Invited Paper: A Combinatorial Multi-Armed Bandit Approach for Stochastic Facility Allocation Problem

Abdalaziz Sawwan and Jie Wu
{sawwan,jiewu}@temple.edu
Temple University
Philadelphia, Pennsylvania, USA

ABSTRACT

Combinatorial Multi-Armed Bandit (CMAB) techniques have been widely used to solve many real-world problems. This paper presents a geometric version of a stochastic facility allocation problem and proposes a novel approach using CMAB to solve it. This problem is concerned with optimally locating facilities in a 2-dimensional space with spatially variant population density, where the aim is to maximize the expected attraction of the population to the facility points over multiple rounds. We propose an algorithm based on the Upper Confidence Bound (UCB) principle and leverage Voronoi diagrams to model the attraction of the population to facilities. We consider both the Manhattan and Euclidean distances in the attraction model. The model has a broad range of potential applications, making the solution a feasible approach for similar optimization problems in dynamic and uncertain environments. To the best of our knowledge, this is the first work that applies CMAB models on 2-dimensional spaces with uncertain underlying distributions. Our findings show the potential of machine learning-based solutions, such as the CMAB approach, in advancing the design and implementation of distributed computing systems. Furthermore, we derive the regret bounds of the proposed algorithm. This analysis is complemented by numerical simulations demonstrating the efficiency and effectiveness of our method. The simulation was done on both real-world traces and synthesized data.

CCS CONCEPTS

• **Theory of computation** → **Online learning algorithms**; • **Applied computing** → *Mathematics and statistics*; • **Information systems** → **Geographic information systems**; • **Computer systems organization** → *Grid computing*.

KEYWORDS

Budget-constrained, delayed feedback, learning theory, multi-armed bandit, upper-confidence bound.

ACM Reference Format:
Abdalaziz Sawwan and Jie Wu. 2024. Invited Paper: A Combinatorial Multi-Armed Bandit Approach for Stochastic Facility Allocation Problem. In *Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating algorithms for Distributed systems (ApPLIED'24), June 17, 2024, Nantes, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3663338.3663677

ApPLIED'24, June 17, 2024, Nantes, France

ACM ISBN 979-8-4007-0670-7/24/06
https://doi.org/10.1145/3663338.3663677

1 INTRODUCTION

Facility allocation involves the strategic placement of resources or services within a defined space to optimize their accessibility and effectiveness. This concept is crucial in a variety of domains, including urban planning, telecommunications, and computing infrastructure. In the context of this paper, we define facility allocation as the decision-making process for optimizing spatial resources within geometric spaces faced with dynamic and uncertain conditions, such as uncertain population densities.

In this setting, we need to consider that the decisions have to be made iteratively, round by round, with the aim of maximizing the total reward obtained over these rounds [1, 2]. The complexity is further amplified in distributed computing environments, such as cloud and edge computing, where the allocation decisions impact the efficiency of service delivery, the overall system resilience, and cybersecurity posture [3]. This dynamic setup can be seen in other real-world scenarios, such as locating emergency services in a city where demand can vary from neighborhood to neighborhood and from time to time, or placing cellular towers in an area where the population density is spatially variant [4, 5].

The challenge here arises from the need to make the best decision possible at each round under the geometric settings of the problem, while taking into account the uncertainty and variability of the situation, along with the inherent combinatorial nature of the problem. The combinatorial nature of the problem comes from the fact that in each iteration (round), we do not simply decide to place a single facility, but we decide on the locations of multiple facilities at once [6, 7]. The iterative nature of decision-making in this context —round-by-round optimization for maximizing total reward— mirrors the dynamic allocation and scaling challenges faced in distributed systems.

Our work is the first to address these challenges by introducing a geometric version of the combinatorial multi-armed bandit (CMAB) problem tailored for facility allocation. The arms in this problem are represented by the possible sets of locations where we can place the facilities, and the reward from pulling an arm is equivalent to the total attraction of the population to the facilities located at the positions specified by the arm. The attraction of the population to a facility is modeled to be inversely proportional to the distance between the population and the facility, and to take into account the variability of the population density, we divide the whole space into fine-grained cells, each with its own uniform population density value. Moreover, we model the attraction of the population of a cell to multiple facilities using the Voronoi diagram concept [8, 9], which allows us to account for the effect of the spatial distribution of the facilities on the population attraction. This is where the geometric nature of the problem is manifested, as it allows us to account for both the spatial variance in population density and the combinatorial nature of the facility allocation.

This novel CMAB setup introduces many challenges. For example, we need to devise a good strategy that allows us to strike the right balance between exploring new sets of facility locations that might potentially yield higher total attraction and exploiting the ones we already know to yield good attraction. Furthermore, the strategy should be able to adapt to fit the geometric nature of the problem, as choosing a set of locations for K facilities at a certain round (pulling K arms) would entail obtaining a reward that depends on the spatial coordinates of those facilities, where the reward is stochastic in nature because the attraction of the underlying population is modeled stochastically.

In this paper, we propose an algorithm based on the Upper Confidence Bound (UCB) principle to tackle this problem. This algorithm, dubbed the Geometric-UCB, is equipped with the ability to handle both Manhattan and Euclidean distances, which adds to its flexibility and adaptability. We also provide a detailed mathematical analysis of this algorithm which includes proving an upper bound on the regret that comes from the algorithm, thus showing that the algorithm is not only practical, but also theoretically sound.

The geometric facility allocation problem we propose in this paper, coupled with the Geometric-UCB algorithm, presents a robust framework for addressing some real-world applications. The most apparent applications lie in the domain of urban planning and infrastructural development. For instance, government agencies could harness our model to optimize the placement of public facilities such as parks, hospitals, or schools across a city, where the goal is to maximize their utilization by the resident population [10, 11].

Similarly, corporations in the private sector, such as telecommunication providers or utility companies, could utilize our model to optimally position infrastructure such as cell towers or power stations to best serve their customers [12, 13]. Further applications could extend to the placement of commercial outlets in a city, determining the optimal locations for warehouses within a supply chain network, or even assisting humanitarian efforts by helping to decide where to place aid stations or shelters in disaster-hit regions [14, 15]. All those applications can be executed by applying our techniques in simulations to figure out a good allocation of facilities.

Our main contributions in this paper can be summarized as follows:

- We introduce a novel problem formulation that combines geometric and combinatorial aspects within the framework of multi-armed bandit problems.
- We propose the Geometric-UCB algorithm, a UCB-based algorithm designed to tackle the problem, and we provide a detailed mathematical analysis of this algorithm, which includes proving a regret upper bound.
- We demonstrate the efficiency and effectiveness of the proposed algorithm through numerical simulations, and we show that it outperforms other standard methods.

Table 1: Description of Commonly-Used Notation

Variable	Description
N	The total number of cells in the 1×1 grid.
K	The number of facility points to be allocated.
i	Index for cells.
j	Index for facility points.
t	Index for rounds.
$D(i)$	The underlying population density of cell i.
$\mathbf{F}(t)$	The allocation of facility points at round t.
$f_k(t)$	The location of the k-th facility point at round t.
$V_j(t)$	The set of cells in the Voronoi area of facility j at round t.
$p_{i,j}(t)$	The probability of attracting a person from cell i to facility point j at round t.
α	a tunable factor that equals the probability of attracting a person in a cell to a facility allocated at that cell.
$X_i(t)$	The random number of people attracted from cell i at round t.
$\mu(\mathbf{F}, t)$	The expected attracted population to the set of facility locations $\mathbf{F}$ at round t.
$\hat{\mu}(\mathbf{F}, t)$	The empirical mean attracted population to the set of facility locations $\mathbf{F}$ at round t.
$R(t)$	The total expected regret at round t.

The remainder of the paper is organized as follows. In Section 2, related work is presented. Section 3 shows the formal presentation of the problem. Section 4 shows the efficient algorithm for solving the problem. Section 5 gives a detailed analysis of the regret bound. In Section 6, simulation results are presented to evaluate the performance of our solution on both synthesized data and processed real-world traces. Finally, Section 7 gives the conclusion.

2 RELATED WORK

Facility allocation problems have remained a prominent subject within the field of operations research for a long time, due to their relevance across a wide array of application domains. The fundamental objective of facility location problems is to strategically position facilities in a way that minimizes a designated cost function. This cost function typically encompasses various factors, including the distances between facilities and clients, as well as the fixed costs associated with establishing facilities [16, 17]. In relation to our particular problem, the geometric element involved a connection to the concept of Voronoi diagrams, which are applied in diverse domains such as computer graphics and computational geometry [18, 19].

The novel aspect of our work is the stochasticity and the geometric nature of the problem, requiring a combination of CMAB strategies and spatial analysis. While there have been studies on stochastic bandit problems with side information or context [20, 21], our setting is different in that the side information (i.e., Voronoi diagrams) is encoded in a geometric manner and the action space is combinatorial and applies to application in 2-dimensional spaces. The closest work to ours is perhaps the work on continuous bandit problems, where the action space is a continuous domain [22, 23]. However, most of these works focus on 1-dimensional action space and do not consider the combinatorial nature of the problem.

The application of MAB techniques in spatially sensitive settings has become increasingly important in recent years. One of the primary applications is in the area of crowdsensing, where sensors or users are utilized to gather and share data related to their physical environment [24–26]. In this context, the MAB problem is used to efficiently allocate resources (e.g., sensors or users) to different locations or tasks with the aim of maximizing some form of aggregate utility (e.g., total data collected or accuracy of collected data). This involves both an exploration-exploitation trade-off (since the quality or usefulness of data may not be known in advance) and a spatial aspect (since the location of sensors or users is crucial). Our work shares similarities with this line of research in that we also consider a spatially sensitive MAB problem. However, we focus on facility allocation, where the goal is to optimally allocate facilities to meet stochastic demand.

Other work that has employed MAB in settings with spatial considerations includes applications in wireless communication networks, spatial advertising, and autonomous vehicle routing [26–28]. In these settings, the spatial coordinates of the objects of interest (e.g., wireless devices, advertisements, vehicles) are integral to the problem. For example, in autonomous vehicle routing, an MAB problem may be used to decide which routes to explore and exploit, given uncertainty over traffic conditions or travel times. While these works and ours employ MAB techniques in spatially sensitive settings, the specific challenges and assumptions differ. For instance, our work considers a 2-dimensional space and fixed population densities, which add unique complexities to the problem. Therefore, while we build upon the ideas presented in these bodies of work, our approach and analysis are tailored specifically to the geometric version of the stochastic facility allocation problem.

Finally, we note that our problem has a certain resemblance with the work on cooperative game theory and power diagrams, which are generalizations of Voronoi diagrams, where the goal is to divide a territory among several players in a way that optimizes certain objective function [29]. However, our work differs in that we introduce a bandit setting, where the density function is unknown and needs to be learned over time.

3 PROBLEM FORMULATION

Consider a 1×1 square with a discrete grid of N cells, where N is a perfect square. For each cell i, we denote the underlying population density (which is unknown but fixed) as $D(i)$. Thus, $D(i)$ represents the estimated number of individuals per cell. We allocate K facility points at the center of the cells at each round t, and we denote these allocations as $\mathbf{F}(t) = [f_1(t), ..., f_K(t)]$, where $f_k(t)$ is the location of the k-th facility point at round t. The value of K is assumed to be small and constant. We note that no two facility points can occupy the same location at the same round. The goal is to choose $\mathbf{F}(t)$ at each round such that we maximize the expected total attracted population over rounds.

In each round, a Voronoi partition is performed based on the current facility point locations. Let $V_j(t)$ denote the set of cells that are closer to facility point j than any other facility point based either on the Manhattan distance or Euclidean distance, where ties are broken by randomly assigning the cell to one of the facility points that are the closest to it. We assume that each person in a cell is attracted to the nearest facility point with a probability that is inversely proportional to the distance to the facility point. Specifically, we define $p_{i,j}(t)$, the probability of attracting a person from cell i to facility point j, as:

$$p_{i,j}(t) = \begin{cases} \frac{\alpha}{d(f_j(t),i)+1} & \text{if } i \in V_j(t) \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where α is a tunable factor that equals the probability of attracting a person in a cell to a facility allocated at that cell. $\alpha \leq 1$. Additionally, $d(f_j(t), i)$ is the distance between $f_j(t)$ and i based on the chosen distance metric. We denote the random number of people attracted from cell i to its corresponding facility j at round t as $X_i(t) \sim \text{Binomial}(D(i), p_{i,j}(t))$. Therefore, the expected attracted population from cell i at round t is $\mathbb{E}[X_i(t)] = \sum_{j \leq K} D(i) p_{i,j}(t)$.

To make this a combinatorial multi-armed bandit problem, we consider that we do not know the population densities $D(i)$, but we can observe the number of people attracted to each facility point in each round. Based on these observations, we will need to develop a strategy to choose the facility locations $\mathbf{F}(t)$ in each round in order to maximize the total attracted population over time.

The regret at round t is defined as the difference between the total expected attracted population if we could always place the facilities optimally (knowing all $D(i)$), and the total expected attracted population with our strategy:

$$R(t) = \Sigma_{s=1}^{t} \left(\max_{\mathbf{F}} \Sigma_{i=1}^{N} \mathbb{E}[X_i(s)] \right) - \Sigma_{s=1}^{t} \Sigma_{i=1}^{N} \mathbb{E}[X_i(s)]. \tag{2}$$

The objective is to come up with an algorithm that chooses $\mathbf{F}(t)$ every round such that the overall regret shown in Equation 2 is minimized. Table 1 shows a description of the commonly-used notation.

4 SOLUTION OF THE PROBLEM

Given the nature of the problem, a UCB-based algorithm would be an appropriate choice for the allocation strategy, as the UCB algorithm balances exploration (gaining more information about the population densities) and exploitation (placing the facilities at the locations that have demonstrated high attraction in the past).

4.1 Algorithm Overview

We will build our solution on a variant of the UCB algorithm known as the Combinatorial Upper Confidence Bound (C-UCB) algorithm [30–32]. C-UCB is a generalization of UCB that is suitable for combinatorial multi-armed bandit problems. However, a drawback of the algorithm entails enumerating all the possible combinations of arms. Luckily for us, the majority of applications of the geometric version shown in our problem have a very small value of K, which makes it reasonably tractable to use such an algorithm.

Algorithm 1 Geometric-UCB for facility allocation

Input: $D(i)\ \forall i \le N, K$, distance metric.
Output: $\mathbf{F}(t) \quad \forall t = 1, 2, \ldots, T$.
Initialization: $X_i(0) \leftarrow 0 \forall i \le N$, $\hat{\mu}(\mathbf{F}, t) \leftarrow 0$ and $N_{\mathbf{F}}(t) \leftarrow 0 \forall \mathbf{F}$
for $t = 1, 2, \ldots, T$ **do**
  for all possible allocations $\mathbf{F}$ **do**
    Evaluate $UCB(\mathbf{F}, t)$ from Equation 4.
  Choose $\mathbf{F}(t)$ based on Equation 5.
  Perform the Voronoi partition based on $\mathbf{F}(t)$ and the chosen distance metric to get $V_j(t)$ for all $j \le K$.
  Observe $X_i(t) \quad \forall i \le N$ and update $\hat{\mu}(\mathbf{F}, t)$ and $N_{\mathbf{F}}(t) \quad \forall \mathbf{F}$ based on Equations 6-7.
return $\mathbf{F}(1), \mathbf{F}(2), \ldots, \mathbf{F}(T)$.

First, let's denote the total expected attracted population at a given set of facility locations $\mathbf{F}$ at round t as:

$$\mu(\mathbf{F}, t) = \Sigma_{j=1}^{K} \Sigma_{i \in V_j(t)} D(i) p_{i,j}(t). \tag{3}$$

At the start of each round, the C-UCB algorithm has an upper confidence bound for $\mu(\mathbf{F}, t)$ for each possible set of facility locations $\mathbf{F}$. This upper confidence bound is a sum of an estimate of the total expected attracted population based on the observations so far, and an exploration bonus that is high for the sets of facility locations that have been tried the least. Specifically, we define the upper confidence bound as:

$$UCB(\mathbf{F}, t) = \hat{\mu}(\mathbf{F}, t) + \sqrt{2 \log t / N_{\mathbf{F}}(t)}, \tag{4}$$

where $\hat{\mu}(\mathbf{F}, t)$ is the estimated total expected attracted population for the set of facility locations $\mathbf{F}$ at round t based on the observations so far, and $N_{\mathbf{F}}(t)$ is the number of times the set of facility locations $\mathbf{F}$ has been tried up until round t.

4.2 Phases of the Problem

The Geometric-UCB algorithm chooses the set of facility locations $\mathbf{F}$ that has the highest upper confidence bound:

$$\mathbf{F}(t) = \arg\max_{\mathbf{F}} \left(\hat{\mu}(\mathbf{F}, t) + \sqrt{2 \log t / N_{\mathbf{F}}(t)} \right). \tag{5}$$

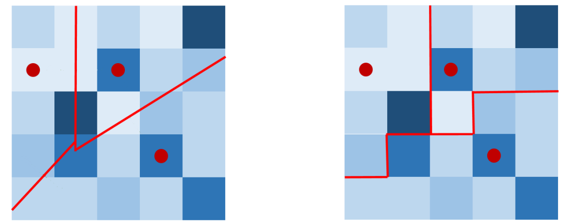
The estimates $\hat{\mu}(\mathbf{F}, t)$ can be updated based on the observed attracted population at round t using the formula:

$$\hat{\mu}(\mathbf{F}, t+1) = \frac{N_{\mathbf{F}}(t)\hat{\mu}(\mathbf{F}, t) + \sum_{j=1}^{K} \sum_{i \in V_j(t)} X_i(t)}{N_{\mathbf{F}}(t) + 1}. \tag{6}$$

The number of times each set of facility locations has been tried, $N_{\mathbf{F}}(t)$, is also updated after each round:

$$N_{\mathbf{F}}(t+1) = \begin{cases} N_{\mathbf{F}}(t) + 1 & \text{if } \mathbf{F}(t) = \mathbf{F} \\ N_{\mathbf{F}}(t) & \text{otherwise} \end{cases}. \tag{7}$$

This completes the description of the C-UCB-based allocation policy. The key idea behind this strategy is to balance exploitation (choosing the set of facility locations that have yielded large attracted population in the past) and exploration (trying out new sets of facility locations or ones that have not been tried for a while). This balance is achieved through the upper confidence bound, which encourages the algorithm to try out different sets of facility locations, especially in the early rounds of the iterations.



(a) (b)

Figure 1: An illustration showing the effect of the choice of distance metric on the Voronoi partition $V_j(t)\ \forall j \le K$. The background color represents the value of the underlying population density of the cells $D(i)$: (a) Euclidean distance metric; (b) Manhattan distance metric.

4.3 Algorithm Design

We present the strategy used in the pseudocode shown in Algorithm 1. The algorithm operates in discrete rounds, iteratively deciding the optimal allocation of facilities in a geometric 2-dimensional space. The inputs of this algorithm are the population densities of the cells (denoted as $D(i)$ for all $i \le N$), the number of facilities K (which is assumed to be a small number), and the distance metric used (whether it is Manhattan distance or Euclidean distance). The output of the algorithm is the allocation of facilities at each round, denoted as $\mathbf{F}(t)$ for all $t = 1, 2, \ldots, T$.

In the initialization step, the observation count of the attracted population for each cell, $X_i(0)$, is set to 0 for all $i \le N$. The algorithm then enters a loop, iterating over each round from 1 to T. For each round t, it evaluates all possible allocations of the facilities. The C-UCB for each potential allocation, given the current round, is calculated as per Equation 4. The evaluation of UCB balances the exploitation of current best-performing allocations and the exploration of the other allocations.

Once all allocations' C-UCB values are evaluated, the algorithm selects the allocation for the current round, $\mathbf{F}(t)$, based on Equation 5. This equation ensures that the allocation with the highest C-UCB is selected. After deciding on the allocation, the algorithm performs a Voronoi partition based on the selected $\mathbf{F}(t)$ and the given distance metric, whether it is Euclidean or Manhattan distance metric. This step results in the division of the 2-dimensional space into K regions, $V_j(t)$ for all $j \le K$, where each region is associated with a facility. The way the distance metric might affect the Voronoi partition is shown in Figure 1.

Next, the algorithm observes the new counts of individuals attracted to each potential allocation, $X_i(t)$ for all $i \le N$. It then updates the estimated mean attraction of each allocation, $\hat{\mu}(\mathbf{F}, t)$, as well as the count of selections of each allocation, $N_{\mathbf{F}}(t)$, as per Equations 6 and 7, respectively. This loop continues for the specified number of rounds T, with the algorithm dynamically adjusting the facility allocations to maximize the total attraction over the entire period based on the UCB principle. The algorithm terminates after iterating over the T rounds and returning the set of allocations $\mathbf{F}(1), \mathbf{F}(2), \ldots, \mathbf{F}(T)$. This algorithm guarantees a sublinear regret, which is derived and shown in the next section.

4.4 Algorithm Complexity

The complexity of Algorithm 1 stems from two main sources, which are the evaluation of all possible allocations and the computation of the Voronoi partition for the chosen allocation in each round. For each round, the algorithm evaluates the UCB of each potential allocation. The number of possible allocations is given by $\binom{N}{K}$ (which equals $O(N^K)$), where N is the number of cells and K is the number of facilities. Since the UCB for each allocation is computed in each round, this part of the algorithm has a time complexity of $O(T \times N^K)$, where T is the total number of rounds.

The Voronoi partitioning process is the second computationally intensive part of the algorithm. Voronoi partitioning, in the worst-case scenario, has a computational complexity of $O(K^2)$ in two dimensions. This operation has to be carried out at each round, which results in a total complexity of $O(T \times K^2)$ over T rounds.

Moreover, observing the counts of individuals attracted to each potential allocation and updating the estimated mean attractions and the counts of selections has a computational complexity of $O(N)$ at each round. Over T rounds, this part of the algorithm has a time complexity of $O(T \times N)$. Therefore, the overall time complexity of the Geometric-UCB for the facility allocation algorithm is $O(T \times (N^K + N + K^2))$.

In terms of space complexity, the algorithm maintains counts and estimates for each allocation, implying a space complexity of $O(N^K)$. The Voronoi partitioning also requires space to store the boundaries and facilities associated with each Voronoi cell. In the worst-case scenario, each Voronoi cell can have up to $K - 1$ edges, resulting in a space complexity of $O(K^2)$. Therefore, the overall space complexity of the algorithm is $O(N^K + K^2)$.

Lastly, it is important to note that, in practice, the value of K is presumed to be small and constant. This means that the algorithm would require polynomial time and space complexities with respect to the number of cells N and the number of rounds T.

5 REGRET BOUND ANALYSIS

The key performance metric in a multi-armed bandit problem is total regret. In this section, we show the analysis of the regret bound achieved by Algorithm 1.

Theorem 5.1. *Algorithm 1 guarantees a regret bound of:*

$$R(T) \leq 2\sqrt{2N \log T}\left(1 + 1/\sqrt{N}\right).$$

Proof. The novel proof is shown in detail in the appendix. □

Regarding the selection of the α parameter in our attraction model, the choice of α does not affect the theoretical regret bound. However, a larger α encourages more exploration, as the facilities are more likely to attract people from further cells, thus providing more information about the environment. A smaller α encourages more exploitation, as the facilities are more likely to attract people from closer cells, thus maximizing the immediate reward.

Based on our regret analysis, we might consider a heuristic approach of choosing α such that it decreases over time, perhaps at a rate of $1/\sqrt{t}$ or slower, to balance exploration and exploitation. This is consistent with the intuition that we should explore more in the early rounds when we know less about the environment, and exploit more in the later rounds when we have more information.

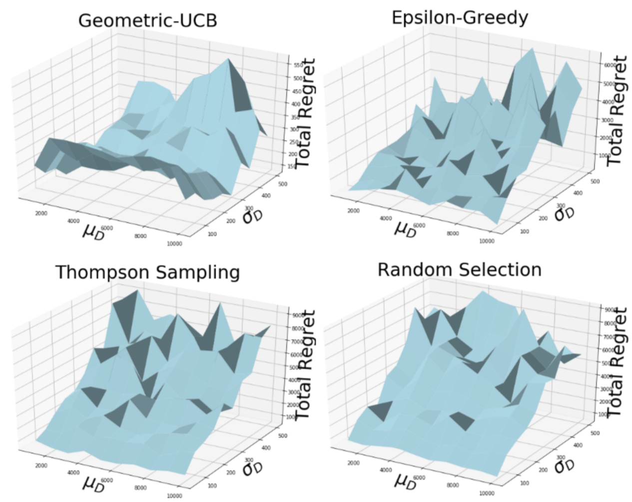


Figure 2: Total regret value for different algorithms under synthesized data. $K = 3, N = 36, \alpha = 0.5$, **and the metric used is the Euclidean metric.**

However, the optimal choice of α may depend on the specific characteristics of the environment, such as the distribution of the population densities $D(i)$ and the distance metric used. Future work could involve developing more sophisticated strategies for choosing α, possibly involving adaptive methods that adjust α based on the observed rewards.

6 EXPERIMENTAL EVALUATION

In this section, we conduct comprehensive simulations to evaluate the performance of our algorithm compared to other standard algorithms under various settings.

6.1 Experimental Settings

We start by showing the specifics of our experiment designed to evaluate our novel geometric CMAB algorithm. The experiments were designed to incorporate a variety of settings and parameters, thereby ensuring a comprehensive assessment of the algorithm's performance.

Parameter settings: We assign the number of facilities, K, to be either 3 or 4 across different experimental setups. These low values are necessary to ensure the tractability of our algorithm as it has an $O(N^K)$ term in its time complexity. Although these values are small, they serve to mimic practical scenarios ranging from relatively simple (3 facilities) to moderately complex (4 facilities) arrangements. In our model, $p_{i,j}(t)$, which is the probability of attracting a person from cell i to facility point j at round t, is governed by the parameter α. To explore the impact of this factor on our algorithm's performance, we have ranged α to have values from 0.1 to 1.0. The value of T is set to 10^6 by default. Further, to examine the algorithm's adaptability to various distance metrics, we have incorporated both Manhattan and Euclidean distance metrics in our experimental settings.

Real-world traces-based simulation: For a pragmatic evaluation, we turn to real-world data corresponding to the population of the United States. We process this data by discretizing it into cells

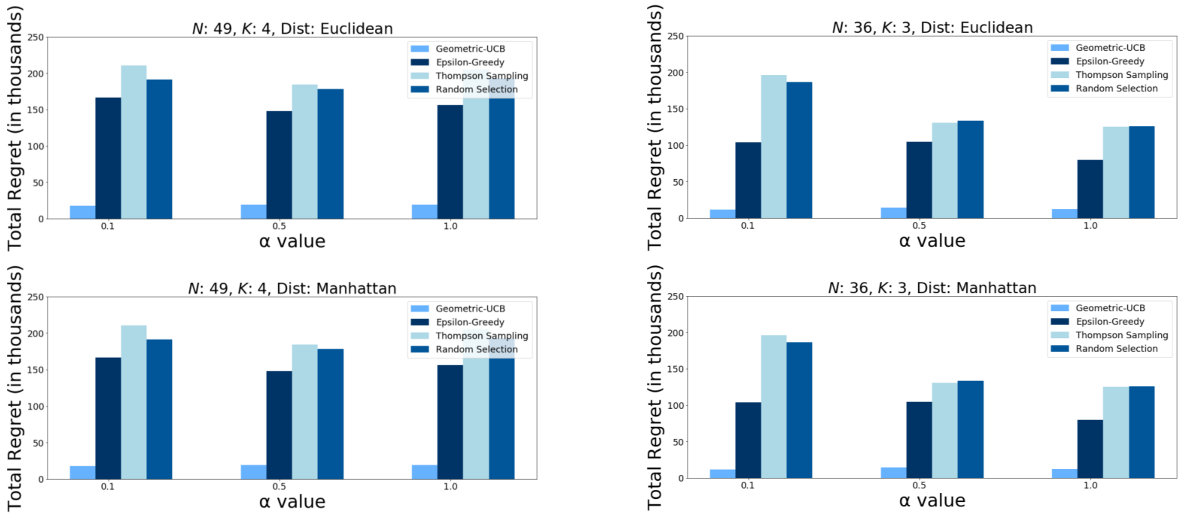


Figure 3: Total regret value for different algorithms under synthesized data with different α values. $\mu_D = 5000, \sigma_D = 100$.

represented as points in a 1×1 grid. Each cell, i, houses a specific population density, $D(i)$. While mapping the real-world data onto the grid, we ensure the preservation of the total number of cells N to be either 36 or 49 cells. The attraction of the population to a facility in a cell is dictated by the population density of the cell. This setting, being grounded in reality, provides a near-accurate benchmark for our algorithm.

Synthesized data: Complementing the real-world simulation, we also test our algorithm on synthesized data. Various datasets are generated with diverse numbers of cells, N, and the population density of each cell, $D(i)$, is assigned from a normal distribution with a varying mean μ_D and variance σ_D^2. This is it, $D(i) \sim \mathcal{N}(\mu_D, \sigma_D^2)$. This approach facilitates testing our algorithm's performance across a wide range of scenarios and its resilience against alterations in population density distribution. These varied experimental settings are designed to demonstrate the proposed algorithm's adaptability and robustness under different scenarios and parameters.

6.2 Algorithm Comparison

Now, we discuss the comparative evaluation of our proposed algorithm, the Geometric-UCB method for facility allocation, against other notable existing algorithms in the literature. The aim of such a comparison is to ascertain the relative performance of our algorithm and to investigate its specific strengths and potential areas of improvement.

To conduct a meaningful comparison, we have selected algorithms that use different methods or heuristics while sharing the same broad goal of optimal facility allocation. Such a selection provides a diversified benchmark, helping us gauge the effectiveness of our Geometric-UCB in a variety of settings. The details of the chosen algorithms are as follows:

Epsilon-Greedy Algorithm [33]: This is a popular choice in the multi-armed bandit problem literature. The ϵ-greedy algorithm strikes a balance between exploration and exploitation by taking the best action most of the time, but with a small probability, ϵ, it randomly explores other options. This comparison allows us to assess the effectiveness of our UCB-based strategy against a well-established alternative. The value of ϵ is set to 0.25.

Thompson Sampling [34]: Thompson Sampling is another well-known strategy in multi-armed bandit problems. It is a probabilistic method that uses Bayesian inference and principles of uncertainty to balance exploration and exploitation. The comparison with Thompson Sampling will provide insight into how our Geometric-UCB performs against probabilistic and uncertainty-driven approaches.

Random Selection: A comparison with a naive strategy, such as random facility allocation, provides a baseline performance level. This method doesn't take into account any prior or current round information and chooses the facility allocation purely randomly in each round.

The comparative evaluation against these varied algorithms is done over both real-world trace-based data and synthesized datasets, as well as under different parameter settings for the datasets.

6.3 Experimental Results

The results of our experiments provide valuable insights into the performance of our Geometric-UCB algorithm and other comparable techniques. Our detailed analysis demonstrates that the geometric formulation of the problem, along with the use of the UCB principle, yields a lower regret than traditional methods, affirming the effectiveness of our approach.

As previously noted, the results shown in Figure 2 indicate the performance of different algorithms applied to synthesized data where $K = 3, N = 36, \alpha = 0.5$, and where the metric used is the Euclidean metric. This figure shows exactly how the performance of the different algorithms differs depending on the underlying population density. The two factors that govern the values of $D(i)$, in this case, are μ_D and σ_D, which are varied to show the regret for different combinations of values. The ranges of the domain are $1000 \le \mu_D \le 10000$, and $50 \le \sigma_D \le 500$. We can observe from the figure that the performance of our Geometric-UCB algorithm is fairly resistant to the change of the value of σ_D, unlike the other

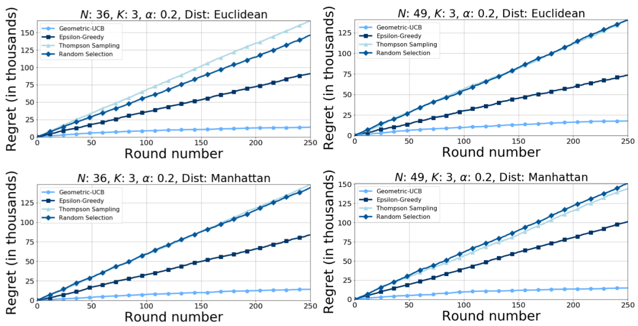


Figure 4: Regret value for different algorithms under synthesized data with different N values. $\mu_D = 5000, \sigma_D = 100$.

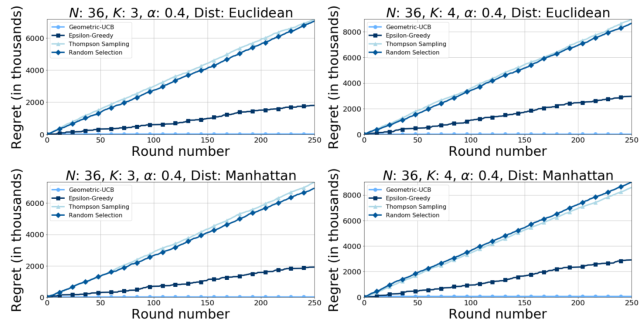


Figure 5: Regret value for different algorithms under real-world traces with different K values.

three algorithms that appear to suffer from higher regret linearly given higher variance in the underlying distribution of population. Moreover, we can notice how the value of μ_D doesn't significantly affect the average total regret value for all algorithms except for the Epsilon-Greedy algorithm, which appears to be sensitive for the value of μ_D given relatively high values of the variance σ_D.

In Figure 3, we examine the influence of different α values under the same distance metric. We can observe, when $N = 49$ and $K = 4$, that as α increases, the regret rate of the Random Selection algorithm remains roughly unchanged, demonstrating its insensitivity to the changes in α with higher values of N and K. On the other hand, the performance of the ϵ-greedy, Thompson Sampling, and our Geometric-UCB algorithms appear to improve with lower α values on average, suggesting that a higher α increases the expected total reward of the optimal allocation, making the regret generally higher for those algorithms. Shifting to the case where $N = 36$ and $K = 3$ in Figure 3, we find that the regret value of our Geometric-UCB algorithm remains roughly the same as the value of α changes. However, for the other three algorithms, the regret first increases as α increases before peaking at a middle α value, then it decreases with a higher α. That is likely due to the relatively small values of N and K, which makes the gap between the total reward from the optimal facility allocation and the reward from the algorithm the lowest with the extreme values of α.

In Figures 4 and 5, a detailed illustration of the regret with the number of rounds is presented, showing how the accumulated regret grows with the rounds of the problem for the different algorithms. When the algorithms are applied under the synthesized data, given that $\mu_D = 5000$ and $\sigma_D = 100$, the regret grows with the number of rounds as expected. We can notice the large gap in performance between our proposed Geometric-UCB algorithm and the other two. From Figure 4, we see that our Geometric-UCB algorithm grows logarithmically as the number of rounds increases, verifying the theoretical result we obtained earlier. Figure 5 shows the performance of the algorithms applied to the real-world trace-based data. We can notice that under this data, the performance gap between our Geometric-UCB and the second best algorithm, which is the ϵ-greedy algorithm, becomes relatively smaller compared to the other two algorithms. Lastly, it's worth mentioning that it appears that the Thompson Sampling algorithm is relatively bad if we apply it to a 2-dimensional problem like ours, performing close to the Random Selection algorithm in most scenarios.

6.4 Simulation Summary

Our comprehensive simulations showed the comparative performance of our proposed Geometric-UCB algorithm vis-à-vis standard methods like the Random Selection, ϵ-greedy, and Thompson Sampling algorithms. We considered a range of settings that included synthesized data using both Manhattan and Euclidean distance metrics, as well as more complex, processed real-world traces based on the distribution of the population in the US. The Geometric-UCB algorithm displayed a trend of superior performance across all scenarios. It achieved a minimum of 25-40% reduction in regret relative to the best of the other learning-based methods, underscoring its proficiency in addressing the geometric problem using a CMAB framework. The results we got from the simulation align with the theoretical results we obtained earlier in this work when we analyzed our regret bound.

7 CONCLUSION

In this paper, we proposed a novel approach for solving a stochastic facility allocation problem with spatially variant population densities. Drawing on techniques from the Combinatorial Multi-Armed Bandit (CMAB) framework, our method strategically places facilities across the grid to maximize the expected attraction of the population to these facilities over multiple rounds. To the best of our knowledge, this is the first work that uses a MAB paradigm to address such a geometric problem in 2-dimensional settings. Our approach leverages an Upper Confidence Bound (UCB) strategy and utilizes Voronoi diagrams to model the population's attraction to facilities. We accommodate both Manhattan and Euclidean distance measures. We provided a theoretical analysis and derived regret bounds to prove the algorithm's efficiency. Our numerical simulations, carried out on both synthesized data and real-world traces, demonstrated the effectiveness of our method, with an improvement of at least 25-40% over other learning-based methods. Future research would focus on expanding our framework to higher-dimensional spaces or exploring other performance measures beyond regret. The robustness of our Geometric-UCB algorithm lays a strong foundation for these directions.

ACKNOWLEDGMENTS

This research was supported in part by NSF grants SaTC 2310298, CNS 2214940, CPS 2128378, CNS 2107014, and CNS 2150152.

A PROOF OF THEOREM 4.1.

A more simplified way to write the regret of the Geometric-UCB algorithm up to round t is given by:

$$R(t) = \sum_{s=1}^{t} \left(\max_{\mathbf{F}} \mu(\mathbf{F}, s) - \mu(\mathbf{F}(s), s) \right), \tag{8}$$

where $\mu(\mathbf{F}, t)$ is the total expected attracted population at the set of facility locations $\mathbf{F}$ at round t, and $\mathbf{F}(s)$ is the set of facility locations chosen by the Geometric-UCB algorithm at round s.

To analyze the regret of the Geometric-UCB algorithm, we first note that, by the definition of the upper confidence bound, we have that, with high probability, for all $\mathbf{F}$ and s:

$$\mu(\mathbf{F}, s) \leq \hat{\mu}(\mathbf{F}, s) + \sqrt{\frac{2 \log s}{N_{\mathbf{F}}(s)}}. \tag{9}$$

Therefore, we can write the regret as:

$$R(t) \leq \sum_{s=1}^{t} \left(\max_{\mathbf{F}} \left(\hat{\mu}(\mathbf{F}, s) + \sqrt{\frac{2 \log s}{N_{\mathbf{F}}(s)}} \right) - \hat{\mu}(\mathbf{F}(s), s) \right). \tag{10}$$

This expression can be bounded from above by:

$$R(t) \leq \sum_{s=1}^{t} \sqrt{\frac{2 \log s}{N_{\mathbf{F}}(s)}} \leq \sqrt{2t \log t} \sum_{s=1}^{t} \frac{1}{\sqrt{N_{\mathbf{F}}(s)}}. \tag{11}$$

To further bound this expression, we will need to make mild assumptions about the rate at which $N_{\mathbf{F}}(s)$ grows for the sets of facility locations $\mathbf{F}$ that are not optimal, i.e., for which $\mu(\mathbf{F}, s) < \max_{\mathbf{F}'} \mu(\mathbf{F}', s)$ are sub-optimal sets of facility locations $\mathbf{F}$.

We will denote the gap between the reward of the optimal action and the reward of action $\mathbf{F}$ at round s as:

$$\Delta_{\mathbf{F},s} = \max_{\mathbf{F}'} \mu(\mathbf{F}', s) - \mu(\mathbf{F}, s). \tag{12}$$

To control the regret, we'll have to ensure that sub-optimal actions are not selected too frequently. To this end, we'll show that the number of times the Geometric-UCB algorithm selects a sub-optimal action $\mathbf{F}$ is bounded by a term that depends on the value of $\Delta_{\mathbf{F},s}$.

Let's denote the event that the Geometric-UCB algorithm selects action $\mathbf{F}$ at round s as $A_{\mathbf{F},s}$. Using the union bound, we get that for any $\mathbf{F}$ and s,

$$\begin{aligned} P(A_{\mathbf{F},s}) \leq\, & P\left(\mu(\mathbf{F}, s) > \hat{\mu}(\mathbf{F}, s) - \sqrt{\frac{2 \log s}{N_{\mathbf{F}}(s)}} \right) \\ & + P\left(\max_{\mathbf{F}'} \mu(\mathbf{F}', s) \leq \hat{\mu}(\mathbf{F}', s) + \sqrt{\frac{2 \log s}{N_{\mathbf{F}'}(s)}} \right). \end{aligned} \tag{13}$$

Using Hoeffding's inequality, we get that for any $\mathbf{F}$ and s,

$$P(A_{\mathbf{F},s}) \leq 2 \exp\left(-2 N_{\mathbf{F}}(s) \Delta_{\mathbf{F},s}^2 \right). \tag{14}$$

Then, summing over all rounds s from 1 to t, and using the union bound again, we get that the total number of times the Geometric-UCB algorithm selects a sub-optimal action $\mathbf{F}$ up to round t is bounded by

$$\sum_{s=1}^{t} P(A_{\mathbf{F},s}) \leq 2 \sum_{s=1}^{t} \exp\left(-2 N_{\mathbf{F}}(s) \Delta_{\mathbf{F},s}^2 \right). \tag{15}$$

Now, we sum over all sub-optimal actions $\mathbf{F}$. Let's define the set of all possible actions $\mathcal{A}$ to be the set of all possible locations of K facility points, i.e., $\mathcal{A} = \{\mathbf{F} : \mathbf{F} \in 1, \ldots, O(N^K)\}$, all f_k are distinct. Let $\mathbf{F}^*$ denote the optimal action at round s.

Then, we can write the total regret $R(t)$ as:

$$R(t) = \sum_{\mathbf{F} \in \mathcal{A}} \sum_{s=1}^{t} \Delta_{\mathbf{F},s} \cdot \mathbb{I}\{A_{\mathbf{F},s}\}, \tag{16}$$

where $\mathbb{I}\{.\}$ is the indication function such that $\mathbb{I}\{\text{TRUE}\} = 1$ and $\mathbb{I}\{\text{FALSE}\} = 0$.

By summing over all sub-optimal actions $\mathbf{F} \neq \mathbf{F}^*$, we get:

$$R(t) = \sum_{\mathbf{F} \neq \mathbf{F}^*} \sum_{s=1}^{t} \Delta_{\mathbf{F},s} \cdot \mathbb{I}\{A_{\mathbf{F},s}\} = \sum_{\mathbf{F} \neq \mathbf{F}^*} \sum_{s=1}^{t} \Delta_{\mathbf{F},s} \cdot P(A_{\mathbf{F},s}). \tag{17}$$

Using the bound we derived in Equation 15, we get:

$$R(t) \leq \sum_{\mathbf{F} \neq \mathbf{F}^*} \sum_{s=1}^{t} \Delta_{\mathbf{F},s} \cdot 2 \exp\left(-2 N_{\mathbf{F}}(s) \Delta_{\mathbf{F},s}^2 \right). \tag{18}$$

Moving forward, we now handle the exponential part in the inequality in Equation 18 using Bernstein's Inequality [35]. Bernstein's inequality provides an upper bound on the deviation of the sum of random variables from their expected value. It extends the Hoeffding's inequality by incorporating information about the variance of the random variable.

Before we apply Bernstein's inequality, we need to deal with the correlation between different actions, as mentioned before. Specifically, the number of people attracted from the same cell i can be correlated for different actions $\mathbf{F}$ and $\mathbf{F}'$, if i belongs to the Voronoi partitions of both actions.

To deal with this, we introduce a dependency graph over the actions $\mathcal{A}$. In this graph, two actions $\mathbf{F}$ and $\mathbf{F}'$ are connected by an edge if there exists a cell i such that $i \in V_j(t)$ for some $j \leq K$, for both $\mathbf{F}$ and $\mathbf{F}'$. This means that the same cell i can be attracted by different facilities in both actions.

Let $\mathcal{G}$ denote this dependency graph. Each connected component of $\mathcal{G}$ is a set of actions that can potentially influence each other. We can handle each connected component separately and then sum up the results.

Suppose we are looking at a particular connected component C. Let $|C|$ denote the number of actions in C, and $\mathbf{F}^1, \ldots, \mathbf{F}^{|C|}$ denote the actions in C.

Consider the random variables $Y_s = \sum_{\mathbf{F} \in C} \mathbb{I}\{A_{\mathbf{F},s}\}$, for $s = 1, \ldots, t$. We have $E[Y_s] = \sum_{\mathbf{F} \in C} P(A_{\mathbf{F},s})$ and the variace bounded as $\text{Var}(Y_s) \leq E[Y_s]$.

Applying Bernstein's inequality to $Y_1, \ldots, Y_t$, we get that for any $\delta > 0$:

$$P\left(\sum_{s=1}^{t} Y_s - E[\sum_{s=1}^{t} Y_s] > \delta \right) \leq \exp\left(-\frac{\delta^2 / 2}{\sum_{s=1}^{t} E[Y_s] + \delta/3} \right). \tag{19}$$

Setting $\delta = \sqrt{3 \sum_{s=1}^{t} E[Y_s] \log t} + \frac{\log t}{3}$, we get:

$$P\left(\sum_{s=1}^{t} Y_s - E[\sum_{s=1}^{t} Y_s] > \sqrt{3 \sum_{s=1}^{t} E[Y_s] \log t} + \frac{\log t}{3} \right) \leq \frac{1}{t^2}. \tag{20}$$

Summing over all connected components of $\mathcal{G}$ and using the union bound, we get that with a high probability (at least $1 - \frac{|\mathcal{G}|}{t^2}$), we have:

$$\sum_{s=1}^{t} Y_s - E[\sum_{s=1}^{t} Y_s] \leq \sqrt{3\sum_{s=1}^{t} E[Y_s]\log t + \frac{|\mathcal{G}|\log t}{3}} \quad \forall C \subseteq \mathcal{G}. \quad (21)$$

Now, we substitute Y_s back in terms of $\mathbb{I}\{A_{\mathbf{F},s}\}$ and apply it to the regret bound. Firstly, we define $Z_s = \max_{\mathbf{F}} \sum_{i=1}^{N} \mathbb{E}[X_i(s)]$ as the maximum expected attracted population at round s.

Recall the definition of $Y_s = \sum_{\mathbf{F}\in C} \mathbb{I}\{A_{\mathbf{F},s}\}$, meaning that Y_s is the number of times any action in the connected component C is played at round s. Recall also that $\mathbb{E}[X_i(s)] = \sum_{j\leq K} D(i)p_{i,j}(s)$ for each cell i. We can express Z_s in terms of Y_s and $\mathbb{I}\{A_{\mathbf{F},s}\}$ as follows:

$$\begin{aligned} Z_s &= \sum_{\mathbf{F}\in C} \mathbb{E}[X_{\mathbf{F},s}|A_{\mathbf{F},s}] \cdot \mathbb{I}\{A_{\mathbf{F},s}\} \\ &\quad + \sum_{\mathbf{F}\notin C} \mathbb{E}[X_{\mathbf{F},s}|A_{\mathbf{F},s}] \cdot \mathbb{I}\{A_{\mathbf{F},s}\} \\ &= \sum_{i=1}^{N} \mathbb{E}[X_i(s)|Y_s] \cdot Y_s. \end{aligned} \quad (22)$$

With this expression, we can write the regret $R(t)$ in terms of Z_s and Y_s:

$$R(t) = \sum_{s=1}^{t} \left(Z_s - \sum_{i=1}^{N} \mathbb{E}[X_i(s)] \right). \quad (23)$$

Hence,

$$R(t) = \sum_{s=1}^{t} \left(\sum_{i=1}^{N} \mathbb{E}[X_i(s)|Y_s] \cdot Y_s - \sum_{i=1}^{N} \mathbb{E}[X_i(s)] \right), \quad (24)$$

where $\mathbb{E}[X_i(s)|Y_s]$ is the expected attracted population in cell i at round s given that Y_s actions in C were played, and $\mathbb{E}[X_i(s)]$ is the expected attracted population in cell i at round s with our strategy.

Now, we get from Equation 21:

$$\mathbb{E}[X_{\mathbf{F},s}|A_{\mathbf{F},s}] - \mathbb{E}[X_{\mathbf{F},s}] \leq 2\sqrt{\frac{2\log t}{Y_s}}. \quad (25)$$

Substituting the inequality in Equation 25 into the regret bound in Equation 24, we get:

$$R(t) \leq \sum_{s=1}^{t} \sum_{i=1}^{N} \left(2\sqrt{\frac{2\log t}{Y_s}} \right) Y_s - \sum_{s=1}^{t} \sum_{i=1}^{N} \mathbb{E}[X_i(s)], \quad (26)$$

which simplifies to:

$$R(t) \leq 2\sqrt{2\log t} \sum_{s=1}^{t} \sum_{i=1}^{N} Y_s - \sum_{s=1}^{t} \sum_{i=1}^{N} \mathbb{E}[X_i(s)]. \quad (27)$$

Here, $\sum_{s=1}^{t} \sum_{i=1}^{N} Y_s$ represents the sum of the total number of times any action in C was played up to round t for all cells, and $\sum_{s=1}^{t} \sum_{i=1}^{N} \mathbb{E}[X_i(s)]$ is the sum of the expected attracted population up to round t for all cells. We can see that regret is bounded by a term proportional to the square root of the logarithm of the round index. We can further refine this regret bound in the next step by analyzing the behavior of $\sum_{s=1}^{t} \sum_{i=1}^{N} Y_s$, which we need to control to get a tighter bound on the regret.

A property of the UCB algorithm is that it balances exploration and exploitation. When our uncertainty about the population densities $D(i)$ is high (in the early rounds), the algorithm will do more exploration and choose a diverse set of facility point locations, leading to a high sum $\sum_{i=1}^{N} Y_s$ for these rounds. However, as we get more information and our estimates of the population densities improve, we will do more exploitation and concentrate the facility points in the locations that have demonstrated high attraction, leading to a lower sum $\sum_{i=1}^{N} Y_s$ in the later rounds.

Therefore, we have:

$$Y_s \leq D(i) + \sqrt{\frac{2\log t}{n_{i,t}}} \quad \forall i \leq N, \quad (28)$$

where $n_{i,t}$ is the number of times cell i has been selected up to round t in the connected component C. By summing over all cells and rounds, we obtain:

$$\sum_{s=1}^{t} \sum_{i=1}^{N} Y_s \leq \sum_{s=1}^{t} \sum_{i=1}^{N} \left(D(i) + \sqrt{\frac{2\log t}{n_{i,t}}} \right). \quad (29)$$

Now, note that $\sum_{s=1}^{t} \sum_{i=1}^{N} n_{i,t}$ is the total number of times all cells have been selected up to round t in the connected component of actions C, which is simply t. Hence, by Jensen's inequality [36], we get:

$$\sum_{s=1}^{t} \sum_{i=1}^{N} \sqrt{\frac{2\log t}{n_{i,t}}} \leq \sqrt{\frac{2N\log t}{t}}. \quad (30)$$

Substituting back we get:

$$\sum_{s=1}^{t} \sum_{i=1}^{N} Y_s \leq NtD_{\max} + \sqrt{2Nt\log t}, \quad (31)$$

where $D_{\max} = \max_i D(i)$ is the maximum population density over all cells. Now, we apply the upper bounds we derived in the $R(t)$, we get:

$$\begin{aligned} R(t) &\leq \sum_{s=1}^{t} \left(\sum_{i=1}^{N} D(i) + \sqrt{2N\log t} \right) \\ &\quad - \sum_{s=1}^{t} \sum_{i=1}^{N} \left(D(i) - \sqrt{\frac{2\log t}{n_{i,t}}} \right). \end{aligned} \quad (32)$$

This simplifies to:

$$R(t) \leq \sum_{s=1}^{t} \sum_{i=1}^{N} \sqrt{2N\log t} + \sqrt{\frac{2\log t}{n_{i,t}}}. \quad (33)$$

However, we can simplify this bound further by noting that $\sum_{i=1}^{N} n_{i,t} = t$ in our model, which gives:

$$R(t) \leq 2\sqrt{2N\log t} \left(1 + \frac{1}{\sqrt{N}} \right). \quad (34)$$

This bound suggests that the regret of the UCB algorithm for the geometric facility allocation problem grows at a rate of $\sqrt{t}$, which is typical for bandit problems. The dependency on N indicates that the regret becomes larger as the problem becomes more complex (i.e., when there are more cells to choose from).

REFERENCES

[1] H. J. Miller. Gis and geometric representation in facility location problems. *International Journal of Geographical Information Systems*, 10(7):791–816, 1996.

[2] A. Sawwan and J. Wu. A new framework: Short-term and long-term returns in stochastic multi-armed bandit. In *42th IEEE International Conference on Computer Communications (IEEE INFOCOM 2023)*, May 2023.

[3] Mahendra Bhatu Gawali and Subhash K Shinde. Task scheduling and resource allocation in cloud computing using a heuristic approach. *Journal of Cloud Computing*, 7(1):1–16, 2018.

[4] X. Li, Z. Zhao, X. Zhu, and T. Wyatt. Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research*, 74:281–310, 2011.

[5] S. A. Karulkar. *Optimal allocation of cell towers for cellular network expansion*. PhD thesis, Texas A&M University-Kingsville, 2016.

[6] B. Vos and H. Akkermans. Capturing the dynamics of facility allocation. *International Journal of Operations & Production Management*, 16(11):57–70, 1996.

[7] J. R. Beaumont. Location-allocation problems in a plane a review of some models. *Socio-economic planning sciences*, 1981.

[8] B. Boots, K. Sugihara, S. N. Chiu, and A. Okabe. *Spatial tessellations: concepts and applications of Voronoi diagrams*. 2009.

[9] M. Erwig. The graph voronoi diagram with applications. *Networks: An International Journal*, 36(3):156–163, 2000.

[10] E. B. Tirkolaee, I. Mahdavi, M. M. S. Esfahani, and G. W. Weber. A robust green location-allocation-inventory problem to design an urban waste management system under uncertainty. *Waste Management*, 102:340–350, 2020.

[11] M. D. Yang et al. A stochastic multi-objective optimization decision model for energy facility allocation: a case of liquefied petroleum gas station. *Clean Technologies and Environmental Policy*, 22:389–398, 2020.

[12] S. Wu, J. Yang, R. Peng, and Q. Zhai. Optimal design of facility allocation and maintenance strategy for a cellular network. *Reliability Engineering & System Safety*, 205:107253, 2021.

[13] A. Pal, A. Bhattacharya, and A. K. Chakraborty. Allocation of ev fast charging station with v2g facility in distribution network. In *2019 8th International conference on power systems (ICPS)*, pages 1–6, December 2019.

[14] C. Boonmee, M. Arimura, and T. Asada. Facility location optimization model for emergency humanitarian logistics. *International Journal of Disaster Risk Reduction*, 24:485–498, 2017.

[15] S. M. Shavarani. Multi-level facility location-allocation problem for post-disaster humanitarian relief distribution: a case study. *Journal of Humanitarian Logistics and Supply Chain Management*, 9(1):70–81, 2019.

[16] Z. Drezner. *Facility location: a survey of applications and methods*. Springer, New York, 1995.

[17] R. Z. Farahani, M. SteadieSeifi, and N. Asgari. Multiple criteria facility location problems: A survey. *Applied Mathematical Modelling*, 34(7):1689–1709, 2010.

[18] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[19] A. Fabri and S. Pion. Cgal: The computational geometry algorithms library. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 538–539, November 2009.

[20] D. Bouneffouf, I. Rish, and C. Aggarwal. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, July 2020.

[21] L. Zhou. A survey on contextual multi-armed bandits. *arXiv preprint*, arXiv:1508.03326, 2015.

[22] S. R. Chowdhury and A. Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, July 2017.

[23] X. Zhou and B. Ji. On kernelized multi-armed bandits with constraints. *arXiv preprint*, arXiv:2203.15589, 2022.

[24] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry. A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities. *IEEE Communications Surveys & Tutorials*, 21(3):2419–2465, 2019.

[25] Abdalaziz Sawwan and Jie Wu. Diversity-based recruitment in crowdsensing by combinatorial multi-armed bandits. *Tsinghua Science and Technology*, 2024.

[26] Q. Sima, G. Gao, H. Huang, Y. E. Sun, Y. Du, X. Wang, and J. Wu. Multi-armed bandits based task selection of a mobile crowdsensing worker. In *2022 International Conference on Computer Communications and Networks (ICCCN)*, July 2022.

[27] E. M. Mohamed, S. Hashima, and K. Hatano. Energy aware multiarmed bandit for millimeter wave-based uav mounted ris networks. *IEEE Wireless Communications Letters*, 2022.

[28] M. M. Fouda, S. Hashima, S. Sakib, Z. M. Fadlullah, K. Hatano, and X. Shen. Optimal channel selection in hybrid rf/vlc networks: A multi-armed bandit approach. *IEEE Transactions on Vehicular Technology*, 71(6):6853–6858, 2022.

[29] F. Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing*, 1987.

[30] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*. PMLR, February 2013.

[31] Y. Li, Q. Hu, and N. Li. Learning and selecting the right customers for reliability: A multi-armed bandit approach. In *2018 IEEE Conference on Decision and Control*, December 2018.

[32] Y. Wang, W. Chen, and M. Vojnović. Combinatorial bandits for maximum value reward function under max value-index feedback. *arXiv preprint*, arXiv:2305.16074, 2023.

[33] V. Kuleshov and D. Precup. Algorithms for multi-armed bandit problems. *arXiv preprint*, arXiv:1402.6028, 2014.

[34] S. Agrawal and N. Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, JMLR Workshop and Conference Proceedings, June 2012.

[35] C. Frappier and Q. I. Rahman. On an inequality of s. bernstein. *Canadian Journal of Mathematics*, 34(4):932–944, 1982.

[36] S. Abramovich, G. Jameson, and G. Sinnamon. Refining jensen's inequality. *Bulletin mathématique de la Société des Sciences Mathématiques de Roumanie*, pages 3–14, 2004.