Correlated Combinatorial Bandits for Online Resource Allocation

Samarth Gupta*, Jinhang Zuo*, Carlee Joe-Wong, Gauri Joshi and Osman Yağan
Carnegie Mellon University
Pittsburgh, PA, USA

ABSTRACT

We study a sequential resource allocation problem where, at each round, the decision-maker needs to allocate its limited budget among different available entities. In doing so, the decision-maker obtains the reward for each entity in that round. The goal of the decision-maker is to maximize the expected cumulative reward or equivalently minimize *cumulative regret* over a total of T rounds. Sequential resource allocation can be modeled as a combinatorial bandit by viewing the allocation of a budget to an entity as a base arm. In the context of resource allocation, the rewards received under different budget allocations are likely to be correlated. We propose a novel correlated combinatorial bandit framework that explicitly models such correlations. We develop a novel Correlated-UCB algorithm for online resource allocation, which yields significantly reduced regret relative to correlation-agnostic algorithms. In certain cases, our proposed algorithm even achieves bounded regret, which is an order-wise reduction in the regret relative to the correlation-agnostic approach, which incurs logarithmic regret under all scenarios. We validate these performance gains through experiments on several applications such as online power allocation across wireless channels, job scheduling in multi-server systems and online channel assignment for the slotted ALOHA protocol.

CCS CONCEPTS

• **Theory of computation → Online learning algorithms**; **Online learning theory**; **Regret bounds**.

KEYWORDS

Resource allocation, multi-armed bandits

ACM Reference Format:
Samarth Gupta*, Jinhang Zuo*, Carlee Joe-Wong, Gauri Joshi and Osman Yağan. 2022. Correlated Combinatorial Bandits for Online Resource Allocation. In *The Twenty-third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '22), October 17–20, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3492866.3549727

1 INTRODUCTION

1.1 Background and Motivation

Resource allocation is a fundamental challenge that arises in wide ranging applications, including wireless networks [1, 2], computer systems [3], multi-server scheduling [4] and financial optimization [5]. In the case of financial optimization, the company needs to decide the investment of its limited financial budget across different products with the goal of maximizing its overall revenue. In the context of power allocation in multi-channel wireless systems, the goal is to maximize the throughput of the system by allocating the power across different available channels. In such problems, the task is to distribute a limited *budget* (i.e., money, power, etc.) among available *entities* (i.e., product teams, channel etc.) with the objective of maximizing the *reward* attained (i.e., revenue, throughput, etc.). These budget allocation problem can be framed as

$$\begin{aligned} \underset{S=(a_1,a_2,\ldots a_K)}{\text{maximize}} \quad & \sum_{k=1}^{K} f_k(a_k) \\ \text{subject to} \quad & \sum_{k=1}^{K} a_k \le Q, a_k \in \mathcal{A}, \end{aligned} \tag{1}$$

with S being the budget allocation vector $(a_1, a_2, \ldots a_K)$ and Q representing the total available budget. The function $f_k(a_k)$ represents the reward attained from entity k upon allocating a budget of a_k to entity k. This budget is selected from a set $\mathcal{A}$, which may or may not be countable. Depending on the problem setting, the reward functions f_k may or may not be known. For instance, under the financial optimization example, the company distributes its total budget of Q among K different products with the goal of maximizing the total revenue, which is the sum of revenue $f_k(a_k)$ from individual products. In this example, the reward function $f_k(a_k)$ may not be known. In the power allocation problem for wireless systems, a total power of Q needs to be distributed across K different channels, and the throughput at each channel depends on the power allocated to that channel and is typically known as a function of the power allocated to the channels.

Moreover, in these problems, the reward obtained upon allocating a budget of a_k to entity k may be random and may depend on the underlying randomness associated with entity k. For instance, the revenue of the product may depend on the underlying unknown demand/market factors. Similarly, in the power allocation problem, with allocated power a_k, the throughput at channel k is $\log\left(1 + \frac{a_k}{X_k}\right)$, where X_k is the background noise associated with channel k and is random. As a result, the problem of budget allocation would now be

$$\begin{aligned} \underset{S=(a_1,a_2,\ldots a_K)}{\text{maximize}} \quad & \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right] \\ \text{subject to} \quad & \sum_{k=1}^{K} a_k \le Q, a_k \in \mathcal{A}. \end{aligned} \tag{2}$$

In this scenario, the optimization problem can be solved if $\mathbb{E}[f_k(a_k, X_k)]$ is known for all (a_k, k) pairs, i.e., the mean reward of each entity k is known at all budget allocations a_k for entity k. In view of this, we refer to (2) as the *offline budget allocation*

* indicates equal contribution.

MobiHoc '22, October 17–20, 2022, Seoul, Republic of Korea

ACM ISBN 978-1-4503-9165-8/22/10.
https://doi.org/10.1145/3492866.3549727

problem. In practice, the reward function may be unknown and the X_k's may be random variables with unknown distributions. For instance, in the financial optimization, the reward obtained for a given budget allocation a_k for product k may depend on underlying market conditions X_k, and one may not know the corresponding reward function f_k. As a result, $\mathbb{E}[f_k(a_k, X_k)]$ remains unknown. In the power allocation example, X_k corresponds to the background noise, which is a latent variable whose distribution is unknown, and correspondingly one does not know $\mathbb{E}[f_k(a_k, X_k)]$ a priori.

Motivated by this, we study the online resource allocation problem, where the goal is to sequentially decide a budget allocation $\boldsymbol{S_t} = (a_{1,t}, a_{2,t} \dots a_{K,t})$ for each round t, so as to maximize the cumulative reward attained over a total of T rounds. To perform this allocation, there is a need to estimate $\mathbb{E}[f_k(a_k, X_k)]$ for each (a_k, k) pair and subsequently use these estimates to decide a budget allocation $\boldsymbol{S_t}$ that generates the maximum possible reward in round t. When deciding a budget allocation $\boldsymbol{S_t}$, the decision-maker has two conflicting goals. Firstly, the allocation $\boldsymbol{S_t}$ should try to gather as much information as possible about the unknown reward distributions (exploration), and secondly the allocation should try to maximize the reward in each round (exploitation).

Resource allocation as a combinatorial bandit problem. In order to balance this exploration-exploitation trade-off, we can view the online resource allocation problem as a combinatorial multi-armed bandit (CMAB) problem, which is a variant of the classical multi-armed bandit (MAB) problem [6, 7]. Under the classical multi-armed bandit framework, the decision-maker is faced with M different base arms whose distributions are unknown and the goal is to maximize the long-term cumulative reward over a total of T rounds by selecting one amongst the available M base arms in each round t and observing its reward. Under the combinatorial bandit framework [8], the decision-maker can select multiple base arms in a given round from a given pre-defined set and observe the reward for each of the selected base arms. By viewing the allocation of budget a_k to entity k as a base arm (a_k, k), we can view the online resource allocation problem as a combinatorial bandit problem [8]. The underlying distribution of the reward of each base arm (a_k, k), i.e., $f_k(a_k, k)$, is unknown, and the goal is to maximize the cumulative reward over a total of T rounds by selecting K different base arms in each round t, i.e., one corresponding to each entity k. Upon the budget allocation, we receive rewards for all the base arms selected in round t, which is then used to decide the budget allocation in round $t + 1$. By modeling the resource allocation problem as a CMAB problem, we can use the existing CMAB algorithms to solve the resource allocation. However, these algorithms do not exploit the structural correlations in reward functions $f_k(a_k, X_k)$. Taking advantage of these correlations is the main challenge of our work.

1.2 Main Contributions

Novel correlated combinatorial bandit framework for online resource allocation. The combinatorial bandit framework described above considers the reward obtained for different base arms to be independent of each other. However, in the context of resource allocation, the rewards may be correlated in two ways. i) the rewards received for one entity k at budget i and for the same entity k at budget j are likely to be correlated. For instance, in the power allocation example, the throughput observed at channel k under power i gives some information on what the throughput would have been if power j were allocated to channel k. ii) the rewards received across two different entities may also be correlated. In the financial optimization example, the revenue obtained from product k under budget i may give some information on what the revenue would have been at product ℓ under budget j. This may occur if the sales of two products are related to one another. In this work, we model such correlations through *pseudo-rewards*, which are upper bounds on conditional expected reward of each base-arm (j, ℓ) given reward sample of base-arm (i, k). In the financial optimization example, this amounts to the knowledge of the form "what is the maximum revenue the company can expect from product ℓ at budget j given the observed revenue of product k under budget i?". The details of this framework are presented in Section 2.

Correlated and Combinatorial UCB. For this novel framework, we propose the correlated upper confidence bound algorithm for online resource allocation. It makes use of the correlations across base-arms to select an allocation $\boldsymbol{S_t}$ that balances the task of gaining information about the reward distributions of $f_k(a_k, X_k)$ for each (a_k, k) pair and maximizing the expected reward in round t based on the available information. More specifically, it computes an upper confidence bound on $\mathbb{E}[f_k(a_k, X_k)]$ for each (a_k, k) pair through the reward samples observed of $f_k(a_k, X_k)$ till round t. These reward samples may be obtained *directly* from the past reward samples of base arm (a_k, k) or *indirectly* through the pseudo-rewards of base arm (a_k, k) from the past reward samples of other base arms (j, ℓ). These upper confidence bounds on $\mathbb{E}[f_k(a_k, X_k)]$ are then used to select an allocation $\boldsymbol{S_t}$ to be played in round $t + 1$. The proposed algorithm is detailed in Section 3 of the paper. As our proposed approach makes use of the correlation information in the selection of $\boldsymbol{S_t}$, as opposed to prior work that is correlation-agnostic, we observe significant performance gains.

Reduction in cumulative regret through correlations. We evaluate our proposed algorithm in terms of the *cumulative regret*, which is defined as the difference between the total reward obtained by our online algorithm and the total reward obtained by the optimal offline solution, where the offline problem has complete knowledge about the joint distribution of X. We introduce novel proof techniques to analyze the regret, and show that the regret of our proposed algorithm is $C \cdot \mathrm{O}(\log T)$ where $0 \leq C \leq KA$, with A denoting the size of the set $\mathcal{A}$ from which budget a_k is allocated to each entity. We prove such results by jointly handling two key complexities, i) the correlations in reward across different base arms, and ii) selecting multiple base arms in a round to maximize the underlying objective function in the presence of budget constraints. The performance gains are a significant improvement over approaches that are agnostic to correlation [9], which have a regret of the form of $KA \cdot \mathrm{O}(\log T)$. In a lot of practical settings, $C = 0$, which implies that our proposed algorithm achieves a *bounded regret*. This is an order-wise improvement over correlation-agnostic approaches as shown in Section 4 of our paper. Our novel analysis technique, in particular Claim 1, is of independent interest as it can be used to analyse the generic combinatorial bandit framework [8] in an alternate manner.

Synthetic experiments on real-world problems. We validate the performance of our algorithm by evaluating it on three practical

problems in Section 6. We conduct experiments for i) the power allocation problem in wireless systems, ii) channel assignment in slotted ALOHA protocol and iii) scheduling of jobs in a multi-server system. For all the three problems, we see that using our correlated and combinatorial UCB algorithm achieves significant improvement in performance relative to correlation agnostic approaches.

1.3 Related Works

The classical offline resource allocation problem, i.e., the setting where the distributions of $f_k(a_k, X_k)$ are known, has been extensively studied for decades [1, 10, 11] and has been applied in several application settings such as financial optimization [5], wireless systems [1, 2], scheduling in multi-server systems [12], etc. Recently, the online resource allocation problem has attracted much attention as the distribution of rewards $f_k(a_k, X_k)$ is typically unknown in practice [9, 13–15]. First, the online resource allocation problem was studied in a setting where the reward functions $f_k(a_k, X_k)$ were assumed to be linear [13, 16]. This was extended by [15], as they assume the reward functions to be concave. More recently, [9] studied this problem in the most general setting by placing no restriction on the type of reward functions $f_k(a_k, X_k)$.

In [9], the online resource allocation is modeled as a combinatorial multi-armed bandit problem by viewing the allocation of budget a_k to entity k as a base arm. Subsequently, they extend the UCB algorithm for combinatorial bandits [17] to the online resource allocation problem. The action space $\mathcal{A}$ in [9] is allowed to be countable, unlike [17] which restricted the action space to be binary. A drawback of the approach in [9] is that it considers the rewards corresponding to different base arms to be independent of each other, and it does not make use of the fact that the reward obtained from one base arm may give some information on what the reward would have been for a different base arm.

In this paper, we fill this gap by proposing our correlated combinatorial bandit framework to study the online resource allocation in the most general setting. To the best of our knowledge, this is the first work that models the correlation in a *combinatorial* bandit framework. The idea of capturing correlations in reward across different arms was previously studied in the context of classical multi-armed bandits, i.e., the setting where only one base-arm is played in each round t, in [18, 19]. Another closely related line of work, where only one base-arm is played in each round, is that of structured bandits [20–22], where mean rewards corresponding to different base arms are related to one another through a hidden parameter θ. While mean rewards between different arms are related to one another in structured bandits, they are not necessarily correlated. Due to this, the correlated bandit framework [18, 19] fits better to the problem setting of online resource allocation where reward realizations are known to be correlated. We extend this idea of correlated bandits to the combinatorial bandit framework, where multiple base-arms may be played in each round t, and propose the correlated UCB algorithm for online resource allocation. The extension is non-trivial as the classical multi-armed bandit and combinatorial bandit often require different design of algorithms and regret analysis due to selection of multiple base arms within provided constraints as opposed to the selection of the single base arm in each round t. Upon doing so, we are able to exploit the correlations to obtain significant performance improvements as demonstrated in Sections 4 and 6. To the best of our knowledge, this is the first work to show that O(1) regret can be achieved in certain online resource allocation problems.

2 PROBLEM SETUP

2.1 Offline Resource Allocation

Consider the offline resource allocation problem where a decision-maker splits the available budget among K different entities. For each entity $k \in [K]$, the decision-maker needs to decide a budget $a_k \in \mathcal{A}$, where $\mathcal{A}$ is the feasible budget space. Notice that the budget space $\mathcal{A}$ could be either discrete (e.g., $\mathbb{N}$) or continuous (e.g., $\mathbb{R}_{\geq 0}$). We focus on the discrete action space first and then consider the case of continuous action space separately in Section 5. We denote the overall budget allocation vector as $\boldsymbol{S} = (a_1, \cdots, a_K)$. We consider a general reward function $f_k(a_k, X_k)$ for each entity k, where X_k (which can be discrete or continuous) is a hidden random variable that reflects the random fluctuation of the obtained reward within entity k. We also consider m general constraints, denoted as $h_i(\boldsymbol{S}) \leq 0, i = 1, 2, \cdots, m$.

For the offline setting where the distribution $\boldsymbol{D} = (D_1, \ldots, D_K)$ of $\boldsymbol{X} = (X_1, \ldots X_K)$ is known, our goal is to maximize the expected total reward collected from all entities, which we denote by $r(\boldsymbol{S}, \boldsymbol{D}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right]$. This problem can be formulated as

$$\begin{aligned} \underset{\boldsymbol{S}=(a_1,\cdots,a_K)}{\text{maximize}} \quad & \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right] \\ \text{subject to} \quad & a_k \in \mathcal{A}, \forall k \in [K]; h_i(\boldsymbol{S}) \leq 0, i = 1, 2, \cdots, m. \end{aligned} \tag{3}$$

The above formulation is a general version of (1) and (2), which contain just one constraint $h_1(\boldsymbol{S}) = \sum_k a_k - Q$. We could have more complex constraints on $\boldsymbol{S}$ through $h_i(\boldsymbol{S})$, e.g., $max_k a_k - W \leq 0$. These constraints on budgets are known to the decision-maker. For instance, if $f_k(a_k, X_k)$ is convex over a_k, $h_i(\boldsymbol{S})$ is convex over $\boldsymbol{S}$, and $\mathcal{A}$ is a convex set, it becomes a convex optimization problem that might be solved exactly; if $\mathcal{A}$ is a discrete set, it can be a NP-hard combinatorial optimization problem.

As the reward functions $f_k(\dot{)}$ may not be known in practice (e.g., the financial optimization example in Section 1), we do not specify the exact form of the reward functions $f_k(a_k, X_k)$ and consider them to be unknown. We assume that there exists an offline approximation oracle $\mathcal{A}$, which outputs an allocation $\mathcal{S}^O$ such that $r(\mathcal{S}^O, \boldsymbol{D}) \geq \alpha \cdot \text{opt}(\boldsymbol{D})$, where α is the approximation ratio and $\text{opt}(\boldsymbol{D}) = \sup_{\boldsymbol{S}} r(\boldsymbol{S}, \boldsymbol{D})$ is the optimal solution to the budget allocation problem. The oracle can output such an allocation if $\mathbb{E}[f_k(a_k, X_k)]$ is known for all $(a_k, k) \in \mathcal{A} \times \mathcal{K}$.

2.2 Online Resource Allocation as a Combinatorial Bandit Problem

Now we introduce the online version of the resource allocation, which is a sequential decision making problem. In each round t, we allocate $a_{k,t}$ budget to each entity k, subject to the budget constraints, $h_i(\boldsymbol{S}) \leq 0, i = 1, 2, \cdots, m$. We then obtain $f_k(a_{k,t}, X_{k,t})$ reward from each entity k, where $X_{k,t}$ is sampled from an unknown distribution D_k. The total reward obtained in round t is $\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})$. Our goal is to accumulate as much total reward as possible through this sequential budget allocation.

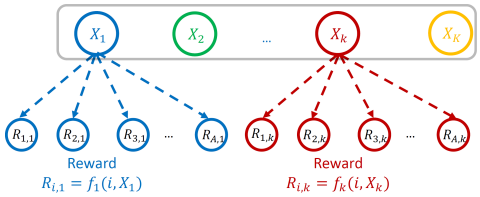


Figure 1: The rewards corresponding to a base arm (i,k), i.e., budget i to entity k, are a function of the allocated budget i and underlying randomness X_k associated with entity k. The rewards for base arms (i,k) and (j,k), i.e., different budget allocations within entity k, are correlated through X_k. There may be also correlation in the rewards across different entities if $X_1, X_2, \dots X_K$ are correlated.

We denote the overall budget allocation in round t as $\boldsymbol{S_t} = (a_{1,t}, a_{2,t}, \cdots, a_{K,t})$ and the joint distribution of all $X_{k,t}$'s as $\boldsymbol{D} = (D_1, D_2, \dots, D_K)$. We define the expected total reward obtained in round t as $r(\boldsymbol{S_t}, \boldsymbol{D}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right]$. We consider a learning algorithm π that makes the budget allocation $\boldsymbol{S_t^\pi}$ in round t. We can measure the performance of π by its (expected) regret, which is the difference in expected cumulative reward between always taking the best offline allocation and taking the budget allocation selected by algorithm π. The best offline allocation can be obtained through the offline oracle O, which knows the underlying joint distribution $\boldsymbol{D}$, and attains $r(\boldsymbol{S_t^O}, \boldsymbol{D}) \geq \alpha \cdot \text{opt}(\boldsymbol{D})$. In view of that, we use the following approximation regret for T rounds:

$$Reg_\alpha^\pi(T; \boldsymbol{D}) = T \cdot \alpha \cdot \text{opt}(\boldsymbol{D}) - \sum_{t=1}^{T} r(\boldsymbol{S_t^\pi}, \boldsymbol{D}). \tag{4}$$

Since the obtained reward $f_k(a_k, X_k)$ of entity k is determined by the allocated budget a_k, following the combinatorial multi-armed bandit framework [8], we can view allocating budget i to entity k as a base arm and denote it as (i,k). The overall budget allocation $\boldsymbol{S_t}$ can be considered as a super arm that consists of multiple base arms. For each base arm (i,k), we denote the expected reward of playing it as $\mu_{i,k} = \mathbb{E}_{X_{k,t} \sim D_k}\left[f_k(i, X_{k,t})\right]$. We can rewrite the expected total reward obtained in round t:

$$r(\boldsymbol{S_t}, \boldsymbol{D}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right] = \sum_{k=1}^{K} \sum_{i \in \mathcal{A}} \mu_{i,k} \cdot \mathbb{1}\{a_{k,t} = i\}, \tag{5}$$

Note that the expected total reward depends only on the mean rewards of base arms (i,k), therefore it can be re-written as

$$r(\boldsymbol{S_t}, \boldsymbol{\mu}) = \sum_{k=1}^{K} \sum_{i \in \mathcal{A}} \mu_{i,k} \cdot \mathbb{1}\{a_{k,t} = i\}. \tag{6}$$

If the mean rewards $\mu_{i,k}$ of individual base arms (i,k) were known, then one can use the offline oracle to obtain the optimal budget allocation in each round. As the mean rewards of individual base arms are unknown, they need to be estimated from the historical observations until round t. The mean reward of the base arm (i,k) can be estimated either through the past samples in which budget i was allocated to entity k, or through the side information collected from other observations. We discuss the latter next.

2.3 Proposed Correlated Combinatorial Bandit Framework

In several application settings, there may be some information on the knowledge of reward functions $f_k(a_k, X_k)$. As a result, the knowledge of the reward from one base arm (i,k) may provide some information on the reward that would have been obtained from entity k if budget j was allocated to entity k. This is illustrated in Figure 1. For instance, in the power allocation example, where the objective is to allocate the total power Q among K different channels to maximize the total throughput, the throughput at channel k is given by $\log\left(1 + \frac{a_{k,t}}{X_{k,t}}\right)$. Here, $a_{k,t}$ represents the power allocated in channel k and X_k denotes the hidden noise in channel k at round t. As the expression of throughput, i.e., the reward function $f(a_k, X_k)$, is known, the throughput in channel k at power i provides some information on what the reward would have been if power j was allocated to channel k. More generally, rewards obtained from one base arm (i,k) may provide some information on the reward of another base arm (j,ℓ). As a result, the rewards corresponding to different base arms are correlated. We capture the presence of such correlations in the form of *pseudo-rewards*, as defined below:

Definition 1 (Pseudo-Reward). *Suppose that we sample base arm (i,k) and observe reward r. We call a quantity $s_{(j,\ell),(i,k)}(r)$ as the pseudo-reward of base arm (j,ℓ) with respect to base arm (i,k) if it is an upper bound on the conditional expected reward of (j,ℓ), i.e.,*

$$\mathbb{E}[f_\ell(j, X_\ell) \mid f_k(i, X_k) = r\ ] \leq s_{(j,\ell),(i,k)}(r). \tag{7}$$

For convenience, we set $s_{(j,\ell),(j,\ell)}(r) = r, \forall j, \ell$.

When no information is known, pseudo-rewards between two base arms are not known, then they can be set equal to the maximum possible reward. This makes our formulation quite general and subsumes the correlation agnostic CMAB framework studied in [9]. The connection will be made explicit through Remark 1 in Section 3. Next, we show how the pseudo-rewards can be evaluated.

Obtaining pseudo-rewards from reward correlations within the same entity. These pseudo-rewards can be evaluated easily in several different practical settings. For instance, if the form of the functions $f_k(a_k, X_k)$ is known, then the pseudo-reward of base arm (j,k) with respect to base arm (i,k) can be obtained as

$$s_{(j,k),(i,k)}(r) = \max_x f_k(j, x) \quad \text{s.t. } f_k(i, x) = r. \tag{8}$$

Note that pseudo-rewards can be obtained even in the scenario where only probabilistic upper and lower bounds on $f_k(a_k, X_k)$ are known, i.e., $\underline{f}_k(a_k, X_k) \leq f_k(a_k, X_k) \leq \bar{f}_k(a_k, X_k)$ w.p. $1 - \kappa$. In this scenario, we can construct pseudo-rewards as follows:

$$s_{(j,k),(i,k)}(r) = (1-\kappa)^2 \times \left(\max_{\{X_k : \underline{f}_k(i,X_k) \leq r \leq \bar{f}_k(i,X_k)\}} \bar{f}_k(j, X_k) \right) + (1 - (1-\kappa)^2) \times M, \tag{9}$$

where M is the maximum possible reward that a base arm can provide. We evaluate this pseudo-reward by first identifying the range of values within which X_k lies based on the reward with probability $1 - \kappa$. The maximum possible reward of the base arm (j,k) within the identified range of X_k is then computed with probability $1 - \kappa$. Due to this, with probability $(1-\kappa)^2$, the conditional reward of

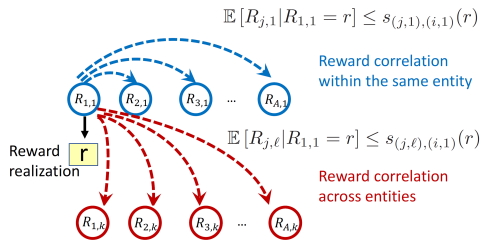


Figure 2: Upon observing a reward r from a base arm, pseudo-rewards $s_{(j,\ell),(i,k)}(r)$, give us an upper bound on the conditional expectation of the reward from base arm (j,ℓ) given that we observed reward r from arm (i,k). Reward received for entity k at a given budget i may provide some information on what the reward would have been if budget j were allocated to entity k, leading to correlations within entity. The rewards of different entities may also be correlated.

base arm (j,k) is at most $\max_{X_k: \underline{f}_k(i,X_k) \le r \le \bar{f}_k(i,X_k)} \bar{f}_k(j,X_k)$. As the maximum possible reward is M otherwise, we get (9).

Obtaining pseudo-rewards from reward correlation across entities. In the most general scenario, there may be knowledge of reward correlations across entities as shown in Figure 2. This can occur if the random variables X_k and X_ℓ, i.e., the hidden random variables corresponding to two different entities k and ℓ, are correlated. These correlations can be incorporated in our framework through pseudo-rewards $s_{(j,\ell),(j,k)}$, which are an upper bound on the conditional expected reward. For instance, in the application of financial optimization, the company may invest its total budget among different products. As the performance of different products are likely to be correlated, the reward feedback under budget i for product k may inform something about the reward feedback for product ℓ under budget j. Such correlations can be modeled through pseudo-rewards, which may either be known from domain knowledge or from previously performed controlled experiments. For example, based on previously performed experiments, it may be known that the expected reward obtained from product ℓ under budget j is at most y whenever the reward obtained for product k under budget i is x. Note that in this modeling, one does not need to explicitly capture what the inherent randomness X_k represents and its corresponding values. This is a key strength of our proposed framework, as in several applications X_k could be hard to interpret and model. For instance, in the financial optimization example, X_k may represent underlying market conditions, which are complex, and subsequently the reward functions $f_k(a_k,k)$ are also unknown. Even in such settings, the pseudo-reward based framework allows one to capture the correlation across different base arms.

3 PROPOSED ALGORITHM

We now propose the correlated-Upper Confidence Bound algorithm for resource allocation (corr-UCB-RA) that uses existing correlation in rewards across base arms to maximize the long-term cumulative reward. Before describing our algorithm, we first review the UCB algorithm for resource allocation (UCB-RA) proposed in [9].

3.1 The UCB Algorithm for Resource Allocation

In order to solve the online resource allocation problem, the UCB-RA algorithm maintains a set of base arms $\{(k,a) \mid k \in [K], a \in \mathcal{A}\}$, where the total number of base arms is equal to KA, with A denoting the size of the discrete set $\mathcal{A}$. If the mean reward of each base arm were known, then the resource allocation problem can be easily solved by the use of the available offline oracle O, which produces an allocation S_t^O such that $r(S_t^O, \boldsymbol{\mu}) \ge \alpha \cdot \text{opt}(\boldsymbol{\mu})$. As the underlying mean rewards of the base arms are unknown, the UCB-RA algorithm maintains the empirical mean $\hat{\mu}_{i,k}(t)$ for each base arm (i,k) at round t. Using these empirical means, it then computes an upper confidence bound (UCB) index for each base arm (i,k) as

$$U_{i,k}(t) = \hat{\mu}_{(i,k)}(t) + \sqrt{\frac{2\log t}{n_{(i,k)}(t)}},$$

where $n_{(i,k)}(t)$ denotes the number of times budget i was allocated to entity k. UCB-RA algorithm then feeds these upper confidence indices of the base arms to the available offline oracle (i.e., substitute $\boldsymbol{\mu}$) and obtains an allocation $S_t = (a_{1,t}, a_{2,t}, \ldots a_{K,t})$. It then uses this allocation for round t and observes the feedback of $f_k(a_{k,t}, X_{k,t}) \forall k$. Note that the upper confidence indices are large if base arm (i,k) has a large empirical mean reward or if it has been sampled fewer times relative to other base arms.

3.2 The Proposed Correlated-UCB Algorithm for Resource Allocation

Under the correlated combinatorial bandit framework, the pseudo-reward for base arm (j,ℓ) with respect to the base arm (i,k) provides an estimate on the reward of base arm (j,ℓ) based on the reward obtained from base arm (i,k). We now define the notion of empirical pseudo-reward, which can be used to obtain an *optimistic estimate* of $\mu_{(j,\ell)}$ through just reward samples of base arm (i,k).

Definition 2 (Empirical and Expected Pseudo-Reward). *After t rounds, a base arm (i,k) is sampled $n_{(i,k)}(t)$ times. Using $n_{(i,k)}(t)$ reward realizations, we construct the empirical pseudo-reward $\hat{\phi}_{(j,\ell),(i,k)}(t)$ for (j,ℓ) with respect to base arm (i,k) as follows.*

$$\hat{\phi}_{(j,\ell),(i,k)}(t) \triangleq \frac{\sum_{\tau=1}^{t} \mathbb{1}_{(i,k)\in\mathcal{S}_t}\, s_{(j,\ell),(i,k)}(f_k(i,X_{k,\tau}))}{n_{(i,k)}(t)}, \tag{10}$$

$$(j,\ell) \in \mathcal{K} \times \mathcal{A} \setminus \{(i,k)\}. \tag{11}$$

The expected pseudo-reward of (j,ℓ) with respect to (i,k) is

$$\phi_{(j,\ell),(i,k)} \triangleq \mathbb{E}\left[s_{(j,\ell),(i,k)}(f_k(i,X_k))\right]. \tag{12}$$

For convenience, we set $\hat{\phi}_{(i,k),(i,k)}(t) = \hat{\mu}_{(i,k)}(t)$ and $\phi_{(i,k),(i,k)} = \mu_{(i,k)}$. Note that the empirical pseudo-reward $\hat{\phi}_{(j,\ell),(i,k)}(t)$ is defined with respect to base arm (i,k) and it is only a function of the rewards observed by sampling base arm (i,k).

Definition 3 (PseudoUCB Index $U_{(j,\ell),(i,k)}(t)$). *We define the PseudoUCB Index of base arm (j,ℓ) with respect to base arm (i,k) as*

$$U_{(j,\ell),(i,k)}(t) \triangleq \hat{\phi}_{(j,\ell),(i,k)}(t) + \sqrt{\frac{2\log t}{n_{(i,k)}(t)}}. \tag{13}$$

The algorithmic blocks for UCB-RA and Corr-UCB-RA are presented in the appendix of the full paper at www.andrew.cmu.edu/user/gaurij/corr_comb_bandits.pdf.

Furthermore, we define $U_{(j,\ell)}(t) = \min_{(i,k)} U_{(j,\ell),(i,k)}(t)$*, the tightest of the KA upper bounds for base arm* (j,ℓ)*.*

At each round, the algorithm computes these pseudo-UCB indices $U_{(j,\ell)}$ for each base arm (j,ℓ). These indices are then fed to the oracle to obtain the budget allocation vector $\boldsymbol{S_t}$ at round t. At the end of each round we update the empirical pseudo-rewards $\hat{\phi}_{(j,\ell),(i,k)}(t)$ for all (j,ℓ), the empirical reward for arm $(i,k) \in \mathcal{S}_t$, where $\mathcal{S}_t$ denotes the set of base arms played in round t. The description of this algorithm is given in the appendix.

Remark 1 (Reduction to Combinatorial Multi-Armed Bandits). *When all pseudo-reward entries are unknown, then all pseudo-reward entries can be filled with the maximum possible reward for each base arm, that is,* $s_{(i,k),(j,\ell)}(r) = M\ \forall r, \ell, k, i, j$*. In that case, the proposed Corr-UCB-RA algorithm reduces to the UCB-RA algorithm.*

4 REGRET BOUNDS AND ANALYSIS

4.1 Main Results

We now characterize the performance of our proposed algorithm in terms of regret (See Eq. (4)). Here, $r(\boldsymbol{S_t^\pi}, \boldsymbol{D})$ represents the expected total reward obtained in round t, which can be written as,

$$r(\boldsymbol{S_t}, \boldsymbol{\mu}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right] = \sum_{k=1}^{K}\sum_{a\in\mathcal{A}} \mu_{i,k} \cdot \mathbb{1}\{i = a_{k,t}\}. \quad (14)$$

For the regret analysis, we assume without loss of generality that the rewards are between 0 and 1 for all base arms (i,k). Furthermore, we denote the oracle's optimal budget allocation vector as $\boldsymbol{S}^*$, i.e., the allocation vector that provides an α-optimal solution to the offline resource allocation problem, where $\mathbb{E}[f_k(a_k, X_k)]$ is known for all base arms. For simplicity, we assume that there is a unique solution $\boldsymbol{S}^*$ to the offline resource allocation problem. Correspondingly, we denote the set of base arms selected in $\boldsymbol{S^*}$ as the set of optimal base arms $\mathcal{S}^*$. To bound the regret, we rely on two properties of $r(\boldsymbol{S}, \mu)$.

Property 1. *(Monotonicity). The expected reward of playing any action* $\boldsymbol{S_t}$ *is monotonically increasing with respect to the expectation vector of base arms, i.e., if for all* $(i,k) \in \mathcal{A}\times\mathcal{K}$*, if* $\mu_{i,k} \le \mu'_{i,k}$*, then we have* $r(\boldsymbol{S_t}, \boldsymbol{\mu}) \le r(\boldsymbol{S_t}, \boldsymbol{\mu'})\ \forall \boldsymbol{S_t}$*.*

Property 2. *(Bounded Smoothness).* $\exists$ *an increasing function* $g(.)$ *such that, if* $\boldsymbol{S_t}$ *is the super-arm selected in round* t *and* $||\boldsymbol{\mu}_{\boldsymbol{S_t}} - \boldsymbol{\mu'}_{\boldsymbol{S_t}}||_\infty < \lambda$*, then*

$$|r(\boldsymbol{S_t}, \boldsymbol{\mu}) - r(\boldsymbol{S_t}, \boldsymbol{\mu'})| < g(\lambda).$$

Here, the infinity norm between $\boldsymbol{\mu}_{\boldsymbol{S_t}}$ *and* $\boldsymbol{\mu'}_{\boldsymbol{S_t}}$ *is defined as* $\max_{(i,k)\in\mathcal{S}_t} |\mu_{(i,k)} - \mu'_{(i,k)}|$ *with* $\mathcal{S}_t$ *denoting the set of base arms played in round* t*.*

It is easy to see that both properties hold from the definition of $r(\boldsymbol{S_t}, \boldsymbol{\mu})$ in Eq. (5). Before stating our main result for the correlated UCB algorithm, we first review the regret bound under the UCB-RA algorithm [9].

The full proofs and the intermediate Lemmas are available in the appendix of the full paper at www.andrew.cmu.edu/user/gaurij/corr_comb_bandits.pdf.

Lemma 1. *The regret for UCB-RA algorithm is upper bounded as*

$$Reg_\alpha(T, \boldsymbol{D}) \le \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \Delta_{min}^{(i,k)} \left(\frac{8\log T}{\left(g^{-1}\left(\Delta_{min}^{(i,k)}\right)\right)^2}\right) + 4KA\Delta_{max}$$
$$= KA \cdot \mathrm{O}(\log T) + \mathrm{O}(1), \quad (15)$$

$$\text{with } \Delta_{min}^{(i,k)} = r(\boldsymbol{S^*}, \boldsymbol{\mu}) - \max(r(\boldsymbol{S}, \boldsymbol{\mu}) | \boldsymbol{S} \in \mathcal{S}_B, (i,k) \in \mathcal{A}\times\mathcal{K}),$$
$$\Delta_{max}^{(i,k)} = r(\boldsymbol{S^*}, \boldsymbol{\mu}) - \min(r(\boldsymbol{S}, \boldsymbol{\mu}) | \boldsymbol{S} \in \mathcal{S}_B, (i,k) \in \mathcal{A}\times\mathcal{K}),$$
$$\Delta_{max} = \max_{(i,k)\in\mathcal{A}\times\mathcal{K}} \Delta_{max}^{(i,k)},$$

where $\mathcal{S}_B$ *is the set of all sub-optimal actions and* $\boldsymbol{S}^*$ *is the oracle's optimal allocation.*

The result follows from the intuition that after the UCB indices of all the base arms are relatively *close* to their true mean rewards, the algorithm selects the budget allocation $\boldsymbol{S}^*$ with high probability. Under the UCB-RA algorithm, each base arm needs to be sampled $\mathrm{O}(\log T)$ times to ensure that the UCB indices are *close* to their true means. Due to which, the regret of UCB-RA algorithm is of the form of $KA \cdot \mathrm{O}(\log T)$. We formalize this intuition for both the UCB-RA and our proposed Corr-UCB-RA algorithms through the following claim. This claim is a novel contribution of our work and it provides an alternative methodology to analyse the generic combinatorial bandit formulation [8] as well.

Claim 1. *If* $U_{(i,k)} \ge \mu_{(i,k)}, \forall (i,k) \in \mathcal{K}\times\mathcal{A}$ *and the UCB-RA and Corr-UCB-RA algorithms select a budget allocation* $\boldsymbol{S_t}$ *at round* t*,*

$$\mu_{(i,k)} \le U_{(i,k)} < \bar{\mu}_{(i,k)} \quad \forall (i,k) \in \mathcal{S}_t,$$

then $\boldsymbol{S_t}$ *is equal to the oracle's optimal allocation* $\boldsymbol{S}^*$*. Here, the thresholds* $\bar{\mu}_{(i,k)}$ *are defined as*

$$\bar{\mu}_{(i,k)} = \mu_{(i,k)} + g^{-1}(\Delta_{min}^{(i,k)}).$$

Using this claim, we will show regret bounds for our proposed Corr-UCB-RA algorithm. To state our results, we first define the notion of competitive and non-competitive base arms.

Definition 4 (Competitive and Non-Competitive base arms). *If* $\phi_{(j,\ell),(i,k)} \le \bar{\mu}_{(j,\ell)}$ *for some* $(i,k) \in \mathcal{S}^*$ *then base arm* (j,ℓ) *is called Non-competitive, otherwise it is called Competitive. Here,* $\mathcal{S}^*$ *denotes the set of base arms played in the oracle's optimal budget allocation vector* $\boldsymbol{S^*}$*. Furthermore, we define the pseudo-gap of a base arm* (j,ℓ) *as* $\bar{\Delta}_{(j,\ell)} = \bar{\mu}_{(j,\ell)} - \max_{(i,k)\in\mathcal{S}^*} \phi_{(j,\ell),(i,k)}$*.*

Note that the pseudo-gap is greater than zero for non-competitive base arms and is less than or equal to zero for competitive base arms. The definition of pseudo-gap is useful to state our regret bounds. Intuitively, a base arm (j,ℓ) is non-competitive if it can be inferred that the mean reward of (j,ℓ) is smaller than the threshold $\bar{\mu}_{(j,\ell)}$ through just the samples of a base arm belonging to the oracle's optimal budget allocation $\mathcal{S}^*$. In what follows, we refer to the total number of competitive base arms as C and the set of competitive base arms as $\boldsymbol{C}$. As mentioned earlier, the Corr-UCB-RA algorithm selects the budget allocation $\boldsymbol{S}^*$ with high probability if the indices of base arms $U_{(i,k)}$ are *close* to their true means. In the presence of correlations, we show that this can be achieved by sampling

competitive base arms $O(\log T)$ times and non-competitive base arms only $O(1)$ times. This occurs as the non-competitive base arms can be identified as sub-optimal based on samples of optimal base arms. We formalize this intuition to get the following regret bound for our Corr-UCB-RA algorithm.

Theorem 1 (Upper Bound on Cumulative Regret). *The expected cumulative regret of the Correlated-UCB algorithm for resource allocation is upper bounded as*

$$Reg_{\alpha}(T, \boldsymbol{D}) \leq \sum_{(i,k)\in C} \Delta_{max}^{(i,k)} \left(\frac{8\log T}{\left(g^{-1}\left(\Delta_{min}^{(i,k)}\right)\right)^2} + 2 \right) + \sum_{(i',k')\in\mathcal{K}\times\mathcal{A}\setminus\{C\}} \Delta_{max}^{(i',k')} (4KAt_0 + 6(KA)^3) + 2(KA)^2\Delta_{max} \quad (16)$$

$$= C \cdot O(\log T) + O(1), \quad (17)$$

where $C \subseteq \mathcal{K}\times\mathcal{A}$ is set of competitive base arms with cardinality C and $t_0 = \inf\left\{\tau \geq 2 : g^{-1}\left(\Delta_{min}^{(i,k)}\right) \geq 4\sqrt{\frac{2K\log\tau}{\tau}} \quad \forall (i,k), \bar{\Delta}_{(i,k)} \geq 4\sqrt{\frac{2K\log\tau}{\tau}} \quad \forall (i,k) \in \mathcal{A}\times\mathcal{K}\setminus C\right\}$.

We now present a proof of our Claim 1, which is then used to provide a proof sketch of Theorem 1. The proof of Claim 1 is of independent interest as well as these techniques can be used to analyse the regret of the UCB algorithm in generic combinatorial bandits as well (e.g., the combinatorial UCB algorithm in [8]).

4.2 Proof Sketch

Proof of Claim 1. In total there are $|K|\times|A|$ base arms. Index these base arms with indices z in the set $\{1, 2, \ldots |K|\times|A|\}$ such that $\Delta_{\min}^{(1)} \geq \Delta_{\min}^{(2)} \geq \ldots \Delta_{\min}^{(z)} \geq \ldots \geq \Delta_{\min}^{(|K|\times|A|)}$.

We consider a case where $\mu_z \leq U_z(t) < \mu_z + g^{-1}(\Delta_{\min}^{(z)}) \quad \forall z \in \mathcal{S}_t$ and $U_z > \mu_z \forall z$. Define y to be the smallest index such that base arm y is selected in $\boldsymbol{S_t}$. From the definition of base arm y and through Property 2 we have,

$$||U_{\boldsymbol{S_t}}(t) - \mu_{\boldsymbol{S_t}}||_\infty < g^{-1}(\Delta_{\min}^{(y)}) \Rightarrow |r(\boldsymbol{S_t}, \boldsymbol{U(t)}) - r(\boldsymbol{S_t}, \boldsymbol{\mu})| < \Delta_{\min}^{(y)}. \quad (18)$$

As $U_z(t) > \mu_z \ \forall z$, we have the following from the monotonicity condition (Property 1),

$$r(\boldsymbol{S_t}, \boldsymbol{\mu}) + \Delta_{\min}^{(y)} > r(\boldsymbol{S_t}, \boldsymbol{U(t)}) \geq r(\boldsymbol{S^*}, \boldsymbol{U(t)}) \geq r(\boldsymbol{S^*}.\boldsymbol{\mu}) \quad (19)$$

The third inequality arises from the monontonicity condition as $U_z > \mu_z, \forall z$. This shows that if $\mu_z \leq U_z(t) < \mu_z + g^{-1}(\Delta_{\min}^{(1)}), \forall z \in S_t$ and $U_z > \mu_z, \forall z$, the expected reward for the budget allocation S_t,

$$r(\boldsymbol{S_t}, \boldsymbol{\mu}) > r(\boldsymbol{S^*}, \boldsymbol{\mu}) - \Delta_{\min}^{(y)}. \quad (20)$$

As base arm y is selected in $\boldsymbol{S_t}$, then by definition of $\Delta_{min}^{(y)}$,

$$\max(r(\boldsymbol{S_t}, \boldsymbol{\mu}) | \boldsymbol{S_t} \in \mathcal{S}_B, (i,k) = y \in \mathcal{S}_t) \leq r(\boldsymbol{S^*}, \boldsymbol{\mu}) - \Delta_{\min}^{(y)}, \quad (21)$$

which shows that the maximum reward that can be attained if the allocation $\boldsymbol{S_t}$ was sub-optimal and base arm y was selected is upper bounded by $r(\boldsymbol{S^*}, \boldsymbol{\mu}) - \Delta_{\min}^{(y)}$. Upon comparing (21) and (20), we conclude that if $\mu_z \leq U_z(t) < \mu_z + g^{-1}(\Delta_{\min}^{(z)}) \quad \forall z \in \mathcal{S}_t$ and $U_z > \mu_z \forall z$, then the budget allocation vector $\boldsymbol{S_t}$ is equal to $\boldsymbol{S^*}$, which is the oracle's unique optimal solution.

Proof of Theorem 1. We now discuss the regret analysis of Theorem 1. In order to bound the regret, we first define the notion of a *responsible* base arm.

Definition 5 (Responsible). *A base arm (i,k) is said to be responsible at round t, if*

(1) It was selected in round t and
(2) $U_{(i,k)}(t) \geq \bar{\mu}_{(i,k)}$

By Claim 1, if a sub-optimal budget allocation was selected in round t, it implies that either $U_{(i,k)}(t) < \mu_{(i,k)}$ for some $(i,k) \in \mathcal{K}\times\mathcal{A}$ or at least one of the selected base arms in $\boldsymbol{S_t}$ was responsible. Therefore, the expected number of rounds in which a sub-optimal allocation was played (referred to as bad rounds) can be upper bounded by

$$\mathbb{E}[\text{Bad rounds}(T)] \leq \sum_{(i,k)} \left[\mathbb{E}[r_{(i,k)}(T)] + \mathbb{E}[n_{U_{(i,k)}<\mu_{(i,k)}}(T)]\right], \quad (22)$$

with $r_{(i,k)}(T)$ denoting the number of rounds for which base arm (i,k) is responsible up until round T and $n_{U_{(i,k)}<\mu_{(i,k)}}(T)$ representing the number of rounds in which $U_{(i,k)}(t) < \mu_{(i,k)}$ for some (i,k) until round T. This inequality arises as a result of the union bound and linearity of expectation. Moreover, whenever arm (i,k) is responsible in round t, the regret incurred in that round can be upper bounded by by $\Delta_{\max}^{(i,k)}$ (by definition of $\Delta_{\max}^{(i,k)}$ in Lemma 1). In scenarios where $U_{(i,k)}(t) < \mu_{(i,k)}$ for some (i,k), the regret incurred in that round can be upper bounded by $\Delta_{\max}$ (by definition of $\Delta_{\max}$ in Lemma 1). Using this observation, we have

$$\mathbb{E}[\text{Reg}(T)] \leq \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[r_{(i,k)}(T)] \times \Delta_{\max}^{(i,k)} + \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[n_{U_{(i,k)}<\mu_{(i,k)}}(T)] \times \Delta_{\max}. \quad (23)$$

Using Hoeffding's inequality, it can be shown that the second term is upper bounded by an $O(1)$ constant, the details are presented in Lemma 7 in the appendix. To bound the regret in (23), we bound $\mathbb{E}\left[r_{(i,k)}(T)\right]$ separately for non-competitive and competitive base arms. More specifically, we show that $\mathbb{E}\left[r_{(i,k)}(T)\right]$ is upper bounded by an $O(1)$ constant for non-competitive base arms and is $O(\log T)$ for competitive base arms. There are two key components to show upper bounds on $\mathbb{E}\left[r_{(i,k)}(T)\right]$ for non-competitive base arm (i,k). Suppose the base arm is non-competitive with respect to (j,ℓ), i.e., $\phi_{(i,k),(j,\ell)} < \bar{\mu}_{(i,k)}$ and $(j,\ell) \in \mathcal{S}^*$.

(1) The probability of base arm (i,k) being responsible in round t jointly with the event that $n_{j,\ell}(t) > \frac{2t}{3}$ is *small*.

$$\Pr\left((resp_{(i,k)}(t), n_{(j,\ell)}(t) \geq \frac{2t}{3}\right) \leq t^{-3} \quad \forall t > 3KAt_0.$$

This occurs as upon obtaining a *large* number of samples of base arm (j,ℓ), the expected pseudo-reward of base arm (i,k) is smaller than $\bar{\mu}_{(i,k)}$ with high probability. As a result, the probability that base arm (i,k) is responsible is *small*. (See Lemma 4.)

(2) The probability that a sub-optimal budget allocation is made for more than $\frac{t}{3}$ times till round t is upper bounded as,

$$\Pr\left(T^{\text{sub-opt}}(t) \geq \frac{t}{3}\right) \leq 6(KA)^2\left(\frac{t}{3KA}\right)^{-2} \quad \forall t > 3KAt_0,$$

We show this in Lemma 9 through Lemma 6,8 by showing that $r_{(i,k)}(T)$, which is the number of rounds for which base arm (i,k) is responsible till round T, is smaller than $\frac{t}{3KA}$ with high probability. Additionally, $n_{U_{(i,k)}<\mu_{(i,k)}}(T)$, representing the number of rounds in which $U_{(i,k)}(t) < \mu_{(i,k)}$ for some (i,k) till round T, is smaller than $\frac{t}{3}$ with high probability. Using these two arguments (1) and (2) above, we bound the expected times a non-competitive base arm (i,k) is responsible until round t in Lemma 10 as

$$\mathbb{E}\left[r_{(i,k)}(T)\right] \leq 3KAt_0 + \sum_{t=3KAt_0}^{T} t^{-3} + 6(KA)^2\left(\frac{t}{3KA}\right)^{-2} \leq \mathrm{O}(1). \tag{24}$$

Next, we bound the term $\mathbb{E}\left[r_{(i,k)}(T)\right]$ for competitive sub-optimal arms. We do so in Lemma 11, by showing that after base arm (i,k) has been sampled $\mathrm{O}(\log T)$ times, the probability of base arm being responsible at round t decays as t^{-2} and as a result $\mathbb{E}\left[r_{(i,k)}(T)\right]$ is $\mathrm{O}(\log T)$. This combined with (24), leads to Theorem 1.

4.3 Discussion on the Regret Bound

Competitive and Non-competitive base arms. Recall that a base arm (i,k) is said to be non-competitive if the expected pseudo-reward of base arm (i,k) with respect to some base $(j,\ell) \in \mathcal{S}^*$ is smaller than $\bar{\mu}_{(i,k)}$. Note that the optimal set of arms $\mathcal{S}^*$, reward distributions of individual base arms are unknown at the beginning and as a result the Corr-UCB-RA initially does not know which base arms are competitive and non-competitive.

Reduction in the effective set of base arms. Upon comparison with the regret of the UCB-RA algorithm, from Lemma 1, we see that our proposed algorithm reduces the regret from $KA \times \mathrm{O}(\log T)$ to $C \times \mathrm{O}(\log T)$, since only C out of the total KA need to be sampled $\mathrm{O}(\log T)$ times before the condition in Claim 1 is met with high probability. As a result, the Corr-UCB-RA only explores C out of the KA base arms explicitly and effectively reduces the problem with KA base arms to one with C base arms.

Bounded regret in certain settings. Whenever the set $\mathcal{C}$ is empty, the proposed Corr-UCB-RA algorithm achieves bounded regret, which is an order-wise improvement over the regret of correlation agnostic UCB-RA algorithm. One scenario in which this can occur is if the functions $f_k(\cdot)$ are invertible with respect to X_k given a_k. More generally, whenever the sub-optimal base arms can be identified as sub-optimal through just the samples of optimal base arms, we get a bounded regret. Note that the algorithm initially has no knowledge about the optimality/sub-optimality of base arms and in such cases it identifies them by sampling the sub-optimal base arms only $\mathrm{O}(1)$ times.

5 CONTINUOUS BUDGET SETTING

So far we have studied the resource allocation problem under the assumption that the set $\mathcal{A}$ from which budget a_k for each entity k is allocated is a countable set. In this section, we discuss settings where $\mathcal{A}$ is uncountable. One instance where this could occur is if $a_k \in \mathbb{R}$. In such scenarios, it is still possible to design an algorithm while achieving bounded regret in some cases.

Reward functions are invertible. Suppose the reward functions $f_k(a_k, X_k)$ are invertible in X_k and are known to the algorithm. In this case, it is possible to estimate X_k directly from the reward samples of entity k. Therefore, one can maintain an empirical mean $\hat{X}_k(t)$ for each entity. This empirical mean can then be used to evaluate the upper confidence bound indices for base arm (i,k) as

$$U_{(i,k)}(t) = f_k(i, \hat{X}_k(t)) + \sqrt{\frac{2\log T}{n_k(t)}},$$

where $n_k(t) = \sum_j n_{j,k}(t)$ and $\hat{X}_k(t) = \frac{\sum_{\tau=1}^{t} g^{-1}_{a_k(\tau),k}(r_k(\tau))}{n_k(t)}$. Here $g_{a_k(\tau),k}(X_k(t)) = f_k(a_k(\tau), X_k(t))$ and $r_k(\tau)$ is the reward attained from entity k at round τ.

One can then use these UCB indices to obtain an allocation $\boldsymbol{S_t}$ from the offline oracle as done in Corr-UCB-RA algorithm, which will then be used to select the action in the next round. Using techniques in Section 4.2, it can be shown that this algorithm in cases where reward functions are invertible will lead to an $\mathrm{O}(1)$ regret. This occurs as the information about the sub-optimal base arms can be obtained through the samples of the optimal action.

Non-invertible reward functions. In scenarios where reward functions are non-invertible, it is still possible to extend the Corr-UCB-RA algorithm. This can be done by discretizing the budget space and making assumptions about Lipschitz continuity as done in [9]. Specifically, the regret is affected by the discretization granularity and [9] provided an optimized value for it. After the discretization, we can use Corr-UCB-RA on the countable action set.

6 EXPERIMENTAL RESULTS

To validate the effectiveness of our algorithm, we conduct experiments on three applications with synthetic and real data. First, we consider a dynamic user allocation problem in wireless networks, where we need to allocate new incoming users to different wireless access points with unknown number of existing users. We evaluate our algorithm in the setting with non-invertible reward function. Next, we study an online server assignment problem, where the servers need to be assigned to different job streams with unknown job arrival rates. Different from the first application, the reward function of this problem is invertible, so it is possible to obtain $\mathrm{O}(1)$ regret. However, we also study a partial feedback setting for this application, which leads to sublinear regret. Finally, we apply our algorithm to an online water filling problem [23] that is essential to the power allocation in OFDM systems [24]. It is a continuous budget allocation problem with invertible reward functions, and we study its partial feedback setting as well. Since the forms of the reward functions are known in all applications, we can directly compute the pseudo-rewards defined in Eq. (8).

6.1 Dynamic User Allocation

In this section, we apply our corr-UCB-RA algorithm to a dynamic user allocation problem in wireless networks. Our goal is maximize the total throughput of wireless access points (APs) by allocating new incoming users to them. The number of existing users associated to each AP is time-varying, which affects the traffic load on the AP. We assume each user has a fixed traffic load of 0.2 and consider

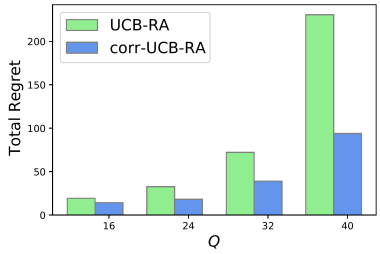


Figure 3: Comparison between regret of UCB-RA and Corr-UCB-RA as a function Q (new incoming users) for the application of dynamic user allocation.

the well-known ALOHA protocol [25] for each AP. We consider K APs and Q new incoming users at each round. Let X_k denote the number of existing users in each AP k and a_k denote the number of new users allocated to it. Note that we assume all the users of an AP will leave when the round ends, so a_k in the current round will not affect X_k in the future rounds. Our goal is to maximize the total throughput of all APs:

$$\max_{a_k} \sum_{k=1}^{K} 0.2(X_k + a_k)e^{-0.2(X_k+a_k)}, \quad \text{s.t.} \sum_{i=k}^{K} a_k = Q, a_k \in \mathbb{N}.$$

We extract $\{X_k\}$, the number of existing users in each AP, from a real-world dataset [26]. We choose 4 APs (91, 92, 94, 95) on the 3rd floor of Building 3 on campus, and record their associated users from 13:00 to 16:00 on March 2, 2015. The detailed distribution of the number of existing users on different access points can be found in the Appendix. In our experiment, at each round, we first sample $\{X_k\}$ from the extracted distribution, then allocate $Q = 8$ new users to these four APs. Since the throughput function is non-invertible, our algorithm cannot directly infer X_k from the observed throughput of each AP and needs to maintain the pseudoUCB indices of base arms as explained in Section 3. We compare it with the UCB-RA algorithm. Figure 4a shows the average regrets with 95% confidence interval over 20 experiments. The result is consistent with our analysis in Section 4: corr-UCB-RA achieves 25% less regret than correlation agnostic UCB-RA algorithm. This occurs as the corr-UCB-RA algorithm is able to make use of the correlations between the reward of base arms to incur a regret of $C \cdot \mathrm{O}(\log T)$ as opposed to $KA \cdot \mathrm{O}(\log T)$. We also show the relationship between Q and the total regret after 2000 rounds in Figure 3: with the increase of Q, the total regret of corr-UCB-RA increases much more slowly than that of UCB-RA.

6.2 Online Server Assignment

We consider 4 independent job streams (i.e., $K = 4$) with unknown expected job arrival rates $\boldsymbol{\lambda} = (0.2, 0.4, 0.6, 0.8)$. For each job stream k, the realized job arrival rate X_k follows a uniform distribution $U(\lambda_k - 0.1, \lambda_k + 0.1)$. We assume each job stream has one initial server to ensure it is a stable system with bounded expected waiting time. There are 8 additional servers (i.e., $Q = 8$) to be assigned and we denote the number of additional servers allocated to stream k as a_k. We assume the service rate of all servers as 1, and our goal is to minimize the average expected waiting time of all job streams:

$$\min_{a_k} \sum_{k=1}^{K} \frac{1}{1 - \frac{X_k}{a_k+1}} \cdot \frac{X_k}{\sum_{k=1}^{K} X_k}, \quad \text{s.t.} \sum_{i=k}^{K} a_k \le Q, a_k \in \mathbb{N}.$$

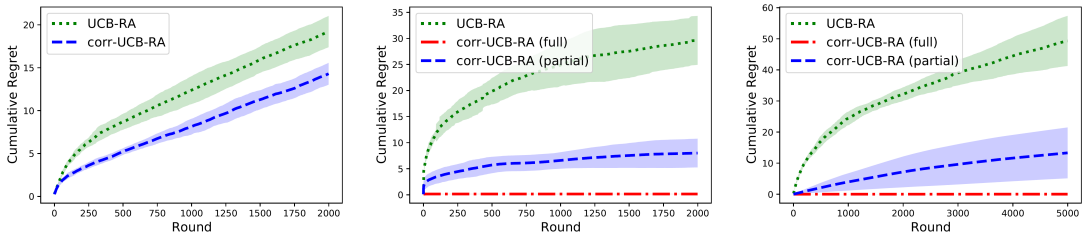
We consider both the full feedback and the partial feedback settings. In the full feedback setting, we assume the waiting times of all job stream are always observable. Since the waiting time function is invertible, our algorithm can directly infer $\{X_k\}$ and update the pseudo-rewards of other base arms as per (8). Notice that our goal is to minimize the expected waiting time, so we need to maintain the lower confidence bound (LCB) indices of all base arms, instead of the UCB indices for reward maximization and correspondingly pseudo-rewards would be lower bounds on conditional expected reward. In the partial feedback setting, the waiting time can only be observed when $a_k \ge 1$, i.e., at least one additional server is assigned to stream K. When no server is assigned to stream K, the pseudo-reward of other assignments with respect to such an assignment is set to the minimum possible reward. We repeat the experiment 20 times and Figure 4b shows the average regrets with 95% confidence interval. In the full feedback setting, corr-UCB-RA obtains $\mathrm{O}(1)$ regret as there is no cost for inferring $\{X_k\}$. In the partial feedback setting, corr-UCB-RA has to balance between the actions of $a_k = 0$ and $a_k \ge 0$, which incurs a sublinear regret. It still outperforms UCB-RA due to the utilization of correlation information.

6.3 Online Water Filling

We finally consider the water filling problem where a total amount of one unit power has to be assigned to 4 communication channels, i.e., $Q = 1, K = 4$, with the objective of maximizing the total throughput. The throughput of the k^{th} channel is given by $\log(X_k + a_k)$, where a_k represents the power allocated to channel k and X_k represents the floor above the baseline at which power can be added to the channel. It can be written as a convex optimization problem:

$$\max_{a_k} \sum_{k=1}^{K} \log(X_k + a_k), \quad \text{s.t.} \sum_{i=k}^{K} a_k \le Q, a_k \ge 0.$$

For the online water filling problem, the $\{X_k\}$ are unknown and need to be learned. For each channel k, we assume the expectation $\mu_k = \mathbb{E}[X_k]$ is uniformly sampled from $[0.8, 1.2]$, and the realization of X_k follows a uniform distribution $U(\mu_k - 0.5, \mu_k + 0.5)$. As it is a online continuous resource allocation problem, we choose UCB-RA algorithm with discretization granularity 0.2 (i.e., $\mathcal{A} = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$) as the baseline. Similar to the online server assignment problem, we consider both the full feedback and the partial feedback settings. In the full feedback setting, the throughput $\log(X_k + a_k)$ is always observable. Since the reward function is invertible, our algorithm can directly infer $\{X_k\}$ and update the pseudoUCB indices as described in Section 5. In the partial feedback setting, we assume $\log(X_k + a_k)$ can be observed only if $a_k \ge 0.2$. For channel k with $a_k < 0.2$, we update the pseudo-rewards of other base arms with the maximum possible rewards. We repeat the experiment 20 times and Figure 4c shows the average regrets with 95% confidence interval. We see that corr-UCB-RA algorithm achieves significantly reduced regret relative to UCB-RA in both the full feedback and the partial feedback settings. For the full feedback case, corr-UCB-RA obtains $\mathrm{O}(1)$ regret as the water filling reward



(a) Dynamic user allocation. **(b) Online server assignment.** **(c) Online water filling.**

Figure 4: Performance comparison between the Corr-UCB-RA and the UCB-RA algorithm for different application problems.

function is invertible and there is no cost in inferring $\{a_k\}$. For the partial feedback case, since the minimal power needs to be 0.2 to observe the throughput, corr-UCB-RA needs to balance between the actions of $a_k < 0.2$ and $a_k \geq 0.2$, due to which it incurs a sublinear regret. The regret is still smaller than UCB-RA as it makes use of the available correlation information.

7 CONCLUDING REMARKS

In this paper, we study the problem of sequential resource allocation by modeling it through a combinatorial bandit framework, where the allocation of a budget to an entity is considered as a base arm. In several practical settings, rewards received under different budget allocations are often correlated. We propose a novel correlated combinatorial bandit framework to tackle the online resource allocation problem. In particular, we model the correlations through *pseudo-rewards*, which represent an upper bound on the conditional expected reward of a budget-entity pair. Using the knowledge of these pseudo-rewards, we propose the correlated UCB algorithm for resource allocation (Corr-UCB-RA) which incurs a regret of $C \cdot \mathrm{O}(\log T)$ as opposed to $KA \cdot \log T$ regret attained by prior correlation agnostic approach in [9]. The value of C can be much smaller than KA and can even be 0 in certain settings, under which our proposed Corr-UCB-RA algorithm attains $\mathrm{O}(1)$ regret. These results are validated by our experimental results on multiple different application settings. While we study this problem in the context of online resource allocation, the algorithm and analysis could be easily extended to the general combinatorial bandit framework [8] as well. An interesting future direction is to learn correlations in an online manner. As multiple base arms are sampled in each round, it is possible to learn correlation information on the go and subsequently use them for budget allocation in the future rounds.

ACKNOWLEDGMENTS

This work was supported by NSF CCF-2007834, NSF CNS-2103024, ONR Grant N000142112547, ONR Grant N000142112128, and CMU IoT@CyLab.

REFERENCES

[1] D. Julian, M. Chiang, D. O'Neill, and S. Boyd, "Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks," in *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 477–486, IEEE, 2002.
[2] A. Lozano, A. M. Tulino, and S. Verdú, "Optimum power allocation for parallel gaussian channels with arbitrary input distributions," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 3033–3051, 2006.
[3] W. W. Chu, "Optimal file allocation in a multiple computer system," *IEEE Transactions on Computers*, vol. 100, no. 10, pp. 885–889, 1969.
[4] K. S. Reddy, S. Moharir, and N. Karamchandani, "Resource pooling in large-scale content delivery systems," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1617–1630, 2019.
[5] D. N. Kleinmuntz, *20 Resource Allocation Decisions*. Citeseer, 2007.
[6] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.
[7] S. Bubeck, N. Cesa-Bianchi, *et al.*, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.
[8] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *International conference on machine learning*, pp. 151–159, PMLR, 2013.
[9] J. Zuo and C. Joe-Wong, "Combinatorial multi-armed bandits for resource allocation," in *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–4, IEEE, 2021.
[10] C. Joe-Wong, S. Sen, T. Lan, and M. Chiang, "Multiresource allocation: Fairness–efficiency tradeoffs in a unifying framework," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1785–1798, 2013.
[11] N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens, "Near optimal online algorithms and fast approximation algorithms for resource allocation problems," *Journal of the ACM (JACM)*, vol. 66, no. 1, pp. 1–41, 2019.
[12] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Schieber, "A unified approach to approximating resource allocation and scheduling," *Journal of the ACM (JACM)*, vol. 48, no. 5, pp. 1069–1090, 2001.
[13] T. Lattimore, K. Crammer, and C. Szepesvári, "Linear multi-resource allocation with semi-bandit feedback.," in *NIPS*, pp. 964–972, 2015.
[14] A. Verma, M. K. Hanawal, A. Rajkumar, and R. Sankaran, "Censored semi-bandits: A framework for resource allocation with censored feedback.," in *NeurIPS*, 2019.
[15] X. Fontaine, S. Mannor, and V. Perchet, "An adaptive stochastic optimization algorithm for resource allocation," in *ALT*, pp. 319–363, PMLR, 2020.
[16] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 1466–1478, 2012.
[17] W. Chen, Y. Wang, Y. Yuan, and Q. Wang, "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1746–1778, 2016.
[18] S. Gupta, G. Joshi, and O. Yağan, "Correlated multi-armed bandits with a latent random source," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3572–3576, IEEE, 2020.
[19] S. Gupta, S. Chaudhari, G. Joshi, and O. Yağan, "Multi-armed bandits with correlated arms," *IEEE Transactions on Information Theory*, vol. 67, no. 10, pp. 6711–6732, 2021.
[20] T. Lattimore and R. Munos, "Bounded regret for finite-armed structured bandits," in *Advances in Neural Information Processing Systems*, pp. 550–558, 2014.
[21] S. Gupta, S. Chaudhari, S. Mukherjee, G. Joshi, and O. Yağan, "A unified approach to translate classical bandit algorithms to the structured bandit setting," 2018.
[22] R. Combes, S. Magureanu, and A. Proutiere, "Minimal exploration in structured stochastic bandits," in *Advances in Neural Information Processing Systems*, pp. 1763–1771, 2017.
[23] Y. Gai and B. Krishnamachari, "Online learning algorithms for stochastic water-filling," in *2012 Information Theory and Applications Workshop*, pp. 352–356, 2012.
[24] A. Narasimhamurthy, M. Banavar, and C. Tepedelenlioglu, *OFDM Systems for Wireless Communications*. Morgan & Claypool, 2010.
[25] N. Abramson, "The aloha system: Another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference*, pp. 281–285, 1970.
[26] L. Pajevic, G. Karlsson, and V. Fodor, "CRAWDAD dataset kth/campus (v. 2019-07-01)." Downloaded from https://crawdad.org/kth/campus/20190701/eduroam, July 2019. traceset: eduroam.