Heterogeneous Multi-Agent Bandits with Parsimonious Hints

Amirmahdi Mirfakhar[1], Xuchuang Wang[1], Jinhang Zuo[2], Yair Zick[1], Mohammad Hajiesmaili[1]

[1]University of Massachusetts Amherst, [2]City University of Hong Kong
smirfakhar@umass.edu, xuchuangw@gmail.com, jinhang.zuo@cityu.edu.hk, yzick@umass.edu, hajiesmaili@cs.umass.edu

Abstract

We study a hinted heterogeneous multi-agent multi-armed bandits problem (HMA2B), where agents can query low-cost observations (hints) in addition to pulling arms. In this framework, each of the M agents has a unique reward distribution over K arms, and in T rounds, they can observe the reward of the arm they pull only if no other agent pulls that arm. The goal is to maximize the total utility by querying the minimal necessary hints without pulling arms, achieving time-independent regret. We study HMA2B in both centralized and decentralized setups. Our main centralized algorithm, GP-HCLA, which is an extension of HCLA, uses a central decision-maker for arm-pulling and hint queries, achieving $O(M^4K)$ regret with $O(MK\log T)$ adaptive hints. In decentralized setups, we propose two algorithms, HD-ETC and EBHD-ETC, that allow agents to choose actions independently through collision-based communication and query hints uniformly until stopping, yielding $O(M^3K^2)$ regret with $O(M^3K\log T)$ hints, where the former requires knowledge of the minimum gap and the latter does not. Finally, we establish lower bounds to prove the optimality of our results and verify them through numerical simulations.

1 Introduction

The multi-agent multi-armed bandit (MA2B) problem (Liu and Zhao 2010; Anandkumar et al. 2011) is a sequential decision making task consisting of $K \in \mathbb{N}^+$ arms and $M \in \mathbb{N}^+$ agents. In each of the total $T \in \mathbb{N}^+$ decision rounds, each agent selects one arm to pull and observes its reward if no other agent pulls the same arm (called no *collision*). This model has applications in wireless communication (Jouini et al. 2009, 2010), caching (Xu, Tao, and Shen 2020; Xu and Tao 2020), and edge computing (Wu et al. 2021). Among various models in MA2B, the heterogeneous multi-agent multi-armed bandit (Bistritz and Leshem 2018; Shi et al. 2021) is a more realistic variant for these applications where agents have different reward distributions over the arms, e.g., in a wireless communication scenario where agents have different channel qualities due to different geographical locations. In this heterogeneous MA2B model, the optimal action of all agents is a bipartite matching (between agents and arms) that maximizes the total reward, called the *optimal matching*. An algorithm's performance is evaluated by *regret*, the difference between the accumulative reward of keeping to choose the optimal matching in all decision rounds and the total reward of the bandit algorithm. A smaller expected regret implies a better algorithm.

Recently, learning-augmented approaches are emerging, e.g., Lykouris and Vassilvitskii (2021); Bamas, Maggiori, and Svensson (2020); Bhaskara et al. (2023). This stream of research studies how to assist an algorithm with *hints* (a.k.a., predictions) queried from existing ML models, e.g., large language model (Achiam et al. 2023), deep convolutional neural network (Krizhevsky, Sutskever, and Hinton 2017), and deep reinforcement learning (François-Lavet et al. 2018).

In this paper, we study the utilization of hint information in heterogeneous multi-agent multi-armed bandits. In addition to receiving feedback from pulling arms, agents can sequentially query hints about the potential rewards of other arms, assisting in their decision-making process. We call the model *Hinted Heterogeneous Multi-Agent Multi-Armed Bandits* (HMA2B). Specifically, we consider a simple and accurate hint mechanism where agents can query the reward of an arm without pulling it, with no regret incurred from the queried hint. Despite assuming accurate hints, this model poses challenges, such as balancing hint queries and arm-pullings while accounting for agent heterogeneity and potential future collisions. In addition to minimizing regret, we aim to reduce *hint complexity*, the total number of queried hints, as querying hints, such as via the GPT-4 API (Achiam et al. 2023), can be costly. Efficiently leveraging hints is crucial in scenarios where hint costs are significantly lower than the costs of taking actions. For instance, in labor markets, structured, low-cost interviews provide hints to improve applicant-role matching, reducing the risk of human resource misallocation. Similarly, in radio channel assignments, test signals serve as hints to allocate high-bandwidth channels effectively, preventing delays and disruptions in critical applications like disaster recovery, where drones depend on reliable communication channels.

We study two scenarios of HMA2B: *centralized* and *decentralized* setups. In the centralized setup, an omniscient decision-maker determines which arm each agent should pull or query hints from, similar to decision-making in hiring processes where the employer has access to the applicants' information to decide which of them to interview and which

Algorithm	D/C	Regret	Queried Hints	Communication
HCLA (Algorithm 1)	C	$O\left(MK^{2M}\right)$	$O\left(MK^{M}\log T\right)$	N/A
G-HCLA (Algorithm 5)	C	$O\left(M^{4}K\right)$	$O\left(M^{2}K\log T\right)$	N/A
GP-HCLA (Algorithm 2)	C	$O\left(M^{4}K\right)$	$O\left(MK\log T\right)$	N/A
HD-ETC (Algorithm 3)†	D	$O\left(M^{3}K^{2}\right)$	$O\left(M^{3}K\log(MT)\right)$	$O\left(\log T\right)$
EBHD-ETC (Algorithm 4)	D	$O\left(M^{3}K^{2}\right)$	$O\left(M^{3}K\log(MT)\right)$	$O\left(\log T\right)$

Table 1: Regret, Queried Hints, and Communication Bounds for Centralized and Decentralized Algorithms ('C' and 'D' stand for centralized and decentralized algorithms respectively, † indicates that HD-ETC relies on the knowledge of minimum gap)

to hire. In the decentralized setup, agents independently decide their actions through collision-based communications, e.g., in radio channel allocation to stations.

Designing an algorithm that achieves time-independent regret with a linear number of hints in T is straightforward. However, reducing the queried hints to a sub-linear number in T is challenging. To tackle this, we first analyze the fundamental limits of hint complexity in the centralized setup and propose GP-HCLA, a fine-tuned algorithm based on the advanced kl-UCB algorithm (Cappé et al. 2013), which achieves asymptotically optimal hint complexity (Appendix H). In decentralized setups, the essence of communication lies in the absence of a central decision maker, requiring collision-based signaling (Wang et al. 2020), where making or avoiding collisions encode '1' or '0' information bit. This method introduces inaccuracies from sending decimal statistics in binary and additional regret due to delayed exploration while balancing communication to determine the optimal matching. To address these challenges, we propose HD-ETC and EBHD-ETC, which achieve relatively similar bounds on hint complexity and regret. These algorithms use a round-robin hint querying strategy combined with an *Explore-then-Commit* (Garivier, Ménard, and Stoltz 2019) approach until a stopping condition is met. Finally, we discuss the optimality of the results in both centralized and decentralized setups in Appendix H.

1.1 Contributions

For the centralized setup (Section 3), we propose two algorithms: HCLA and GP-HCLA. Both use empirical means to select a matching to pull and kl-UCB indices (Cappé et al. 2013) to identify another matching, querying a hint if the latter has a higher value. Additionally, we analyze an intermediate algorithm, G-HCLA (Appendix A), which operates similarly to HCLA but differs from GP-HCLA in how it selects the matching to hint after deciding to query. As summarized in Table 1, both GP-HCLA and G-HCLA—extensions of HCLA—achieve time-independent regret with an asymptotically optimal number of hints. We further prove that the upper bound on the hint complexity for GP-HCLA is tight, with both GP-HCLA and G-HCLA matching the established lower bounds. In the decentralized setup (Section 4), we introduce two algorithms: HD-ETC and EBHD-ETC. Both divide the time horizon into three phases: *exploration*, *communication*, and *exploitation*, with a key difference in how they transition to the exploitation phase. In the exploitation phase, no further communication, exploration, or hint querying occurs, and the two algorithms handle this transition differently. In HD-ETC, agents know the minimum gap—the smallest utility difference between the optimal and other matchings—and the time horizon T, allowing them to switch to exploitation at a fixed time step T_0. Conversely, EBHD-ETC does not require this knowledge, using an edge elimination strategy to determine the transition point, which makes it a random variable. This results in slightly higher hint queries and regret compared to HD-ETC. We provide regret bounds for both algorithms that align with the lower bounds, accounting for uncertainties due to delayed communication.

1.2 Related Works

Heterogeneous MMAB (HMMAB) HMMAB is one of the standard models in multi-player multi-armed bandits with collision literature; to name a few, Rosenski, Shamir, and Szlak (2016); Boursier and Perchet (2019); Mehrabian et al. (2020); Bistritz and Leshem (2018); Shi et al. (2021). Among them, Bistritz and Leshem (2018) was the first to study the HMMAB, where they proposed a decentralized algorithm with $O(\log^2 T)$ regret. Later on, the regret bound of this model was improved to $O(M^3K\log T)$ by Mehrabian et al. (2020) and further to $O(M^2K\log T)$ by Shi et al. (2021) that is the state-of-the-art result. We are the first to introduce the hint mechanism to HMMAB.

Bandits with Hints Learning algorithms with hints (or predictions) are part of the emerging literature on learning-augmented methods, as seen in works like (Lykouris and Vassilvitskii 2021; Purohit, Svitkina, and Kumar 2018; Mitzenmacher and Vassilvitskii 2022), etc. The hint mechanism was initially explored in the basic stochastic multi-armed bandits model by Yun et al. (2018). Later, Lindståhl, Proutiere, and Johnsson (2020) examined a more realistic hint mechanism, which includes failure noise, for the same model. Additionally, Bhaskara et al. (2023) investigated the impact of hints in adversarial bandits. We are the first to study the hint mechanism in a multi-agent scenario.

2 Hinted Heterogeneous Multi-Agent Multi-Armed Bandits

Basic model A Hinted Heterogeneous Multi-agent Multi-Armed Bandit (HMA2B) model consists of a set of K *arms*

$\mathcal{K}$ and a set of M *agents* $\mathcal{M}$, such that $M < K$. Agents have heterogeneous *rewards* for arms. That is, for each agent $m \in \mathcal{M}$, each arm $k \in \mathcal{K}$ is associated with a Bernoulli reward random variable $X_{m,k}$ with mean $\mu_{m,k} \coloneqq \mathbb{E}[X_{m,k}]$.

The heterogeneous reward means are represented by a matrix $\boldsymbol{\mu} \in [0,1]^{M \times K}$, where each of its rows is denoted by $\boldsymbol{\mu}_m = (\mu_{m,k})_{k \in \mathcal{K}} \in [0,1]^K$.

Reward feedback Suppose that $T \in \mathbb{N}^+$ denotes the total number of decision rounds. At each *time step* $t \in \{1, 2, \ldots, T\}$, every agent m chooses an arm $k_m(t)$ to pull. The arms requested by the agents construct a bipartite graph characterized by M nodes (agents) on one side and K nodes (arms) on the other comprising M edges, ensuring that each node on the agent side is connected to exactly one arm. Let us define $\mathcal{G}$ as the set of all such graphs. Denote $G(t) \coloneqq (m, k_m(t))_{m \in \mathcal{M}}$ as the bipartite graph representing the arm pulling graph of the agents at time step t. We consider the *collision* setting (Boursier and Perchet 2019; Shi et al. 2021): that is, if there exist other agents pulling the arm $k_m(t)$ at time step t, then agent m gets a reward of zero; otherwise, agent m gets a reward $X_{m,k_m(t)}(t)$ sampled from the reward distribution of arm $k_m(t)$, or formally, $r_m(t) \coloneqq X_{m,k_m(t)}(t) \mathbb{1}\{\forall m' \neq m : k_{m'}(t) \neq k_m(t)\}$. This induces the optimal action to be a matching.

Given a matching $G \in \mathcal{G}$ and a reward mean matrix $\boldsymbol{\mu} \in [0,1]^{M \times K}$, we define the expected utility as

$$U(G; \boldsymbol{\mu}) \coloneqq \mathbb{E}\left[\sum_{m \in \mathcal{M}} r_m\right] = \sum_{m \in \mathcal{M}} \mu_{m,k_m^G} \mathbb{1}\{\forall m' \neq m : k_{m'}^G \neq k_m^G\},$$

where k_m^G denotes the matched arm of agent m under matching G. We denote the matching with the highest utility as the optimal matching $G^* \coloneqq \max_{G \in \mathcal{G}} U(G; \boldsymbol{\mu})$. We assume that G^* is *unique*, i.e., there does not exist any $G \neq G^*$ in $\mathcal{G}$ such that $U(G; \boldsymbol{\mu}) = U(G^*; \boldsymbol{\mu})$.

Hint mechanism At each time slot t, besides the pulled arm $k_m(t)$, agent m can query another arm $k_m^{\text{hint}}(t)$ and observe the arm's reward realization $X_{m,k_m^{\text{hint}}(t)}(t)$ without regret cost. The hint graph then is denoted by $G^{\text{hint}}(t)$ and $k_m^{G^{\text{hint}}(t)}$ is the arm agent m queried a hint for in it. These hint observations do not impact the accumulative reward and regret, and the agent can decide whether to query for a hint, denoted by the indicator function $\ell_m^\pi(t) \coloneqq \mathbb{1}\{\text{agent } m \text{ query a hint at } t \text{ under policy } \pi\}$. We denote $L^\pi(T) \coloneqq \mathbb{E}\left[\sum_{m \in \mathcal{M}} \sum_{t=1}^T \ell_m^\pi(t)\right]$ as the total number of times of agents querying hints, and we want to design a learning policy π minimizes the $L^\pi(T)$ while maintaining low regret.

Regret. We aim to find a policy π that maximizes the cumulative reward of all agents by determining $G(t)$ at each time step for T rounds. To evaluate the performance of π, we define the *regret* of a policy as the difference between the total reward of all agents under the optimal matching G^* in all decision rounds and the total reward of all agents following the policy π, as follows,

$$R^\pi(T) \coloneqq \sum_{t=1}^T U(G^*; \boldsymbol{\mu}) - \mathbb{E}\left[U(G(t); \boldsymbol{\mu})\right], \tag{1}$$

where the expectation is taken over the randomness of the policy π. Last, we define the important parameter, the minimum gap, which is crucial and appears in our regret analysis. The *minimum gap* here represents the minimum difference between the utility of any matching G and G^*, i.e., $\Delta_{\min}^{\text{match}} \coloneqq \min_{G \neq G^* \in \mathcal{G}} U(G^*; \boldsymbol{\mu}) - U(G; \boldsymbol{\mu})$.

Main goal and motivating examples Our goal is to design learning policies that use hints—one per agent at a time—to reduce the large regret bounds established in previous works (Shi et al. 2021; Mehrabian et al. 2020; Wang et al. 2020; Boursier and Perchet 2019) to a preferably time-independent regret, while minimizing the number of hints queried. We assume that hints are sampled from the same distributions as the rewards from pulling arms. Our algorithms query these hints strategically, only when exploring a sub-optimal matching is necessary before committing to the optimal one. This approach minimizes the costs of direct exploration and improves performance by separating the exploration of sub-optimal matchings from the exploitation of the optimal one.

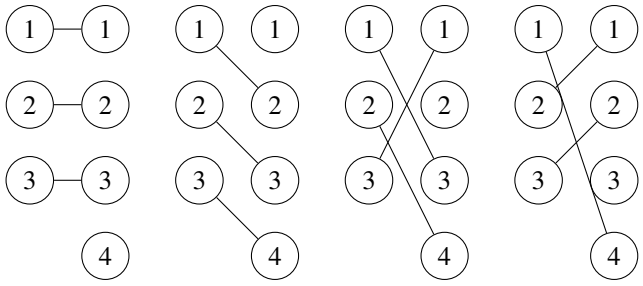
In practical scenarios, hints are typically much cheaper than direct actions. For instance, in labor markets, a low-cost interview process can provide valuable insights into candidate suitability without the high costs of hiring mistakes. Similarly, in communication networks, using test signals to estimate bandwidth needs can prevent wasting high-quality channels on low-demand stations. These examples demonstrate how the hint-based approach in `HMA2B` can improve decision-making across various applications.

3 Algorithms for Centralized Hinted Heterogeneous Multi-Armed Bandits

In the *Centralized Hinted Heterogeneous Multi-Armed Bandit* (`C_HMA2B`) setup, we consider an *omniscient* decision maker who selects both the matching and the hint graph at each round. The agents then follow the decision maker's instructions to pull arms and query hints. We propose two learning policies for this setup: the *Hinted Centralized Learning Algorithm* (`HCLA`) and the *Generalized Projection-based Hinted Centralized Learning Algorithm* (`GP-HCLA`).

Under both policies, the decision maker treats each matching $G \in \mathcal{G}$ as a super arm for hint inquiries. However, the handling of observations differs between the two: in `HCLA`, observations are maintained for each matching, while in `GP-HCLA`, they are treated at the edge level. This distinction allows us to reduce the potentially exponential regret relative to the size of $\mathcal{G}$ to a polynomial regret upper bound in the number of edges, MK.

We first introduce the statistics maintained by agents in `HCLA` and `GP-HCLA`, aiding the central decision maker in deciding when and how to query hints. Next, we describe `HCLA` as a baseline for designing `GP-HCLA`, our main algorithm. We also present an intermediate algorithm, `G-HCLA`,



(a) R_1 (b) R_2 (c) R_3 (d) R_4

Figure 1: Set of covering matchings $\mathcal{R}$ for $M = 3$ and $K = 4$: R_1, R_2, R_3 and R_4 are depicted in (a), (b), (c) and (d).

as a direct extension of HCLA. Finally, we detail GP−HCLA, which requests hints more efficiently than G−HCLA. G−HCLA is further discussed in Appendix A.

3.1 Preliminaries

Beyond the generic empirical means matrix $\hat{\boldsymbol{\mu}}$, the decision maker employs kl−UCB indices $\boldsymbol{d}$ (Cappé et al. 2013) as upper confidence bounds for $\boldsymbol{\mu}$ in the C_HMA2B setup to determine when to query a hint. These indices are defined as:

$$\hat{\mu}_G(t) \coloneqq \frac{\sum_{t'=1}^{t} \mathbb{1}\{G(t') = G\} U(G(t'); \boldsymbol{r}(t'))}{N_G^{\pi}(t)}, \tag{2}$$

$$\hat{\mu}_{m,k}(t) \coloneqq \frac{\sum_{t'=1}^{t} \mathbb{1}\{(m,k) \in G(t')\} r_m(t')}{N_{m,k}^{\pi}(t)}, \tag{3}$$

$$d_G(t) \coloneqq \sup\left\{q \geq 0 : N_G^{\pi}(t)\, \mathrm{kl}\left(\hat{\mu}_G(t), q\right) \leq f(t)\right\}, \tag{4}$$

$$d_{m,k}(t) \coloneqq \sup\left\{q \geq 0 : N_{m,k}^{\pi}(t)\, \mathrm{kl}\left(\hat{\mu}_{m,k}(t), q\right) \leq f(t)\right\}, \tag{5}$$

for any matching $G \in \mathcal{G}$ and edge $(m,k) \in \mathcal{M} \times \mathcal{K}$, respectively, where kl is the Kullback-Leibler divergence, and $f(t) = \log t + 4 \log\log t$. Here, $N_G^{\pi}(t)$ and $N_{m,k}^{\pi}(t)$ represent the number of times matching G or edge (m,k) has been pulled or hinted.

Before detailing the algorithm, we define a fixed set $\mathcal{R} \coloneqq \{R_1, \ldots, R_K\}$ of K pairwise edge-disjoint matchings that cover all edges $(m,k) \in \mathcal{M} \times \mathcal{K}$, referred to as *covering matchings*. For uniquely labeled agents and arms in $[M]$ and $[K]$, $R_i \in \mathcal{R}$ is the matching where agent m is paired with arm $(m + i - 1) \bmod K$, as shown in Figure 1 for $M = 3$ and $K = 4$. By pulling or hinting each covering matching in $\mathcal{R}$ at least once, agents can observe all $G \in \mathcal{G}$ at least once. This set serves as a *hint pool*, from which all hint graphs G^{hint} will be selected.

3.2 Warm-up: The HCLA Algorithm

As noted earlier, the HCLA algorithm treats each $G \in \mathcal{G}$ as a super arm and maintains separate statistics: empirical mean $\hat{\mu}_G(t)$, kl−UCB index $d_G(t)$, and counters $N_G^{\text{HCLA}}(t)$. At each time step t, the central decision maker selects a matching $G(t)$ with the maximum empirical mean $\hat{\mu}_G(t)$ and another matching $G'(t)$ with the maximum $d_G(t)$ (Lines 3–4). If $d_{G'(t)}(t) > \hat{\mu}_{G(t)}(t)$, the decision maker chooses $G^{\text{hint}}(t)$ as either $G'(t)$ or a uniformly at random chosen matching $G_2^{\text{hint}}(t)$, each with probability $1/2$. It then queries a hint from

Algorithm 1: Hinted Centralized Learning Algorithm (HCLA)

Input: agent set $\mathcal{M}$, arm set $\mathcal{K}$, number of agents M, matching set $\mathcal{G}$, time horizon T

1: **Initialization:** $t \leftarrow 0$, $\hat{\mu}_G(t) \leftarrow 0$, $d_G(t) \leftarrow 0$, $N_G(t) = 0$ for each matching $G \in \mathcal{G}$
2: **for** $t \in [T]$ **do**
3: $\quad G(t) \leftarrow \arg\max_{G \in \mathcal{G}} \hat{\mu}_G(t)$
4: $\quad G'(t) \leftarrow \arg\max_{G \in \mathcal{G}} d_G(t)$
5: $\quad$ **if** $d_{G'(t)}(t) > \hat{\mu}_{G(t)}(t)$ **then**
6: $\quad\quad G_1^{\text{hint}}(t) \leftarrow G'(t)$
7: $\quad\quad G_2^{\text{hint}}(t) \leftarrow$ pick a matching out of $\mathcal{G}$ uniformly at random
8: $\quad\quad G^{\text{hint}}(t) \leftarrow \begin{cases} G_1^{\text{hint}}(t), & \text{w.p. } \frac{1}{2} \\ G_2^{\text{hint}}(t), & \text{w.p. } \frac{1}{2} \end{cases}$
9: $\quad\quad$ Each agent m asks for a hint from $k_m^{G^{\text{hint}}(t)}$
10: $\quad\quad$ Update $\hat{\mu}_{G^{\text{hint}}(t)}(t+1)$ according to the observation of $G^{\text{hint}}(t)$
11: $\quad$ Each agent m pulls $k_m^{G(t)}$
12: $\quad$ Update $\hat{\mu}_{G(t)}(t+1)$ and according to the reward observation of $G(t)$
13: $\quad$ Update $d_G(t+1)$ for all $G \in \mathcal{G}$

$G^{\text{hint}}(t)$ and updates $\hat{\mu}_{G^{\text{hint}}(t)}(t+1)$ based on the hint observation (Lines 5–10). Finally, the decision maker pulls $G(t)$, updates $\hat{\mu}_{G(t)}(t+1)$ with the reward observation, and recalculates $d_G(t+1)$ for all $G \in \mathcal{G}$ (Lines 11–13). The detailed pseudocode of the HCLA algorithm is provided in Algorithm 1.

Next, we present the upper bounds for the time-independent regret and the number of queried hints $L^{\text{HCLA}}(T)$ for the HCLA algorithm in Theorem 1. The detailed proof is presented in Appendix B.1.

Theorem 1. *For $0 < \delta < \frac{\Delta_{\min}^{match}}{2}$ and policy $\pi = HCLA$, the policy π has*

1. *time-independent regret $R^{\pi}(T) \in O\left(MK^{2M}\right)$,*
2. *hint complexity $L^{\pi}(T) \in O\left(\frac{MK^M \log T}{\Delta^{\text{kl}}}\right)$,*

where $\Delta^{\text{kl}} = \mathrm{kl}(U(G^*; \boldsymbol{\mu}) - \Delta_{\min}^{match} + \delta, U(G^*; \boldsymbol{\mu}) - \delta)$.

The regret of HCLA is time-independent, but the exponential constants in its regret and hint upper bounds are unsatisfactory. To address this, we propose a new algorithm called GP−HCLA, which provides a more refined analysis while maintaining the same hint inquiry and arm-pulling approach but using observations differently.

3.3 The GP−HCLA Algorithm

We present GP−HCLA in Algorithm 2. The GP−HCLA algorithm follows steps similar to HCLA to identify $G(t)$ and $G'(t)$. However, unlike HCLA, the central decision maker maintains statistics $\hat{\mu}_{m,k}(t)$ and $d_{m,k}(t)$ for each edge $(m,k) \in \mathcal{M} \times \mathcal{K}$. It then defines $d_G(t) \coloneqq \sum_{(m,k) \in G} d_{m,k}(t)$, with a slight abuse of notation, enabling the use of the Hungarian algorithm (Kuhn 1955), which finds the matching with maximum additive utility in a

Algorithm 2: Generalized Projection-based Hinted Centralized Learning Algorithm (GP−HCLA)

Input: agent set $\mathcal{M}$, arm set $\mathcal{K}$, time horizon T,

1: **Initialization:** $t \leftarrow 0$, $\hat{\boldsymbol{\mu}}_m(t) \leftarrow \mathbf{0}$, $\boldsymbol{d}_m(t) \leftarrow \mathbf{0}$, $\boldsymbol{N}_m^{\text{GP-HCLA}}(t) \leftarrow \mathbf{0}$ for each agent $m \in \mathcal{M}$
2: **for** $t \in T$ **do**
3: $G(t) \leftarrow \texttt{Hungarian}\,(\hat{\boldsymbol{\mu}}(t))$
4: $G'(t) \leftarrow \texttt{Hungarian}\,(\boldsymbol{d}(t))$
5: **if** $U(G'(t); \boldsymbol{d}(t)) > U(G(t); \hat{\boldsymbol{\mu}}(t))$ **then**
6: $(m,k) \leftarrow \arg\min_{(m',k') \in G'(t)} N_{m',k'}^{\text{GP-HCLA}}(t)$
7: $G_1^{\text{hint}}(t) \leftarrow \{R \in \mathcal{R} : (m,k) \in R\}$
8: $G_2^{\text{hint}}(t) \leftarrow$ pick a matching out of $\mathcal{R}$ uniformly at random
9: $G^{\text{hint}}(t) \leftarrow \begin{cases} G_1^{\text{hint}}(t), & \text{w.p. } \frac{1}{2} \\ G_2^{\text{hint}}(t), & \text{w.p. } \frac{1}{2} \end{cases}$
10: Each agent m asks for a hint from $k_m^{G^{\text{hint}}(t)}$
11: Each agent m pulls $k_m^{G(t)}$
12: Update $\hat{\boldsymbol{\mu}}_m(t+1)$, $\boldsymbol{N}_m^{\text{GP-HCLA}}(t+1)$, and $\boldsymbol{d}_m(t+1)$ for each agent m according to new observations

weighted bipartite graph. Accordingly, GP−HCLA utilizes Hungarian to compute $G(t)$ and $G'(t)$, where the weights of the edges (m,k) are $\hat{\mu}_{m,k}(t)$ and $d_{m,k}(t)$, respectively (Lines 3–4). The decision maker employs a distinctly different approach from HCLA for selecting $G^{\text{hint}}(t)$ and updating the statistics after each observation, whether from pulling an arm or querying a hint. As in HCLA, the algorithm queries for a hint if $U(G'(t); \boldsymbol{d}(t)) > U(G(t); \hat{\boldsymbol{\mu}}(t))$ (Line 5). However, instead of querying a hint directly from $G'(t)$, GP−HCLA *projects* $G'(t)$ onto a matching in $\mathcal{R}$, a set of pairwise edge-disjoint covering matchings. By projection, we mean mapping $G'(t) \in \mathcal{G}$ to a matching in $\mathcal{R}$, which contains K covering matchings and is exponentially smaller. During this step, the algorithm selects the matching $G_1^{\text{hint}}(t)$ from $\mathcal{R}$ that contains the edge $(m,k) \in G'(t)$ with the fewest $N_{m,k}^{\text{GP-HCLA}}(t)$, and a second matching $G_2^{\text{hint}}(t)$, chosen uniformly at random from $\mathcal{R}$. The hint graph $G^{\text{hint}}(t)$ is then set to either $G_1^{\text{hint}}(t)$ or $G_2^{\text{hint}}(t)$, each with probability $1/2$ (Lines 6–10).

In Theorem 2, we provide the bound for the regret and the asymptotically optimal bound for the number of hints. The detailed proof is presented in Appendix B.2.

Theorem 2. *For $0 < \delta < \frac{\Delta_{\min}^{match}}{2}$ and policy $\pi =$ GP−HCLA, the policy π has*

1. *time-independent regret $R^\pi(T) \in O\left(M^4 K\right)$ regret,*
2. *hint complexity $L^\pi(T) \in O\left(\frac{MK \log T}{\Delta^{\text{kl}}}\right)$,*

where $\Delta^{\text{kl}} = \text{kl}(U(G^*; \boldsymbol{\mu}) - \Delta_{\min}^{match} + \delta, U(G^*; \boldsymbol{\mu}) - \delta)$.

Theorem 2 highlights the impact of maintaining edge-level statistics in GP−HCLA, reducing the exponential time-independent regret bound to a polynomial. It also shows that projection in hint inquiries minimizes hints, achieving asymptotic optimality (matching the lower bound given by Theorem 7 in the Appendix). We study G−HCLA, an extension of HCLA, which updates statistics like GP−HCLA but skips projection, using $G^{\text{hint}}(t)$ as in HCLA. Theorem 6 (Appendix) shows G−HCLA can have up to M-times higher hint complexity than GP−HCLA, highlighting the importance of projection. Experiments (Appendix H, Figure 3b) confirm GP−HCLA outperforms G−HCLA on small problem instances. The exact tightness of this gap remains open due to the complexity of the kl−UCB index.

4 Algorithms for Decentralized Hinted Heterogeneous Multi-Armed Bandits

We study the *Decentralized Hinted Heterogeneous Multi-Armed Bandits* (D_HMA2Bs), where no central decision maker coordinates agents to avoid collisions while learning the optimal matching G^*. Theorem 3 demonstrates that sub-linear regret is unattainable in a decentralized setup without agents sharing statistics, making communication essential in D_HMA2Bs. To enable communication, agents intentionally collide to exchange statistics like $\hat{\boldsymbol{\mu}}$s, while non-colliding agents continue pulling their assigned arms k_m^G from the matching $G \in \mathcal{G}$ without interference (Shi et al. 2021; Wang et al. 2020). Communication order is determined by unique agent ranks, as discussed below.

Theorem 3 (Necessity of Communication)**.** *No decentralized learning algorithm can achieve sub-linear instance-independent regret in HMA2Bs without communication.*

Building on Theorem 3, we propose a cooperative learning framework for the *Hinted Decentralized Explore-then-Commit* (HD−ETC) and *Elimination-Based Hinted Decentralized Explore-then-Commit* (EBHD−ETC) algorithms. These divide time into *Initialization*, *Exploration*, and *Communication* phases, where agents request hints in a round-robin manner until meeting a *stopping condition*, after which they transition to the *Exploitation* phase with no further hints or communication.

4.1 Decentralized Learning Framework

We first outline the common framework for the HD−ETC and EBHD−ETC algorithms. Both divide the T decision-making rounds into alternating exploration and communication phases. A counter ρ tracks exploration epochs, and $N_{m,k}^\rho$ records the number of times agent m has pulled or hinted at arm k by the start of epoch ρ. The decentralized learning framework for HMA2B consists of four phases:

Initialization phase: Assigning unique ranks among the agents. The detailed rank assignment procedure and analysis follows Wang et al. (2020) (detailed in Appendix D).

Exploration phase: Agents use the gathered statistics to identify the best matching G^ρ at the start of each epoch ρ using Hungarian algorithm. They then commit to their corresponding arm $k_m^{G^\rho}$ for K rounds until the epoch ends. At the end of epoch ρ, agents signal the communication phase by creating collisions on arms pulled by other agents.

Communication phase: Before each exploration epoch ρ, agents transmit their statistics $\hat{\boldsymbol{\mu}}$ to others, denoted as $\hat{\boldsymbol{\mu}}^\rho$. This communication is realized via the intentional collision signals, where a collision represents a '1' and its absence a '0' information bits (Boursier and Perchet 2019), with agents

relying on their unique ranks to identify senders and receivers. Since $\hat{\boldsymbol{\mu}}^{\rho}$ often contains decimal values, agents transmit a quantized version, $\tilde{\boldsymbol{\mu}}^{\rho}$, optimized for binary communication at the cost of minor information loss. To further reduce communication length and minimize information loss, agents employ the *Differential* communication (Shi et al. 2021), sending only the differences $\tilde{\boldsymbol{\delta}}^{\rho} = \tilde{\boldsymbol{\mu}}^{\rho} - \tilde{\boldsymbol{\mu}}^{\rho-1}$ at the start of epoch ρ. This method reduces communication-induced regret through $t^{\rho}_{\text{com}} \in O(M^2K)$ communication rounds. It enables agents to synchronize actions and exchange critical information efficiently via the $\text{Send2All}(\tilde{\boldsymbol{\delta}}^{\rho})$ routine, detailed in Appendix E.

Exploitation phase: Agents stop communicating, exploring, and querying for hints after a specific time T_0^{π}, which depends on the policy π being used. After that, they agree on a matching G'^{*} and commit to it for the rest of the time.

Unlike Shi et al. (2021) employing exponentially increasing exploration epoch lengths summing to $O(\log T)$ epochs, our approach simplifies this by assigning each epoch the same length K, resulting in potentially $O\left(\frac{T}{K}\right)$ epochs. However, with stop conditions, our algorithms reduce the number of epochs and transition to the exploitation phase while maintaining $O(\log T)$ exploration epochs.

Hint inquiry mechanism `HD-ETC` and `EBHD-ETC` employ a round-robin approach for querying hints, setting $G^{\text{hint}}(t) = R_{(t\%K)+1}$, and follow an Explore-then-Commit exploration style. By evenly distributing hint queries over K rounds, this method reduces communication costs and prevents time-dependent regret. In comparison, the decentralized `HCLA` queries hints on demand, requiring constant communication and potentially incurring linear regret.

Regret decomposition We decompose the regret as follows to analyze its components separately:

$$R^{\pi}(T) = R^{\pi_{\text{rank}}}(T) + R^{\pi_{\text{exp}}}(T) + R^{\pi_{\text{com}}}(T),$$

where $R^{\pi_{\text{rank}}}(T)$, $R^{\pi_{\text{exp}}}(T)$, and $R^{\pi_{\text{com}}}(T)$ represent the regret due to 'rank assignment,' 'exploration,' and 'communication,' respectively, under policy π.

Under this framework, we introduce the `HD-ETC` and `EBHD-ETC` algorithms in the following sections.

4.2 Warm-Up: The `HD-ETC` algorithm

The `HD-ETC` algorithm builds on the learning framework in Section 4.1, extending the Explore-then-Commit (`ETC`) method in bandits literature. To follow this method, agents uniformly query hints for covering matchings $R \in \mathcal{R}$, Lines 9–11, until time step $T_0^{\texttt{HD-ETC}}$, determined by the assumed knowledge of $\Delta^{\text{match}}_{\min}$. At $T_0^{\texttt{HD-ETC}}$ where ρ is the index of the last exploration epoch, agents run $\texttt{Hungarian}\,(\tilde{\boldsymbol{\mu}}^{\rho})$ to identify the matching G'^{*}, which they commit to for all $t > T_0^{\texttt{HD-ETC}}$, i.e., $G(t) = G'^{*}$ (Lines 18–20).

Theorem 4 establishes that with a properly chosen $T_0^{\texttt{HD-ETC}}$, which depends on $\Delta^{\text{match}}_{\min}$, the algorithm achieves time-independent exploration regret while ensuring asymptotically optimal hint and communication usage. Detailed proofs are provided in Appendix F.1.

Algorithm 3: Hinted Decentralized Explore then Commit (`HD-ETC`) : agent m

Input: agent m, agent set $\mathcal{M}$, arm set $\mathcal{K}$, number of agents M, time horizon T, time threshold for hint inquiry $T_0^{\texttt{HD-ETC}}$

1: **Initialization:** $t \leftarrow 0$, $\rho \leftarrow 0$, $\hat{\boldsymbol{\mu}}_m(t) \leftarrow 0$, $\boldsymbol{N}_m^{\texttt{HD-ETC}}(t) \leftarrow 0$, $\tilde{\boldsymbol{\mu}}^{\rho}_{m'} \leftarrow 0$ for each $m' \in \mathcal{M}$
2: **while** $t < T_0^{\texttt{HD-ETC}}$ **do**
3: **for** each epoch ρ **do**
4: $G^{\rho} \leftarrow \texttt{Hungarian}\,(\tilde{\boldsymbol{\mu}}^{\rho})$
5: $t_0 \leftarrow t$
▷ `Exploration Phase`
6: **for** $t \le t_0 + K$ **do**
7: Pull the arm $k_m^{G^{\rho}}$
8: Update $\hat{\mu}_{m,k_m^{G\rho}}(t+1)$ and $N^{\texttt{HD-ETC}}_{m,k_m^{G\rho}}(t+1)$
▷ `Hint Inquiry`
9: $G^{\text{hint}}(t) \leftarrow R_{(t\%K)+1}$
10: Ask for a hint from $k_m^{G^{\text{hint}}(t)}$
11: Update $\hat{\mu}_{m,k_m^{G^{\text{hint}}(t)}}(t+1)$ and $N^{\texttt{HD-ETC}}_{m,k_m^{G^{\text{hint}}(t)}}(t+1)$
12: $t \leftarrow t+1$
▷ `Communication Phase`
13: **for** $k \in [K]$ **do**
14: $\tilde{\delta}^{\rho+1}_{m,k} \leftarrow \tilde{\mu}^{\rho+1}_{m,k} - \tilde{\mu}^{\rho}_{m,k}$
15: $\text{Send2All}(\tilde{\delta}^{\rho+1}_{\text{m,k}})$
16: $t \leftarrow t + t^{\rho+1}_{\text{com}}$
17: $\rho \leftarrow \rho + 1$
▷ `Exploitation Phase`
18: $G'^{*} \leftarrow \texttt{Hungarian}\,(\tilde{\boldsymbol{\mu}}^{\rho})$
19: **while** $t \le T$ **do**
20: Pull the arm $k_m^{G'^{*}}$

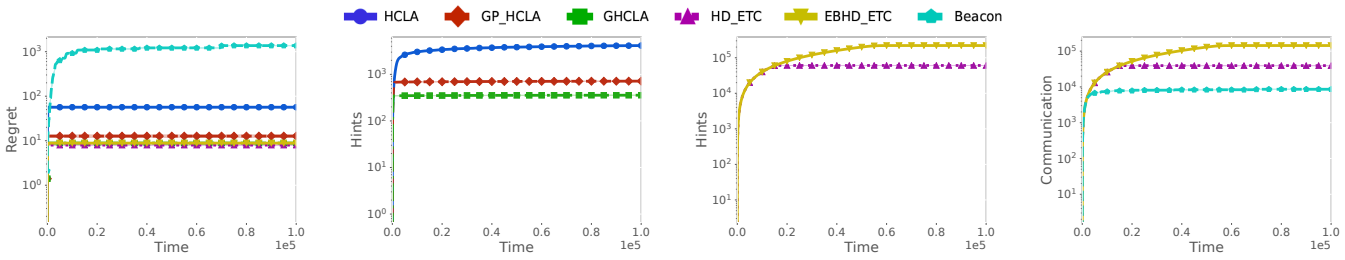
Theorem 4. *Assuming knowing the minimum gap $\Delta^{match}_{\min}$, for the policy $\pi =$ HD-ETC and $T_0^{\pi} = \frac{9M^2K\log(2MT)}{(\Delta^{match}_{\min})^2}$, the policy π has*

1. *exploration regret $R^{\pi_{exp}}(T) \in O\left(M^3K^2\right)$.*
2. *hint complexity $L^{\pi}(T) \in O\left(MT_0^{\pi}\right)$*
3. *communication regret $R^{\pi_{com}}(T) \in O\left(M^2T_0^{\pi}\right)$.*

Although `HD-ETC` performs well, it assumes agents know $\Delta^{\text{match}}_{\min}$, an unrealistic requirement in many settings. To overcome this, we propose `EBHD-ETC`, an elimination-based algorithm that achieves similar bounds without relying on the minimum gap. The simplicity of the Explore-then-Commit structure in `HD-ETC` necessitates knowledge of $\Delta^{\text{match}}_{\min}$ to determine the stopping time $T_0^{\texttt{HD-ETC}}$. Removing this assumption requires a more advanced algorithm design. In the next section, we introduce an elimination-based approach within the decentralized learning framework that operates without this gap assumption.

4.3 The `EBHD-ETC` algorithm

In `EBHD-ETC`, agents transition into the exploitation phase differently compared to `HD-ETC`. Accordingly, each agent maintains a set of active edges $\mathcal{C}^{\rho}$, which includes edges likely to be in G^{ρ} for the upcoming epoch ρ, initially



(a) Exploration Regret (b) Centralized Hint Complexity (c) Decentral. Hint Complexity (d) Communication Regret

Figure 2: Figure 2a plots $R^{\pi}(T)$ and $R^{\pi_{\text{exp}}}(T)$ for both centralized and decentralized setups. Figures 2b and 2c reflects the $L^{\pi}(T)$ for centralized and decentralized algorithms respectively and Figure 2d shows $R^{\pi_{\text{com}}}(T)$ for decentralized algorithms.

$\mathcal{C}^0 = \{(m,k) \in \mathcal{M} \times \mathcal{K}\}$. The set $\mathcal{C}^\rho$ maintained by each agent is the same across the others, determined by $\tilde{\boldsymbol{\mu}}$. Agents enter the exploitation phase when $|\mathcal{C}^\rho| = M$ (Line 2). Edge removal from $\mathcal{C}^\rho$ is guided by the error variable ϵ^ρ, defined as $\epsilon^\rho := \sqrt{\frac{\log\left(\frac{2}{\eta}\right)}{\rho}}$, with $\eta = \sqrt{\frac{2}{MT^2}}$. At the beginning of each epoch, agents determine G^ρ and check whether there exists a matching $G^\rho_{(m,k)} \neq G^\rho$ such that

$$U(G^\rho, \tilde{\boldsymbol{\mu}}^\rho) - U(G^\rho_{(m,k)}, \tilde{\boldsymbol{\mu}}^\rho) \leq 4M\boldsymbol{\epsilon}^\rho, \tag{6}$$

for all $(m,k) \in \mathcal{C}^\rho$. Each $G^\rho_{(m,k)}$ is constructed as $G^\rho_{(m,k)} := \{(m,k)\} \cup \texttt{Hungarian}\left(\tilde{\boldsymbol{\mu}}^\rho_{-m,-k}\right)$, where $\tilde{\boldsymbol{\mu}}^\rho_{-m,-k}$ is the matrix $\tilde{\boldsymbol{\mu}}^\rho$ with row m and column k removed.

Agents update $\mathcal{C}^{\rho+1}$ by removing edges (m,k) for which $G^\rho_{(m,k)}$ does not satisfy (6) (Lines 6–9).This process continues until the stopping condition $|\mathcal{C}^\rho| = M$ is met. At that point, agents transition to the exploitation phase, committing to G'^*, which corresponds to either $\texttt{Hungarian}\,(\tilde{\boldsymbol{\mu}}^\rho)$ or the matching formed by the edges in $\mathcal{C}^\rho$ (both refer to the same matching) for the remainder of the rounds (Lines 11–13).

Theorem 5. *The policy $\pi = \texttt{EBHD-ETC}$ has*

1. *exploration regret $R^{\pi_{exp}}(T) \in O\left(M^3K^2\right)$.*
2. *hint complexity $L^{\pi}(T) \in O\left(\frac{M^3K\log\left(T\sqrt{2M}\right)}{\left(\Delta^{match}_{\min}\right)^2}\right)$*
3. *communication regret $R^{\pi_{com}}(T) \in O\left(\frac{M^4K\log\left(T\sqrt{2M}\right)}{\left(\Delta^{match}_{\min}\right)^2}\right)$.*

Theorem 5 proves that EBHD-ETC achieves mostly similar but slightly weaker bounds in terms of coefficients compared to HD-ETC, even without the unrealistic assumption of knowing $\Delta^{\text{match}}_{\min}$. A proof is given in Appendix F.2.

5 Experiments

We executed the algorithms HCLA, GP-HCLA, G-HCLA, HD-ETC, and EBHD-ETC with $M = 4$, $K = 4$, and $\Delta^{\text{match}}_{\min} \leq 0.18$, averaging regret and hint complexity over 50 replications for 10^5 rounds. We also benchmarked the Beacon algorithm (Shi et al. 2021) and compared its regret $R^{\text{Beacon}_{\text{exp}}}(T)$ with ours in Fig. 2a, showing that our hint-augmented algorithms significantly reduce regret. Figs. 2b and 2c demonstrate that our centralized algorithms achieve better hint-query efficiency than HD-ETC and EBHD-ETC while maintaining similar regret. Fig. 2d highlights Beacon's advantage in communication regret due to its exponentially growing epoch lengths. These results confirm the effectiveness of centralized algorithms in balancing regret and hint complexity. Further extended experiments are presented in Appendix G.

Algorithm 4: Elimination-Based Hinted Decentralized Explore then Commit (EBHD-ETC) : agent m

Input: agent m, agent set $\mathcal{M}$, arm set $\mathcal{K}$, number of agents M, time horizon T

1: **Initialization:** $t \leftarrow 0$, $\rho \leftarrow 0$, $\hat{\boldsymbol{\mu}}_m(t) \leftarrow 0$, $\boldsymbol{N}^{\text{HD-ETC}}_m(t) \leftarrow 0$, $\tilde{\boldsymbol{\mu}}^\rho_{m'} \leftarrow 0$ for each $m' \in \mathcal{M}$, $\mathcal{C}^\rho \leftarrow \{(m,k) \in \mathcal{M} \times \mathcal{K}\}$
2: **while** $|\mathcal{C}^\rho| > M$ **do**
3: $\quad G^\rho \leftarrow \texttt{Hungarian}\,(\tilde{\boldsymbol{\mu}}^\rho)$
4: $\quad$ Execute Lines 6-16 of Algorithm 3
5: $\quad \mathcal{C}^{\rho+1} \leftarrow \emptyset$
▷ Updating Active Edges Set
6: $\quad$ **for** $(m,k) \in \mathcal{C}^\rho$ **do**
7: $\quad\quad G^\rho_{(m,k)} \leftarrow (m,k) \cup \texttt{Hungarian}\left(\tilde{\boldsymbol{\mu}}^\rho_{-m,-k}\right)$
8: $\quad\quad$ **if** $U(G^\rho; \tilde{\boldsymbol{\mu}}^\rho) - U(G^\rho_{(m,k)}; \tilde{\boldsymbol{\mu}}^\rho) \leq 4M\boldsymbol{\epsilon}^\rho$ **then**
9: $\quad\quad\quad \mathcal{C}^{\rho+1} \leftarrow \mathcal{C}^{\rho+1} \cup (m,k)$
10: $\quad \rho \leftarrow \rho + 1$
▷ Exploitation Phase
11: $G'^* \leftarrow \texttt{Hungarian}\,(\tilde{\boldsymbol{\mu}}^\rho)$ ▷ or $G'^* \leftarrow \mathcal{C}^\rho$
12: **while** $t \leq T$ **do**
13: $\quad$ Pull the arm $k^{G'^*}_m$

6 Conclusion

In this paper,we studied how hints enhance learning in the HMA2B problem, with heterogeneous rewards and collisions. We proposed both centralized and decentralized algorithms, analyzing their regret and hint usage. An interesting future work is extending these methods to two-sided matching markets, where both sides have preferences, and ties are broken due to them while collision occurs. This could enable the design of decentralized algorithms without communication to learn matchings with the same regret and hint optimality.

Acknowledgments

The work of Mohammad Hajiesmaili is supported by NSF CNS-2325956, CAREER-2045641, CPS-2136199, CNS-2102963, and CNS-2106299. The work of Jinhang Zuo was supported by CityU 9610706. Xuchuang Wang is the corresponding author.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anandkumar, A.; Michael, N.; Tang, A. K.; and Swami, A. 2011. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4): 731–745.

Bamas, E.; Maggiori, A.; and Svensson, O. 2020. The primal-dual method for learning augmented algorithms. *Advances in Neural Information Processing Systems*, 33: 20083–20094.

Bhaskara, A.; Cutkosky, A.; Kumar, R.; and Purohit, M. 2023. Bandit online linear optimization with hints and queries. In *International Conference on Machine Learning*, 2313–2336. PMLR.

Bistritz, I.; and Leshem, A. 2018. Distributed multi-player bandits-a game of thrones approach. *Advances in Neural Information Processing Systems*, 31.

Boursier, E.; and Perchet, V. 2019. SIC-MMAB: Synchronisation involves communication in multiplayer multi-armed bandits. *Advances in Neural Information Processing Systems*, 32.

Cappé, O.; Garivier, A.; Maillard, O.-A.; Munos, R.; and Stoltz, G. 2013. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 1516–1541.

François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M. G.; Pineau, J.; et al. 2018. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4): 219–354.

Garivier, A.; Ménard, P.; and Stoltz, G. 2019. Explore first, exploit next: The true shape of regret in bandit problems. *Mathematics of Operations Research*, 44(2): 377–399.

Jouini, W.; Ernst, D.; Moy, C.; and Palicot, J. 2009. Multi-armed bandit based policies for cognitive radio's decision making issues. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*, 1–6. IEEE.

Jouini, W.; Ernst, D.; Moy, C.; and Palicot, J. 2010. Upper confidence bound based decision making strategies and dynamic spectrum access. In *2010 IEEE International Conference on Communications*, 1–5. IEEE.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.

Lindståhl, S.; Proutiere, A.; and Johnsson, A. 2020. Predictive bandits. In *2020 59th IEEE Conference on Decision and Control (CDC)*, 1170–1176. IEEE.

Liu, K.; and Zhao, Q. 2010. Distributed learning in multi-armed bandit with multiple players. *IEEE transactions on signal processing*, 58(11): 5667–5681.

Lykouris, T.; and Vassilvitskii, S. 2021. Competitive caching with machine learned advice. *Journal of the ACM (JACM)*, 68(4): 1–25.

Mehrabian, A.; Boursier, E.; Kaufmann, E.; and Perchet, V. 2020. A practical algorithm for multiplayer bandits when arm means vary among players. In *Proceedings of the 23th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1211–1221. PMLR.

Mitzenmacher, M.; and Vassilvitskii, S. 2022. Algorithms with predictions. *Communications of the ACM*, 65(7): 33–35.

Purohit, M.; Svitkina, Z.; and Kumar, R. 2018. Improving online algorithms via ML predictions. *Advances in Neural Information Processing Systems*, 31.

Rosenski, J.; Shamir, O.; and Szlak, L. 2016. Multi-player bandits–a musical chairs approach. In *International Conference on Machine Learning*, 155–163. PMLR.

Shi, C.; Xiong, W.; Shen, C.; and Yang, J. 2021. Heterogeneous multi-player multi-armed bandits: Closing the gap and generalization. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 22392–22404.

Wang, P.-A.; Proutiere, A.; Ariu, K.; Jedra, Y.; and Russo, A. 2020. Optimal algorithms for multiplayer multi-armed bandits. In *Proceedings of the 2020 International Conference on Artificial Intelligence and Statistics (AISTATS)*, 4120–4129.

Wu, B.; Chen, T.; Ni, W.; and Wang, X. 2021. Multi-agent multi-armed bandit learning for online management of edge-assisted computing. *IEEE Transactions on Communications*, 69(12): 8188–8199.

Xu, X.; and Tao, M. 2020. Decentralized multi-agent multi-armed bandit learning with calibration for multi-cell caching. *IEEE Transactions on Communications*, 69(4): 2457–2472.

Xu, X.; Tao, M.; and Shen, C. 2020. Collaborative multi-agent multi-armed bandit learning for small-cell caching. *IEEE Transactions on Wireless Communications*, 19(4): 2570–2585.

Yun, D.; Proutiere, A.; Ahn, S.; Shin, J.; and Yi, Y. 2018. Multi-armed bandit with additional observations. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1): 1–22.