{\omega} \in \Sigma} F_{\boldsymbol{\mu}}(\boldsymbol{\omega}),$$

where $\hat{\boldsymbol{\omega}}(t)$ is the empirical allocation of arm draws up to round t, and

$$F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \inf_{\boldsymbol{\lambda} \in \mathsf{Alt}(\boldsymbol{\mu})} \sum_{k=1}^{K} \frac{\omega_k (\mu_k - \lambda_k)^2}{2}.$$

A Computational Challenge for Matching Lower Bound

A standard way [Garivier and Kaufmann, 2016] consists of:

- (sampling rule) pull an action by tracking $\arg\max_{\boldsymbol{\omega}\in\Sigma} F_{\hat{\boldsymbol{\mu}}(t)}(\boldsymbol{\omega})$
- (stopping rule) Generalized Likelihood Ratio Test (GLRT):

$$\tau = \inf\left\{ t : tF_{\hat{\boldsymbol{\mu}}(t)}(\hat{\boldsymbol{\omega}}(t)) > \ln\left(\frac{t}{\delta}\right) + o(1) \right\} \tag{2}$$

- (decision rule) return $\hat{\boldsymbol{\imath}} \leftarrow \boldsymbol{\imath}^\star(\hat{\boldsymbol{\mu}}(\tau))$

where $\hat{\boldsymbol{\mu}}(t)$ is the estimate of $\boldsymbol{\mu}$, and $\hat{\boldsymbol{\omega}}(t)$ is empirical pulled arm allocation up to t.

The value of $F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) := \inf_{\boldsymbol{\lambda}\in \mathrm{Alt}(\boldsymbol{\mu})} \sum_{k=1}^{K} \frac{\omega_k(\mu_k-\lambda_k)^2}{2}$ is the most confusing parameter (MCP).

Difficulty in identifying MCP

Recall $\mathrm{Alt}(\boldsymbol{\mu}) = \{\boldsymbol{\lambda} \in \Lambda : \boldsymbol{\imath}^\star(\boldsymbol{\lambda}) \neq \boldsymbol{\imath}^\star(\boldsymbol{\mu})\}$

$\Rightarrow$ Prior approach is to solve $|\mathcal{X}| - 1$ many convex programs by partitioning

$\mathrm{Alt}(\boldsymbol{\mu}) = \cup_{\boldsymbol{x}\neq \boldsymbol{\imath}^\star(\boldsymbol{\mu})}\{\boldsymbol{\lambda} \in \Lambda : \langle \boldsymbol{\imath}^\star(\boldsymbol{\mu}) - \boldsymbol{x}, \boldsymbol{\lambda}\rangle < 0\}$

However, $|\mathcal{X}| = \mathcal{O}(2^K)$!

Other Statistically Optimal Sampling Rules

- FWS [Wang et al., 2021]: apply the first-order methods to maximize $F_{\boldsymbol{\mu}}(\boldsymbol{\omega})$, where the gradient is derived by a MCP (Envelop theorem)
- CombGame [Jourdan et al., 2021]: need a MCP for chasing a saddle point.
- ACC [Qin and You, 2023]: need a MCP to get its best challenger.

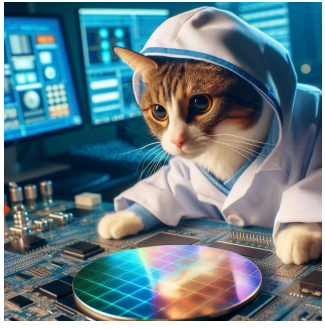


Figure 1: MCP in pure exploration resembles the role of the semi-conductor in modern technology.

Our Perturbed Frank-Wolfe Sampling (P-FWS)

- P-FWS uses perturbed Frank-Wolfe based algorithm to reach the optimal allocation
- All P-FWS needs are
  - (i) linear maximization (standard),
  - (ii) (ε, θ)-MCP, where $\varepsilon > 0, \theta \in (0, 1)$ (established in this work).

(ε, θ)-MCP

Return $\hat{F}$ s.t.

$$\mathbb{P}[F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \leq \hat{F} \leq (1 + \varepsilon) F_{\boldsymbol{\mu}}(\boldsymbol{\omega})] \geq 1 - \theta$$

How to get (ε, θ)-MCP?

Let Us Observe MCP

For each suboptimal action $\boldsymbol{x} \neq i^\star(\boldsymbol{\mu})$, define

$$f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) := \inf_{\boldsymbol{\lambda} \in \mathbb{R}^K : \langle \boldsymbol{i}^\star(\boldsymbol{\mu}) - \boldsymbol{x}, \boldsymbol{\lambda} \rangle < 0} \sum_{k=1}^{K} \frac{\omega_k (\mu_k - \lambda_k)^2}{2}. \tag{3}$$

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

$$f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \max_{\alpha \geqslant 0} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha) \text{ (strong duality)}$$

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

$$\Rightarrow F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \min_{\boldsymbol{x} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})} f_{\boldsymbol{x}}(\boldsymbol{\omega}, \boldsymbol{\mu}) = \min_{\boldsymbol{x} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})} \max_{\alpha \geqslant 0} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha) \tag{4}$$

(4) motivates us a design:

- $\boldsymbol{x}$-player employs a regret minimization algorithm to minimize $g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha)$
- α-player employs a regret minimization algorithm to maximize $g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}, \alpha)$

Two-players Game

However, the choices are limited since...

1. $\{\boldsymbol{x} \neq \boldsymbol{i}^\star(\boldsymbol{\mu})\}$ is a discrete set which consist of $\mathcal{O}(2^K)$ elements
2. $\min\max = \max\min$ may not hold
3. *equilibrium* action $\boldsymbol{x}_e$ s.t. $F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) = \max_{\alpha \geqslant 0} g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}_e, \alpha)$ need a maximizer in α
4. we want it to be computationally efficient, i.e. approximate $\boldsymbol{x}_e$ in polynomial time

Recall

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

Figure 2: Two-cats Game

Our (ε, θ)-MCP

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

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

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

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

(Best-Response) $\alpha^{(n)} \in \underset{\alpha \geqslant 0}{\arg\max}\, g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(n)}, \alpha)$

if $\sqrt{n} > \dfrac{c_\theta(1+\varepsilon)}{\varepsilon \hat{F}}$, *where* $\hat{F} = \min_n g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(n)}, \alpha^{(n)})$ **then return** $\hat{F}$;

end

Main Results

Theorem for (ε, θ)-MCP

Theorem 1

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

- $\mathbb{P}\Big[F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \leqslant \hat{F} \leqslant (1+\varepsilon)F_{\boldsymbol{\mu}}(\boldsymbol{\omega})\Big] \geqslant 1-\theta$
- the number of calls to linear maximization $(\boldsymbol{i}^\star(\cdot))$ is polynomial

P-FWS: the First poly (K)-time Optimal Algorithm

Theorem 2

Let $\boldsymbol{\mu} \in \Lambda$ and $\delta \in (0,1)$. P-FWS is δ-PAC, and

(i) $\limsup_{\delta \to 0} \frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau]}{\ln(1/\delta)} \leq \left(\sup_{\boldsymbol{\omega} \in \Sigma} \inf_{\boldsymbol{\lambda} \in \mathsf{Alt}(\boldsymbol{\mu})} \sum_{k=1}^{K} \frac{\omega_k (\mu_k - \lambda_k)^2}{2} \right)^{-1}$

(ii) $\mathbb{E}_{\boldsymbol{\mu}}[\tau] = \text{poly}\,(K)$ in moderate confidence regime

(iii) the expected number of calls for linear maximization upper bounded by poly (K)

Note: To our best knowledge, (ii) is even not yet established in unstructured BAI.

Figure 3: A cat wins three medals in *Olympicat*.

Conclusion and Future Works

Conclusion: P-FWS closes the statistical-computational gap for combinatorial BAI by exploiting the structural properties in lower bound, and it show strong numerical performance.

Table 1: 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 [Jourdan et al., 2021] with our (ε, θ)-MCP	OOM

Future works: It remains largely unexplored whether one can close the computational-statistical gap for other distributions.

	reward distr.	comput. & stat. efficient	stat. optimal
comb. BAI semi-bandit	Gaussian	[Tzeng et al., 2023]	[Jourdan et al., 2021, Wang et al., 2021]
	Bernoulli	open	
comb. BAI bandit	Gaussian Bernoulli	open	[Wang et al., 2021]
regret min.	Gaussian Bernoulli	open	[Combes et al., 2017]

Table 2: Computational-statistical gap in combinatorial semi-bandits.

Reference i

Combes, R., Magureanu, S., and Proutiere, A. (2017).
Minimal exploration in structured stochastic bandits.
In *Proc. of NeurIPS*.

Du, Y., Kuroki, Y., and Chen, W. (2021).
Combinatorial pure exploration with full-bandit or partial linear feedback.
In *Proc. of AAAI*.

Garivier, A. and Kaufmann, E. (2016).
Optimal best arm identification with fixed confidence.
In *Proc. of COLT*.

Jourdan, M., Mutnỳ, M., Kirschner, J., and Krause, A. (2021).
Efficient pure exploration for combinatorial bandits with semi-bandit feedback.
In *Proc. of ALT*.

Reference ii

Katz-Samuels, J., Jain, L., Jamieson, K. G., et al. (2020).
An empirical process approach to the union bound: Practical algorithms for combinatorial and linear bandits.
In *Proc. of NeurIPS*.

Qin, C. and You, W. (2023).
Dual-directed algorithm design for efficient pure exploration.
arXiv preprint arXiv:2310.19319.

Tzeng, R.-C., Wang, P.-A., Proutiere, A., and Lu, C.-J. (2023).
Closing the computational-statistical gap in best arm identification for combinatorial semi-bandits.
In *Proc. of NeurIPS*.

Wang, P.-A., Tzeng, R.-C., and Proutiere, A. (2021).
Fast pure exploration via frank-wolfe.
In *Proc. of NeurIPS*.

Typical Assumptions in Combinatorial BAI

Throughout this work, we assume

(i) $\forall \boldsymbol{\mu} \in \Lambda$, the best action $\boldsymbol{i}^\star(\boldsymbol{\mu})$ is unique, where Λ denotes all possible parameters.

(ii) $\forall k \in [K]$, arm-k reward distribution is $\nu_k = \mathcal{N}(\mu_k, 1)$.

(iii) $\forall \boldsymbol{v} \in \mathbb{R}^K$, a maximizing action $\boldsymbol{i}^\star(\boldsymbol{v}) \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{v} \rangle$ can be found in polynomial time.

Computational Cost of Our (ε, θ)-MCP

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

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

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

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

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

(Computational Cost Per Iteration)

- $\boldsymbol{x}^{(n)}$ is evaluated in $\max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_0$ calls to linear maximization $\boldsymbol{i}^\star(\cdot)$
- $\alpha^{(n)}$ is evaluated in $\mathcal{O}(1)$

Termination Condition

The termination condition

$$\textbf{If } \sqrt{n} > \frac{c_\theta(1+\varepsilon)}{\varepsilon \hat{F}} \text{ where } \hat{F} = \min_n g_{\boldsymbol{\omega},\boldsymbol{\mu}}(\boldsymbol{x}^{(n)}, \alpha^{(n)}) \textbf{ then return } \hat{F} \tag{5}$$

is designed such that $\mathbb{P}\Big[F_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \leqslant \hat{F} \leqslant (1+\varepsilon)F_{\boldsymbol{\mu}}(\boldsymbol{\omega})\Big] \geqslant 1-\theta$ holds.

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

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

Empirical Evaluation on $\mathcal{X}$ as the Set of Spanning Trees

All the experiments[1] are performed on a Macbook Air with 16 GB memory.

Table 3: 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` [Jourdan et al., 2021] with our (ε, θ)-`MCP`	$1\,277$

Table 4: 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` [Jourdan et al., 2021] with our (ε, θ)-`MCP`	OOM

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