

Multi-product Influence Maximization in Billboard Advertisement

Dildar Ali

Department of Computer Science and
Engineering, Indian Institute of
Technology Jammu
Jammu, Jammu & Kashmir, India
2021rcs2009@iitjammu.ac.in

Rajibul Islam

Department of Computer Science and
Engineering, Gandhi Institute for
Technological Advancement
Bhubaneswar, Odisha, India
rajibulislam1604@gmail.com

Suman Banerjee

Department of Computer Science and
Engineering, Indian Institute of
Technology Jammu
Jammu, Jammu & Kashmir, India
suman.banerjee@iitjammu.ac.in

ABSTRACT

Billboard Advertisement has emerged as an effective out-of-home advertisement technique where the goal is to select a limited number of slots and play advertisement content over there with the hope that this will be observed by many people, and effectively, a significant number of them will be influenced towards the brand. Given a trajectory and a billboard database and a positive integer k , how can we select k highly influential slots to maximize influence? In this paper, we study a variant of this problem where a commercial house wants to make a promotion of multiple products, and there is an influence demand for each product. We have studied two variants of the problem. In the first variant, our goal is to select k slots such that the respective influence demand of each product is satisfied. In the other variant of the problem, we are given with ℓ integers $k_1, k_2, \dots, k_\ell$, the goal here is to search for ℓ many set of slots $S_1, S_2, \dots, S_\ell$ such that for all $i \in [\ell]$, $|S_i| \leq k_i$ and for all $i \neq j$, $S_i \cap S_j = \emptyset$ and the influence demand of each of the products gets satisfied. We model the first variant of the problem as a multi-submodular cover problem and the second variant as its generalization. For solving the first variant, we adopt the bi-criteria approximation algorithm, and for the other variant, we propose a sampling-based approximation algorithm. Extensive experiments with real-world trajectory and billboard datasets highlight the effectiveness and efficiency of the proposed solution approach.

CCS CONCEPTS

• Information Systems → Computational Advertising;

KEYWORDS

Billboard Advertisement, Influence Provider, Advertiser, Regret Minimization

1 INTRODUCTION

Creating and maximizing influence is an effective marketing strategy, and most commercial houses spend 7 – 10% of total revenue on advertisements. Among many, one of the effective approaches is billboard advertisement because of its guaranteed return on investment, ease of use, etc. In this advertising technique, a limited number of influential billboard slots are used in the hope that if some attractive advertisement content is displayed, it will create an influence among people who observe the advertisement content. The key problem that arises in this context is how to select a limited number of influential slots so that the influence is maximized.

This problem has been referred to as the Influential Billboard Slot Selection Problem. There exist several studies on this problem.

Our Observation. Existing studies in billboard advertising scenarios share a common objective: (a) helping advertisers achieve maximum influence under budget constraints in a single or multi-advertiser setting [9, 17, 20], and (b) minimizing the regret of an influence provider by effective allocation of slots to advertisers [3, 4, 8, 22]. It is standard practice that multiple advertisers submit a campaign proposal to the influence provider, specifying the influence demand and the corresponding payment. A more challenging, yet unexplored, problem is how to allocate slots across multiple advertisers having multiple products under a given budget to maximize total influence from the influence provider’s perspective.

Motivation. Advertisers promote multiple heterogeneous products, each targeting a different customer base. Billboard advertisements typically optimize slot selection to maximize the individual influence of the advertiser or allocate slots to minimize the influence provider’s regret. However, there is a practical need for commercial houses to satisfy distinct influence demands across multiple products simultaneously. This requires a new formulation to allocate billboard slots within a unified budget while maximizing overall influence across all products.

Our Problem. We introduce the Multi-Product Influential Billboard Slot Selection Problem in two variants: (1). Common Slot Selection, in which we select a fixed set of slots (within budget) that jointly satisfy the influence demands of all products. (2). Disjoint Slot Selection, where we select disjoint sets of slots, one for each product, ensuring that the individual influence demand of every product is met and the total cost is minimized.

Hardness. Both problem variants are computationally hard, generalizing the NP-hard Multi-Submodular Cover Problem [14]. The disjoint variant further extends this to the Generalized Multi-Submodular Cover Problem by adding non-overlapping slot constraints and separate budgets per product, increasing complexity.

Our Solutions. We proposed two different solution approaches to solve the slot selection problem. For the Common Slot Selection Problem, we develop a continuous greedy-based bi-criteria approximation algorithm that leverages the submodular structure of the influence function to yield a provable approximation guarantee. For the Disjoint Slot Selection Problem, we design a sampling-based randomized algorithm. This algorithm randomly samples permutations of slot-product assignments and selects the one with minimal cost while satisfying all influence constraints. We perform a sample

complexity analysis using Hoeffding's inequality [16], establishing bounds on the number of samples needed to achieve near-optimal performance with high probability.

General Applicability. The proposed framework and algorithms are applicable to any multi-product, out-of-home advertisement setting. This includes billboard networks, public displays, transit advertising, and digital signage, where multiple product campaigns must be scheduled concurrently under budget constraints.

Relevant Studies. Existing studies on computational problems in the context of billboard advertisement can be classified into two categories: Influential Slot Selection and Regret Minimization. In the influential slot selection problem, we are given the trajectory information (location of a group of people across different time-stamps), billboard slot information (location, duration, and cost), and a limited budget. The task here is to select a limited number of influential billboard slots to maximize the influence. Existing literature includes a pruned submodularity graph-based approach [1, 6], a branch and bound-based framework [2], greedy-based solutions [7], and a Co-operative Co-Evolutionary approach proposed by Wang et al. [19], etc. On the other direction of studies, it is assumed that there exists an influence provider who has access to a number of billboards and a number of advertiser approaches for an influence demand in exchange for some budget. Now, the task of the influence provider is to allocate slots to the advertisers in such a way that minimizes their total regret. There exists some study on this problem, which includes some local search-based solution approaches by Zhang et al. [22], regret minimization under zonal influence constraint by Ali et al. [3–5]. However, to our knowledge, there is no literature on the slot selection problem that considers the existence of multiple products that advertisers advertise.

Our Contributions. In practice, a commercial house often advertises multiple diverse products, each targeting different audiences. Thus, billboard slot selection must account for multi-product promotion needs. To the best of our knowledge, there does not exist any study that addresses this problem. To bridge this gap, in this paper, we study the Multi-Product Influential Billboard Slot Selection Problem in two variants. In the first variant, it asks to choose a subset of k billboard slots such that the influence demand of each product gets satisfied and the cost of selection gets minimized. Here, all the selected slots are common, and all of them will be used for promoting the products. We refer to the other variant of the problem as the Disjoint Multi-Product Slot Selection Problem. In this problem, we are given a trajectory and billboard database, and a set of ℓ integers. This problem asks to choose nonintersecting ℓ many subsets of slots bounded by their respective cardinality, such that the total selection cost is minimized and the influence demand of each product gets satisfied. We have developed approximation algorithms for these problems. In particular, we make the following contributions in this paper:

- We extend the influential slot selection problem to the setting where multiple products are to be advertised. We study two related problems in this direction.
- As both problems are NP-hard, we propose approximation algorithms to solve these problems.

- To address the scalability issues, we have developed efficient heuristic solutions.
- A number of experiments have been carried out to show the effectiveness and efficiency of the proposed solution approaches.

Organization of the Paper. The rest of the paper has been organized as follows. Section 2 describes the required preliminary concepts and defines the problem formally. The proposed solution approaches have been described in Section 3. Section 4 describes the experimental evaluation. Finally, Section 5 presents the concluding remarks of our study.

2 BACKGROUND AND PROBLEM DEFINITION

2.1 Trajectory and Billboard Database

A trajectory database contains the location information of a set of people in a city across different time stamps. In the context to our problem, a trajectory database $\mathcal{D}$ contains tuples of the form $(u_i, \text{loc}, [t_a, t_b], \mathcal{P}(u_i))$ and this signifies that the person u_i is at the location loc for the duration t_a to t_b . Here, $\mathcal{P}(u_i)$ denotes the set of products in which user u_i will be interested in. For any person u_i , $\text{loc}_{[t_x, t_y]}(u_i)$ denotes the locations of u_i during the time interval $[t_x, t_y]$. Let $\mathcal{U}$ denote the set of people whose movement data is contained in $\mathcal{D}$. Let $T_1 = \min_{\tau \in \mathcal{D}} t_a$ and $T_2 = \max_{t \in \mathcal{D}} t_b$ and we say that the trajectory database $\mathcal{D}$ contains the movement data from time stamp T_1 to T_2 . A billboard database $\mathcal{B}$ contains the information about billboard slots. Typically, this contains the tuples of the following form $(b_{id}, s_{id}, \text{loc}, \text{duration})$ where b_{id} and s_{id} denote the billboard id and slot id. loc and duration signify the location of the billboard and the duration of the slot.

2.2 Billboard Advertisement

Let a set of m billboards $B = \{b_1, b_2, \dots, b_m\}$ be placed in different locations of a city. For simplicity, we assume that all the billboards can be leased for a multiple of a fixed duration Δ , which is called a 'slot' and has been stated in Definition 2.1.

Definition 2.1 (Billboard Slot). A billboard slot is defined by a tuple consisting of two entities, the billboard ID and the duration, that is, $(b_{id}, [t, t + \Delta])$.

The set of all billboard slots is denoted by $\mathcal{BS}$, i.e., $\mathcal{BS} = \{(b_i, [t, t + \Delta]) : i \in \{1, 2, \dots, m\} \text{ and } t \in \{T_1, T_1 + \Delta, T_1 + 2\Delta, \dots, T_2 - \Delta\}\}$. It can be observed that $|\mathcal{BS}| = m \cdot \frac{T}{\Delta}$ where $T = T_2 - T_1$. For simplicity, we assume that T is perfectly divisible by Δ . For any slot $s_j \in \mathcal{BS}$, b_{s_j} denotes the corresponding billboard and $[t_{s_j}^s, t_{s_j}^f]$ denotes the time interval for the slot s_j where $t_{s_j}^f = t_{s_j}^s + \Delta$. Now, we state the notion of the influence probability of a billboard slot in Definition 2.2.

Definition 2.2 (Influence Probability of a Billboard slot). Given a slot $s_j \in \mathcal{BS}$ and a person $u_i \in \mathcal{U}$, the influence probability of s_j on u_i is denoted by $Pr(s_j, u_i)$ and can be computed using the following conditional equation.

$$Pr(s_j, u_i) = \begin{cases} \frac{\text{size}(s_j)}{\max_{s_k \in \mathcal{BS}} \text{size}(s_k)}, & \text{if } \text{loc}_{[t_{s_j}^s, t_{s_j}^f]}(u_i) = \text{loc}(s_j) \\ 0, & \text{otherwise} \end{cases}$$

Now, we define the influence of a subset of billboard slots in 2.3.

Definition 2.3 (Influence of Billboard Slots). Given a trajectory database $\mathcal{D}$, and a subset of billboard slots $\mathcal{S} \subseteq \mathcal{BS}$, the influence of $\mathcal{S}$ can be defined as the expected number of trajectories that are influenced, which can be computed using Equation 1.

$$\mathcal{I}'(\mathcal{S}) = \sum_{u_i \in \mathcal{U}} \left[1 - \prod_{s_j \in \mathcal{S}} (1 - \text{Pr}(s_j, u_i)) \right] \quad (1)$$

Here, $\text{Pr}(b_j, u_i)$ denotes the influence probability of the billboard slot b_j on the people u_i , and $\mathcal{I}'(\mathcal{S})$ denotes the influence of the billboard slots of $\mathcal{S}$. $\mathcal{I}'(\cdot)$ is the influence function which maps each possible subset of billboard slots to its corresponding influence value, i.e., $\mathcal{I}' : 2^{\mathcal{BS}} \rightarrow \mathbb{R}_0^+$ where $\mathcal{I}'(\emptyset) = 0$.

Now, it is important to observe that every product may not be relevant to every trajectory user. For effective advertisement, it is important to influence the relevant people for a particular product. Consider $\mathcal{P} = \{1, 2, \dots, p\}$ denotes the set of products. From the trajectory database, for every trajectory user, we have the information about the products for which he is relevant. For a given product, $j \in [p]$, we state the notion of Product Specific Influence for a given billboard slot in Definition 2.4.

Definition 2.4 (Product Specific Influence). Given a specific product and a subset of billboard slots $\mathcal{S} \subseteq \mathcal{BS}$, the product-specific influence of $\mathcal{S}$ for the product j is denoted by $\mathcal{I}_j(\mathcal{S})$ and can be computed using Equation 2.

$$\mathcal{I}_j(\mathcal{S}) = \sum_{u_i \in \mathcal{U}_k} \left[1 - \prod_{s_j \in \mathcal{S}} (1 - \text{Pr}(s_j, u_i)) \right] \quad (2)$$

Here, $\mathcal{U}_k$ denotes the set of users relevant to the k -th product and $\mathcal{I}$ is the product specific influence function, i.e., $\mathcal{I} : 2^{\mathcal{BS}} \times \mathcal{P} \rightarrow \mathbb{R}_0^+$.

2.3 Submodular Set Functions and Continuous Extensions

A function $f : 2^N \rightarrow \mathbb{R}$ is submodular if for every $A \subseteq B \subseteq N$, $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$ such that $e \in N$ and $e \notin B$. Such functions arise in many applications. The continuous extension of submodular set functions has played an important role in algorithmic aspects. The idea is to extend a discrete set function $f : 2^N \rightarrow \mathbb{R}$ to continuous space $[0, 1]^N$. Here, we are mainly concerned with multi-linear extension motivated by maximization problems and refer the interested reader to Calinescu et al. [13], Vondrak [18] for a detailed discussion.

The multi-linear extension is a real-valued set function $f : 2^N \rightarrow \mathbb{R}$, denoted by F and defined as follows:

$$F(x) = \sum_{\mathcal{S} \subseteq N} f(\mathcal{S}) \prod_{i \in \mathcal{S}} x_i \prod_{j \notin \mathcal{S}} (1 - x_j) \quad (3)$$

where $x \in [0, 1]^N$, and the random subset $\mathcal{S} \subseteq N$ is drawn by including each element $i \in N$ independently with probability x_i , i.e., $\text{Pr}[i \in \mathcal{S}] = x_i$ independently for each i .

For any two vectors $p, q \in [0, 1]^N$, we use $p \vee q$ and $p \wedge q$ to denote the coordinate-wise maximum and minimum, respectively of p and q . We also make use of the notation $w_e(t) \leftarrow F(y(t) \vee 1_e) - F(y(t))$, where 1_e is the characteristic vector of set $\{e\}$.

2.4 Problem Definition

Common and Disjoint Multi-Product Slot Selection Problem.

As mentioned previously, we have studied the Multi-Product Slot Selection Problem in two variants. First, we talk about the variant where a fixed number of slots will be selected for all the products. In this problem we are given with a trajectory and billboard database (which includes the cost function $w : \mathcal{BS} \rightarrow \mathbb{R}^+$), and ℓ positive integers $k_1, k_2, \dots, k_\ell \in \mathbb{Z}^+$, the task is to find out a subset of the slots $\mathcal{S} \subseteq \mathcal{BS}$ such that the total cost gets minimized and for each product $j \in [\ell]$, their influence demand constraint gets satisfied, i.e., for all $j \in [\ell]$, $\mathcal{I}_j(\mathcal{S}) \geq k_j$. From a computational point of view, this problem can be posed as follows:

COMMON MULTI-PRODUCT SLOT SELECTION PROBLEM

Input: Billboard $\mathcal{B}$ and Trajectory Database $\mathcal{D}$, Cost Function $w : \mathcal{BS} \rightarrow \mathbb{R}^+$, ℓ integers $k_1, k_2, \dots, k_\ell \in \mathbb{Z}^+$ for influence demand constraint.

Problem: Select a subset of the slots $\mathcal{S} \subseteq \mathcal{BS}$ such that $\sum_{v \in \mathcal{S}} w(v)$ gets minimized and the constraints $f_j(\mathcal{S}) \geq k_j$ are satisfied.

In the Disjoint Multi-Product Slot Selection Problem, along with the trajectory and billboard database, and the cost function, we are given with two sets of ℓ integers $p_1, p_2, \dots, p_\ell \in \mathbb{Z}^+$ and $\sigma_1, \sigma_2, \dots, \sigma_\ell \in \mathbb{Z}^+$. The task here is to select ℓ many subsets of slots $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_\ell \subseteq \mathcal{BS}$ such that for all $j \in [\ell]$, $|\mathcal{S}_j| \leq k_j$, for all $i \neq j$, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$, and for all $j \in [\ell]$, $\mathcal{I}_j(\mathcal{S}_j) \geq \sigma_j$. From a computational point of view, this problem can be posed as follows:

DISJOINT MULTI-PRODUCT SLOT SELECTION PROBLEM

Input: Billboard $\mathcal{B}$ and Trajectory Database $\mathcal{D}$, Two sets of ℓ integers $p_1, p_2, \dots, p_\ell \in \mathbb{Z}^+$ and $\sigma_1, \sigma_2, \dots, \sigma_\ell \in \mathbb{Z}^+$.

Problem: Select ℓ many subsets of slots $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_\ell \subseteq \mathcal{BS}$ such that:

1. For all $j \in [\ell]$, $|\mathcal{S}_j| \leq k_j$
2. For all $i \neq j$, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$
3. For all $j \in [\ell]$, $\mathcal{I}_j(\mathcal{S}_j) \geq \sigma_j$

As our problem is closely related to the Multi-submodular Cover Problem, first, we describe and generalize it.

Multi-Submodular Cover Problem. In Multi-Submodular Cover Problem, we are given with a ground set $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$, a cost function $w : \mathcal{V} \rightarrow \mathbb{R}^+$ which maps each ground set element to its corresponding cost, a set of ℓ submodular functions $f_1, f_2, \dots, f_\ell$, and a set of ℓ integers $k_1, k_2, \dots, k_\ell \in \mathbb{Z}^+$. This problem asks to choose a subset $\mathcal{S}$ of ground set elements, i.e., $\mathcal{S} \subseteq \mathcal{V}$, such that the total cost of the chosen elements gets minimized and for any $i \in [\ell]$, $f_i(\mathcal{S}) \geq k_i$, i.e., the constraints on the functional values of the submodular functions get satisfied. This is an optimization problem which can be posed as follows:

$$\mathcal{S}^* \leftarrow \underset{\substack{\mathcal{S} \subseteq \mathcal{V} \\ \forall j \in [\ell], f_j(\mathcal{S}) \geq k_j}}{\operatorname{argmin}} \sum_{v \in \mathcal{S}} w(v) \quad (4)$$

Here, S^* denotes the optimal solution for this problem. From the computational point of view, this problem can be posed as follows:

MULTI-SUBMODULAR COVER PROBLEM

Input: The ground set $X = \{x_1, x_2, \dots, x_n\}$, the cost function $w : X \rightarrow \mathbb{R}^+$, ℓ many submodular functions $f_1, f_2, \dots, f_\ell$ defined on the ground set X , ℓ many positive integers $k_1, k_2, \dots, k_\ell \in \mathbb{Z}^+$.

Problem: Select a subset of the ground set elements $S \subseteq X$ which minimizes $\sum_{x \in S} w(x)$ but satisfies $f_j(S) \geq k_j$ for all $j \in [\ell]$.

An instance of the Multi-Submodular Cover Problem is said to be r -sparse if each ground set element can be involved in a maximum of r objectives. This problem has been studied by Chekuri et al. [14], and they proposed a randomized bi-criteria approximation algorithm which runs in polynomial time. The formal result has been stated in Theorem 2.5.

THEOREM 2.5. *There exists a randomized bi-criteria approximation algorithm that for a given instance of the Multi-Submodular Cover Problem it produces a set $S \subseteq X$ such that (i) For all $j \in [\ell]$, $f_j(S) \geq (1 - \frac{1}{e} - \epsilon)k_j$ and (ii) $\mathbb{E}[w(S)] \geq O(\frac{1}{e} \ln r)$.*

GENERALIZED MULTI-SUBMODULAR COVER PROBLEM

Input: The ground set $X = \{x_1, x_2, \dots, x_n\}$, the cost function $w : X \rightarrow \mathbb{R}^+$, ℓ many submodular functions $f_1, f_2, \dots, f_\ell$ defined on the ground set X , two sets of ℓ positive real numbers each $k_1, k_2, \dots, k_\ell \in \mathbb{Z}^+$ and $\sigma, \sigma_2, \dots, \sigma_\ell \in \mathbb{R}^+$.

Problem: select ℓ many subsets of the ground set $S_1, S_2, \dots, S_\ell \subseteq X$ such that:

- For all $i, j \in [\ell]$ and $i \neq j$, $S_i \cap S_j = \emptyset$
- For all $i \in [\ell]$, $\sum_{x \in S_i} w(x) \leq k_i$
- For all $i \in [\ell]$, $f_i(S_i) \geq k_i$

Generalized Multi-Submodular Cover Problem. In this problem we are given with a ground set of n elements $X = \{x_1, x_2, \dots, x_n\}$, a cost function w that maps each ground set element to its corresponding cost; i.e., $w : X \rightarrow \mathbb{R}^+$, ℓ many submodular functions $f_1, f_2, \dots, f_\ell$ defined on the ground set X , and two sets of ℓ many real numbers $k_1, k_2, \dots, k_\ell \in \mathbb{R}^+$ and $\sigma, \sigma_2, \dots, \sigma_\ell \in \mathbb{R}^+$. The task is to select ℓ many subsets of the ground set $S_1, S_2, \dots, S_\ell \subseteq X$ such that: (i) For all $i, j \in [\ell]$ and $i \neq j$, $S_i \cap S_j = \emptyset$, (ii) For all $i \in [\ell]$, $\sum_{x \in S_i} w(x) \leq k_i$, and (iii) For all $i \in [\ell]$, $f_i(S_i) \geq k_i$.

3 PROPOSED SOLUTION APPROACHES

In this section, we describe the proposed solution approaches. Section 3.1 and 3.2 contain the methodologies for the Common and Disjoint Multi-Product Slot Selection Problem, respectively.

3.1 Common Multi-Product Slot Selection

We analyze the continuous greedy algorithm for general monotone, submodular functions. At each step t , in Line No. 3 to 5, the marginal

gain for each $e \in \mathcal{BS}$ is computed. Next, in Line No. 6, solve the linear program to get a direction vector, and for each slot in $\mathcal{BS}$, each coordinate of $y(t)$ is updated in Line No. 7 to 9. Finally, after time T , it returns the fractional solution $y(T)$. Parameter T is the stopping criterion of Algorithm 1 as mentioned in Theorem 3.1. It is important to note that in Line 8, we modify the direction y . The definition of δ implies that δ^{-1} is at least n^5 and is divisible by T^{-1} . This ensures that after $T\delta^{-1}$ iterations, t will be exactly equal to T .

THEOREM 3.1. [15] *For any normalized monotone submodular function $f : 2^N \rightarrow \mathbb{R}^+$, a solvable polytope $\mathcal{P} \in [0, 1]^N$ and stopping time $T > 0$, Algorithm 1 finds a point $x \in [0, 1]^N$, such that $F(x) \geq [1 - e^{-T} - O(1)] \cdot f(OPT)$.*

Algorithm 1: Continuous Greedy Algorithm

Data: Trajectory Database $\mathcal{D}$, Billboard Database $\mathcal{B}$, Influence Functions $I_j()$, Billboard Slots $\mathcal{BS}$

Result: A solution of vector $\mathcal{Y}(T)$

```

1 Initialize  $t \leftarrow 0$ ,  $\mathcal{Y}(0) \leftarrow 1_0$ ,  $n \leftarrow |\mathcal{BS}|$ ,  $\delta \leftarrow T((n^5 T))^{-1}$ ;
2 while  $t < T$  do
3   for each  $e \in \mathcal{BS}$  do
4      $w_e(t) \leftarrow F(y(t) \vee 1_e) - F(y(t))$ 
5   end
6    $I(t) \leftarrow \text{argmax}\{x \cdot w(t) \mid x \in \mathcal{P}\}$ ;
7   for each  $e \in \mathcal{BS}$  do
8      $y_e(t + \delta) \leftarrow y_e(t) + \delta I_e(t) \cdot (1 - y_e(t))$ 
9   end
10   $t \leftarrow t + \delta$ 
11 end
12 return  $y(T)$ ;
```

We address the *Common Multi-Product Slot Selection Problem* in Algorithm 2, where a single slot set must satisfy the influence demands of multiple products under budget. We propose a bi-criteria approximation algorithm combining continuous greedy and randomized rounding. First, in Line No. 1, we scale each product's influence function I_j such that $I_j(\mathcal{BS}) = 1$, and set $k_j \leftarrow 1$, and in Line No. 2, we define the polytope $\mathcal{P}$. In Line No. 3, using Algorithm 1, we solve the multilinear extension over the matroid polytope to obtain a fractional solution $x \in [0, 1]^{|\mathcal{BS}|}$. In the rounding step, Line No. 5 to 7, sample $\ell = \lceil \log_{1/(1-\epsilon)}(r) \rceil$ random subsets from x to form S . Next, in Line No. 8 to 14 for any product j where $I_j(S)$ is below $(1 - \frac{1}{e} - 2\epsilon)k_j$, add slots greedily by marginal gain until the bound $(1 - \frac{1}{e})k_j$ is met.

THEOREM 3.2. [14] *Let r be the maximum number of functions any slot contributes to (i.e., sparsity), and let $\epsilon > 0$. Then with high probability, the Algorithm 2 returns a set S such that for all $j \in [\ell]$, $I_j(S) \geq (1 - \frac{1}{e} - \epsilon) \cdot k_j$, and the expected cost satisfies $\mathbb{E}[w(S)] = O(\frac{1}{\epsilon} \log r) \cdot OPT$.*

Complexity Analysis. Let $n = |\mathcal{BS}|$ be the number of billboard slots and ℓ the number of products. The Continuous Greedy phase runs in $O(n^2 \ell / \epsilon)$ time due to gradient computation over a polytope of size n for each of ℓ influence functions. The rounding and union of $\ell = O(\log r / \epsilon)$ random subsets take $O(n \log r / \epsilon)$ time. In the worst case, the repair step examines all n slots per product, taking $O(n\ell)$ time. So the overall time complexity is $O\left(\frac{n^2 \ell}{\epsilon} + n\ell\right)$. The additional space requirement for Algorithm 2 is $O(n)$ for maintaining the fractional vector and final slot set.

Algorithm 2: Bi-criteria Approximation Algorithm for Multi-Product Influence Maximization

Data: Trajectory Database $\mathcal{D}$, Billboard Slots $\mathcal{BS}$, Influence Functions $\{I_j(\cdot)\}_{j=1}^h$, Cost Function $w : \mathcal{BS} \rightarrow \mathbb{R}^+$, Sparsity r , Approximation Parameter $\epsilon > 0$, Influence Thresholds $\{k_j\}_{j=1}^h$

Result: A set of billboard slots $\mathcal{S} \subseteq \mathcal{BS}$ satisfying all influence constraints approximately

- 1 **Normalize:** For each $j \in [h]$, set $I_j(\mathcal{BS}) = 1$ and scale $k_j \leftarrow 1$;
- 2 Let $\mathcal{P} \subseteq [0, 1]^{|\mathcal{BS}|}$ be the constraint polytope;
- 3 $x \leftarrow \text{ContinuousGreedy}(\mathcal{BS}, \mathcal{P}, T = 1)$;
- 4 $\alpha \leftarrow 1 - \epsilon$, $\ell \leftarrow \lceil \log_{1/\alpha} r \rceil$, $\mathcal{S} \leftarrow \emptyset$;
- 5 **for** $t = 1$ **to** ℓ **do**
- 6 Sample $S_t \subseteq \mathcal{BS}$ by including each $i \in \mathcal{BS}$ independently with probability $x(i)$;
- 7 $\mathcal{S} \leftarrow \mathcal{S} \cup S_1 \cup S_2 \cup \dots \cup S_\ell$;
- 8 **for** $j = 1$ **to** h **do**
- 9 **if** $I_j(\mathcal{S}) < (1 - \frac{1}{e} - 2\epsilon) \cdot k_j$ **then**
- 10 $M_j \leftarrow \emptyset$;
- 11 **while** $I_j(M_j) < (1 - \frac{1}{e}) \cdot k_j$ **do**
- 12 $u^* \leftarrow \arg \max_{u \in \mathcal{BS} \setminus M_j} [I_j(M_j \cup \{u\}) - I_j(M_j)]$;
- 13 $M_j \leftarrow M_j \cup \{u^*\}$;
- 14 $\mathcal{S} \leftarrow \mathcal{S} \cup M_j$;
- 15 **return** $\mathcal{S}$

3.2 Disjoint Multi-Product Slot Selection

In this section, we describe the proposed solution approach for the Disjoint Multi-Product Slot Selection Problem. In this problem, we are given a budget and the influence demand for every product. The goal here is to select billboard slots for each product within the budget such that the influence demand for every product is satisfied. Also, a slot can be used for at most one product only. In our proposed methodology, for every possible permutation of products and slots, we perform the following task. As long as the remaining budget of the current advertiser being processed is positive and his influence demand is not satisfied, then in each iteration, we select a slot that makes the maximum possible marginal gain with the existing set of slots that have been chosen for the advertiser. This process terminates once the influence demand is satisfied or the advertiser's budget is exhausted. If the budget of the advertiser is exhausted, then it means that, for the current permutation of advertisers and slots, for this advertiser, the influence demand can not be satisfied. Hence, this does not lead to a feasible solution, and we mark it as an infeasible solution. This process is continued for all possible permutations of advertisers and slots. Among all the feasible solutions stored in $\mathcal{M}$, we return one with the least cost. Algorithm 1 describes the whole process in the form of pseudocode.

One point to highlight here is that the number of possible permutations will be of $O(\ell! \cdot n!)$. By starling's approximation, we can write it as $O(\ell^\ell \cdot n^n)$. For moderate values of ℓ and n (in practice, the value of n could be excessively large), the number of possible permutations could be large. If we consider all of them in the computation, then the execution time requirement will not be practically feasible. To mitigate this problem, in this paper, we sample out independently at random a subset of them. Now, it is an important question what the sample size should be such that the error in the estimation could be bounded with high probability. This leads to the sample complexity analysis, which has been stated subsequently.

3.2.1 Sample Complexity Analysis. We can use Hoeffding's Inequality to provide a sample bound that guarantees finding a

close to optimal solution with high probability [?]. Theorem 2 describes the statement.

THEOREM 3.3. *Let $X_1, X_2, \dots, X_n$ be independent and identically distributed random variables such that for any $i \in [n]$, $a_i \leq X_i \leq b_i$. Let $\bar{X} = \frac{1}{n} \sum_{i \in [n]} X_i$ and $\mu = \mathbb{E}[\bar{X}]$. For any $\epsilon > 0$, the following inequality holds*

$$\Pr[|\bar{X} - \mu| \geq \epsilon] \leq 2 \cdot \exp\left(-\frac{2n^2\epsilon^2}{\sum_{i \in [n]} (b_i - a_i)^2}\right) \quad (5)$$

So for estimation, the upper and lower bound of the solution cost of each feasible solution present in the sample is required. We prove the same in Proposition 3.4.

PROPOSITION 3.4. *For any feasible solution $\mathcal{S} = (S_1, S_2, \dots, S_\ell)$ contained in $\mathcal{M}$ the cost of solution will lie in between 0 and $\sum_{s \in \mathcal{BS}} w(s)$.*

PROOF. For any advertiser $a_i \in \mathcal{A}$, its influence demand $\sigma_i \geq 0$ and for any slot $s_j \in \mathcal{BS}$, its cost $w(s_j) \geq 0$. Now, it can be observed that the lower bound holds only when for all the advertisers, their influence demand is all 0. In that case, every advertiser will be assigned the empty set, i.e., for all $i \in [\ell]$, $S_i = \emptyset$, and hence, the solution cost will be 0. In any other case, the cost of the solution will be strictly greater than 0. The upper bound occurs when the allocated slots to the advertisers, i.e., $S_1, S_2, \dots, S_\ell$, is a partition of $\mathcal{BS}$. This means (i) For all $i, j \in [\ell]$, $S_i \cap S_j = \emptyset$, (ii) $\bigcup_{i \in [\ell]} S_i = \mathcal{BS}$. In this case, the cost of the solution will be $\sum_{s \in \mathcal{BS}} w(s)$. In any other case, the cost of the solution will be strictly less than $\sum_{s \in \mathcal{BS}} w(s)$. Hence, for any solution $\mathcal{S} = (S_1, S_2, \dots, S_\ell)$, the cost of the solution will be in the range in between 0 and $\sum_{s \in \mathcal{BS}} w(s)$. $\square$

Now, we prove Theorem 3.5, which will prove the lower bound on the sample size for which Algorithm 1 will provide a close to optimal solution with high probability.

THEOREM 3.5. *For any $\epsilon, \delta \in (0, 1)$ if the sample size is greater than equals to $\frac{\ln(\frac{2}{\delta}) \cdot w(\mathcal{BS})^2}{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2}$ then the probability error in computation will be strictly less than ϵ with probability at least $(1 - \delta)$.*

PROOF. Let, $\mathcal{Y}$ denote the set of sampled solutions. W^* and $W^{\mathcal{A}}$ denote the optimal solution and the best solution among the solutions in Sample, respectively. Now, by applying Hoeffding's Inequality and Proposition 3.4 we gave the following inequality:

$$\Pr[|W^* - W^{\mathcal{A}}| \geq \epsilon \cdot W^*] \leq 2 \cdot \exp\left(-\frac{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2 \cdot |\mathcal{Y}|}{w(\mathcal{BS})^2}\right) \quad (6)$$

We want to establish the sample size such that $\Pr[|W^* - W^{\mathcal{A}}| < \epsilon \cdot W^*] \geq 1 - \delta$. By simplifying, we have the following:

$$2 \cdot \exp\left(-\frac{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2 \cdot |\mathcal{Y}|}{w(\mathcal{BS})^2}\right) \leq \delta$$

$$\ln\left(\frac{2}{\delta}\right) \leq \frac{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2 \cdot |\mathcal{Y}|}{w(\mathcal{BS})^2}$$

This leads to the lower bound on the sample size, which is

$$|\mathcal{Y}| \geq \frac{\ln(\frac{2}{\delta}) \cdot w(\mathcal{BS})^2}{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2}$$

□

Algorithm 3: Randomized Algorithm for Multi-Product Influence Maximization

Data: Billboard Slots $\mathcal{BS}$, Cost Function $w : \mathcal{BS} \rightarrow \mathbb{R}^+$, The influence functions $I_1, I_2, \dots, I_\ell, v_1, v_2, \dots, v_\ell \in \mathbb{R}^+$, and $\sigma_1, \sigma_2, \dots, \sigma_\ell \in \mathbb{R}^+$
Result: Non-intersecting sets of slots $S_1, S_2, \dots, S_\ell \subseteq \mathcal{BS}$

```

1  $a \leftarrow \ell!$ ,  $b \leftarrow |\mathcal{BS}|$ ;
2 for  $All p \in [a]$  AND  $q \in [b]$  do
3    $M[p, q] \leftarrow$  Empty Solution;
4    $Flag[p, q] \leftarrow 1$ ;
5 for  $All p \in [a]$  AND  $q \in [b]$  do
6    $\mathbb{S} \leftarrow \emptyset$ ;
7   for  $i = 1$  to  $\ell$  do
8      $S_i \leftarrow \emptyset$ ;
9   for  $i = 1$  to  $\ell$  do
10    while  $v_i > 0$  AND  $I(S_i) < \sigma_i$  do
11       $s^* \leftarrow \underset{s \in \mathcal{BS} \setminus \mathbb{S}}{\operatorname{argmax}} I(S_i \cup \{s\}) - I(S_i)$ ;
12       $S_i \leftarrow S_i \cup \{s^*\}$ ,  $\mathbb{S} \leftarrow \mathbb{S} \cup \{s^*\}$ ;
13       $v_i \leftarrow v_i - w(s^*)$ ;
14    if  $I(S_i) < \sigma_i$  then
15       $Flag[p, q] \leftarrow 0$ ;
16      break;
17 if  $Flag[p, q] = 0$  then
18    $M[p, q] \leftarrow$  Not a Feasible Solution;
19 else
20    $M[p, q] \leftarrow [\{S_1, S_2, \dots, S_\ell\}, \sum_{s \in S_1 \cup S_2 \cup \dots \cup S_\ell} w(s)]$ ;
21  $(S_1^*, S_2^*, \dots, S_\ell^*) \leftarrow \underset{(S_1, S_2, \dots, S_\ell) \in M}{\operatorname{argmin}} \sum_{s \in S_1 \cup S_2 \cup \dots \cup S_\ell} w(s)$ ;
22 return  $(S_1^*, S_2^*, \dots, S_\ell^*)$ ;
```

3.2.2 Computational Complexity Analysis. Consider $|\mathcal{X}|$ denotes the sample size used to implement Algorithm 1. Both the for loops from 2 to 4 and from 5 to 20 will execute $O(|\mathcal{X}|)$. Again the for loops from 7 to 8 and 9 to 16 will execute for $O(\ell)$ times. Now, it is important to analyze how many times the while loop from 10 to 16 will execute. Consider the minimum selection cost among all the billboards is denoted by w_{min} , i.e., $w_{min} = \min_{bs_j \in \mathcal{BS}} w(bs_j)$.

Let $B = \{k_1, k_2, \dots, k_\ell\}$ are the budgets of the advertisers. Among these, let B_{max} denote the maximum budget among all the advertisers, i.e., $B_{max} = \max_{i \in [\ell]} k_i$. It can be observed that the maximum

$\frac{B_{max}}{w_{min}}$ number of slots can be allocated to an advertiser. Hence, the while loop can be iterated at max $O(\frac{B_{max}}{w_{min}})$ times. Within the while loop, the main computation is the marginal influence gain computation of a non-allocated slot, and this requires $O(n \cdot t)$ time, where n and t denote the number of billboard slots and the tuples in the trajectory database, respectively. Hence, the time requirement for the execution of the while loop will be of $O(\frac{B_{max}}{w_{min}} \cdot n \cdot t)$. All the remaining statements within the for loop from 5 to 20 will take $O(1)$ time. Hence, the time requirement till Line 20 will be of $O(|\mathcal{X}| \cdot \ell \cdot \frac{B_{max}}{w_{min}} \cdot n \cdot t)$. In the worst case, it may so happen that all the sampled solutions are feasible solutions. In Line 21, we find the best solution among all the feasible solutions in $\mathcal{X}$. It will take $O(|\mathcal{X}|)$. Hence, the time requirement to execute of Algorithm 1 is of $O(|\mathcal{X}| \cdot \ell \cdot \frac{B_{max}}{w_{min}} \cdot n \cdot t)$. In addition space consumed by Algorithm 1 is to store the feasible solutions which will be of $O(|\mathcal{X}| \cdot n)$, to store the sets $S_1, S_2, \dots, S_\ell$ and $\mathbb{S}$ and both of them will take $O(n)$

space. Hence total extra space consumed by $O(|\mathcal{X}| \cdot n + n)$ which is equal to $O(|\mathcal{X}| \cdot n)$. Hence, Theorem 3.6 holds.

THEOREM 3.6. *The sampling-based approach will take $O(|\mathcal{X}| \cdot \ell \cdot \frac{B_{max}}{w_{min}} \cdot n \cdot t)$ time and $O(|\mathcal{X}| \cdot n + n)$ space to execute.*

4 EXPERIMENTAL EVALUATIONS

This section describes the experimental evaluation of the proposed solution approaches. Initially, we start by describing the datasets.

4.1 Dataset Descriptions.

We used two real-world datasets: New York City (NYC)¹ and Los Angeles (LA)², both adopted in previous studies [1, 4? ?]. The NYC dataset contains 227,428 check-ins (Apr 2012–Feb 2013), while the LA dataset includes 74,170 check-ins across 15 streets. Billboard data for both cities is sourced from LAMAR³, comprising 1,031,040 slots in NYC and 2,135,520 in LA. In Table 1 $|\mathcal{T}|$, $|\mathcal{U}|$, and Avg_{dist} denote the number of trajectories, the number of unique users, and the average distance between trajectories. Similarly, in Table 2 $|\mathcal{B}|$, $|\mathcal{BS}|$, and Avg_{dist} denote the number of billboards, the number of billboard slots, and the average distance between the billboards placed across the city. The descriptions of the datasets are summarized in Table 1 and Table 2.

Dataset	$ \mathcal{T} $	$ \mathcal{U} $	Avg_{dist}
NYC	227428	1083	3.12 km
LA	74170	2000	0.61 km

Table 1: Trajectory Datasets

Dataset	$ \mathcal{B} $	$ \mathcal{BS} $	Avg_{dist}
NYC	716	1031040	15.07 km
LA	1483	2135520	10.26 km

Table 2: Billboard Datasets**4.2 Key Parameters.**

All key parameters used in our experiments are summarized in Table 3. The default parameter settings are highlighted in bold.

Table 3: Key Parameters

Parameter	Values
α	40%, 60%, 80%, 100%, 120%
β	1%, 2%, 5%, 10%, 20%
$ \mathcal{P} $	100, 50, 20, 10, 5
ϵ	0.01, 0.05, 0.1 , 0.15, 0.2
λ	25m, 50m, 100m , 125m, 150m

Advertiser. Given an advertiser $\mathcal{A}$, with a set of products $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$, the advertiser can be represented as $(\mathcal{A}, p_i, \sigma_i, \mathcal{B})$, where p_i is the i^{th} product, σ_i and $\mathcal{B}$ is the corresponding influence demand and budget.

Demand Supply Ratio. The ratio $\alpha = \sigma^{\mathcal{P}} / \sigma^*$ represents the proportion of global influence demand ($\sigma^{\mathcal{P}} = \sum_{i=1}^n \sigma_i$) to total influence supply ($\sigma^* = \sum_{b \in \mathcal{BS}} I(b)$). We evaluate five values of θ : 40%, 60%, 80%, 100%, and 120%.

Advertiser Individual Demand. The ratio $\beta = \sigma^{\mathcal{P}''} / \sigma^*$ denotes the ratio of average individual demand to total influence supply, where $\sigma^{\mathcal{P}''} = \sigma^{\mathcal{P}} / |\mathcal{P}|$ is the average influence demand per product. This value β adjusts the demand of individual products.

¹<https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

²<https://github.com/Ibtihal-Alablani>

³<http://www.lamar.com/InventoryBrowser>

Product Demand. The demand for each product is generated as $\sigma = \lfloor \omega \cdot \sigma^* \cdot \beta \rfloor$, where ω is a random factor between 0.8 and 1.2 to simulate varying product payments.

Billboard Cost. The outdoor advertising companies like LAMAR and JCDcuax do not disclose the actual cost of renting a billboard slot. In existing studies [1, 9, 10, 12, 21], the cost of a billboard slot is proportional to its influence. Following this, we calculated the cost of a billboard slot ‘bs’ using $\lfloor \delta \cdot \frac{I(bs)}{10} \rfloor$, where $\delta \in [0.8, 1.1]$ and $I(bs)$ denotes the influence of billboard slot ‘bs’.

Advertiser Budget. Following existing studies [3, 4, 10, 11, 22], we set each product’s payment proportional to its influence demand: $\mathcal{L}_i = \lfloor \eta \cdot \sigma_i \rfloor$, where η is a random factor in [0.9, 1.1] to model payment variations. The total advertiser budget is then $\mathcal{B} = \sum_{i=1}^n \mathcal{L}_i$.

Environment Setup. All Python code is executed on an HP Z4 workstation with 64 GB RAM and a 3.50 GHz Intel Xeon(R) CPU.

4.3 Baseline Methodologies.

Random Allocation (RA). In this approach, billboard slots are selected uniformly at random and assigned to products until both the influence demand and budget constraints are met.

Top-k Allocation. In this approach, billboard slots are first sorted by their influence values. The highest-influence slots are allotted to the products sequentially until the influence requirements and budget constraints are met.

4.4 Goals of Our Experimentation

- **RQ1:** Varying α, β , how do the satisfied product change?
- **RQ2:** Varying α, β , how do the total budget is utilized?
- **RQ3:** Varying α, β , how do the computational time change?
- **RQ4:** Varying ϵ and λ , how do the influence quality change?

4.5 Results and Discussions

Varying α and β Vs. Influence. From Figure 1(a,b) and Figure 2(a,b), it is clear that with the increase of demand supply ratio (α), within the budget of each product, the influence value increases. When the α value is 100% or 120%, the influence demand is higher for each product and requires a larger number of slots to satisfy. In the case of the ‘RA’ approach, in the NYC dataset, it satisfies the influence demand of 100 products with the lowest cost compared to the ‘Random’ and ‘Top-K’ approaches. In the case of the ‘BCA’ approach, the influence gain is higher compared to the baseline methods like ‘Random’ and ‘Top-K’; however, it is less than the ‘RA’ approach. This happens because in the ‘BCA’ approach, we find a common set of slots that can satisfy all the products. The difference in influence is 20% to 25% more in the NYC dataset and 8% to 10% more than the baselines in the LA dataset. Now, in case of the LA dataset in Figure 2, similar observations were found for all the proposed and baseline methods. In the experimental setup, when $\alpha \geq 100\%$ and $\beta \geq 10\%$, the influence demand is greater than the influence supply in those cases, and the influence demand of all the products is not satisfied. In these cases, ‘BCA’ methods perform better than the baselines; however, the ‘RA’ approach fails to provide a feasible solution.

Varying α and β Vs. No. of Slots. With an increase in α and β , the number of slots allocated for each product increases due to an increase in the demand for influence, as shown in Figure 1(c,d) and Figure 2 (c,d). Among the baseline approaches, the ‘Top-k’ allocates fewer slots than the ‘Random’ approach. As in the ‘Top-k’, influential slots are stored in descending order and allocated to each product. The ‘BCA’ approach allocates more slots than the ‘RA’ and baseline methods, as it generates a single set of slots to satisfy all the product’s demand. However, ‘RA’ allocates a reasonable number of slots to satisfy the influence demand for all the advertisers.

Efficiency Test. From Figure 1(e,f) and Figure 2(e,f), we have three main observations. First, the running time of ‘RA’ is longer than that of the ‘BCA’ and baseline methods because ‘RA’ considers a large number of permutations of slots and products. Second, with increasing values of α and β , the computational time for all methods increases. Among the baseline approaches, ‘Top-k’ takes more time compared to the ‘Random’ approach. Third, the ‘BCA’ approach takes more time compared to the baselines; however, it is less than the ‘RA’ approach. With an increase of α and β , the computation time change is less because the initially selected slots using continuous greedy are significant, such that they satisfy most of the product’s influence demand. So, further allocation using greedy based on marginal gain requires less time to compute.

Product Vs. Budget. From Figure 1(g,h) and Figure 2(g,h), we have two observations. First, with increasing demand supply ratio (α) and average individual demand (β), the influence of demand for each product increases, and a larger number of slots are allocated. Hence, the allocation cost of products increases. Second, in terms of cost, ‘RA’ outperforms ‘Random’, ‘Top-k’, and ‘BCA’ methods while providing maximum influence. ‘RA’ uses the minimum cost to satisfy the influence demand for all the products.

Varying α, β Vs. Products. We have experimented with different values of α, β , and the number of products to show that different cases occur due to the varying demand of the products. Although we have reported the experimental results of two settings: $\alpha = 40\%, 60\%, 80\%, 100\%, 120\%$ and $\beta = 1\%, 2\%$, and the number of products ($|\ell|$) is 100 and 50. However, the other cases are (1) when varying $\alpha, \beta = 5\%$, and $|\ell| = 20$ (2) when varying $\alpha, \beta = 10\%$, and $|\ell| = 10$ (3) when varying $\alpha, \beta = 20\%$, and $|\ell| = 5$. In all these three cases, some of the product’s influence demand is unsatisfied, and the ‘RA’ methods will not provide any feasible solution due to a higher demand than supply. However, the ‘BCA’, ‘Top-k’, and ‘Random’ provide solutions where some products are unsatisfied, and among them, ‘BCA’ outperforms baseline methods.

Additional Discussion. The additional parameters used in our experiments are the distance parameter λ and the approximation parameter ϵ . The λ determines the distance from which a billboard slot can influence a number of trajectories. The ϵ controls the accuracy of approximation. The smaller $\epsilon \in (0, 1)$ provides a better approximation, but increases runtime. In our experiments, we set $\epsilon = 0.1$ and $\lambda = 100m$ as the default setting. Due to space limitations, we have only reported the results for the default settings.

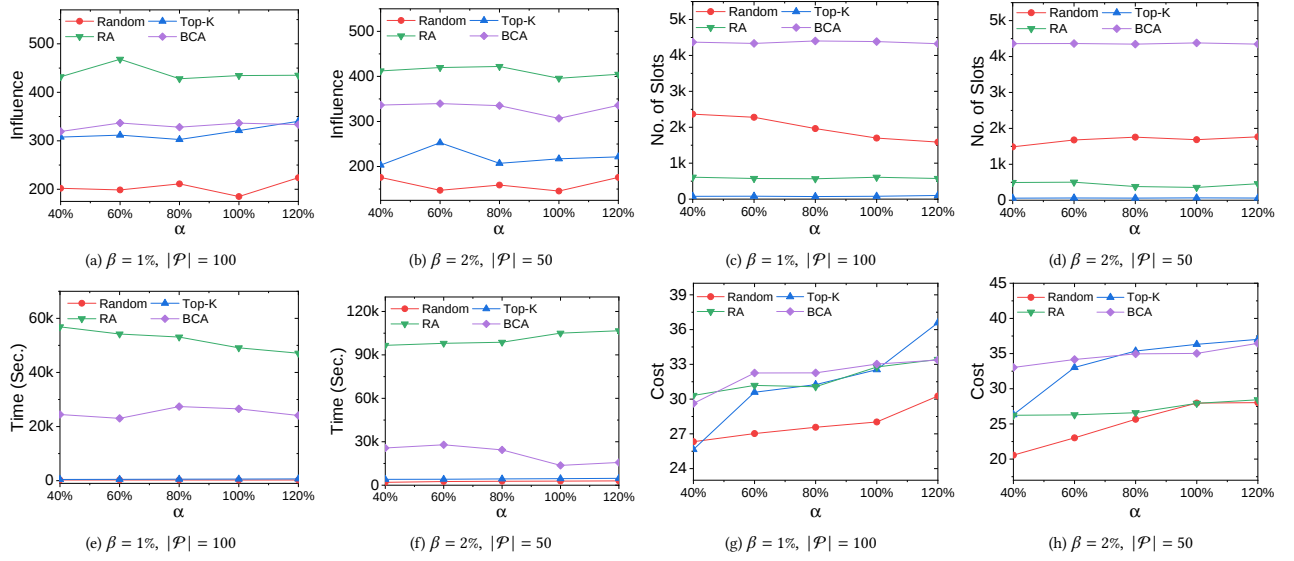


Figure 1: Plots for NYC dataset: (a, b) α vs. Influence, (c, d) α vs. No. of Slots, (e, f) α vs. Time, and (g, h) α vs. Cost

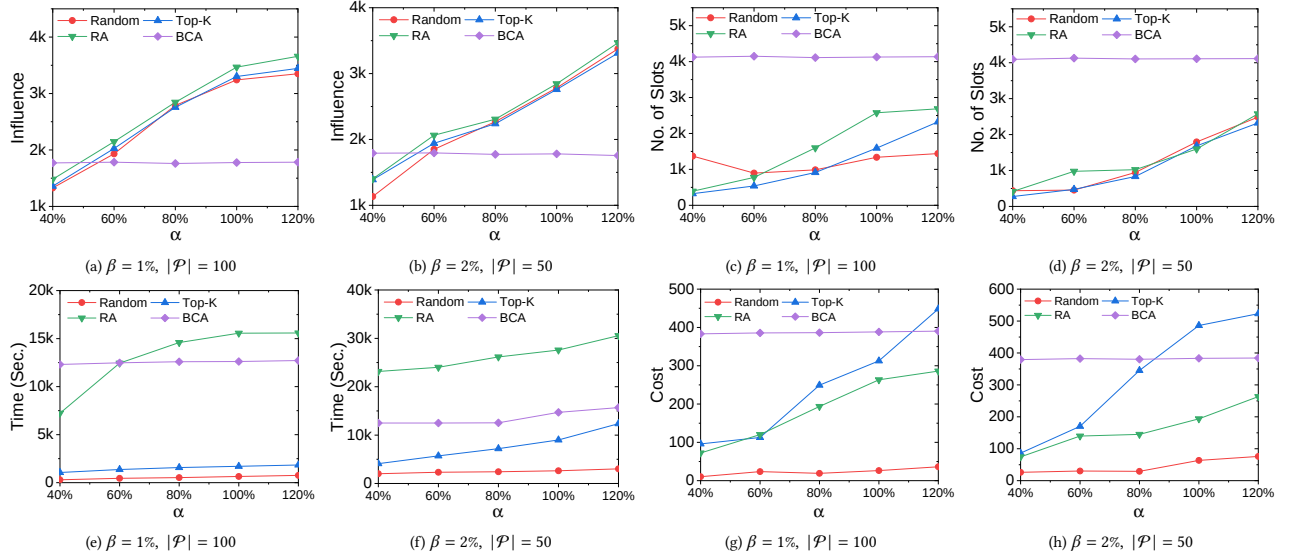


Figure 2: Plots for LA dataset: (a, b) α vs. Influence, (c, d) α vs. No. of Slots, (e, f) α vs. Time, and (g, h) α vs. Cost

5 CONCLUDING REMARKS

In this paper, we have studied the Multi-Product Influence Maximization Problem, where the goal is to maximize the influence for multiple products. As different trajectory user has affinity towards different products, it is a challenging problem. In this paper, we study this problem in two variants. In the first case, we select a set of k slots for maximizing the aggregated influence, whereas in the second case, we are interested in selecting a non-intersecting set of slots depending upon the given budget. We modeled this problem as a submodular multi-cover problem and its generalized version. We have adopted the bi-criteria approximation algorithm

for solving the first variant, and for the second variant, we have proposed a sampling-based randomized algorithm. The experimental evaluation with real-world datasets has demonstrated the superior performance of the proposed solution approaches.

REFERENCES

- [1] Dildar Ali, Suman Banerjee, and Yamuna Prasad. 2022. Influential billboard slot selection using pruned submodularity graph. In *International Conference on Advanced Data Mining and Applications*. Springer, 216–230.
- [2] Dildar Ali, Suman Banerjee, and Yamuna Prasad. 2024. Influential Billboard Slot Selection Under Zonal Influence Constraint. In *European Conference on Advances in Databases and Information Systems*. Springer, 93–106.
- [3] Dildar Ali, Suman Banerjee, and Yamuna Prasad. 2024. Minimizing regret in billboard advertisement under zonal influence constraint. *arXiv preprint*

- arXiv:2402.01294 (2024).
- [4] Dildar Ali, Suman Banerjee, and Yamuna Prasad. 2024. Regret Minimization in Billboard Advertisement under Zonal Influence Constraint. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*. 329–336.
 - [5] Dildar Ali, Suman Banerjee, and Yamuna Prasad. 2024. Toward regret-free slot allocation in billboard advertisement. *International Journal of Data Science and Analytics* (2024), 1–25.
 - [6] Dildar Ali, Suman Banerjee, and Yamuna Prasad. 2025. Influential billboard slot selection using spatial clustering and pruned submodularity graph. *Data Science and Engineering* (2025), 1–22.
 - [7] Dildar Ali, Suman Banerjee, and Yamuna Prasad. 2025. Influential slot and tag selection in billboard advertisement. In *International Conference on Database and Expert Systems Applications*. Springer, 276–290.
 - [8] Dildar Ali, Ankit Kumar Bhagat, Suman Banerjee, and Yamuna Prasad. 2023. Efficient algorithms for regret minimization in billboard advertisement (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 16148–16149.
 - [9] Dildar Ali, Harishchandra Kumar, Suman Banerjee, and Yamuna Prasad. 2024. An effective tag assignment approach for billboard advertisement. In *International Conference on Web Information Systems Engineering*. Springer, 159–173.
 - [10] Cigdem Aslay, Wei Lu3 Francesco Bonchi, Amit Goyal, and Laks VS Lakshmanan. 2015. Viral Marketing Meets Social Advertising: Ad Allocation with Minimum Regret. *Proceedings of the VLDB Endowment* 8, 7 (2015).
 - [11] Cigdem Aslay, Francesco Bonchi Laks VS Lakshmanan, and Wei Lu. 2017. Revenue Maximization in Incentivized Social Advertising. *Proceedings of the VLDB Endowment* 10, 11 (2017).
 - [12] Suman Banerjee, Bithika Pal, and Mamata Jenamani. 2020. Budgeted influence maximization with tags in social networks. In *Web Information Systems Engineering–WISE 2020: 21st International Conference, Amsterdam, The Netherlands, October 20–24, 2020, Proceedings, Part I* 21. Springer, 141–152.
 - [13] Grui Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. 2007. Maximizing a submodular set function subject to a matroid constraint. In *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 182–196.
 - [14] Chandra Chekuri, Tanmay Inamdar, Kent Quanrud, Kasturi Varadarajan, and Zhao Zhang. 2022. Algorithms for covering multiple submodular constraints and applications. *Journal of combinatorial optimization* 44, 2 (2022), 979–1010.
 - [15] Moran Feldman, Joseph Naor, and Roy Schwartz. 2011. A unified continuous greedy algorithm for submodular maximization. In *2011 IEEE 52nd annual symposium on foundations of computer science*. IEEE, 570–579.
 - [16] Wassily Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association* 58, 301 (1963), 13–30.
 - [17] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 137–146.
 - [18] Jan Vondrák. 2007. Submodularity in combinatorial optimization. (2007).
 - [19] Liang Wang, Zhiwen Yu, Bin Guo, Dingqi Yang, Lianbo Ma, Zhidan Liu, and Fei Xiong. 2022. Data-driven targeted advertising recommendation system for outdoor billboard. *ACM Transactions on Intelligent Systems and Technology (TIIST)* 13, 2 (2022), 1–23.
 - [20] Ping Zhang, Zhifeng Bao, Yuchen Li, Guoliang Li, Yipeng Zhang, and Zhiyong Peng. 2020. Towards an optimal outdoor advertising placement: When a budget constraint meets moving trajectories. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 5 (2020), 1–32.
 - [21] Yipeng ZHANG, Yuchen LI, Zhifeng BAO, Songsong MO, and Ping ZHANG. 2019. Optimizing impression counts for outdoor advertising.(2019). In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, Alaska*. 4–8.
 - [22] Yipeng Zhang, Yuchen Li, Zhifeng Bao, Baihua Zheng, and HV Jagadish. 2021. Minimizing the regret of an influence provider. In *Proceedings of the 2021 International Conference on Management of Data*. 2115–2127.