Online Learning with Probing for Sequential User-Centric Selection

Tianyi Xu [a,†,*], **Yiting Chen**[b,†], **Henger Li**[a], **Zheyong Bian**[c], **Emiliano Dall'Anese**[b] **and Zizhan Zheng**[a]

[a]Tulane University, United States
[b]Boston University, United States
[c]University of Houston, United States

Abstract. We formalize sequential decision–making with information acquisition as the Probing-augmented User-Centric Selection (PUCS) framework, where a learner first probes a subset of arms to obtain side information on resources and rewards, and then assigns K plays to M arms. PUCS encompasses practical scenarios such as ridesharing, wireless scheduling, and content recommendation, in which both resources and payoffs are initially unknown and probing incurs cost. For the offline setting (known payoff distributions), we present a greedy probing algorithm with a constant-factor approximation guarantee of $\zeta = (e-1)/(2e-1)$. For the online setting (unknown payoff distributions), we introduce OLPA, a stochastic combinatorial bandit algorithm that achieves a regret bound of $\mathcal{O}\big(\sqrt{T} + \ln^2 T\big)$. We also prove an $\Omega(\sqrt{T})$ lower bound, showing that the upper bound is tight up to logarithmic factors. Numerical results using two real-world datasets demonstrate the effectiveness of our solutions.

1 Introduction

Stochastic multi-armed bandits (MAB) [16] and Multi-player MABs (MP-MAB) [1] have been extensively studied as general frameworks for sequential decision making, where the goal is to maximize the cumulative reward by sequentially choosing from a set of options, or "arms" [16, 2]. These models have been applied in a variety of domains, including personalized recommendation [18], financial portfolio management [14], and dynamic resource allocation [1].

To cope with complex real-world scenarios, the user-centric selection problem has recently been introduced as an extension of MP-MABs [7]. Consider context recommendation as an example. The decision-maker recommends multiple pieces of content (mapped to arms in MP-MABs) and selects a subset of slots (plays) to allocate to these arms (assignment decision). Unlike traditional MP-MABs, the user-centric selection problem allows each arm to be associated with stochastic units of resources, and multiple plays can pull the same arm. Applications of the user-centric selection problem are vast and include personalized content delivery [18] and resource allocation in ridesharing platforms [7].

As another example, consider the user-centric selection problem in ridesharing [7]. Pickup locations correspond to arms, and drivers can be modeled as plays. There is a moving cost to a pickup location that is related to the reward. Multiple drivers can drive to the same location, but there is a limit on the maximum number of requests (resources) that can be hosted together at any location. This example clearly illustrates the user-centric selection problem, but in practice, some information is unknown. For instance, in this example, the distribution of passenger requests in a certain area is unknown. The real-time road conditions are also unknown. These unknown factors can significantly impact the accuracy of decision-making. To this end, the classic MAB framework relies purely on instantaneous feedback received after each decision round to obtain a desired exploration vs. exploitation trade-off.

In such an uncertain environment, probing is a promising approach for acquiring additional information when searching for the best alternative under uncertainty. It was initially studied in economics [25] and has found applications in database query optimisation [22, 10, 19]. Recently, it has been used to obtain real–time road traffic conditions (subject to a cost) before making vehicle-routing decisions [3], and for collecting link-quality information for scheduling in wireless access points [26, 27].

In this work, we propose a unified framework that integrates probing strategies with assignment decisions in the context of user-centric selection. We term this abstraction the Probing-augmented User-Centric Selection (PUCS) framework. PUCS explicitly couples the probing decision with the subsequent play-to-arm assignment, enabling us to quantify both the value *and* the cost of information acquisition. Concretely, in every round the decision maker first probes a budget-limited subset of arms to observe their resources and rewards, and then assigns the K plays to the M arms accordingly. In the offline case where the reward and resource distributions are known a priori, we derive a greedy probing algorithm that achieves a constant approximation factor by exploiting the submodularity of the objective function. For the more challenging online setting, we develop a combinatorial bandit algorithm and prove a regret bound of $\mathcal{O}\big(\sqrt{T} + \ln^2 T\big)$.

We remark that [30] also considers probing in a similar MP-MAB setting. However, they do not consider multiple resources for arms and assume that two or more plays assigned to the same arm can lead to a collision. In addition, they assume the rewards follow a Bernoulli distribution, while we consider general distributions. Further, they do not consider a probing budget as we do.

Contributions. The primary contributions of this work are as follows:

- We introduce the *PUCS* framework for jointly choosing which arms to probe and how to assign plays in user-centric selection

* Corresponding author. Email: txu9@tulane.edu.
† Equal contribution of the authors.

problems.

- For the offline setting with known distributions, we design a greedy probing algorithm that admits a constant-factor approximation guarantee $\zeta = (e-1)/(2e-1)$.
- For the online setting with unknown distributions, we propose a two-phase stochastic combinatorial bandit algorithm (OLPA) and establish a regret bound $\mathcal{O}\big(\sqrt{T} + \ln^2 T\big)$ together with an $\Omega(\sqrt{T})$ lower bound.
- Extensive experiments on real-world datasets demonstrate the empirical effectiveness of our approach over strong baselines.

2 Related Work

Sequential decision-making and online learning have been extensively studied in various contexts. Classical approaches such as multi-armed bandits [2, 4] provide foundational strategies for balancing exploration and exploitation. Recent advancements have extended these models to more complex scenarios, including combinatorial bandits [8], and multi-player MAB [15] which are more relevant to our sequential user-centric selection problems.

Sequential user-centric selection problems can be viewed as a variant and extension of the multi-player multi-armed bandit (MP-MAB) framework. MP-MAB models are designed to handle situations where multiple players compete for shared arms, requiring strategies that account for both coordination and competition among players [29, 21]. These models are particularly effective in scenarios like recommendation systems or resource allocation tasks, where multiple users interact with a common set of options. Unlike traditional MP-MABs, the user-centric selection problem allows each arm to be associated with stochastic units of resources and multiple plays can pull the same arm [7].

A ridesharing example is considered in [7] about the sequential user-centric selection problem. In this scenario, pickup locations can be mapped to arms, and drivers can be modeled as plays. The cost of moving to a pickup location is related to the reward, and the ride requests arriving at each arm can be modeled as the resource. Multiple drivers can choose to drive to the same location. However, this approach overlooks the unknown information of the real world, such as real-time traffic information, which can significantly impact decision-making.

To address these unknown factors in real-world systems, one can actively probe the environment and acquire side information before acting. Selecting which items to probe is, however, computationally intractable in general—the underlying decision problems are typically NP-hard [11]. Notably, the work of Golovin and Krause on adaptive submodularity [12] has provided theoretical guarantees for greedy algorithms in adaptive settings. This insight has been used in diverse applications, including active learning [13] and network monitoring [17], where limited probing gathers critical information before decisions are made.

Recent studies [3], have considered probing strategies at a certain cost to obtain information about multiple road conditions. However, they primarily focus on the probing itself without considering how to make decisions afterward. [26, 27] explored the use of probing to gather additional information before making decisions in the context of a wireless network access point serving users. However, their work is limited to a single-player setting and is only applied in the context of wireless networks. [30] proposed a general framework that incorporates observation (probing) into MP-MABs, but they only considered probing rewards from a Bernoulli distribution and could not apply their approach to sequential user-centric selection problems. In contrast, we introduce the PUCS framework, which jointly models probing and assignment in sequential user-centric selection problems, and we consider general distributions. Within this framework we design both offline and online algorithms and provide theoretical guarantees for each.

3 Probing-Augmented User-Centric Selection (PUCS) Framework

In this section, we formalize the Probing-augmented User-Centric Selection (PUCS) problem. We first describe the sequential user–centric model without probing and then show how probing is integrated on top of it.

3.1 Arm, Play, and Resource Model

Consider a sequential user-centric decision problem [7] over $T \in \mathbb{N}^+$ time slots, which is a variant of the classic multi-play multi-armed bandits problem. There are a set of $M \in \mathbb{N}^+$ arms and a set of $K \in \mathbb{N}^+$ plays. In each time slot, the decision is to assign the K plays denoted by $[K] := \{1, 2, \ldots, K\}$ to the arm set denoted by $[M] := \{1, 2, \ldots, M\}$. Each play can only be assigned to one arm and different plays can be assigned to the same arm.

Each arm $m \in [M]$ has $D_{t,m}$ units of resource in time slot t, where $D_{t,m}$ is an i.i.d. sample from a distribution with support $\{1, 2, \ldots, D_{\max}\}$. Let $\mathbf{p}_m = [p_{m,d} : \forall d \in \{1, 2, \ldots, D_{\max}\}]$ denote the probability mass function of $D_{t,m}$, where $p_{m,d} = \mathbb{P}[D_{t,m} = d]$. We define a probability mass matrix $\mathbf{P} \in \mathbb{R}^{M \times D_{\max}}$ with $\mathbf{P}_{m,d} = p_{m,d}$.

Each play assigned to an arm consumes one unit of resource for that arm. If the play k gets one unit of resource from arm m in time slot t, it receives a reward $R_{t,m,k}$, again an i.i.d. sample from an unknown distribution. We denote the expectation of $R_{t,m,k}$ as:

$$\mu_{m,k} = \mathbb{E}\left[R_{t,m,k}\right], \quad \forall t \in [T], m \in [M], k \in [K].$$

We define the reward mean matrix as $\boldsymbol{\mu} \in \mathbb{R}^{M \times K}$ with $\boldsymbol{\mu}_{m,k} = \mu_{m,k}$. We denote the m-th row of $\boldsymbol{\mu}$ by $\boldsymbol{\mu}_m$ and hence $\boldsymbol{\mu}_m = \begin{bmatrix} \mu_{m,1} & \ldots & \mu_{m,K} \end{bmatrix}$. We further let $\{F_{m,k}\}_{m \in [M], k \in [K]}$ denote the cumulative distribution function (CDF) of rewards for each arm-play pair. As shown below, $\{F_{m,k}\}$ (rather than just $\boldsymbol{\mu}$) are needed to evaluate the objective functions, because the objective functions depend on the entire distribution of rewards, not just their expected values.

Thus, the resource and reward associated with arm $m \in \{1, 2, \ldots, M\}$ is characterized by a list of the random vectors $(D_m, R_{m,1}, ..., R_{m,K})$, where $D_m = [D_{t,m} : \forall t \in \{1, 2, \ldots, T\}]$ and $R_{m,k} = [R_{t,m,k} : \forall t \in \{1, 2, \ldots, T\}], \forall k \in [K]$. We assume that $D_{t,m}$ and $R_{t,m,k}$, $\forall m \in [M], \forall k \in [K]$, are i.i.d. from each other and across time slots.

Let $C_{t,m} \subseteq [K]$ denote the collection of plays that pull arm m in time slot t. The action profile of all plays is denoted by $\mathbf{C}_t := \{C_{t,m} : m \in [M]\}$. Let $|C_{t,m}|$ denote the number of plays assigned to arm m in time t.

Below we give two examples of the above model.

- Ridesharing Services: Plays can be mapped to cars, arms to pickup locations, resources to passengers, and rewards to fares. The decision maker must decide which cars to send to which locations and multiple cars can be sent to the same location.
- Content Recommendation: Plays can be mapped to recommendation slots, arms to different content pieces, resources to user engagement (e.g., clicks or views), and rewards to the engagement level (e.g., watch time or read depth). The decision maker needs

to assign content to recommendation slots and the same piece of content can be assigned to multiple slots.

3.2 Probing Model

In the original user-centric selection problem [7], similar to the classic multi-armed bandits setting, the instantaneous reward associated with an arm and the number of resource units available at an arm is observed only when the arm is played (i.e., assigned to a play). A key observation of this paper is that probing can often be utilized to gather additional on-demand information in practice. For example, in ridesharing, probing can be used to check real-time traffic conditions [3] or get more accurate information about passenger availability and fare rates. Similarly, in content recommendation, probing in the form of A/B tests or surveys can be used to gauge user interest before making broader recommendations.

Probing differs fundamentally from prediction: it measures the current state directly (e.g., actual queue length, current link quality), avoiding model-drift errors that accumulate in time-varying systems. However, probes consume limited system resources such as driver detours, sensing bandwidth, or user attention. We therefore assume a per-round budget that restricts the learner to probe at most I arms, modeling the realistic "sense-some-but-not-all" constraint used in prior work on adaptive sensing and opportunistic measurement [3, 27]. Because the platform (e.g., Transportation Network Company (TNC) backend [5], recommender engine, WLAN controller) controls where to send drivers, which articles to A/B-test, or which channels to sense, it can decide which subset S_t of arms to probe in each round, consistent with existing deployments.

To this end, we assume that in each time slot t, the decision maker can probe a subset of arms $S_t \subseteq [M]$ before the arm-play assignment is made. For each probed arm $m \in S_t$, the decision maker immediately observes the realization of both $D_{t,m}$ and $R_{t,m,k}$, for all $k \in [K]$. This is a reasonable assumption in practice. For example, a single probe at a pickup location reveals its current vacancy and provides an estimate of the nearby traffic conditions, which determines the pickup cost.

Let $N_{t,m}$ denote the realization of $D_{t,m}$, and $X_{t,m,k}$ the realization of $R_{t,m,k}$, and let $\mathbf{X}_{t,m} := [X_{t,m,k} : \forall k \in [K]]$. In each time slot, the decision maker first probes a subset of arms $S_t \subseteq [M]$ and based on the values of $N_{t,m}$ and $\mathbf{X}_{t,m}$, $\forall\ m \in S_t$, the decision maker assigns K plays to M arms and each play gets a reward if it receives one unit of resource from the assigned arm.

We then derive the total rewards from an arm m by distinguishing two cases. First, assume that m is in the probing set S_t. If play k receives one unit of resource from arm m, the reward received by play k is the same as the probed value, which is $X_{t,m,k}$. That is, we assume that probing is accurate and the random variable $R_{t,m,k}$ stays the same within a time round. Similarly, we assume the random variable $D_{t,m}$ stays the same within a time round. That is, the decision maker observes the actual amount of resources, $N_{t,m}$, through probing if $m \in S_t$. If $N_{t,m} \geq |C_{t,m}|$, each play receives one unit of resource and the total rewards returned by arm m is $\sum_{k \in C_{t,m}} X_{t,m,k}$. If $N_{t,m} < |C_{t,m}|$, only $N_{t,m}$ plays get resource, and the remaining $|C_{t,m}| - N_{t,m}$ plays in $C_{t,m}$ do not get resources. We further assume that the plays associated with larger rewards are given priority. Formally, let $\mathbf{X}_{t,m}(C_{t,m}) := \{\mathbf{X}_{t,m,k} : k \in C_{t,m}\}$ and let $\mathbf{X}^{\text{sort}}_{t,m}$ be the sequence generated by sorting $\mathbf{X}_{t,m}(C_{t,m})$ from largest to smallest. Then the total rewards returned by arm m is $\sum_{1 \leq i \leq N_{t,m}} \mathbf{X}^{\text{sort}}_{t,m,i}$ if $N_{t,m} < |C_{t,m}|$. To summarize, if $m \in S_t$, the total rewards returned by arm m is

$$\mathcal{R}^{\text{prob}}_m(C_{t,m}; \mathbf{X}_{t,m}, N_{t,m}) = \sum_{i=1}^{j} X^{\text{sort}}_{t,m,i}$$
$$j := \min\{N_{t,m}, |C_{t,m}|\}$$

We then consider the case when arm m is not in the probing set S_t. In this case, if play k receives one unit of resource from arm m, the reward received by play k is a sample of $R_{t,m,k}$. Further, the value of $D_{t,m}$ (i.e., the actual number of resources of arm m in time slot t) is unknown, although we assume it is fixed within a time round. If $D_{t,m} \geq |C_{t,m}|$, each play receives one unit of resource, and the expected total rewards returned by arm m is $\sum_{k \in C_{t,m}} \mu_{m,k}$. If $D_{t,m} < |C_{t,m}|$, only $D_{t,m}$ units of resource are allocated, and $|C_{t,m}| - D_{t,m}$ plays in $C_{t,m}$ do not get resources. In this case, we assume that the plays associated with larger **expected** rewards are given priority. Formally, let $\boldsymbol{\mu}_m(C_{t,m}) := \{\mu_{m,k} : k \in C_{t,m}\}$ and $\boldsymbol{\mu}^{\text{sort}}_m$ be the sequence generated by sorting $\boldsymbol{\mu}_m(C_{t,m})$ from largest to smallest. Then the expected total rewards returned by arm m is $\sum_{1 \leq i \leq D_{t,m}} \boldsymbol{\mu}^{\text{sort}}_{m,i}$ if $D_{t,m} < |C_{t,m}|$. From the assumed independence between $D_{t,m}$ and $(R_{t,m,1}, ..., R_{t,m,K})$, it can be shown that the total expected reward returned by arm m is

$$\begin{aligned}
&\mathcal{R}_m(C_{t,m}; \boldsymbol{\mu}_m, \mathbf{p}_m) \\
:=&\mathbb{E}\left[\sum_{i=1}^{j} \boldsymbol{\mu}^{\text{sort}}_{m,i}\right],\ j := \min\{D_{t,m}, |C_{t,m}|\} \\
=&\sum_{d=1}^{|C_{t,m}|} \left(p_{m,d} \sum_{i=1}^{d} \boldsymbol{\mu}^{\text{sort}}_{m,i}\right) + \sum_{d=|C_{t,m}|+1}^{D_{\max}} p_{m,d} \left(\sum_{i=1}^{|C_{t,m}|} \boldsymbol{\mu}^{\text{sort}}_{m,i}\right).
\end{aligned}$$

by adapting a similar argument in [7]. In the special case that $\mu_{m,k_1} = \mu_{m,k_2}$ for any $k_1, k_2 \in [K]$, the above definition of $\mathcal{R}_m(C_{t,m}; \boldsymbol{\mu}_m, \mathbf{p}_m)$ reduces to the definition in [7].

To model the probing overhead, we further assume that up to I arms can be probed in a single time round and the final total rewards obtained for probing S_t and action profile $\mathbf{C}_t$ is

$$\mathcal{R}^{\text{total}}_t = (1 - \alpha(|S_t|)) \left(\sum_{m \in S_t} \mathcal{R}^{\text{prob}}_m + \sum_{m \in [M] \setminus S_t} \mathcal{R}_m\right)$$

where $\alpha : \{0, 1, ..., I\} \mapsto [0, 1]$ is a non-decreasing function and $\alpha(0) = 0$, $\alpha(I) = 1$. Here $\alpha(S)$ captures the probing overhead as the percentage of reward loss, which is monotonic with the size of S. This is appropriate because the probing impact often scales with the system's overall performance. In many scenarios, the opportunity cost of probing - such as time, resources, or energy - depends on the potential rewards that could have been achieved without probing. A similar approach has been adopted in [30, 27].

3.3 Objective and Regret

We then define the objective of our Probing-augmented User-Centric Selection (PUCS) problem. Recall that in each time slot t, the decision maker first picks a subset of arms S_t to probe and observes the instantaneous rewards and amount of resources for each arm in the probed set. It then determines the set of plays $C_{t,m}$ assigned to each arm m.

We first consider the offline case where $\mathbf{p}_m$ and the reward CDFs $\{F_{m,k}\}_{m \in [M], k \in [K]}$ are known a priori. In this case, it

suffices to consider each time slot separately and we drop the subscript t to simplify the notation. We first observe that when the probing set S is fixed and the observations from probing are given, the optimal assignment can be obtained. Formally, let $h_{\text{total}}(S, \{N_m, \mathbf{X}_m\}_m, \{\mathbf{p}_m, \boldsymbol{\mu}_m\}_{m\notin S})$ denote the maximum expected reward in a time slot with probing set S and the given realizations of rewards and resources. That is,

$$\begin{aligned} &h_{\text{total}}(S, \{N_m, \mathbf{X}_m\}_{m\in S}, \{\mathbf{p}_m, \boldsymbol{\mu}_m\}_{m\notin S}) \\ &:= \max_{\substack{C_m\subseteq[K]\\ m\in[M]}} \left(\sum_{m\in M\setminus S} \mathcal{R}_m(C_m; \boldsymbol{\mu}_m, \mathbf{p}_m) \right. \\ &\qquad \left. + \sum_{m\in S} \mathcal{R}_m^{\text{prob}}(C_m; \mathbf{X}_m, N_m) \right) \\ &\text{s.t.} \quad C_{m_1} \cap C_{m_2} = \emptyset,\ \forall m_1 \neq m_2,\ m_1, m_2 \in [M] \end{aligned} \tag{1}$$

where the constraints ensure that each play is assigned to at most one arm. This problem can be solved by adapting Algorithm 1 in [7] that considers the special case when S is empty. This is further elaborated in Observation 1 in the Appendix of the full version [28].

From the above discussion, the joint probing and assignment problem simplifies to finding an optimal probing set S. Let $f(S)$ denote the the optimal total expected reward that can be obtained for a given probing set S, that is

$$f(S) := \mathbb{E}_{\substack{X_{t,m,k}\sim R_{t,m,k}\\ N_{t,m}\sim D_{t,m}\\ m\in S,\ i\in[K]}} h_{\text{total}}\left(S, \{N_m, \mathbf{X}_m\}_m, \{\mathbf{p}_m, \boldsymbol{\mu}_m\}_{m\notin S}\right).$$

Then the offline problem is to finding a probing set S to maximize $R(S) := (1 - \alpha(|S|))f(S)$, that is,

$$S^* := \arg\max_{S\subseteq[M]} R(S) \tag{2}$$

Finding the optimal solution to this problem is hard for general reward and resource distributions. Fortunately, there is an efficient approximation algorithm with a constant approximation factor as we show in the next section by exploring the structure of $f(S)$. Specifically, we say that an offline algorithm is an ζ-approximation $(0 < \zeta \leq 1)$ if it achieves an expected $R(S_t)$ of at least $\zeta R(S_t^*)$.

In the more challenging online setting with unknown $\mathbf{p}_m$ and $\boldsymbol{\mu}_m$, we quantify the performance of a probing policy via the regret $\mathcal{R}_{\text{regret}}(T)$, which is defined as the difference between the cumulative reward of the optimal probing policy and the cumulative reward of our probing strategy. In this scenario, it is challenging to achieve sublinear regret with respect to the optimal S_t^* as commonly considered in the MAB literature. Instead, our objective is to achieve sublinear ζ-approximation regret, defined as follows [9]:

$$\mathcal{R}_{\text{regret}}(\zeta, T) = \sum_{t=1}^{T} \zeta R(S_t^*) - \sum_{t=1}^{T} R(S_t) \tag{3}$$

4 The Offline Setting

In the offline PUCS setting, the probability mass matrix $\mathbf{P}$ and the true CDF $\{F_{m,k}\}_{m\in[M],k\in[K]}$ of rewards for each arm-play pair are known. This offline case is an important contribution of our work, as it establishes the first constant-factor approximation guarantee and serves as the foundation for the online algorithm developed later. We aim to determine the optimal probing set that maximizes the expected total reward, which is in the equation (2). This maximization problem is computationally challenging; in fact, similar problems in related settings have been shown to be NP-hard in prior work [11]. Consequently, we design a surrogate objective function $f_{\text{prob}}(S)$ that captures the marginal value of probing. We further prove that this surrogate is both monotonic and submodular. Leveraging these properties, a greedy algorithm (Algorithm 1) provably achieves a constant-factor approximation to the optimal offline solution.

4.1 Offline Greedy Probing

For small problems, the optimal solution for the offline setting can be obtained by solving Problem (2) with an exhaustive search. For larger instances, however, a more efficient solution is needed. In this section, we derive a greedy algorithm that provides a constant factor approximation. One work [23] states that a simple greedy algorithm with an objective function that is monotone and submodular can obtain an approximation factor of an approximation factor of $1 - 1/e$. Based on this, we construct our objective function $f_{\text{prob}}(S)$ that is monotone and submodular shown in Lemma 3 and Lemma 4 and design an Offline Greedy Probing algorithm to obtain the approximation solution.

Let h_{prob} denote the value of the optimal assignment for the special case where all the plays can only select the probed arms. Formally, given a probing set S and the realization of rewards $\mathbf{X}_m$ and number of resources N_m, $h_{\text{prob}}(S, \{\mathbf{X}_m, N_m\}_{m\in S})$ is defined as

$$\begin{aligned} h_{\text{prob}} &:= \max_{\substack{C_m\subseteq[K],\\ m\in S}} \left(\sum_{m\in S} \mathcal{R}_m^{\text{prob}}(C_m; \mathbf{X}_m, N_m) \right) \\ &\text{s.t. } C_{m_1} \cap C_{m_2} = \emptyset,\ \forall m_1 \neq m_2, m_1, m_2 \in S \end{aligned} \tag{4}$$

h_{prob} has an optimal assignment policy based on Observation 1 in the Appendix of the full version [28]. We then define

$$f_{\text{prob}}(S) := \mathbb{E}_{\substack{X_{m,k}\sim R_{m,k}\\ N_m\sim D_m\\ m\in S,\ k\in[K]}} h_{\text{prob}}(S, \{\mathbf{X}_m, N_m\}_{m\in S}).$$

Here $R_{m,k}$ denotes a random reward whose CDF is $F_{m,k}$, so the expectation in $f_{\text{prob}}(S)$ is taken with respect to $F_{m,k}$; this is precisely how $\{F_{m,k}\}$ enter Algorithm 1. $f_{\text{prob}}(S)$ is monotone and submodular shown in Lemma 3 and Lemma 4, which is used for proving Theorem 1.

Similarly, we use f_{unprobed} to denote the value of the optimal assignment for the case where all the plays can only select the unprobed arms. Formally, given a probing set S , we have

$$\begin{aligned} f_{\text{unprobed}}(S) &:= \max_{\substack{C_m\subseteq[K],\\ m\in[M]\setminus S}} \left(\sum_{m\in M\setminus S} \mathcal{R}_m(C_m; \boldsymbol{\mu}_m, \mathbf{p}_m) \right) \\ &\text{s.t. } C_{m_1} \cap C_{m_2} = \emptyset,\ \forall m_1 \neq m_2, m_1, m_2 \in [M] \setminus S \end{aligned} \tag{5}$$

We note that if $S = \emptyset$, $R(S) = f(S) = f_{\text{unprobed}}(S)$. With these notations, we are ready to explain our greedy probing algorithm for the offline setting (Algorithm 1). It starts by initializing I empty sets S_i for $i = 0$ to $I - 1$, where I is the probing budget (lines 1-2). In each iteration from 1 to $I - 1$, the algorithm selects the arm m that, when added to the current set S_{i-1}, provides the maximum marginal increase in the expected reward function f_{prob} (line 4). The algorithm

Algorithm 1 Offline Greedy Probing

Input: $\mathbf{P}$, $\{F_{m,k}\}_{m\in[M],k\in[K]}$
Output: S^{pr}: the probing set

1: **for** $i = 0$ to $I-1$ **do**
2: $\quad S_i \leftarrow \emptyset$
3: **for** $i = 1$ to $I-1$ **do**
4: $\quad m \leftarrow \underset{m\in[M]\setminus S_{i-1}}{\operatorname{argmax}} \left(f_{\text{prob}}(S_{i-1}\cup\{m\}) - f_{\text{prob}}(S_{i-1})\right)$
5: $\quad S_i \leftarrow S_{i-1}\cup\{m\}$
6: $j \leftarrow \arg\max_i \left((1-\alpha(i))f_{\text{prob}}(S_i)\right)$
7: $S^{\mathrm{pr}} \leftarrow S_j$
8: **if** $(1-\alpha(j))f_{\text{prob}}(S_j) < f_{\text{unprobed}}(\emptyset)$ **then**
9: $\quad S^{\mathrm{pr}} \leftarrow \emptyset$

then updates the probing set S_i to include the newly selected arm (line 5). It determines the optimal probing set S^{pr} by selecting the set that maximizes the adjusted reward function $(1-\alpha(i))f_{\text{prob}}(S_i)$, where $\alpha(i)$ captures the probing cost as a function of the set size (lines 6-7). If the adjusted reward for the selected probing set S_j is less than the reward obtained without probing, $f_{\text{unprobed}}(\emptyset)$, then the final probing set S^{pr} is set to be empty (lines 8-9).

To show that the greedy algorithm is nearly optimal, we first establish the following properties. The proofs are given in the Appendix of the full version [28].

Lemma 1. *Given a probing set S, it holds that*

$$f(S) \le f_{prob}(S) + f_{unprobed}(S).$$

Lemma 2. *$f_{unprobed}(S)$ is monotonically decreasing, i.e., for any $S \subseteq T \subseteq [M]$, $f_{unprobed}(S) \ge f_{unprobed}(T)$.*

Lemma 3. *$f_{prob}(S)$ is monotonically increasing, i.e., for any $S \subseteq T \subseteq [M]$, $f_{prob}(S) \le f_{prob}(T)$.*

Lemma 4. *$f_{prob}(S)$ is submodular.*

Lemma 5. *Let S_i be the i-th probing set found by Algorithm 1 line 5 , $\tilde{S}^*$ be the one that maximizes $(1-\alpha(|S_i|))f(S_i)$ (Algorithm 1 line 6), S^{pr} be the final output of Algorithm 1 lines 7-9. Then it holds that $f_{unprobed}(\emptyset) \le R(S^{pr})$ and $(1-\alpha(|\tilde{S}^*|))f_{prob}(\tilde{S}^*) \le R(S^{pr})$.*

Theorem 1. *Let $S^* := \arg\max_{S\subseteq[M]} R(S)$. Algorithm 1 outputs a subset S^{pr} such that $R(S^{pr}) \ge \zeta R(S^\star)$ where $\zeta = \frac{e-1}{2e-1}$.*

Lemma 4 and Theorem 1 are our main contributions. In conclusion, our algorithm effectively balances the trade-off between the benefit of probing more arms and the associated costs, ensuring that the selected probing strategy obtains nearly optimal reward in the offline setting.

4.2 Time Complexity Analysis

For each of the I iterations (the probing budget), the algorithm computes the marginal gain in f_{prob} for up to M arms, with each evaluation taking $O(M)$ time and each time to compute f_{prob}, which is related to compute a maximum weighted matching needs $O((MK)^3)$. After the iterations, computing $f_{\text{unprobed}}(\emptyset)$ using [7, Algorithm 1] has a complexity of $O((MK)^3)$. The total complexity is $O(I\cdot M\cdot (MK)^3)$ for the iterations, plus $O((MK)^3)$ for the post-selection computation. If f_{prob} requires sample-based estimation for general distribution, the complexity becomes $O(IMW(MK)^3)$, where W is number of samples.

5 The Online Setting

The online case constitutes our second main contribution and complements the offline analysis. Here the probability–mass matrix $\mathbf{P}$, the reward mean matrix $\boldsymbol{\mu}$, and the true CDF $\{F_{m,k}\}_{m\in[M],k\in[K]}$ of rewards for each arm–play pair are unknown. The learner now faces two coupled challenges: (i) deciding where to probe under the probing overhead, and (ii) continually updating resource and reward estimates—together with their confidence bounds—so that the offline greedy routine (Alg. 1) can be invoked each round and the subsequent assignment remains optimistic-for-exploration. To address these challenges, we embed the offline greedy algorithm (Alg. 1) inside a two-phase online algorithm (OLPA) and derive regret guarantees under general reward distributions.

5.1 Parameters

We use the history from steps 1 to t to maintain four empirical summaries for each arm–play pair. The resource probability $\hat{p}^{(t)}_{m,d} = \frac{1}{t}\sum_{i=1}^{t}\mathbf{1}\{D_{m,i}=d\}$ (the empirical probability that arm m realizes resource level d), the observation count $n^{(t)}_{m,k} = \sum_{i=1}^{t}\mathbf{1}\{R_{i,m,k}\neq \text{null}\}\mathbf{1}\{k\in C_{i,m}\}$ (how many times (m,k) has been observed), the empirical mean $\hat{\mu}^{(t)}_{m,k} = \big(\sum_{i=1}^{t}\mathbf{1}\{R_{i,m,k}\neq\text{null}\}R_{i,m,k}\mathbf{1}\{k\in C_{i,m}\}\big)/n^{(t)}_{m,k}$ (average observed reward), and the empirical CDF $\hat{F}^{(t)}_{m,k}(x) = \big(\sum_{i=1}^{t}\mathbf{1}\{R_{i,m,k}\neq\text{null}\}\mathbf{1}\{R_{i,m,k}\le x\}\mathbf{1}\{k\in C_{i,m}\}\big)/n^{(t)}_{m,k}$ (distribution of observed rewards) are updated online for all $m\in[M]$, $k\in[K]$, $d\in[D_{\max}]$, and $x\in\mathbb{R}$. By [20, Lemma 9], we construct a confidence interval for empirical mean $\hat{\mu}^{(t)}_{m,k}$ with high probability as stated below.

Lemma 6. *For any $m\in[M]$, $k\in[K]$ and $\delta\in(0,1)$,*

$$\mathbb{P}\big[\forall t\ge 1,\ \mu_{m,k}-\hat{\mu}^{(t)}_{m,k}\ge\epsilon^{(t)}_{m,k}\big]\le\delta,$$

where $\epsilon^{(t)}_{m,k} = \sqrt{\big(1+n^{(t)}_{m,k}\big)\frac{\ln(\sqrt{n^{(t)}_{m,k}+1}/\delta)}{2\,(n^{(t)}_{m,k})^2}}$ if $n^{(t)}_{m,k}>0$ and $\epsilon^{(t)}_{m,k}=+\infty$ otherwise.

By incorporating the confidence interval, the algorithm can balance exploration and exploitation, ensuring that it does not overly rely on potentially inaccurate estimates of the rewards.

5.2 Online Learning for Joint Probing and Assignment (OLPA)

The contribution of our online algorithm (OLPA) (Algorithm 2) is to incorporate the greedy probing idea from Algorithm 1 to implement a two-phase update mechanism for joint probing and assignment. In the probing phase, the agent selects a subset of arms to probe based on the expected information gain within the budget (by invoking Alg. 1 on the current estimates). In the assignment phase, the agent assigns plays to arms optimally using the realized rewards/resources for probed arms and *UCB scores* (i.e., mean estimates combined with confidence bounds) for unprobed arms under the same feasibility constraints.

Initialization (lines 1-5): The algorithm initializes estimates for resource probabilities $\hat{p}^{(t)}_{m,d}$, mean rewards $\hat{\mu}^{(t)}_{m,k}$, play counts $n^{(t)}_{m,k}$, confidence bounds $\epsilon^{(t)}_{m,k}$, and CDFs $\hat{F}^{(t)}_{m,k}(\mathbf{x})$ for all arms and plays; unseen pairs start with $n^{(t)}_{m,k}=0$ (hence large $\epsilon^{(t)}_{m,k}$) to ensure valid UCB behavior at cold start.

Update Function (lines 6-14): The UPDATEESTIMATES function refines these estimates *after both probing and assignment* in each round, and refreshes $\epsilon_{m,k}^{(t)}$ according to Lemma 6 so that subsequent decisions remain guided by statistically valid confidence bounds.

Probing Phase (lines 16-17): At each time step t, the algorithm selects a probing set S_t^{pr} using current estimates, maximizing the surrogate marginal-value objective (Sec. 4) based on $\hat{\mathbf{P}}^{(t)}$ and $\{\hat{F}_{m,k}^{(t)}\}$, subject to the probing overhead/budget.

Assignment Phase (lines 19-20): After probing, the algorithm determines the play–arm assignment $\mathbf{C}_t$ that maximizes the expected total reward: realized outcomes from probed arms are combined with UCB scores derived from $\hat{\mu}_{m,k}^{(t)}$ and $\epsilon_{m,k}^{(t)}$ for unprobed arms, preserving feasibility and exploration.

Iteration: The process repeats over the time horizon T, with estimates updated after probing and after assignment each round; the regret guarantee in Theorem 2 is obtained under these updates.

Remark on novelty. OLPA departs from existing multi-play bandit methods in two key aspects: (i) it uses a custom UCB assignment rule whose confidence radius $\epsilon_{m,k}^{(t)}$ captures the joint uncertainty of rewards and remaining resources; (ii) it couples that UCB rule with our offline greedy probing routine, forming a unified two-phase policy that first selects which arms to probe under overhead and then assigns plays using UCB scores.

5.3 Theoretical Analysis

We provide a theoretical analysis of our online algorithm (OLPA), demonstrating that it achieves sublinear regret and near-optimal performance under reasonable assumptions. We have the following theorem.

Theorem 2. *The regret of Algorithm 2 is bound by*

$$\mathcal{R}_{\textit{regret}}(\frac{e-1}{2e-1}, T) \leq \frac{\sqrt{2}}{2} M\delta \left(\ln \frac{KT}{\delta}\right)^2 + 4D_{\max} K \left(\sum_{m\in[M]} \max_{k\in C_m} \mu_{m,k}\right) \sqrt{2\ln\frac{2}{\delta}}\sqrt{T}.$$

Comparison with the no-probing variant. We note that the constant in the $\sqrt{T}$-term of Theorem 2 arises from a loose upper bound used in the proof: $(1-\alpha(|S_t|)) \cdot \left(\sum_{m\in[M]\setminus S_t} |\mathbf{C}_m|\right) \cdot \left(\sum_{m\in[M]\setminus S_t} \max_{k\in C_m} \mu_{m,k}\right) \leq K \left(\sum_{m\in[M]} \max_{k\in C_m} \mu_{m,k}\right)$. The inequality is tight only when $S_t = \varnothing$ (probing disabled), so our worst-case guarantee coincides with that of a no-probing algorithm; whenever $|S_t| > 0$ the left-hand side is strictly smaller, implying a reduced constant bound while the $\widetilde{O}(\sqrt{T} + \ln^2 T)$ rate is preserved.

Experiments (Section 6) show that this reduced constant leads to clearly lower cumulative regret, confirming that probing delivers practical improvements even though both variants share the same asymptotic order. The regret scales linearly with K and logarithmically with M, which is consistent with multi-play bandit settings. The complete proof, incorporating probing and UCB estimate errors, is in the Appendix of the full version [28].

Lower bound. As a complement to the upper regret bound, the regret is lower bounded by $\Omega(\sqrt{T})$ in the worst-case scenario. The proof, based on a standard two-environment construction, is in the Appendix of the full version [28] and shows that our upper bound is tight up to constant factors.

Algorithm 2 Online Learning for Joint Probing and Assignment (OLPA)

1: $\hat{p}_{m,d}^{(t)} \leftarrow 0, \forall m \in [M], d \in [D_{\max}]$
2: $\hat{\mu}_{m,k}^{(t)} \leftarrow 0, \forall m \in [M], k \in [K]$
3: $n_{m,k}^{(t)} \leftarrow 0, \forall m \in [M], k \in [K]$
4: $\epsilon_{m,k}^{(t)} \leftarrow +\infty, \forall m \in [M], k \in [K]$
5: $\hat{F}_{m,k}^{(t)}(\mathbf{x}) \equiv 1, \forall x \in \mathbb{R}, m \in [M], k \in [K]$
6: **function** UPDATEESTIMATES(M, $\mathbf{X}_{t,m}$, $N_{t,m}$)
7: $\quad \hat{p}_{m,d}^{(t)} \leftarrow \frac{(t-1)\hat{p}_{m,d}^{(t-1)} + \mathbf{1}\{D_{m,t}=N_{t,m}\}}{t}$, $\forall m \in [M], d \in [D_{\max}]$
8: $\quad n_{m,k}^{(t)} \leftarrow n_{m,k}^{(t-1)} + \mathbf{1}\{k \in C_{t,m}\}$, $\forall m \in [M], k \in [K]$
9: $\quad \hat{\mu}_{m,k}^{(t)} \leftarrow \frac{(n_{m,k}^{(t-1)}\hat{\mu}_{m,k}^{(t-1)} + \mathbf{X}_{t,m}\mathbf{1}\{k\in C_{t,m}\})}{n_{m,k}^{(t)}}$, $\forall m \in [M], k \in [K]$
10: $\quad \hat{F}_{m,k}^{(t)}(\mathbf{x}) \leftarrow \frac{(n_{m,k}^{(t-1)}\hat{F}_{m,k}^{(t-1)}(\mathbf{x}) + \mathbf{1}\{R_{t,m,k}\leq \mathbf{X}_{t,m}\}\mathbf{1}\{k\in C_{t,m}\})}{n_{m,k}^{(t)}}$, $\forall x \in \mathbb{R}, m \in [M], k \in [K]$
11: $\quad$ **for** $m \in [M]$ **do**
12: $\quad\quad$ **for** $k \in [K]$ **do**
13: $\quad\quad\quad$ **if** $n_{m,k}^{(t)} \neq 0$ **then**
14: $\quad\quad\quad\quad \epsilon_{m,k}^{(t)} \leftarrow \sqrt{\left(1+n_{m,k}^{(t)}\right)\frac{\ln\left(\sqrt{n_{m,k}^{(t)}+1}/\delta\right)}{2\left(n_{m,k}^{(t)}\right)^2}}$
15: **for** $t = 1, 2, \ldots, T$ **do**
16: $\quad S_t^{\text{pr}} \leftarrow$ Algorithm 1 $\left(\hat{\mathbf{P}}^{(t)}, \{\hat{F}_{m,k}^{(t)}\}_{m\in[M],k\in[K]}\right)$
17: $\quad$ Probe each arm $m \in S_t^{\text{pr}}$ and get $(\mathbf{X}_{t,m}, N_{t,m})$
18: $\quad$ UPDATEESTIMATES(S_t^{pr}, $\mathbf{X}_{t,m}$, $N_{t,m}$)
19: $\quad$ Determine the optimal action profile $\mathbf{C}_t = \arg\max \mathcal{R}_t^{\text{total}}(S_t^{\text{pr}}, \{N_{t,m}, \mathbf{X}_{t,m}\}_{m\in S}, \{\hat{\mathbf{p}}_m^{(t)}, \hat{\boldsymbol{\mu}}_m^{(t)} + \boldsymbol{\epsilon}_m^{(t)}\}_{m\notin S}, \mathbf{C})$
20: $\quad$ Do assignment by the optimal action profile, get arm set S_{mat}, $(\mathbf{X}_{t,m}, N_{t,m})$
21: $\quad$ UPDATEESTIMATES(S_{mat}, $\mathbf{X}_{t,m}$, $N_{t,m}$)

6 Experiments

In this section, we evaluate our algorithm and compare it with baselines for the ridesharing problem. We utilize two real-world datasets: the NYYellowTaxi 2016 dataset [24] and the Chicago Taxi Trips 2016 dataset [6]. The NYYellowTaxi dataset contains trip records, including pickup and dropoff locations, passenger counts, and trip durations in New York City from 3/22/2016 to 3/31/2016. For the Chicago Taxi Trips 2016 dataset, we randomly select a subset of the data spanning from 1/9/2016 to 9/29/2016. These two datasets allow us to evaluate our approach across different urban environments.

Given the similarity between the two datasets, we apply the same data processing method for both. The geographical coordinates of pickup locations are discretized into bins of 0.01 degrees for both latitude and longitude. The frequency of passenger counts within these bins is normalized to derive the probability mass function (PMF), representing the distribution of passengers in each grid cell. Vehicle locations are randomly pre-sampled within the dataset bounds and fixed throughout the experiments. Figure 2 visualizes the resulting Manhattan street network (left) and one example environment with pre-sampled vehicles, pickup requests, and the discretization

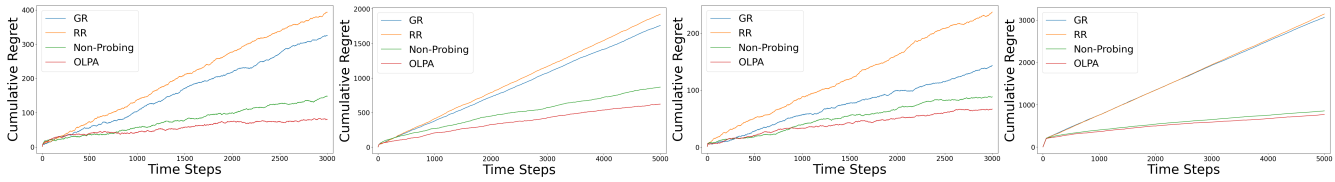

Figure 1: (a): Arms number $M = 3$, plays number $K = 2$, Bernoulli distribution for reward, $D_{\max} = 5$. (b): Arms number $M = 5$, plays number $K = 3$, Bernoulli distribution for reward, $D_{\max} = 7$. (c): Arms number $M = 3$, plays number $K = 2$, General distribution for reward, $D_{\max} = 5$. (d): Arms number $M = 10$, plays number $K = 6$, General distribution for reward, $D_{\max} = 7$. Data from NYYellowTaxi 2016.

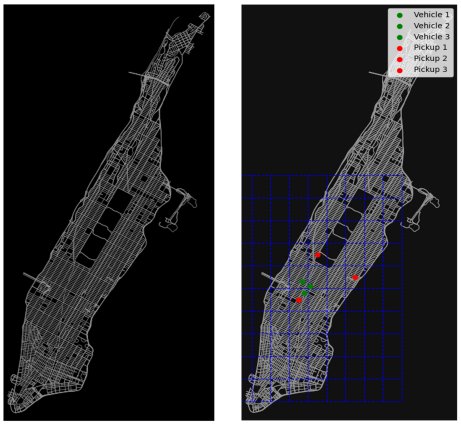

Figure 2: Visualization of the NYYellowTaxi dataset and the environment setting. Left: Manhattan road network used to sample vehicle and pickup locations. Right: an example of environment configuration.

Table 1: Cumulative regret at 1000/2000/3000 steps on the Chicago Taxi Trips dataset.

Set.	M	K	Distr.	Alg.	1000	2000	3000
(a)	3	2	Bernoulli	OLPA	67.67	112.09	151.40
				GR	117.79	220.55	309.40
				Non–Probing	87.84	122.22	170.62
				RR	120.16	223.61	327.28
(b)	5	3	Bernoulli	OLPA	133.15	225.79	292.60
				GR	330.30	655.82	1002.47
				Non–Probing	178.82	288.53	357.26
				RR	292.88	580.72	831.95
(c)	3	2	General	OLPA	37.71	72.86	94.27
				GR	184.84	361.42	538.00
				Non–Probing	54.01	89.28	118.10
				RR	183.80	359.63	537.37
(d)	10	6	General	OLPA	1961.19	3323.28	4357.97
				GR	1865.01	3546.49	5254.05
				Non–Probing	1967.15	3635.66	5275.73
				RR	1853.63	3545.97	5226.82

grid (right).

The reward is based on the Manhattan distance between a vehicle and a pickup location. The distances are normalized to the [0, 1] range, where closer distances correspond to higher rewards. We consider two types of reward distributions: (1) a Bernoulli distribution with mean $\mu_{m,k}$, representing the normalized reward for arm m and play k, assumed i.i.d. across t, m, and k; and (2) a discrete distribution with support $[0.1, 0.4, 0.7, 1.0]$, where probabilities for each level are derived from the distribution of normalized rewards. Cumulative regret is computed using Equation (3), where $R(S^\star)$ is obtained through exhaustive search. Specific details are provided in the Appendix of the full version [28].

6.1 Baselines

We compare OLPA with three baselines. **Non-Probing** (OnLin-ActPrf [7]) disables probing altogether and solves the assignment optimally from current estimates. **RR** (Random Probing, Random assignment) first samples a random number $i \leq I$ of arms, probes them uniformly at random, and then allocates plays by an independent uniform draw—capturing an uninformed exploration strategy. **GR** (Greedy Probing, Random assignment) uses the same greedy probing strategy as Algorithm 2 in our approach. However, instead of performing optimal assignment, it randomly selects arm-play pairs for assignment after probing.

6.2 Results

We compare all algorithms across different settings using the datasets described above. All four panels of Fig. 1 are generated from the NYYellowTaxi 2016 data. Fig. 1a gives the results when there are 3 locations and 2 vehicles, and the reward distribution is i.i.d. Bernoulli. After 800 time steps, our OLPA algorithm significantly outperforms the others, particularly GR and RR, which accumulate much higher regret. In Fig. 1b, we consider a slightly larger setting with 5 locations and 3 vehicles. As the horizon grows, the gap between RR and the other strategies widens, yet OLPA still maintains the lowest regret. Besides the Bernoulli distribution, we also evaluate under the general four-level reward distribution (Fig. 1c), where OLPA continues to lead, followed by Non-Probing and GR, with RR worst of all. Finally, in the largest setting (Fig. 1d) the performance gap between GR, RR, and the other methods widens even further, yet OLPA still achieves the lowest cumulative regret, demonstrating its robustness across problem scales and reward models.

Table 1 reports cumulative regret at 1 k, 2 k, and 3 k steps on the Chicago Taxi Trips dataset. In every configuration OLPA attains the smallest regret—e.g. in setting (b) it reduces regret by over 30% compared to Non-Probing and by a factor of 3+ relative to GR/RR—confirming that probing yields large constant-level gains in practice.

7 Conclusion

We introduce PUCS, a probing-augmented framework for sequential user-centric selection. In the offline case (known distributions) we give a greedy probing algorithm with a constant-factor guarantee; in the online case we couple this routine with a UCB-style assignment to obtain OLPA, whose regret is $\widetilde{O}(\sqrt{T})$ with a strictly smaller leading constant than the no-probing variant. Experiments on two large taxi-trip datasets show consistent gains over strong baselines, indicating that PUCS can benefit practical systems such as content recommendation and ridesharing dispatch.

References

[1] V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards. *IEEE Transactions on Automatic Control*, 32(11): 968–976, 1987.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[3] A. Bhaskara, S. Gollapudi, K. Kollias, and K. Munagala. Adaptive probing policies for shortest path routing. In *Advances in Neural Information Processing Systems*, 2020.

[4] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[5] California Public Utilities Commission. Decision 13-09-045: Decision adopting rules and regulations to protect public safety while allowing new entrants to the transportation industry. https://docs.cpuc.ca.gov/publisheddocs/published/g000/m077/k192/77192335.pdf, Sept. 2013. Rulemaking 12-12-011.

[6] H. Chauhan, N. Gupta, and Z. Haskell-Craig. Understanding human mobility patterns in chicago: an analysis of taxi data using clustering techniques. *arXiv preprint arXiv:2306.12094*, 2023.

[7] J. Chen and H. Xie. An online learning approach to sequential user-centric selection problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[8] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 151–159, 2013.

[9] W. Chen, W. Hu, F. Li, J. Li, Y. Liu, and P. Lu. Combinatorial multi-armed bandit with general reward functions. In *Advances in Neural Information Processing Systems*, 2016.

[10] A. Deshpande, L. Hellerstein, and D. Kletenik. Approximation algorithms for stochastic submodular set cover with applications to boolean function evaluation and min-knapsack. *ACM Transactions on Algorithms (TALG)*, 12(3):1–28, 2016.

[11] A. Goel, S. Guha, and K. Munagala. Asking the right questions: Model-driven optimization using probes. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 203–212, 2006.

[12] D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 2129–2137, 2011.

[13] A. Guillory and J. Bilmes. Interactive submodular set cover. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 415–422, 2010.

[14] X. Huo and F. Fu. Risk-aware multi-armed bandit problem with application to portfolio selection. *Royal Society open science*, 4(11):171377, 2017.

[15] J. Komiyama, J. Honda, and H. Nakagawa. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *International Conference on Machine Learning*, pages 1152–1161. PMLR, 2015.

[16] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[17] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007.

[18] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[19] Z. Liu, S. Parthasarathy, A. Ranganathan, and H. Yang. Near-optimal algorithms for shared filter evaluation in data stream systems. In *ACM SIGMOD*, 2008.

[20] O.-A. Maillard. *Basic Concentration Properties of Real-Valued Distributions*. Doctoral thesis, Université de Lille, France, 2017. HAL Id: tel-01632228.

[21] J. Mo and H. Xie. A multi-player mab approach for distributed selection problems. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 243–254. Springer, 2023.

[22] K. Munagala, S. Babu, R. Motwani, and J. Widom. The pipelined set cover problem. In *International Conference on Database Theory*, pages 83–98. Springer, 2005.

[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14:265–294, 1978.

[24] S. Shah, M. Lowalekar, and P. Varakantham. Neural approximate dynamic programming for on-demand ride-pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[25] M. L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3):641–654, 1979.

[26] T. Xu, D. Zhang, P. H. Pathak, and Z. Zheng. Joint ap probing and scheduling: A contextual bandit approach. In *IEEE Military Communications Conference (MILCOM)*. IEEE, 2021.

[27] T. Xu, D. Zhang, and Z. Zheng. Online learning for adaptive probing and scheduling in dense wlans. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2023.

[28] T. Xu, Y. Chen, H. Li, Z. Bian, E. Dall'Anese, and Z. Zheng. Online learning with probing for sequential user-centric selection. *arXiv preprint arXiv:2507.20112*, 2025. Full version of this paper.

[29] D. Zhou and C. Tomlin. Budget-constrained multi-armed bandits with multiple plays. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[30] J. Zuo, X. Zhang, and C. Joe-Wong. Observe before play: Multiarmed bandit with pre-observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.