Hierarchical Unimodal Bandits

Tianchi Zhao TZHAO7@EMAIL.ARIZONA.EDU
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721, USA

Chicheng Zhang CHICHENGZ@CS.ARIZONA.EDU
Department of Computer Science
University of Arizona
Tucson, AZ 85721, USA

Ming Li LIM@EMAIL.ARIZONA.EDU
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721, USA

Abstract

We study a multi-armed bandit problem with clustered arms and a unimodal reward structure, which has applications in millimeter wave (mmWave) communication, road navigation, etc. More specifically, a set of N arms are grouped together to form C clusters, and the expected reward of arms belonging to the same cluster forms a Unimodal function (a function is Unimodal if it has only one peak, e.g. parabola). First, in the setting when $C = 1$, we propose an algorithm, SGSD (Stochastic Golden Search for Discrete Arm), that has better guarantees than the prior Unimodal bandit algorithm (Yu and Mannor, 2011). Second, in the setting when $C \geq 2$, we develop HUUCB (Hierarchical Unimodal Upper Confidence Bound (UCB) algorithm), an algorithm that utilizes the clustering structure of the arms and the Unimodal structure of the rewards. We show that the regret upper bound of our algorithm grows as $O(\sqrt{CT\log(T)})$, which can be significantly smaller than UCB's $O(\sqrt{NT\log(T)})$ regret guarantee. We perform a multi-channel mmWave communication simulation to evaluate our algorithm. Our simulation results confirm the advantage of our algorithm over the UCB algorithm (Auer et al., 2002) and a two-level policy (TLP) proposed in prior works (Pandey et al., 2007). [1]

1. Introduction

1.1 Motivation

Multi-armed bandit (MAB) problem (Thompson, 1933) models many real-world scenarios where a decision maker learns to take a sequence of arms to maximize reward. Here, the decision maker is given access to an arm set, and takes an arm from the arm set results in a reward drawn from an unknown distribution. The objective of the decision maker is to

[1] This work was partly supported by ONR YIP grant N00014-16-1-2650, and the paper is accepted by ECML-PKDD 2022. The authors would like to thank Zhiwu Guo for his help on drawing figures, and the anonymous reviewers for their helpful comments.

maximize its expected cumulative reward over a time horizon of T. To this end, it faces a tradeoff between exploration and exploitation.

In this work, we consider a multi-armed bandit problem with clustered arms, where the arm space can be partitioned into C clusters, and each cluster's rewards exhibits a Unimodality structure. This arises naturally in various decision problems, as shown in Appendix A:

1.2 Related work

1.2.1 Hierarchical bandit

Hierarchical bandit problem, where the arm space is partitioned into multiple clusters, has been studied in (Nguyen and Lauw, 2014; Jedor et al., 2019; Bouneffouf et al., 2019; Carlsson et al., 2021). These papers give regret bounds under different assumptions on the clustering. Specifically, (Pandey et al., 2007) proposed a Two-level Policy (TLP) algorithm. It divided the arms into multiple clusters. However, their work does not provide a theoretical analysis of the algorithm. (Zhao et al., 2019) proposed a novel Hierarchical Thompson Sampling (HTS) algorithm to solve this problem. The beams under the same chosen channel can be regarded as a cluster of arms in MAB. However, it does not utilize the Unimodal property in each cluster. (Bouneffouf et al., 2019) considered a two-level UCB scheme that the arm set is pre-clustered, and the reward distributions of the arms within each cluster are similar. However, they didn't consider the Unimodal property in each cluster. (Jedor et al., 2019) introduced a MAB setting where arms are grouped in one of three types of categories. Each type has a different ordering between clusters, and our work does not have such assumption among the clusters. (Yang et al., 2022) considered a problem of online clustering: a set of arms can be partitioned into various groups that are unknown. Note that the partition of cluster is time-variant, and we study a different setting where the clusters are pre-specified. (Kumar et al., 2019) addressed the problem of hidden population sampling problem in online social platforms. They proposed a hierarchical Multi-Arm Bandit algorithm (Decision-Tree Thompson Sampling (DT-TMP)) that uses a decision tree coupled with a reinforcement learning search strategy to query the combinatorial search space. However, their algorithm is based on Thompson Sampling, and no theoretical analysis of its regret is given. (Singh et al., 2020) studies a multi-armed bandit problem with dependent arms. When an agent pulls arm i, it not only reveals information about its own reward distribution but also reveal all those arms that belong to the same cluster with arm i, which is not the case in our problem. (Carlsson et al., 2021) proposed a Thompson Sampling based algorithm with clustered arms, and give a regret bound which depends on the number of clusters. However, they do not utilize the Unimodal property as well.

1.2.2 Unimodal bandit

In a Unimodal bandit problem, the expected reward of arms forms a Unimodal function. Here, specialized algorithms have been designed to exploit the Unimodality structure, to achieve faster convergence rate (compared to standard bandit algorithms such as UCB). (Yu and Mannor, 2011) is the first work to propose a Unimodal bandit algorithm for both continuous arm and discrete arm settings. (Combes and Proutiere, 2014) proposed Optimal Sampling for Unimodal Bandits (OSUB), and exploits the Unimodal structure under the discrete arm setting. They provided a regret upper bound for OSUB which does not depend on the number of arms. (Zhang et al., 2021) showed that the effective throughputs of mmWave codebooks possess the Unimodal property and proposed a Unimodal Thompson Sampling (UTS) algorithm to deal with mmWave codebook selection. However, both papers only consider Unimodal property without clustered arms. (Blinn et al., 2021) proposed Hierarchical Optimal Sampling of Unimodal Bandits. The difference with our work is that they use the OSUB algorithm to select an arm in each cluster, and they did not provide a theoretical regret analysis.

1.3 Main Contributions

Our main contributions can be summarized as follows:

1. In the single-cluster setting ($C = 1$), we propose a new Unimodal bandit algorithm, called Stochastic Golden Search with Discrete arm (SGSD), that improves over an existing Unimodal bandit algorithm (Yu and Mannor, 2011), in that it simultaneously achieves gap-dependent and gap-independent regret bounds. In addition, its regret bounds are competitive with UCB, and can sometimes be much better (Shown in Appendix B).

2. In the multi-cluster setting ($C \geq 2$), built on the SGSD, we present a UCB-based, hierarchical Unimodal bandit algorithm, called HUUCB, to solve the MAB with Clustered arms and a Unimodal reward structure (MAB-CU) problem. The key insight is a new setting of reward UCB for each cluster, taking into account the regret incurred for each cluster. We prove a gap-independent regret bound for this algorithm, and show that they can be better compared with the baseline strategy of UCB on the "flattened" arm set.

3. We evaluate our algorithms experimentally in both the single-cluster setting and the multi-cluster setting, using two different datasets (synthetic/simulated).
   (a) In the single-cluster setting, our SGSD algorithm outperforms UCB.
   (b) In the multi-cluster setting, our HUUCB algorithm outperforms UCB with flatten arms, and TLP.

2. Hierarchical Unimodal bandits: Problem Setup

The problem statement is as follows: There are N arms available and each arm j's reward comes from a particular distribution (supported on $[0, 1]$) with an unknown mean μ_j. The arms are partitioned to C clusters, where we denote Cluster_i as the i-th cluster. In each cluster i, we assume that the expected rewards of arm $j \in \text{Cluster}_i$ form a Unimodal function (a function is Unimodal if the function has only one local maximum, e.g. a negative parabola). We assume that every cluster have the same size B, therefore, $N = CB$.

The Multi-armed bandit (MAB) model focuses on the essential issue of balancing the trade-off between exploration and exploitation (Auer et al., 2002). At each time step, the algorithm selects one arm j_t. Then a reward of this arm is independently drawn, and observed by the algorithm. The objective of the algorithm is to gather as much cumulative

Algorithm 1 Hierarchical Unimodal UCB Algorithm

```
Input: $D_H, D_L$
For each cluster $i = 1, \dots, C$: $\hat{\nu}_i(0) = 0, M_i(0) = 0$, initialize $\mathcal{A}_i$, a copy of Alg. 2.
For each arm $j = 1, \dots, N$: $\hat{\mu}_j(0) = 0, m_j(0) = 0$
for $t = 1, 2, ...N$, do
    Play arm $j = t$, and update corresponding $\hat{\mu}_j(t), m_j(t) = 1$ ,
end for
for each cluster i do
    $M_i(t) = \sum_{j \in \text{Cluster}_j} m_j(t)$
    $\hat{\nu}_i(t) = \sum_{j \in \text{Cluster}_i} \frac{m_j}{M_i} \hat{\mu}_j(t)$
end for
for stage $t = N + 1, N + 2, ...,$ do
    Choose the cluster
        $i_t := \arg\max \left\{ \hat{\nu}_i(t) + \sqrt{\frac{2\log(t)}{M_i(t)}} + \frac{D_H}{D_L}\sqrt{\frac{\log(t)}{M_i(t)}} \right\},$    (2)
    Resume $\mathcal{A}_{i_t}$ and run it for one time step, select an arm $j_t \in \text{Cluster}_{i_t}$, and obtain the
    reward of selected arm $r_{j_t}(t)$ at stage $t$
    Update empirical mean rewards and counts for all clusters:
        $(\hat{\nu}_i(t), M_i(t)) = \begin{cases} \left(\frac{\hat{\nu}_i(t-1) \cdot M_i(t-1) + r_{j_t}(t)}{M_i(t-1)+1}, M_i(t-1)+1\right), & i = i_t, \\ \left(\hat{\nu}_i(t-1), M_i(t-1)\right), & i \neq i_t. \end{cases}$
end for
```

reward as possible. The expected cumulated regret can be expressed as (Bubeck and Cesa-Bianchi, 2012):

$$E[R(T)] = \sum_{t=1}^{T} (\mu_{j^*} - \mu_{j_t}) \tag{1}$$

where $j^* = \arg\max_{j \in \{1, \dots, N\}} \mu_j$ is the optimal arm, T is the total running time. Note that the algorithm only observes the reward for the selected arm, also known as the bandit feedback setting.

3. Hierarchical Unimodal UCB Algorithm

In this section, we study the multi-cluster setting $(C \geq 2)$[2]. Existing works such as Two-Level Policy (TLP, (Pandey et al., 2007)) approaches this problem using the following strategy: treat each cluster as a "virtual arm' ', and view the cluster selection problem (which we call *inter-cluster selection*) as a stationary MAB problem. In each step, the TLP algorithm chooses a virtual arm first using UCB, and then an actual arm within the

[2] The single-cluster setting ($C = 1$) is shown in Appendix B

selected cluster using some *intra-cluster arm selection* algorithm. However, due to the nonstationary nature of the rewards within a cluster (as the intra-cluster arm selection algorithm may gradually converge to pulling the cluster's optimal arm), TLP do not have theoretical guarantees.

In contrast, in this section, we propose a Hierarchical Unimodal UCB Algorithm (HU-UCB) (Alg. 1) that has a provable regret guarantee. Our algorithm design follows the "optimism in the face of uncertainty" principle: clusters are chosen according to their optimistic upper confidence bounds on their maximum expected rewards $\nu_i = \max_{j \in \text{Cluster}_i} \mu_j$'s, a property not satisfied by TLP. This ensures a sublinear regret for the cluster selection task. The algorithm proceeds as follows: it first takes into D_H, D_L as inputs, which are the reward gap parameters specified in Assumption 1. Then, the initialization phase (lines 4 to 10) begins by selecting each arm at least once to ensure $M_i(t)$ and $\hat{\nu}_i(t)$ are updated. $M_i(t)$ is number of times that cluster i has been selected and $\hat{\nu}_i(t)$ is the empirical mean value for the cluster i. Once the initialization is completed, the algorithm selects the cluster that maximizes our designed UCB (Equation 2). From the equation, we can see that the UCB for cluster i,

$$\hat{\nu}_i(t) + \sqrt{\frac{2\log(t)}{M_i(t)}} + \frac{D_H}{D_L}\sqrt{\frac{\log(t)}{M_i(t)}}$$

is the sum of three terms. The first term is the empirical mean value of the $M_i(t)$ rewards obtained by pulling the arms in the cluster i. The second term accounts for the concentration between the sum of the noisy rewards and the sum of their corresponding expected rewards. The third term is new and unique to HUUCB – it accounts for the suboptimality of the arm selection in cluster i by SGSD so far, calculated by dividing SGSD's regret $O\left(\frac{D_H}{D_L}\sqrt{M_i(t)}\right)$ by $M_i(t)$. The three terms jointly ensures that the UCB is indeed a high-probability upper bound of ν_i. In line 13, algorithm 1 selects an arm $j_t \in \text{Cluster}_{i_t}$ using $\mathcal{A}_{i_t}$ after selecting a cluster i_t and obtaining the reward r_{j_t} ($\mathcal{A}_{i_t}$ is a copy of Alg. 2 for cluster i_t). Last, in line 14, the algorithm updates the chosen cluster i_t's statistics, empirical reward mean $\hat{\nu}_{i_t}(t)$ and count $M_{i_t}(t)$. Other clusters' statistics remain the same as time step $t-1$.

We have the following regret guarantee of Algorithm 1:

Theorem 1 *If each cluster satisfies assumption 1, the regret of Hierarchical Unimodal UCB is upper bounded by,*

$$E[R(T)] \leq O(\frac{D_H}{D_L}\sqrt{2CT\log(T)}), \tag{3}$$

where C is the number of clusters.

Outline of the proof for Theorem 1: First, we define the event

$$\boldsymbol{E} = \left\{ |\hat{\nu}_i(t) - \nu_i| \leq \sqrt{\frac{2\log(T)}{M_i(t)}} + \frac{D_H}{D_L}\sqrt{\frac{\text{const}}{M_i(t)}}, \forall i, t \right\}.$$

Without loss of generality, assume that Cluster_1 contains the globally optimal arm. The high-level idea of the proof is as follows:

(1) We bound the regret incurred when the algorithm chooses cluster $i \neq 1$ when the event $\boldsymbol{E}$ holds.

(2) We bound the probability of the event $\boldsymbol{E}$ does not happen using Azuma's inequality.

(3) Lastly, we bound the regret incurred when the algorithm chooses optimal cluster Cluster_1 but selects a sub-optimal arm using Theorem 2 (see in Appendix B). The detailed proof is in Appendix D.1.

Remark: Theorem 1 shows that the regret bound depends on the number of clusters (instead of the number of arms) because we incorporate the SGSD algorithm. Compared to the "flattened" UCB algorithm with a total of $N = CB$ arms (B is the number of arms in each cluster), whose regret is $\mathrm{O}(\sqrt{TCB\log(T)})$, when $\frac{D_H}{D_L} \ll \sqrt{B}$, HUUCB has a much better regret.

We can also prove the regret bound of HUUCB with that of a general bandit model selection algorithm (Abbasi-Yadkori et al., 2020; Cutkosky et al., 2021). We regard each cluster i's algorithm as a base algorithm defined in (Abbasi-Yadkori et al., 2020; Cutkosky et al., 2021). In our problem, C is the number of the base algorithm. Then, we define $R_i(T)$ as the regret upper bound for cluster i, represented as

$$R_i(T) \leq O\left(\frac{D_H}{D_L}\log(T)\sqrt{T}\right) \leq \text{const}_1 d_i \log(T)\sqrt{T}, \tag{4}$$

where const_1 is a positive constant independent of T and i, $d_i = \frac{D_H}{D_L}$. According to Theorem 2.1 in (Abbasi-Yadkori et al., 2020), the regret is upper bounded by,

$$E[R(T)] \leq C \max_i R_i(T) \leq C\text{const}_1 d_i \log(T)\sqrt{T}, \tag{5}$$

Comparing (3) and (5), we can see that our result is better than their result (Our result's C term (number of cluster) is in the square root). According to Theorem 1 in (Cutkosky et al., 2021), the regret is upper bounded by,

$$E[R(T)] \leq O\left(\sqrt{CT} + (C^{\frac{1}{2}}(\frac{D_H}{D_L})^2 + \frac{D_H}{D_L} + C^{\frac{1}{2}})\log(T)T^{\frac{1}{2}}\right), \tag{6}$$

Comparing (3) and (6), we can see that our result is better than their result (Our result's $\log(T)$ term is in the square root).

4. Experiments

We aim to answer the following questions through experiments:

1. Can SGSD outperform other algorithms in Unimodal bandit environments?

2. Can HUUCB outperform other hierarchical bandit algorithms (such as TLP) in MAB-CU environments? Meanwhile, we intend to validate whether the simulation result conforms to our theoretical analysis – specifically, does HUUCB's new $\frac{D_H}{D_L}\sqrt{\frac{\log(t)}{M_i(t)}}$ bonus term help in cluster selection?

See Appendix C for more details.

5. Conclusion

In this paper, we formulate the hierarchical Unimodal bandit learning problem, which occurs in many real-world applications. First, we adapt the Stochastic Golden Search (SGS) algorithm into discrete arm settings (SGSD), and we derive its gap-dependent and gap-independent regret bounds. Then, we propose a novel HUUCB algorithm based on the SGSD algorithm. Simulation results show that our algorithm performs better than TLP-UCB. For future work, we are planning to derive a gap-dependent $\log(T)$-style regret bound for the HUUCB algorithm and validate the regret bound in various simulation scenarios.

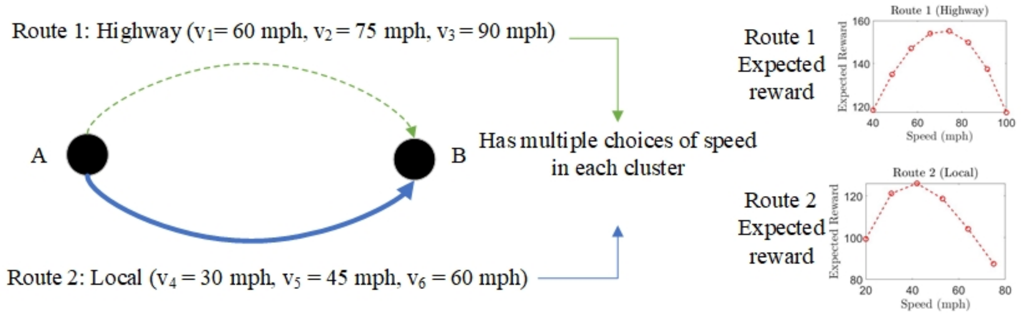


Figure 1: Road navigation example: cluster means the set under each route (The set contains different speed. For cluster (route) 1, it contains $v_1 = 60mph, v_2 = 75mph, v_3 = 90mph$). The safety indices for speeds in route 1 are $p_1 = 0.9, p_2 = 0.8, p_3 = 0.5$. The expected reward values in cluster 1 are $r_1 = 150$, $r_2 = 155$ and $r_3 = 140$. For cluster (route) 2, it contains $v_4 = 30mph, v_5 = 45mph, v_6 = 60mph$). The safety indices for speeds in route 2 are $p_4 = 0.9, p_5 = 0.8, p_6 = 0.5$. The expected reward values in route 2 are $r_4 = 120$, $r_5 = 125$ and $r_6 = 110$. We can see that each cluster's expected reward function has only one peak, which satisfies the Unimodal property.

Appendix A. Unimodal Example

Example 1: Road navigation. A person driving from A to B has the option to choose two routes: highway and local way. After choosing a route, she needs to further choose a speed. In this example, a (route, speed) combination corresponds to an arm, and a route corresponds to a cluster. The expected reward (Utility) is defined as follows: $r_j = v_j + 10 \times p_j$, where v_j denotes velocity for arm j and p_j denotes safety (Sun et al., 2018) for arm j. Note that, if velocity increases, safety will decrease, and thus, each cluster's reward structure is oftentimes Unimodal. See Fig. 1 for a numerical example.

Example 2: Multi-channel mmWave communication. Consider optimal antenna beam selection for a mmWave communication link with multiple frequency channels. Theoretical analysis (Wu et al., 2019) and experimental results (Hashemi et al., 2018) indicate that the received signal strength (RSS) function over the beam space in the channel with a single path (or a dominant line-of-sight path) can be characterized by a Unimodal function. Our goal is to select the best channel and beam combination to maximize the link RSS. *In this example, the arm is the combination of frequency channel and beam, and the reward is the signal strength. We regard the beams under each channel as a cluster.* Our goal is to select the optimal channel and beam for communication in an online manner. In Fig. 2, we provide an illustration of the multi-channel mmWave communication example.

Appendix B. Algorithm for the Single-Cluster setting

We first study the single-cluster setting ($C = 1$), where the problem degenerates to a Unimodal bandit problem (Yu and Mannor, 2011; Combes and Proutiere, 2014). One drawback of prior works is that their guarantees have limited adaptivity: achieving gap-dependent and gap-independent regret bounds require setting parameters differently. In

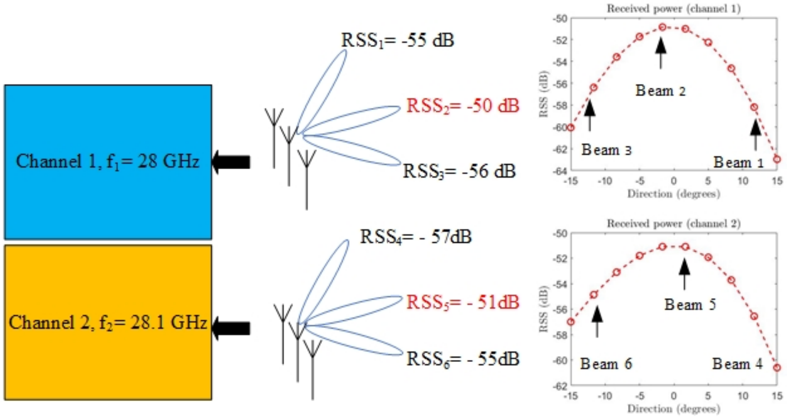


Figure 2: Multi-channel mmWave communication example. There are two channels: $f_1 = 28GHz, f_2 = 28GHz + 100MHz$ (These two frequencies are based on 3GPP TS 38.101-1/2, 38.104-1/2 (Lopez et al., 2019)). For each channel, the algorithm can select three beams. Experimental results in (Hashemi et al., 2018) show that the RSS function over the beam space in a fixed frequency is a Unimodal function.

Algorithm 2 Stochastic Golden Search for discrete arm (SGSD)

```
Parameters: $\epsilon_1, .... > 0$:
Initialize $x_A = 0$,$x_B = \frac{1}{\phi^2}$, $x_c = 1$ ($\phi = \frac{1+\sqrt{5}}{2}$)
for each stage $s = 1, 2, ...S$, do
  if there has more than one discrete arms $j/N$ in $[x_A, x_C]$ then
    Let
      $$x'_B = \begin{cases} x_B - \frac{1}{\phi^2}(x_B - x_A) & x_B - x_A > x_C - x_B \\ x_B + \frac{1}{\phi^2}(x_C - x_B) & otherwise, \end{cases}$$
    Obtain the reward of each continuous point $\{x_A, x_B, x'_B, x_C\}$ according to Alg. 3, each point for $\frac{2}{\epsilon_s^2}\log(8T)$ times, and let $\hat{x}$ be the point with highest empirical mean in this stage
    If $\hat{x} \in \{x_A, x_B\}$ then eliminate interval $(x'_B, x_C]$ and let $x_C = x'_B$,
    else eliminate interval $[x_A, x_B)$ and let $x_A = x_B$
  else
    Break
  end if
  Keep pulling the only discrete arm $j/N$ in $[x_A, x_C]$
end for
```

this work, we provide an algorithm that simultaneously enjoys gap-dependent and gap-independent regret guarantees, which is useful for practical deployment. Our algorithm is built on the SGS algorithm (Yu and Mannor, 2011), and we call it SGS for discrete arm setting algorithm (SGSD), namely, Algorithm 2.

The high level idea of SGSD is to reduce the discrete-arm Unimodal bandits problem to a continuous-arm Unimodal bandits, and use the SGS algorithm in the continuous arm

Algorithm 3 Reward sampling algorithm for an arbitrary continuous point x'

```
1: Input: $x'$
2: Output: a stochastic reward of conditional mean $f(x')$ (Eq. (7))
3: $j = \lfloor Nx' \rfloor$
4: set
   $$l = \begin{cases} j & \text{with probability} \quad j+1-Nx' \\ j+1 & \text{otherwise,} \end{cases}$$
5: $r \leftarrow$ reward of pulling arm $l$
6: **return** $r$
```

setting. Specifically, given a discrete-arm Unimodal bandit problem $\mu_1, \ldots, \mu_N$, we associate every arm j to a point j/N in the $[0,1]$ interval and perform linear interpolation, inducing a function

$$f(x) = \mu_j \cdot (j+1-Nx) + \mu_{j+1} \cdot (Nx-j), x \in [j/N, (j+1)/N) \tag{7}$$

over the continuous interval $[0,1]$, and use SGS to optimize it. Observe that f has minimum at $x^* = j^*/N$, and for $x \in [j/N, (j+1)/N)$, bandit feedback of $f(x)$ can be simulated by pulling arms randomly from $\{j, j+1\}$ (Algorithm 3; see subsequent paragraphs for more details). To this end, it narrows down the sampling interval, maintaining the invariant that with high probability, $j^*/N \in [x_A, x_C]$.

The SGSD algorithm proceeds as follows: first, the algorithm initialize parameters $x_A = 0$,$x_B = \frac{1}{\phi^2}$, $x_c = 1$ (line 2 in Alg. 2, where $\phi = \frac{1+\sqrt{5}}{2}$). In line 4, the algorithm checks the number of discrete arms in the range $[x_A, x_C]$; if only one arm j/N is in the range $[x_A, x_C]$, with high probability, it must be the case that $j = j^*$, i.e. we have identified the optimal arm – in this case, the algorithm breaks the loop and keep pulling that arm (line 12). Then, given three points $x_A < x_B < x_C$ where the distance of x_B to the other two points satisfy the golden ratio. The reason we choose three point is to ensure the elimination of a constant fraction of the sample interval that does not contain j^*/N in each iteration. Note that x_B may be closer to x_A or to x_C depending on the past updating value of the SGSD algorithm. The point x'_B is set in the larger interval between $x_B - x_A$ and $x_C - x_B$ (The updating procedure for x'_B is in Alg. 2's line 5). If we set $x_C - x_A = \ell$, the following equalities hold at any step of SGS algorithm: $x_B - x_A = \frac{\ell}{\phi^2}, x'_B - x_B = \frac{\ell}{\phi^3}, x_C - x'_B = \frac{\ell}{\phi^2}$. Then, we eliminate $[x_A, x_B)$ or (x'_B, x_C), depending on whether the smallest empirical mean value is found in set $\{x_A, x_B\}$ or $\{x'_B, x_C\}$ (Shown in 2's line 7 and 8). Algorithm 2 gives the detail of the algorithm.

Note that we convert the expected rewards of discrete arms into a continuous function, we need to simulate noisy values of f on $\{x_A, x_B, x'_B, x_C\}$ via queries to the discrete arms $\{1, \ldots, N\}$. We use Alg. 3 to calculate such "virtual" instantaneous rewards. Given input arm $x' \in [0,1]$, we determine the range $[j/N, (j+1)/N)$ that x' belongs to (Alg 3's line 3). In each iteration, we obtain its reward by the probabilistic sampling of the two discrete arms in x's neighborhood (where the sampling probability of each neighboring arm is shown in line 4), such that the output reward has expectation $f(x')$ (Shown in Alg. 3's line 4 -line 5).

We make the following assumptions similar to (Yu and Mannor, 2011):

Assumption 1 *(1) μ is strongly Unimodal: there exists a unique maximizer j^* of $\mu_1, \ldots, \mu_N$*[3]*.*

(2) There exist positive constants D_L and $D_H > 0$ such that $\left|\mu_j - \mu_{j+1}\right| \leq D_H$, and $\left|\mu_j - \mu_{j+1}\right| \geq D_L$ for all $j \in \{1, \ldots, N\}$.

Assumption 1.(1) ensures that the continuous function has one peak value. Assumption 1.(2) extends the validate domain becomes $[0, v_{j^*}]$ and $[v_{j^*}, 1]$. This is because each neighbor is connected by linear interpolation. So, our new continuous function has the lowest slope value which is determined by linear interpolation and D_L. Then, we have the following regret bound.

Theorem 2 *Under Assumption 1, the expected regret of Alg. 2, with $\epsilon_s = ND_L\phi^{-(s+3)}$, is:*

$$E[R(T)] \leq O\left(\min\left\{\frac{D_H}{D_L}\log(8T)\sqrt{T}, \frac{D_H}{(D_L)^2}\log(8T)\right\}\right). \tag{8}$$

The proof of the first bound in Theorem 1 is inspired by the analysis of SGS in (Yu and Mannor, 2011) after linear interpolation to reduce the discrete-arm setting to a continuous-arm setting. The second bound is inspired by the proof of Theorem IV.4 in (Yu and Mannor, 2011). From Theorem 1, we can see that the upper bound is independent of the number of arms. However, it depends on the problem-dependent constants (D_L, D_H).

We now compare this regret bound with that of the UCB algorithm (Auer et al., 2002). UCB has a gap-independent regret bound of O$(\sqrt{TN\log(T)})$, and gap-dependent regret bound of $O(\sum_{j\neq j^*}\frac{\log(T)}{\Delta_j})$ (where $\Delta_j = \mu_{j^*} - \mu_j$). Then, we examine UCB's gap-dependent bound in terms of D_H. Note that, the function is a Unimodal function, and the number of arms on either the left or the right side of the optimal arm j^* must be greater than $\frac{N}{2}$. Then, the gap-dependent regret bound of UCB must be larger than $\sum_{j=1}^{N/2}\frac{log(T)}{jD_H} = \frac{\log(T)}{D_H}\sum_{j=1}^{N/2}\frac{1}{j} = \Omega(\log(\frac{N}{2})\cdot\frac{\log(T)}{D_H})$. We therefore see that the regret bound of the UCB algorithm depends on the number of arms in both gap-independent and gap-dependent bounds, which does not apply to SGSD.

Appendix C. Experiments

We aim to answer the following questions through experiments:

1. Can SGSD outperform other algorithms in Unimodal bandit environments?
2. Can HUUCB outperform other hierarchical bandit algorithms (such as TLP) in MAB-CU environments? Meanwhile, we intend to validate whether the simulation result conforms to our theoretical analysis – specifically, does HUUCB's new $\frac{D_H}{D_L}\sqrt{\frac{\log(t)}{M_i(t)}}$ bonus term help in cluster selection?

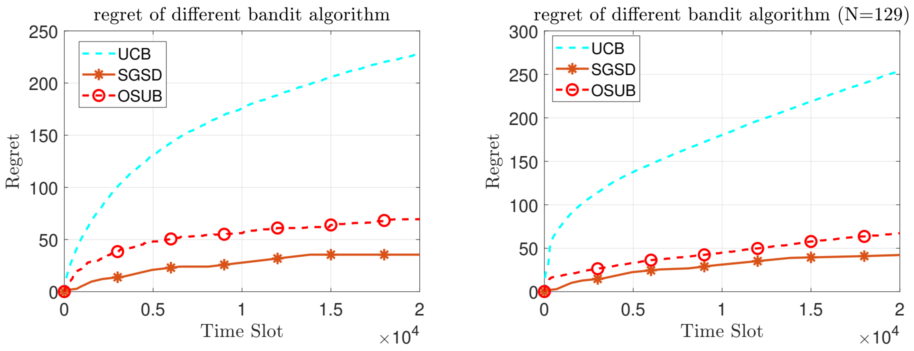
[3] Strong Unimodality means that it only has one optimal arm among the arm set.

To answer these questions, we consider two sets of experiments:

1. Learning in a synthetic Unimodal bandit setting, taken from (Combes and Proutiere, 2014). First, we consider $N = 17$ arms with Bernoulli rewards which $\mu = [0.1, 0.2...0.9, 0.8...0.1]$ and the rewards are Unimodal. Then, we consider $N = 129$, and the expected rewards form a triangular shape as in the previous example $N = 17$ (μ is between $[0.1, 0.9]$). We evaluate three algorithms: our SGSD algorithm, UCB (Auer et al., 2002), and OSUB (Combes and Proutiere, 2014).

2. Bandit learning in the MAB-CU setting. We use a simulated environment of multi-channel mmWave communication. We perform our simulations using MATLAB. Recall from Section 1.1 that in this application, an arm is a combination of channel and beam (chosen by the transmitter), and the reward is the received signal strength (RSS) at the receiver. We regard the beams under the same channel as a cluster. We fix the transmitter (i.e., base station) at location [0,0], and we randomly generate four receiver locations from a disk area with a radius of 10 meters. The base station is equipped with a uniform linear array (ULA) with four antennas, which are separated by a half wavelength. For the wireless channel model, we assume that there either exists only one line-of-sight (LOS) path or one non-line-of-sight (NLOS) path if the LOS path is blocked. We obtain the RSS under channel i and beam j in each time step using Monte-Carlo simulations, following the free-space signal propagation model: $RSS_{ij} = \alpha_i P^{TX} G_j^{RX} G_j^{TX} (\frac{\lambda_i}{4\pi d})^2$ (Molisch, 2012), where α_i is the random path fading amplitude under channel i (since there's a dominant LoS path, α_i is assumed to follow the Rician distribution (Samimi et al., 2016)), G_j^{RX} and G_j^{TX} are the gains of the receive and transmit antennas for beam j (in the directions of angle-of-arrival (AoA) and angle-of-departure (AoD)), respectively, λ_i is the wavelength for channel i ($j \in \text{Cluster}_i$), d is the distance between transmitter and receiver, and P^{TX} is the transmit power. Note that, the AoA, AoD, distance d, and fading α_i are all unknown to the transmitter during the bandit algorithm execution. We denote RSS_{ij} as the reward for beam j under channel i. The system is assumed to operate at 28GHz center carrier frequency (based on the 3GPP TS 38.101-1/2 standard (Lopez et al., 2019)), has a bandwidth of 100 MHz, and uses 16-QAM modulation. We consider two scenarios: 1) two channels (clusters): $[28 \sim 28.1,\ 28.1 \sim 28.2]$ GHz, 2) five channels (clusters): $[27.8 \sim 27.9,\ 27.9 \sim 28,\ 28 \sim 28.1,\ 28.1 \sim 28.2,\ 28.2 \sim 28.3]$ GHz. For each channel, there are a total of 16 beams and we only consider TX beam selection (each beam's width is 5 degrees and the step between adjacent beams' angles is 10 degrees.).

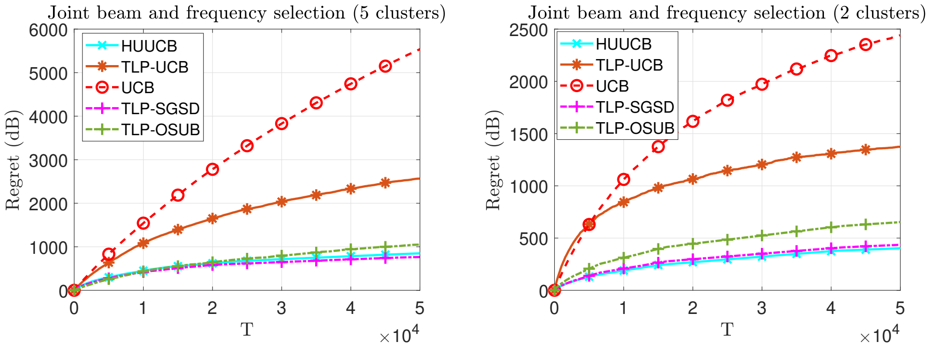
   We evaluate the following algorithms: (1) our HUUCB algorithm; (2) UCB algorithm (Auer et al., 2002) and (3) Instantiations of the Two-level Policy framework (TLP) of (Pandey et al., 2007) using different base algorithms for intra-cluster arm selection. In each step, the TLP algorithm chooses a cluster first, and then an actual arm within the cluster is selected. The key difference between algorithms under TLP framework and our HUUCB algorithm is that, TLP uses an aggressive confidence bound for selecting clusters, which does not follow the "optimism in the face of uncertainty" principle, and does not have theoretical guarantees. We consider TLP composed with three base algorithms: first, UCB, which does not utilize the Unimodal property in each cluster;



(a) Regret vs. time in stationary environments N=17

(b) Regret vs. time in stationary environments N=129

Figure 3: Comparison between UCB and SGSD algorithm under Unimodal setting



(a) Regret for joint beam and frequency (5 clusters)

(b) Regret for joint beam and frequency (2 clusters)

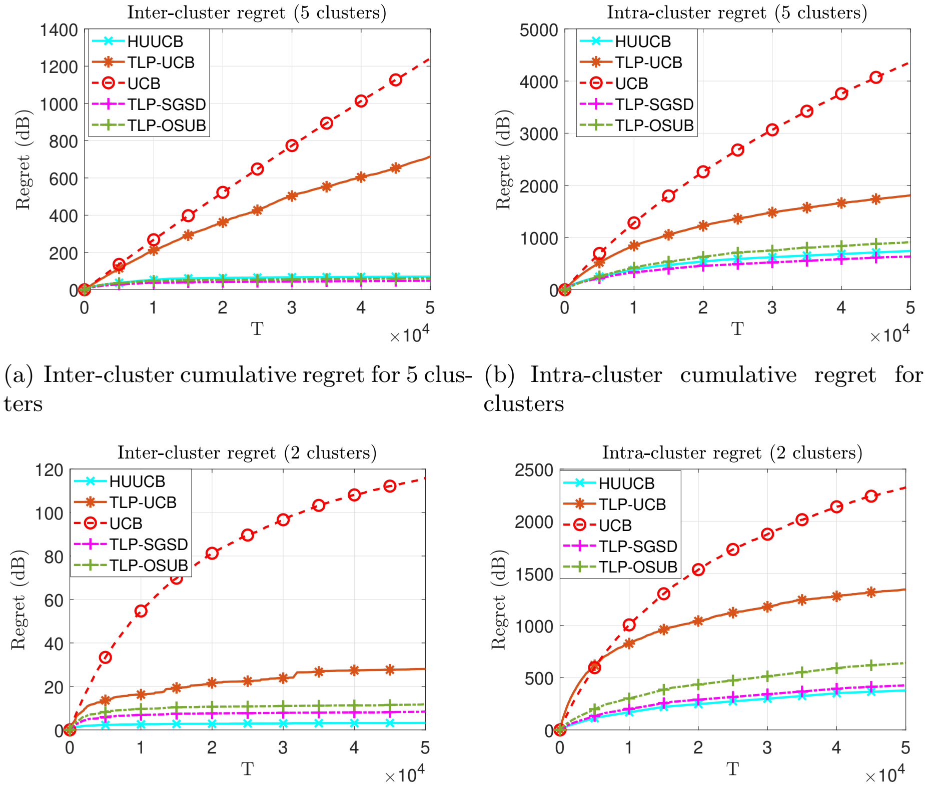
Figure 4: Comparison of cumulative regret among HUUCB and existing algorithms

second, SGSD, our algorithm 2; third, OSUB (Combes and Proutiere, 2014) – we call the resulting algorithms TLP-UCB, TLP-SGSD, and TLP-OSUB respectively.

C.1 Simulation Result

Fig. 3 shows the cumulative regret of our SGSD algorithm. Regrets are calculated averaging over 100 independent runs. SGSD significantly outperforms the UCB algorithm. This is because the UCB algorithm does not utilize the Unimodal property. Meanwhile, the SGSD algorithm has better performance than the OSUB algorithm. We speculate that SGSD's improved performance is due to its use of D_H and D_L, in contrast to OSUB.

Fig. 4(a) shows the cumulative regret of the joint beam and frequency selection with 5 clusters, and Fig. 4(b) shows the same result with 2 clusters. Regrets are calculated averaging over 20 independent runs. From Fig. 4(a) and 4(b), we can see that HUUCB has lower regret than the UCB and TLP-UCB algorithm. This result is consistent with our expectation since the Unimodal property in each cluster can help the algorithm converge faster. Meanwhile, we can see that our HUUCB algorithm has a similar performance as TLP-SGSD and TLP-OSUB algorithms, and has the best performance in the 2-cluster setting.



(a) Inter-cluster cumulative regret for 5 clusters

(b) Intra-cluster cumulative regret for 5 clusters

(c) Inter-cluster cumulative regret for 2 clusters

(d) Intra-cluster cumulative regret for 2 clusters

Figure 5: Comparison of Intra-cluster and inter-cluster cumulative regret

To further examine the advantage of our proposed algorithm over the baseline, we analyze the *inter-cluster* and *intra-cluster* cumulative regret of all algorithms. Intra-cluster cumulative regret is the regret that the algorithm chooses an arm that is not the optimal arm in the currently chosen cluster; formally, $R_{intra}(T) = \sum_{t=1}^{T}(\nu_{i_t} - \mu_{j_t})$. Inter-cluster cumulative regret is the regret that the algorithm chooses a suboptimal cluster, i.e. the cluster that does not contain the optimal arm; formally $R_{inter}(T) = \sum_{t=1}^{T}(\mu_{j^*} - \nu_{i_t})$. It can be seen that the regret can be decomposed as: $R(T) = R_{intra}(T) + R_{inter}(T)$. From Fig. 5, we can see that UCB and TLP-UCB algorithms incur both large inter-cluster cumulative regret and intra-cluster cumulative regret. This is because both algorithms do not fully utilize Unimodal and Hierarchical properties. Meanwhile, we can see that HUUCB has comparable performance to TLP-SGSD. In the 2-cluster setting, HUUCB has better inter-cluster regret than TLP-SGSD - this may be due to the setting of the extra bonus term in HUUCB.

Appendix D. Proof

For convenience, we define $C_H = ND_H, C_L = ND_L$; therefore, $|\mu_j - \mu_{j+1}| \leq C_H|v_j - v_{j+1}|$, and $|\mu_j - \mu_{j+1}| \geq C_L|v_j - v_{j+1}|$ for all $j \in \{1, \ldots, N\}$ (where v_j is the feature for arm j, and $v_j = \frac{j}{N}$).

D.1 Proof of Theorem 2

After linear interpolation, using elementary algebraic identities for ϕ, one can show that setting $x_C - x_A = 1$ the following equalities hold at any step of SGSD

$$x_B - x_A = \frac{1}{\phi^2}, x'_B - x_B = \frac{1}{\phi^3}, x_C - x'_B = \frac{1}{\phi^2} \tag{9}$$

Since either $x_B - x_A$ or $x_C - x'_B$ are eliminated at each stage, at each stage SGSD shrinks the search interval by a factor of $1 - \phi^{-2} = \frac{1}{\phi}$. Let $[x_{A,s}, x_{C,s}]$ be the search interval at the beginning of stage $s+1$, where $x_{A,0} = 0$ and $x_{C,0} = 1$.

Lemma If $\epsilon_s = C_L \phi^{-(s+3)}$ then

$$P(v_{j^*} \notin [x_{A,s}, x_{C,s}]) \leq \frac{s}{T}, \tag{10}$$

Proof Once the interval containing v_{j^*} is eliminated, it is never recovered. Thus, we have

$$\begin{aligned} &P(v_{j^*} \notin [x_{A,s}, x_{C,s}]) \leq P(v_{j^*} \notin [x_{A,s-1}, x_{C,s-1}]) + \\ &P(v_{j^*} \notin [x_{A,s}, x_{C,s}] | v_{j^*} \in [x_{A,s-1}, x_{C,s-1}]), \end{aligned} \tag{11}$$

Let $X_s = \{x_{A,s-1}, x_{B,s-1}, x_{B',s-1}, x_{C,s-1}\}$ where $x_{B,s-1} < x_{B',s-1}$ are computed in line 6 (Alg.1) of stage s. Let $\hat{\mu}_s(x)$ be the sample loss of point $x \in X_s$ in stage s and let $\hat{x}_s = \arg\min_{x \in X_s} \hat{\mu}(x)$. Since at stage s every point in X_s is played $\frac{2}{\epsilon^2}\log(6n)$ times, Hoeffding bounds imply that $|\mu(x) - \hat{\mu}_s(x)| \leq \epsilon_s$ with probability at least $1 - \frac{1}{6n}$ for all $x \in X_s$ simultaneously. Let,

$$x^*_s = \arg\min_{x \in X_s} \mu(x), \tag{12}$$

Now assume $v_{j^*} \in [x_{B',s-1}, x_{C,s-1}]$. Then $v_{j^*} \notin [x_{A,s}, x_{C,s}]$ implies $\hat{\mu}(x_{A,s-1}) < \hat{\mu}(x_{B',s-1})$ or $\hat{\mu}(x_{B,s-1}) < \hat{\mu}(x_{B',s-1})$. Similarly, assume $v_{j^*} \in [x_{A,s-1}, x_{B,s-1}]$. Then $v_{j^*} \notin [x_{A,s}, x_{C,s}]$ implies $\hat{\mu}(x_{B',s-1}) < \hat{\mu}(x_{B,s-1})$ or $\hat{\mu}(x_{C,s-1}) < \hat{\mu}(x_{B,s-1})$. In both cases, we need to compare three values of μ on the same side with respect to v_{j^*} (When $v_{j^*} \in [x_{B,s-1}, x_{B',s-1}]$ we always have $v_{j^*} \in [x_{A,s}, x_{C,s}]$). Hence, we can apply our assumption involving C_L. More precisely, (9) implies that after $s-1$ stages the search interval has size $\phi^{-(s-1)}$ and $\min\{x_{B,s-1} - x_{A,s-1}, x_{B',s-1} - x_{B,s-1}, x_{C,s-1} - x_{B',s-1}\} = \phi^{-(s+2)}$[4]. Hence, introducing

$$\Delta_{s'} = \begin{cases} \min\{x_{B',s-1} - x_{B,s-1}, x_{C,s-1} - x_{B',s-1}\} = \phi^{-(s+2)} & v_{j^*} \in [x_{A,s-1}, x_{B,s-1}] \\ \min\{x_{B,s-1} - x_{A,s-1}, x_{B',s-1} - x_{B,s-1}\} = \phi^{-(s+2)} & v_{j^*} \in [x_{B',s-1}, x_{C,s-1}], \end{cases}$$

[4] In (Bubeck and Cesa-Bianchi, 2012) proof, it applies $\{x_{A,s}, x_{B,s}, x_{B',s}, x_{C,s}\}$ to the Hoeffding bound. However, in stage s, it uses $X_s = \{x_{A,s-1}, x_{B,s-1}, x_{B',s-1}, x_{C,s-1}\}$ to operate the algorithm. Note that, $v_{j^*} \in [x_{A,s-1}, x_{C,s-1}]$ and assumption 1 (2) cannot hold because $C_L|v_j - v_{j+1}| \leq |r(v_j) - r(v_{j+1})|$ for neighboring point $v_j, v_{j+1} \in [0, v_{j^*}]$ or $v_j, v_{j+1} \in [v_{j^*}, 1]$ (It needs the two points in the same side of v_{j^*}. However, $x_{A,s-1}$ and $x_{C,s-1}$ are not in the same side v_{j^*}). We divide into two cases 1)$v_{j^*} \in [x_{A,s-1}, x_{B,s-1}]$ 2) $v_{j^*} \in [x_{B',s-1}, x_{C,s-1}]$. In first case, we only consider the interval $[x_{B,s-1}, x_{C,s-1}]$ and we can satisfy assumption 1 (2) because the interval $[x_{B,s-1}, x_{C,s-1}]$ in the same side of v_{j^*}. Similarly, in second case, it also satisfy assumption 1 (2).

If $v_{j^*} \in [x_{A,s-1}, x_{B,s-1}]$, we have

$$\Delta_{s'} \geq C_L \min\{x_{B',s-1} - x_{B,s-1}, x_{C,s-1} - x_{B',s-1}\} \geq C_L \phi^{-(s+2)} = \epsilon_s \phi^{-1} \tag{13}$$

Similarly, if $v_{j^*} \in [x_{B',s-1}, x_{C,s-1}]$, we have

$$\Delta_{s'} \geq C_L \min\{x_{B,s-1} - x_{A,s-1}, x_{B',s-1} - x_{B,s-1}\} \geq C_L \phi^{-(s+2)} = \epsilon_s \phi^{-1} \tag{14}$$

Next, we calculate the probability of $P(v_{j^*} \notin [x_{A,s}, x_{C,s}] | v_{i^*} \in [x_{A,s-1}, x_{C,s-1}])$. For simplicity, we define event $\boldsymbol{A} : v_{j^*} \notin [x_{A,s}, x_{C,s}]$, event $\boldsymbol{B} : v_{j^*} \in [x_{A,s-1}, x_{C,s-1}]$, sub-case 1: $\boldsymbol{B}_1 : v_{j^*} \in [x_{A,s-1}, x_{B,s-1}]$, sub-case 2: $\boldsymbol{B}_2 : v_{j^*} \in [x_{B,s-1}, x_{B',s-1}]$ and $\boldsymbol{B}_3 : v_{j^*} \in [x_{B',s-1}, x_{C,s-1}]$. we can write

$$\begin{aligned}
&P(\boldsymbol{A}|\boldsymbol{B}) = \frac{P(\boldsymbol{A},\boldsymbol{B})}{P(\boldsymbol{B})} \overset{(a)}{=} \frac{P(\boldsymbol{A},\boldsymbol{B}_1) + P(\boldsymbol{A},\boldsymbol{B}_2) + P(\boldsymbol{A},\boldsymbol{B}_3)}{P(\boldsymbol{B})} \\
&= \frac{P(\boldsymbol{A},\boldsymbol{B}_1)}{P(\boldsymbol{B}_1)} \frac{P(\boldsymbol{B}_1)}{P(\boldsymbol{B})} + \frac{P(\boldsymbol{A},\boldsymbol{B}_2)}{P(\boldsymbol{B}_2)} \frac{P(\boldsymbol{B}_2)}{P(\boldsymbol{B})} + \frac{P(\boldsymbol{A},\boldsymbol{B}_3)}{P(\boldsymbol{B}_3)} \frac{P(\boldsymbol{B}_3)}{P(\boldsymbol{B})} \\
&= P(\boldsymbol{A}|\boldsymbol{B}_1) \frac{P(\boldsymbol{B}_1)}{P(\boldsymbol{B})} + P(\boldsymbol{A}|\boldsymbol{B}_2) \frac{P(\boldsymbol{B}_2)}{P(\boldsymbol{B})} + P(\boldsymbol{A}|\boldsymbol{B}_3) \frac{P(\boldsymbol{B}_3)}{P(\boldsymbol{B})} \\
&\overset{(b)}{\leq} P(\boldsymbol{A}|\boldsymbol{B}_1) + P(\boldsymbol{A}|\boldsymbol{B}_2) + P(\boldsymbol{A}|\boldsymbol{B}_3)
\end{aligned} \tag{15}$$

Eq.(a) is based on the fact $\boldsymbol{B} = \boldsymbol{B}_1 + \boldsymbol{B}_2 + \boldsymbol{B}_3$ and marginal probability. Inequality (b) is based on the fact that $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, $\boldsymbol{B}_3$ are the subset of $\boldsymbol{B}$. So, we have $\frac{P(\boldsymbol{B}_1)}{P(\boldsymbol{B})} \leq 1$, $\frac{P(\boldsymbol{B}_2)}{P(\boldsymbol{B})} \leq 1$ and $\frac{P(\boldsymbol{B}_3)}{P(\boldsymbol{B})} \leq 1$. First, we calculate $P(\boldsymbol{A}|\boldsymbol{B}_1)$. In sub-case 1, $v_{i^*} \notin [x_{A,s}, x_{C,s}]$ implies $\hat{\mu}(x_{B',s-1}) < \hat{\mu}(x_{B,s-1})$ or $\hat{\mu}(x_{C,s-1}) < \hat{\mu}(x_{B,s-1})$. Then, we have

$$\begin{aligned}
&P(\boldsymbol{A}|\boldsymbol{B}_1) \leq P(\hat{\mu}(x_{B',s-1}) < \hat{\mu}(x_{B,s-1})) + P(\hat{\mu}(x_{C,s-1}) < \hat{\mu}(x_{B,s-1})) \\
&\overset{(a)}{\leq} P(\hat{\mu}(x_{B',s-1}) < \mu(x_{B',s-1}) - \frac{\Delta_{s'}}{2}) + P(\mu(x_{B,s-1}) < \hat{\mu}(x_{B,s-1}) - \frac{\Delta_{s'}}{2}) \\
&+ P(\hat{\mu}(x_{C,s-1}) < \mu(x_{C,s-1}) - \frac{\Delta_{s'}}{2}) + P(\mu(x_{B,s-1}) < \hat{\mu}(x_{B,s-1}) - \frac{\Delta_{s'}}{2}) \\
&\leq 4e^{-T_s \Delta_{s'}^2/8} \leq 4e^{-T_s(\epsilon_s \phi^{-1})^2/8} \leq \frac{1}{2T}
\end{aligned} \tag{16}$$

Inequality (a) is based on the following facts

$$\begin{aligned}
&P(\hat{\mu}(x_{B',s-1}) < \hat{\mu}(x_{B,s-1})) \\
&= P(\hat{\mu}(x_{B',s-1}) - \mu(x_{B',s-1}) - \hat{\mu}(x_{B,s-1}) + \mu(x_{B,s-1}) + \mu(x_{B',s-1}) - \mu(x_{B,s-1}) < 0) \\
&\leq P(\hat{\mu}(x_{B',s-1}) - \mu(x_{B',s-1}) + \frac{\mu(x_{B',s-1}) - \mu(x_{B,s-1})}{2} < 0) + \\
&P(\mu(x_{B,s-1}) - \hat{\mu}(x_{B,s-1}) + \frac{\mu(x_{B',s-1}) - \mu(x_{B,s-1})}{2} < 0) \\
&\overset{(a)}{\leq} P(\hat{\mu}(x_{B',s-1}) < \mu(x_{B',s-1}) - \frac{\Delta_{s'}}{2}) + P(\mu(x_{B,s-1}) < \hat{\mu}(x_{B,s-1}) - \frac{\Delta_{s'}}{2})
\end{aligned} \tag{17}$$

Inequality (a) is based on the fact that $\frac{\mu(x_{B',s-1})-\mu(x_{B,s-1})}{2} \leq \frac{\Delta_{s'}}{2}$, and $\frac{\Delta_{s'}}{2}$ makes that the event becomes much easier to smaller than 0 because we adding a smaller number. We can obtain a similar result for $P(\hat{\mu}(x_{C,s-1}) < \hat{\mu}(x_{B,s-1}))$. Due to space limitations, we omit the details. Next, we calculate $P(\boldsymbol{A}|\boldsymbol{B}_2)$. Note that, when $v_{i^*} \in [x_{B,s-1}, x_{B',s-1}]$, we always have $v_{i^*} \in [x_{A,s}, x_{C,s}]$. Then, we have,

$$P(\boldsymbol{A}|\boldsymbol{B}_2) = 0 \tag{18}$$

Last, we calculate $P(\boldsymbol{A}|\boldsymbol{B}_3)$. The procedure to calculate $P(\boldsymbol{A}|\boldsymbol{B}_3)$ is same to the $P(\boldsymbol{A}|\boldsymbol{B}_1)$.Then, we have,

$$P(\boldsymbol{A}|\boldsymbol{B}_3) \leq \frac{1}{2T} \tag{19}$$

Combing (16) (18) (19), we have

$$P(\boldsymbol{A}|\boldsymbol{B}) \leq \frac{1}{T} \tag{20}$$

Substituting this in (11) and recurring gives the desired result. We start by decomposing the pseudo-regret as follows:

$$E[R(T)] \leq \sum_{s=1}^{S} T_s(\min_{x \in A_s} \mu(x) - \mu(v_{i^*})) + \sum_{s=1}^{S}(\sum_{t \in T_s} \mu(x_t) - T_s \min_{x \in A_s} \mu(x)), \tag{21}$$

Using the Lipschitz assumption

$$\max_{x,x' \in A_s} |\mu(x) - \mu(x')| \leq C_H |x_{C,s} - x_{A,s}|, \tag{22}$$

and recalling that $|x_{C,s} - x_{A,s}| \leq \phi^{-s}$, we bound the first term in this decomposition as follows

$$\begin{aligned} &T_s(\min_{x \in A_s} \mu(x) - \mu(v_{i^*})) \leq T_s C_H |x_{C,s} - x_{A,s}| P(v_{i^*} \in [x_{A,s}, x_{C,s}]) \\ &+ T_s C_H P(v_{i^*} \notin [x_{A,s}, x_{C,s}]) \leq \frac{T_s C_H}{\phi^s} + T_s C_H \frac{s}{T} \end{aligned} \tag{23}$$

The second term is controlled similarly,

$$\sum_{t \in T_s} \mu(x_t) - T_s \min_{x \in A_s} \mu(x) \leq T_s C_H |x_{C,s} - x_{A,s}| \leq \frac{T_s C_H}{\phi^s} \tag{24}$$

Hence we get an easy expression for the regret,

$$
\begin{aligned}
E[R(T)] &\leq C_H \sum_{s=1}^{S} T_s(\frac{2}{\phi^s} + \frac{s}{T}) \\
&\leq \frac{C_H}{C_L^2} 8\phi^6 \log(8T) \sum_{s=1}^{S} \phi^{2s}(\frac{2}{\phi^s} + \frac{s}{T})
\end{aligned} \tag{25}
$$

We now compute an upper bound on the number S of stages,

$$
T \geq \sum_{s=1}^{S} T_s = \frac{8\phi^6}{C_L^2} \log(8T) \sum_{s=1}^{S} \phi^{2s} = \frac{8\phi^6}{C_L^2} \log(8T) \frac{\phi^{2S+2} - \phi^2}{\phi^2 - 1} \tag{26}
$$

Solving for T and over approximating, we get

$$
S \leq \frac{1}{2} \log_\phi(1 + C_L^2 T) \tag{27}
$$

Therefore, the sum in (25) is bounded as follows

$$
2\sum_{s=1}^{S} \phi^s + S^2 \leq \frac{2\phi}{\phi - 1}\sqrt{1 + C_L^2 T} + \frac{1}{4} \log_\phi^2(1 + C_L^2 T) \tag{28}
$$

Next, we derive the regret bound when we consider a final stage. In a similar fashion to the proof of algorithm 2, the regret of the s-th iteration is bounded by,

$$
L_s \leq \frac{2C_H}{\phi^s} T_s + T_s C_H \frac{s}{T}, \tag{29}
$$

since the sampling interval is $\frac{1}{\phi^s}$ at the s-th iteration, the sampling interval is at most $\frac{1}{N}$ wide after $\log_\phi(N)$ and hence contains a single arm. Therefore, the algorithm 2 continues picking the same arm after $\log_\phi(N)$ iterations. Observe also that the length of the s-th stage of the algorithm 2 is

$$
T_s = \frac{2}{(\epsilon_s)^2} \log(8T) = \frac{2\phi^{(2s+6)}}{(ND_L)^2} \log(8T) \tag{30}
$$

for all T. Combining equation (29) and (30), we have,

$$
\begin{aligned}
L_s &\leq \frac{2ND_H}{\phi^s} \frac{2\phi^{(2s+6)}}{(ND_L)^2} \log(8T) + \frac{2\phi^{(2s+6)}}{(ND_L)^2} \log(8T) ND_H \frac{s}{T} \\
&= \frac{4\phi^6 D_H \phi^s}{N(D_L)^2} \log(8T) + \frac{2\phi^{(2s+6)} D_H}{N(D_L)^2} \log(8T) \frac{s}{T},
\end{aligned} \tag{31}
$$

Note that the largest iteration time is $\log_\phi(N)$. Hence the total regret is,

$$\begin{aligned} E[R(T)] &\leq \frac{2\phi^6 D_H}{N(D_L)^2}\log(8T)\sum_{s=1}^{\log_\phi(N)}(2\phi^s+\phi^{(2s)}\frac{s}{T}) \\ &\leq \frac{2\phi^6 D_H}{N(D_L)^2}\log(8T)(\frac{2\phi(1-N)}{1-\phi}+\frac{\phi^2\frac{(1-N^2)}{1-\phi^2}\log_\phi(N)}{T}) \\ &\leq \frac{2\phi^6 D_H}{N(D_L)^2}\log(8T)(\frac{2\phi(1-N)}{1-\phi}+\frac{\phi^2\frac{(1-N^2)}{1-\phi^2}\log_\phi(m)}{T}), \end{aligned} \tag{32}$$

Note that, the term $\frac{2\phi(1-N)}{1-\phi}$ is at the order of $\Theta(N)$. The second term becomes to a constant value when $T \geq \phi^2\frac{(1-N^2)}{1-\phi^2}\log_\phi(N)$ (T is time horizon so we can choose a large enough stage). When $T \geq \phi^2\frac{(1-N^2)}{1-\phi^2}\log_\phi(N)$, the regret is,

$$E[R(T)] \leq O(\frac{2\phi^6 D_H}{(D_L)^2}\log(8T)), \tag{33}$$

Combining Inequality (28) (33), we get

$$\begin{aligned} &E[R(T)] \leq \min\{\frac{C_H}{C_L^2}8\phi^6\log(8T)\{\frac{2}{1-\frac{1}{\phi}}\sqrt{1+C_L^2T}+\frac{1}{4}\log_\phi^2(1+C_L^2T)\},O(\frac{2\phi^6 D_H}{(D_L)^2}\log(8T))\} \\ &\leq \min\{O(\frac{C_H}{C_L}\log(8T)\sqrt{T}),O(\frac{2\phi^6 D_H}{(D_L)^2}\log(8T))\} \\ &= \min\{O(\frac{D_H}{D_L}\log(8T)\sqrt{T}),O(\frac{2\phi^6 D_H}{(D_L)^2}\log(8T))\}. \end{aligned} \tag{34}$$

■

D.2 Proof of Theorem 1

Recall that we have C clusters, and each cluster has B arms.

We define $UCB(\nu_i) = \hat{\nu}_i(t)+\sqrt{\frac{2\log(T)}{M_i(t)}}+\frac{D_H}{D_L}\sqrt{\frac{\log(T)}{M_i(t)}} = \hat{\nu}_i(t)+\sqrt{\frac{2\log(T)}{M_i(t)}}+\frac{C_H}{C_L}\sqrt{\frac{\log(T)}{M_i(t)}}$. Where $\hat{\nu}_i(t) = \sum_{b\in i} w_{i,b}(t)\hat{\mu}_{i,b}(t)$, $M_i(t)$ is selected time for cluster i. We define event $\boldsymbol{E}$: $|\hat{\nu}_i(t)-\nu_i| \leq \sqrt{\frac{2\log(T)}{M_i(t)}}+\frac{C_H}{C_L}\sqrt{\frac{\log(T)}{M_i(t)}}$, where $\nu_i = \max_{j\in i}\mu_j$. Conditioning on the event $\boldsymbol{E}$, we have

$$\sum_{t=BC+1}^{T}(\nu_{i^*}-\nu_{i_t}) \leq \sum_{t=BC+1}^{T}(UCB_t(i^*)-\nu_{i_t}) \leq \sum_{t=BC+1}^{T}(UCB_t(i_t)-\nu_{i_t})$$
$$\leq 2\sum_{t=BC+1}^{T}\{\sqrt{\frac{2\log(T)}{M_{i_t}(t)}}+\frac{C_H}{C_L}\sqrt{\frac{\log(T)}{M_{i_t}(t)}}\}$$
$$= 2\sum_{t=BC+1}^{T}\sum_{i=1}^{C}I(i=i_t)\{\sqrt{\frac{2\log(T)}{M_{i_t}(t)}}+\frac{C_H}{C_L}\sqrt{\frac{\log(T)}{M_{i_t}(t)}}\}$$
$$= 2\sum_{i=1}^{C}\sum_{s=1}^{M_i(T)}\{\sqrt{\frac{2\log(T)}{s}}+\frac{C_H}{C_L}\sqrt{\frac{\log(T)}{s}}\} \overset{(a)}{\leq} 2\sum_{i=1}^{C}2\sqrt{2M_i(T)\log(T)}+$$
$$2\frac{C_H}{C_L}\sum_{i=1}^{C}\sum_{s=1}^{M_i(T)}\sqrt{\frac{\log(T)}{s}} \overset{(b)}{\leq} 8\sqrt{2CT\log(T)}+\frac{C_H}{2C_L}\sqrt{CT\log(T)}, \tag{35}$$

Inequality (a) (b) are based on Cauchy-Schwarz inequality. Next, we derive $P(\boldsymbol{E}^c)$. First, we divide $|\hat{\nu}_i(t)-\nu_i|$ into,

$$|\hat{\nu}_i(t)-\nu_i| \leq |\nu_i-\sum_b w_{i,b}(t)\mu_{i,b}|+|\hat{\nu}_i(t)-\sum_b w_{i,b}(t)\mu_{i,b}|, \tag{36}$$

We define event $\boldsymbol{E}_1$ is $|\nu_i-\sum_b w_{i,b}(t)\mu_{i,b}| \leq \frac{C_H}{C_L}\sqrt{\frac{\log(T)}{M_i(t)}}$, and event $\boldsymbol{E}_2$ is $|\hat{\nu}_i(t)-\sum_b w_{i,b}(t)\mu_{i,b}| \leq \sqrt{\frac{2\log(T)}{M_i(t)}}$. Then, we know that if $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ hold, E must hold. Then, the $P(\boldsymbol{E}^c)$ is,

$$P(\boldsymbol{E}^c) \leq P(\boldsymbol{E}_2^c\cap\boldsymbol{E}_1)+P(\boldsymbol{E}_2\cap\boldsymbol{E}_1^c)+P(\boldsymbol{E}_2^c\cap\boldsymbol{E}_1^c)$$
$$= P(\boldsymbol{E}_2^c)P(\boldsymbol{E}_1|\boldsymbol{E}_2^c)+P(\boldsymbol{E}_1^c)P(\boldsymbol{E}_2|\boldsymbol{E}_1^c)+P(\boldsymbol{E}_2^c)P(\boldsymbol{E}_1^c|\boldsymbol{E}_2^c)$$
$$\overset{(a)}{\leq} 2P(\boldsymbol{E}_2^c)+P(\boldsymbol{E}^c), \tag{37}$$

Inequality (a) is based on the fact that $P(\boldsymbol{E}_1|\boldsymbol{E}_2^c)\leq 1$, $P(\boldsymbol{E}_2|\boldsymbol{E}_1^c)\leq 1$ and $P(\boldsymbol{E}_1^c|\boldsymbol{E}_2^c)\leq 1$. Then, according to Azuma's inequality, we derive $P(\boldsymbol{E}_2^c)$

$$P(\boldsymbol{E}_2^c) = P(|\hat{\nu}_i(t)-\sum_b w_{i,b}(t)\mu_{i,b}| > \sqrt{\frac{2\log(T)}{M_i(t)}}) \leq exp(-2\epsilon^2 M_{i,b}(t))$$
$$= exp(-4\frac{\log(T)}{M_{i,b}(t)}M_{i,b}(t)) = T^{-4}, \tag{38}$$

According to lemma 6.4 in (Bubeck and Cesa-Bianchi, 2012), $P(\boldsymbol{E}_1^c)$ is

$$P(\boldsymbol{E}_1^c) \leq T^{-1}, \tag{39}$$

Combining (37), (38) and (39), we get

$$P(\boldsymbol{E}^c) \leq 2T^{-4} + T^{-1} = O(T^{-1}), \tag{40}$$

Therefore, we have:

$$\begin{aligned} Regret(T) &\leq 8\sqrt{2CT\log(T)} + \frac{C_H}{2C_L}\sqrt{CT\log(T)} + P(E^c) \times T \\ &= O(\frac{C_H}{C_L}\sqrt{2CT\log(T)}) = O(\frac{D_H}{D_L}\sqrt{2CT\log(T)}). \end{aligned} \tag{41}$$

References

Yasin Abbasi-Yadkori, Aldo Pacchiano, and My Phan. Regret balancing for bandit and rl model selection. *arXiv preprint arXiv:2006.05491*, 2020.

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

Nathan Blinn, Jana Boerger, and Matthieu Bloch. mmwave beam steering with hierarchical optimal sampling for unimodal bandits. In *ICC 2021-IEEE International Conference on Communications*, pages 1–6. IEEE, 2021.

Djallel Bouneffouf, Srinivasan Parthasarathy, Horst Samulowitz, and Martin Wistub. Optimal exploitation of clustering and history information in multi-armed bandit. *arXiv preprint arXiv:1906.03979*, 2019.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

Emil Carlsson, Devdatt Dubhashi, and Fredrik D Johansson. Thompson sampling for bandits with clustered arms. *arXiv preprint arXiv:2109.01656*, 2021.

Richard Combes and Alexandre Proutiere. Unimodal bandits: Regret lower bounds and optimal algorithms. In *International Conference on Machine Learning*, pages 521–529. PMLR, 2014.

Ashok Cutkosky, Christoph Dann, Abhimanyu Das, Claudio Gentile, Aldo Pacchiano, and Manish Purohit. Dynamic balancing for model selection in bandits and rl. In *International Conference on Machine Learning*, pages 2276–2285. PMLR, 2021.

Morteza Hashemi, Ashu Sabharwal, C Emre Koksal, and Ness B Shroff. Efficient beam alignment in millimeter wave systems using contextual bandits. In *IEEE INFOCOM 2018*, pages 2393–2401. IEEE, 2018.

Matthieu Jedor, Vianney Perchet, and Jonathan Louedec. Categorized bandits. *Advances in Neural Information Processing Systems*, 32, 2019.

Suhansanu Kumar, Heting Gao, Changyu Wang, Kevin Chen-Chuan Chang, and Hari Sundaram. Hierarchical multi-armed bandits for discovering hidden populations. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 145–153, 2019.

Aida Vera Lopez, Andrey Chervyakov, Greg Chance, Sumit Verma, and Yang Tang. Opportunities and challenges of mmwave nr. *IEEE Wireless Communications*, 26(2):4–6, 2019.

Andreas F Molisch. *Wireless communications*, volume 34. John Wiley & Sons, 2012.

Trong T Nguyen and Hady W Lauw. Dynamic clustering of contextual multi-armed bandits. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1959–1962, 2014.

Sandeep Pandey, Deepayan Chakrabarti, and Deepak Agarwal. Multi-armed bandit problems with dependent arms. In *Proceedings of the 24th international conference on Machine learning*, pages 721–728, 2007.

Mathew K Samimi, George R MacCartney, Shu Sun, and Theodore S Rappaport. 28 ghz millimeter-wave ultrawideband small-scale fading models in wireless channels. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pages 1–6. IEEE, 2016.

Rahul Singh, Fang Liu, Yin Sun, and Ness Shroff. Multi-armed bandits with dependent arms. *arXiv preprint arXiv:2010.09478*, 2020.

Mingshun Sun, Ming Li, and Ryan Gerdes. Truth-aware optimal decision-making framework with driver preferences for v2v communications. In *2018 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2018.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Wen Wu, Nan Cheng, Ning Zhang, Peng Yang, Weihua Zhuang, and Xuemin Shen. Fast mmwave beam alignment via correlated bandit learning. *IEEE Transactions on Wireless Communications*, 18(12):5894–5908, 2019.

Junwen Yang, Zixin Zhong, and Vincent YF Tan. Optimal clustering with bandit feedback. *arXiv preprint arXiv:2202.04294*, 2022.

Jia Yuan Yu and Shie Mannor. Unimodal bandits. 2011.

Yi Zhang, Soumya Basu, Sanjay Shakkottai, and Robert W Heath Jr. Mmwave codebook selection in rapidly-varying channels via multinomial thompson sampling. In *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 151–160, 2021.

Tianchi Zhao, Ming Li, and Matthias Poloczek. Fast reconfigurable antenna state selection with hierarchical thompson sampling. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.