Finding Effective Ad Allocations: How to Exploit User History

Matteo Castiglioni
Politecnico di Milano
Milan, Italy
matteo.castiglioni@polimi.it

Alberto Latino
Politecnico di Milano
Milan, Italy
alberto.latino@mail.polimi.it

Alberto Marchesi
Politecnico di Milano
Milan, Italy
alberto.marchesi@polimi.it

Giulia Romano
Politecnico di Milano
Milan, Italy
giulia.romano@polimi.it

Nicola Gatti
Politecnico di Milano
Milan, Italy
nicola.gatti@polimi.it

Chokha Palayamkottai
Locify
Miami, USA
chokha@locify.com

ABSTRACT

A primary source of revenue for web platforms is digital advertising. Platforms typically maximize the effectiveness of advertising campaigns by exploiting user features (*i.e.*, targeted advertising). However, performance can be further improved by leveraging user navigation history. In particular, the advent of new augmented reality platforms encourages users to spend a considerable amount of time in the same virtual environment, opening up the challenge of determining *which ads* to display and at *which time* of their experience. In this paper, we initiate the study of *history-dependent* advertising by providing a user model and optimized ad allocation algorithms. Our model assumes that users move through a series of *scenes* where they are exposed to ads. The performance of an ad may be influenced by various factors, such as the features of the scene in which it is displayed, the externalities of previously observed ads and the possibility that a user has already purchased the promoted product. We analyze the computational complexity of finding an optimal ad allocation for several model flavors and provide practical approximation algorithms with tight theoretical guarantees. We also discuss under which conditions our approximation algorithms are *monotone* according to Myerson's definition, thus leading to truthful auction mechanisms.

KEYWORDS

Ad allocation, Algorithmic advertising, mechanism design

ACM Reference Format:
Matteo Castiglioni, Alberto Latino, Alberto Marchesi, Giulia Romano, Nicola Gatti, and Chokha Palayamkottai . 2024. Finding Effective Ad Allocations: How to Exploit User History. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

In the last decade, artificial intelligence has been one of the main drivers of growth for digital markets. The use of AI tools in digital advertising has become increasingly common, opening up new opportunities that were previously unavailable [6, 13]. Some of the advantages over traditional advertising channels are the possibility of profiling a user from behavioral data [5], targeting ads in a precise way [14] and running auction mechanisms to maximize specific objective functions associated with revenue [15]. AI tools can efficiently optimize these processes managing the vast amount of data and the numerous parameters provided by platforms. Moreover, the continuous innovation of web platforms provides further opportunities to extract value from optimizing these parameters exploiting the structure of new advertising settings. For instance, there is a broad agreement that the advent of metaverse platforms may revolutionize marketing in the next decade [20]. The peculiarity of virtual reality platforms is to offer the user a real-time immersive experience, enabling the possibility to exploit a plethora of additional information on users' behavior to dramatically increase the effectiveness of advertisements. In this environment, it becomes possible not only to track the ads sequentially shown to the user as they move through different locations but also to *control* which ads to display at each stage of their path. It is worth noting that, while cookies can be used to gather information on user navigation history as they browse the web, it is not always possible for a single provider to manage the allocation of all the slots observed by the user. On the contrary, new self-contained environments such as augmented reality platforms enable the opportunity to implement history-dependent advertising, since user experience is not as fragmented as on the web.

To the best of our knowledge, the problem of deciding *which ads* to display to the users and at *which time* of their experience is unexplored so far. In this paper, we propose a user model which extends those currently adopted for search and mobile advertising. In particular, we assume that users traverse several *scenes*, which could be, for example, locations in virtual realities (*i.e.*, sports events, concerts, job meetings, tourist sites, lectures, and conferences). During the traversal, users are targeted with multiple ads, whose performance, usually referred to as *quality*, may depend on the specific scene in which they are displayed. For example, an ad may attract the user attention differently if shown during a sport event or a concert. Furthermore, the ads may be subject to externalities due to their sequential display. In particular, displaying an ad in a scene may raise *negative forward externalities* to other ads shown in future scenes. This happens, for instance, when two ads promoting

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

products that are strategic substitutes are displayed sequentially, as shown by Deng and Pekec [4]. Finally, real-world experiments show that it is unlikely that a user recalls every ad seen in the past [1, 2]. Thus, we assume that the users' behavior is affected only by the ads displayed in the last k scenes, where k is a parameter modeling users' memory, whose value can be estimated from historical data.

Intuitively, the activities of a user in a self-contained platform could be modeled as a graph of scenes with stochastic transitions. An ad allocation defines which ad to display in every graph scene. Despite the graph provides a compact representation, it would require to display the same ad every time the user traverses the same scene independently of the history of previously-observed ads. This static ad allocation may generate frustration and reduce the user's attention. Moreover, disregarding previously-observed ads could lead to heavy negative externalities. These limits push the need for *(non-Markovian) history-dependent policies* whose natural representation is a tree. Thus, we expand the graph by generating a tree composed of the potential paths a user can follow starting from an initial scene and then finding an allocation of ads such that every node of the tree is treated as a different scene. Interestingly, in practical applications with a huge number of paths, the tree size can be bounded by resorting to Monte Carlo sampling techniques.

Related Works. Several works investigate ad allocation problems in related scenarios such as web and mobile advertising and provide attention models describing how users observe the slots in which ads are displayed. A user model needs to address the tradeoff between a sufficiently accurate description of the user behavior and the possibility of designing allocation algorithms running in polynomial time to scale up to real-world applications. The seminal model for sponsored search auctions, called *cascade* and proposed by Kempe and Mahdian [14], assumes that users scan the slots sequentially. The authors also propose some algorithms for special cases of their model, while Farina and Gatti [7] provide practical algorithms for the general case. Fotakis et al. [9] and Gatti et al. [11] propose detailed models incorporating negative externalities between ads. This higher degree of model accuracy comes at the cost of higher computational complexity, as no constant approximation algorithm is possible. Gatti et al. [10] adopt a similar approach in the case of mobile advertising, where the user moves in a geographical area. The aforementioned works assume that the model parameters values are known. In real-world settings, the proposed ad allocation algorithms can be paired with multi-armed bandit techniques to perform an online estimation of those values [5].

Original Contributions. We study the computational complexity of finding optimal ad allocations and provide approximation algorithms with tight theoretical guarantees. We deliver results for different model flavors: qualities may or may not depend on the scene, and externalities may or may not be present (see Table 1). Interestingly, allowing the ads qualities to be scene dependent makes the problem APX-Complete. To address this case, we provide a polynomial-time algorithm with an approximation factor of $1 - 1/e$. Moreover, introducing externalities among the ads makes the problem Poly-APX-Complete, therefore, we provide a polynomial-time algorithm with an approximation factor of $1/(k+1)$, which is tight and shows that the problem is in APX when k is fixed. In particular, the algorithm returns an optimal solution when $k = 0$, *i.e.*, when user behavior does not depend on the ads observed in the previous scenes. Similar upper and lower complexity bounds hold when adopting the model in the general case, *i.e.*, with externalities and scene-dependent qualities. Finally, we show that allocation algorithms disregarding basic user features and Markovian solutions directly defined on the graph provide approximations arbitrarily worse than ours. This suggests that our model is effective in practice, even when combined with simple greedy algorithms that scale up to real-world instances. We also discuss under which conditions our algorithms define *weakly monotone* allocation functions in the sense of Myerson, thus leading to truthful auctions. In particular, we show that our algorithms are weakly monotone, according to Myerson's definition, when the qualities are scene-independent.

Table 1: Summary of our computational complexity results. `SI` = scene-independent quality, `SD` = scene-dependent quality, `NE` = no externalities, `E` = with externalities. The fourth column refers to Myerson's weak monotonicity.

Allocation problem	Complexity	Best apx. ratio	Best weakly monotone apx. ratio
SI-NE	Poly	1	1
SD-NE	APX-Complete	$1-1/e$	—
SI-E	Poly-APX-Complete	$1/(k+1)$	$1/(k+1)$
SD-E	Poly-APX-Complete	$(1-1/e)/(k+1)$	—

2 ADVERTISING MODEL

We introduce the following advertising model. We assume that an user moves through a graph with stochastic transitions in which nodes correspond to scenes in a virtual environment. To capture history-dependent policies, the graph is expanded as follows.

Scenes Tree and User Transitions. Given a graph, a starting scene, and a time horizon, we generate the tree describing the potential paths the user traverses according to a probability distribution over the successors of every scene. We remark that, when adopting history-dependent ad-allocation policies, different ads can be displayed at different traversals of the same scene of the graph depending on the specific history. To better capture history-dependent policies, we treat all the tree nodes corresponding to the same scene in the graph as different scenes. Formally, the *tree of scenes* is represented with $T = (S, \rho)$, where S is the *set of scenes* in which a user can be, $s \in S$ is a scene, and $\rho : S \to \mathcal{P}(S)$ is the *successor function* taking as input a scene $s \in S$ and returning the subset $\rho(s)$ of S composed of all the scenes that are immediate successors of s in the tree; $\mathcal{P}(S)$ denotes the powerset of S. We say that scenes s such that $\rho(s) = \emptyset$ are *terminal.* We denote with $\pi_{s,s'} \in [0, 1]$, where $s \in S, s' \in \rho(s)$, the *transition probability* that a user in scene s moves to immediate successor scene s'. Furthermore, for every non-terminal scene $s \in S$, it holds $\sum_{s' \in \rho(s)} \pi_{s,s'} = 1$. We can model a user who leaves the platform with a non-null probability from scene s by using an immediate successor of s that is terminal. In the case of search advertising, this corresponds to stop observing further slots. We denote with $\boldsymbol{\sigma}$ a generic *ordered sequence* of scenes such that σ_i is the i-th scene of $\boldsymbol{\sigma}$. In particular, $\boldsymbol{\sigma}^s$ is the sequence of scenes from the root node to scene $s \in S$, with $|\boldsymbol{\sigma}^s|$ the length of $\boldsymbol{\sigma}^s$, and σ^s_i is the i-th element of $\boldsymbol{\sigma}^s$, where $i \in [\,|\boldsymbol{\sigma}^s|\,]$.[1] Hence, for every $s \in S$, the root scene corresponds to σ^s_1 and scene s to $\sigma_{|\boldsymbol{\sigma}^s|}$. The *reach probability* of s is $\Pi^s = \prod_{i=1}^{|\boldsymbol{\sigma}^s|-1} \pi_{\sigma^s_i, \sigma^s_{i+1}}$, stating the probability that a user reaches s starting from root σ^s_1.

Ads, Qualities, and Externalities. We denote with A the *set of ads* and with $a \in A$ an ad. For simplicity, we assume that at most one ad can be displayed in every scene. In particular, we denote with $x : S \to A \cup \{a_\emptyset\}$ the *allocation function* taking as input scene $s \in S$ and returning ad $a \in A$ or $a_\emptyset$ allocated to scene s. Ad $a_\emptyset$ is fictitious, meaning that no ad is allocated in that scene. Every ad $a \in A$ allocated in scene $s \in S$ is characterized by a quality $q_{a,s} \in [0, 1]$, that is the user's conversion probability conditioned to the event that scene s has been reached by the user and that no other ad has been displayed before s. For the sake of presentation, whenever we focus on settings in which the quality is scene-independent, we use q_a in place of $q_{a,s}$. Since an empty slot produces no conversion, we set $q_{a_\emptyset,s} = 0$, for every $s \in S$. Furthermore, ads are subject to *forward externalities*, such that the display of ad a allocated in scene s affects the quality of ad a' allocated in scene s' when s precedes s' in the tree. Formally, we model such an externality with $\gamma_{a,a'} \in [0, 1]$, where $a, a' \in A$ and a is allocated in a scene of the tree preceding (immediately or not) the scene where a' is allocated. We assume that $\gamma_{a,a'} \leq 1$ for every $a \neq a' \in A$, while $\gamma_{a,a} = 1$ for every $a \in A$. Notice that, when $\gamma_{a,a'} < 1$, the externality is *negative*, meaning that the display of a before a' negatively affects the quality of a', while, when $\gamma_{a,a'} = 1$, the externality is *neutral*, meaning that the display of a before a' does not affect the quality of a'. By convention, the absence of ads in a scene does not introduce any externalities and therefore $\gamma_{a_\emptyset,a'} = 1$ for every $a' \in A$. In the no-externalities settings, we assume $\gamma_{a,a'} = 1$ for every $a, a' \in A$.

Furthermore, we assume that the user may forget the past ads, thus alleviating the negative effects due to externalities. More precisely, we assume that the user's behavior only depends on the ads seen in the previous $k \in \mathbb{N}$ scenes. Notice that, when $k = 0$, the user forgets every previous ad, while setting $k = \infty$ implies that the user perfectly recalls all the observed ads. The *externality* to which ad a in scene s is subject to is $\Gamma(x, s) = \prod_{i=\max\{1,|\boldsymbol{\sigma}^s|-k\}}^{|\boldsymbol{\sigma}^s|-1} \gamma_{x(\sigma^s_i),x(s)}$ and depends on all the ads displayed in the k scenes preceding s in the sequence $\boldsymbol{\sigma}^s$, whose are $\min\{k, |\,\boldsymbol{\sigma}^s| - 1\}$. The probability that a user converts on an ad a in scene s conditioned to the reach of scene s is $\Gamma(x, s)\, q_{s,a}$. This holds whenever ad a is not displayed in scenes preceding s, since we handle differently the case in which an ad is displayed multiple times along the same path.

Many works on search advertising allow an ad to be displayed only once in the allocation *e.g.*, [14]. However, we think that a more accurate model would exploit the opportunity of showing the same ad multiple times, as it happens in real-world advertising scenarios, since users could convert after having observed the ad several times. Formally, we assume that, if a user converts on ad a in scene s, then that user will never convert again on a when displayed in a scene s' following s. On the other hand, if a user does not convert on ad a in scene s, then the user can convert on the same a in a scene s' following s. This assumption requires adjusting the quality of an ad when displayed multiple times along a single path. In particular, we denote with $H(x, s) \subseteq S$ the subset of scenes s' along sequence $\boldsymbol{\sigma}^s$ in which ad $a = x(s) = x(s')$ is allocated, excluded scene s. We define $\Xi(x, s) = \prod_{s' \in H(x,s)} (1 - \Gamma(x, s')\, q_{x(s'),s'})$ as the probability that the user never converts on ad $a = x(s)$ when allocated in scenes s' strictly before scene s conditioned to the reach of s'. Finally, we denote with $\tilde{q}(x, s) = \Gamma(x, s)\, q_{x(s),s}\, \Xi(x, s)$ the *adjusted quality* of the ad allocated in s given the ads allocated in the previous scenes. Thus, given an ad a allocated in scene s, its *conversion rate* is $\Pi^s\, \tilde{q}(x, s)$. Moreover, its expected value is $\Pi^s\, \tilde{q}(x, s)\, \theta_{x(s)}$, where $\theta_a \in [0, 1]$ is the *value per conversion* of ad a. We can finally compute the *allocation expected value* of x as $\sum_{s \in S} \left(\Pi^s\, \tilde{q}(x, s)\, \theta_{x(s)}\right)$.

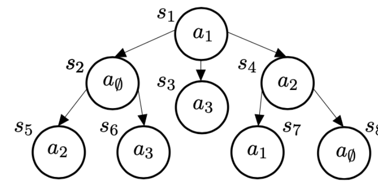

Figure 1: Example of the tree of scenes.

In this paper, we study the ad allocation problem under different flavors of our user model. As in Table 1, the acronyms SI and SD are used to denote the cases where qualities are scene-independent and scene-dependent, respectively. Similarly, NE and E denote settings with no externalities and with externalities among ads, respectively. We provide an example to clarify the functioning of our model.

Example 2.1. Figure 1 shows a setting described by a tree with set of scenes $S = \{s_1, \ldots, s_8\}$ and set of ads $A = \{a_1, a_2, a_3\} \cup a_\emptyset$. The quality of the ads is $q_{a,s} = 0.1$ for all $a \in A$ and $s \in S$, the externalities are $\gamma_{a_1,a_2} = \gamma_{a_1,a_3} = \gamma_{a_1,a_3} = 0.8$, and the values per conversion are $\theta_{a_1} = 0.5$, $\theta_{a_2} = 0.6$ and $\theta_{a_3} = 0.7$. The transition probabilities are $\pi_{s_1,s_2} = \pi_{s_2,s_6} = \pi_{s_4,s_7} = 0.7$, $\pi_{s_1,s_3} = 0.1$, $\pi_{s_1,s_4} = 0.2$, $\pi_{s_2,s_5} = \pi_{s_4,s_8} = 0.3$. Moreover, we set $k \geq 2$. Consider, for instance, scene s_7: the total externality is $\Gamma(x, s_7) = \gamma_{a_1,a_1}\, \gamma_{a_1,a_2} = 0.8$, the adjusted quality is $\tilde{q}(x, s_7) = \Gamma(x, s_7)\, q_{a_1,s_7}\, \Xi(x, s_7) = 0.072$, the expected value is $\Pi^{s_7}\, \tilde{q}(a_1, s_7)\, \theta_{a_1} = 5 \cdot 10^{-3}$. The allocation expected value is $\sum_{s \in S} \left(\Pi^s\, \tilde{q}(x, s)\, \theta_{x(s)}\right) = 107.76 \cdot 10^{-3}$. Notice that if $k = 1$, the value increases to $117.14 \cdot 10^{-3}$ as the negative effect due to the externalities is mitigated further.

Myerson's Weak Monotonicity. When designing allocation algorithms in the following sections, we investigate whether they satisfy Myerson's weak monotonicity property. Indeed, since our history-dependent advertising model is a single-parameter (*i.e.*, θ_a) linear environment, Myerson's weak monotonicity is *necessary* and *sufficient* for the design of a truthful mechanism in dominant strategies [16]. In our case, the property reads as follows.

Definition 2.2. In the single-parameter environment of the history-dependent advertising model, an allocation mechanism M that maps a type profile $(\theta_a)_{a \in A}$ to an allocation x is weakly monotone if for every ad $\hat{a} \in A$ and types $\theta_{a'}$ of the other ads $a' \in A \setminus \{\hat{a}\}$, the allocation mechanism M is such that the term $\sum_{s \in S: x^{\theta_{\hat{a}}}(s) = \hat{a}} \left(\Pi^s\, \tilde{q}(x^{\theta_{\hat{a}}}, s)\right)$ is non-decreasing in $\theta_{\hat{a}}$, where $x^{\theta_{\hat{a}}} = \mathsf{M}((\theta_a)_{a \in A})$ is the allocation returned by the mechanism with type profile $(\theta_a)_{a \in A}$.

[1] We denote with $[n]$ the set $\{1, \ldots, n\}$, where $n \in \mathbb{N}$.

Algorithm 1 Greedy

1: **Inputs**: set of scenes S, set of ads A
2: Initialize $R \leftarrow S$, $x(s) \leftarrow a_\emptyset \ \forall s \in S$
3: **while** $R \neq \emptyset$ **do**
4: $\quad (s^*, a^*) \leftarrow \operatorname{argmax}_{s \in R, a \in A} \Pi^s \, \tilde{q}(x, s) \, \theta_a$
$\quad\quad \triangleright$ Ties are broken according to Definition 3.1
5: $\quad x(s^*) \leftarrow a^*$
6: $\quad R \leftarrow R \backslash s^*$
7: **return** x

3 POLY-TIME ALGORITHM FOR AD ALLOCATION IN THE SI-NE SETTING

We focus on the basic SI-NE case with no externalities and scene-independent quality. This case differs from the allocation problem in ad auctions for two reasons: the allocation may be on a tree instead of a line, and an ad can be displayed multiple times along a single path of the tree. We show that we can design a polynomial-time greedy algorithm facing this setting. Furthermore, this algorithm plays a central role when solving more general settings. The pseudocode is reported in Algorithm 1. We denote with R the subset of scenes that have not been filled yet with an ad in $A \cup \{a_\emptyset\}$. The algorithm works iteratively and, at each step, it chooses a scene-ad pair $(s^*, a^*) \in S \times A$ which maximizes the expected value of allocating an ad in an available scene. We define the following tie-breaking rule to identify the unique pair chosen at each iteration (Line 4) among all the possible value-maximizing pairs.

Definition 3.1. Let $\bar{P}$ be the set of pairs $(\bar{s}, \bar{a})$ returned by

$$\operatorname*{argmax}_{s \in S, a \in A} \Pi^s \, \tilde{q}(x, s) \, \theta_a.$$

Whenever $\bar{P}$ is not a singleton, ties are broken by assigning to (s^*, a^*) any pair $(\bar{s}', \bar{a}')$ such that $|\boldsymbol{\sigma}^{\bar{s}'}|$ is the minimum among all $|\boldsymbol{\sigma}^{\bar{s}}|$ where $(\bar{s}, \bar{a}) \in \bar{P}$, for some $\bar{a} \in A$.

After selecting the value-maximizing pair (s^*, a^*), Algorithm 1 allocates ad a^* to scene s^* (Line 5). Then, scene s^* is removed from the set R which contains the available scenes (Line 6). The algorithm iterates until every scene has been filled with one ad. Finally, it returns the allocation function x. The following theorem shows that Algorithm 1 returns an optimal allocation.

Theorem 3.2. *Algorithm 1 computes an optimal solution to the ad allocation problem in the* SI-NE *setting.*

Proof. As a first step, we show that the value of an allocation x can be decomposed into a component for each possible path.

$$\sum_{s \in S} \left(\Pi^s \, \tilde{q}(x, s) \, \theta_{x(s)} \right) = \sum_{s \in S : \rho(s) = \emptyset} \Pi^s V_s(x),$$

where $V_s(x) = \sum_{s' \in \boldsymbol{\sigma}^s} \tilde{q}_{x,s} \theta_{x(s)}$. To see that, it is sufficient to observe that given an s, it holds

$$\begin{aligned} \Pi^s \, \tilde{q}(x, s) \, \theta_{x(s)} &= \left(\sum_{s' : s \in \boldsymbol{\sigma}^{s'}, \rho(s) = \emptyset} \Pi^{s'} \right) \tilde{q}_{x,s} \theta_{x(s)} \\ &= \sum_{s' : s \in \boldsymbol{\sigma}^{s'}, \rho(s) = \emptyset} \Pi^{s'} \tilde{q}_{x,s} \theta_{x(s)} \end{aligned}$$

and hence

$$\begin{aligned} \sum_{s \in S} \left(\Pi^s \, \tilde{q}(x, s) \, \theta_{x(s)} \right) &= \sum_{s \in S} \sum_{s' : s \in \boldsymbol{\sigma}^{s'}, \rho(s) = \emptyset} \Pi^{s'} \tilde{q}_{x,s} \theta_{x(s)} \\ &= \sum_{s \in S : \rho(s) = \emptyset} \Pi^s \sum_{s' \in \boldsymbol{\sigma}^s} \tilde{q}_{x,s} \theta_{x(s)} \\ &= \sum_{s \in S : \rho(s) = \emptyset} V_s \end{aligned}$$

Then, we observe that thanks to the tie breaking rule in Definition 3.1, the algorithm assigns ads to nodes from the top to the bottom of the tree. Suppose by contradiction that Algorithm 1 assigns an ad a to a node s^1 such that there exists a node $s^2 \neq s^1$ in $\boldsymbol{\sigma}^{s^1}$ that is not assigned, *i.e.*, with $x(s^2) = a_\emptyset$. Then, we have that $\Pi^{s^2} \, \tilde{q}(x, s^2) \theta_a \geq \Pi^{s^1} \, \tilde{q}(x, s^1) \theta_a$ and by the tie breaking rule the ad is assigned to node s^2. Let x^* be the allocation returned by Algorithm 1. Moreover, given a node s, let x' be a different allocation with $x'(s') = x^*(s')$ for all s' that are predecessors of s. Then, the assignment rule in Line 4 of the algorithm implies that

$$\tilde{q}(x^*, s) \, \theta_{x^*(s)} \geq \tilde{q}(x', s) \, \theta_{x'(s)}, \tag{1}$$

where the inequalities follows since the value of assigning any ad to s does not change from the partial allocation x considered by the algorithm and the final allocation x^* (and x') since all the scenes that precede s have already been assigned. Let x^* be the allocation returned by Algorithm 1. We show that this allocation is optimal for each possible path. Formally, given a terminal node $\bar{s}$, *i.e.*, such that $\rho(\bar{s}) = \emptyset$, and an optimal allocation x_0 for the path that terminates in $\bar{s}$, *i.e.*, $x_0 \in \operatorname{argmax}_x V_{\bar{s}}(x)$, we show that $V_{\bar{s}}(x^*) \geq V_{\bar{s}}(x_0)$. This is sufficient to prove the theorem since it implies

$$\begin{aligned} \sum_{s \in S} \left(\Pi^s \, \tilde{q}(x^*, s) \, \theta_{x^*(s)} \right) &= \sum_{s \in S : \rho(s) = \emptyset} \Pi^s V_s(x) \\ &\geq \sum_{s \in S : \rho(s) = \emptyset} \Pi^s \max_x V_s(x) \\ &\geq \max_x \sum_{s \in S : \rho(s) = \emptyset} \Pi^s V_s(x) \\ &= \max_x \sum_{s \in S} \left(\Pi^s \, \tilde{q}(x, s) \, \theta_{x(s)} \right) \end{aligned}$$

Given a terminal node $\bar{s}$, let x_0 be the optimal allocation for the path terminating in $\bar{s}$. We show how to modify iteratively x_0 into x^* without decreasing the value of the allocation. This directly implies that $V_{\bar{s}}(x^*) \geq V_{\bar{s}}(x_0)$ and the optimality of x^*. We iterate over all the $i \in \{1, \ldots, |\, \boldsymbol{\sigma}^{\bar{s}} \,|\}$ and for each i, we build an allocation x_i such that the expected value of x_i is at least the expected value of x_{i-1}. Moreover, the procedure guarantees that for each i it holds $x_i(\sigma_j^{\bar{s}}) = x^*(\sigma_j^{\bar{s}})$ for all $j \leq i$, implying $x_{|S|} = x^*$. The procedure works as follows. We iterate over all the i and given an i we consider three cases. Let S_i be the set of scene s in $\boldsymbol{\sigma}^{\bar{s}} \setminus \boldsymbol{\sigma}^{s_i}$ such that $x_{i-1}(s) = x^*(s_i)$.

Case 1. Suppose that $x^*(\sigma_i^{\bar{s}}) = x_{i-1}(\sigma_i^{\bar{s}})$. Then, setting $x_i = x_{i-1}$ we trivially satisfy the required conditions.

Case 2. Suppose $x^*(\sigma_i^{\bar{s}}) \neq x_{i-1}(\sigma_i^{\bar{s}})$ and $S_i = \emptyset$. Let $x_i(s_i) = x^*(s_i)$ while the allocation x_i is equivalent to x_{i-1} in all the other nodes. Then, the difference between the values of the allocations

x_i and x_{i-1} is $\tilde{q}(x_i, \sigma_i^{\bar{s}})\theta_{x_i(\sigma_i^{\bar{s}})} - \tilde{q}(x_{i-1}, s')\theta_{x_{i-1}(s')}$, where s' is the last node in the path $\boldsymbol{\sigma}^{\bar{s}}$ with $x_{i-1}(s') = x_{i-1}(\sigma_i^{\bar{s}})$ (it may be $\sigma_i^{\bar{s}}$). Moreover,

$$\begin{aligned}\tilde{q}(x_i, \sigma_i^{\bar{s}})\theta_{x_i(\sigma_i^{\bar{s}})} &= \tilde{q}(x^*, \sigma_i^{\bar{s}})\,\theta_{x^*(\sigma_i^{\bar{s}})}\\ &\geq \tilde{q}(x_{i-1}, \sigma_i^{\bar{s}})\,\theta_{x_{i-1}(\sigma_i^{\bar{s}})}\\ &\geq \tilde{q}(x_{i-1}, s')\theta_{x_{i-1}(s')},\end{aligned}$$

where the equality comes from the equivalence between x_i and x^* for all the scenes $\sigma_1^{\bar{s}}, \ldots, \sigma_i^{\bar{s}}$, the first inequality follows from Eq. (1) and the second inequality from the fact that the quality decreases when an ad is displayed multiple times. This proves that the expected value of the allocation x_i is at least the expected value of the allocation x_{i-1}.

Case 3. Suppose $x^*(\sigma_i^{\bar{s}}) \neq x_{i-1}(\sigma_i^{\bar{s}})$ and $S_i \neq \emptyset$. Let $x_i(\sigma_i^{\bar{s}}) = x^*(\sigma_i^{\bar{s}})$ and $x_i(s') = x^*(\sigma_i^{\bar{s}})$, where s' is an arbitrary scene in S_i. Moreover, let x_i be equivalent to x_{i-1} in all the other scenes. Then, every ad appears the same number of times in the path $\boldsymbol{\sigma}^{\bar{s}}$ in x_i and x_{i-1} and hence the expected value of the allocation does not change from x_{i-1} to x_i. This concludes the proof. □

Note that, since the greedy algorithm returns an optimal allocation, it can be used with the Vickrey-Clarke-Groves mechanism [18] to obtain a truthful mechanism in dominant strategies which runs in efficiently (Myerson's weak monotonicity is satisfied). Hence, such mechanism can scale up to real-world settings.

4 THE SD-NE SETTING: DEALING WITH SCENE-DEPENDENT QUALITIES

In this section, we focus on the setting in which the quality of the ads depends on the scene and there are no externalities. Initially, we show that the allocation problem is APX-Hard. Our reduction is based on the satisfiability problem 3-SAT-5 defined as follows.

Definition 4.1. A 3-SAT-5 instance is a 3-SAT instance in which each variable appears in exactly 5 clauses.

As shown by [8], the following theorem holds.

Theorem 4.2. *For some constant $0 < c < 1$, it is NP-Hard to distinguish whether a 3-SAT-5 instance is satisfiable or there is no assignment satisfying a c fraction of the clauses.*

Now, we can prove the following.

Theorem 4.3. *The ad allocation problem in the* SD-NE *setting is APX-Hard.*

Proof. Let $\eta = \max\{c, (1 - \frac{1}{5^5})\}$, where c is the constant factor approximation in Theorem 4.2. Notice that Theorem 4.2 holds even if we replace the approximation factor c with the constant $\eta \geq c$.

Given an instance of 3-SAT-5 with clauses C and variables V, we build an instance of the SD-NE problem as follows. The tree of scenes is composed by a line with a scene s_v for each $v \in V$ in an arbitrary order. Then, it follows a line that includes a scene s_c for each clause $c \in C$ in an arbitrary order. All the transition probabilities $\pi_{s,s'}$ are set to 1. The set of ads A includes two ads a_v and $a_{\sim v}$ for each variable v. Let $\epsilon = 1 - \eta^{1/5}$, and let l denote a literal, *i.e.*, l is a variable or its negation. The qualities of ads are defined as follows: $q_{a_v,s_v} = q_{a_{\sim v},s_v} = 1$ for each $v \in V$, and for each clause $c \in C$ the quality is $q_{a_l,s_c} = \epsilon$ if the literal l belongs to the clause. Every other quality is 0. Finally, let $\theta_a = 1$ for each $a \in A$.

In the following, we show that if there exists an assignment that satisfies all the clauses the utility is at least $|V| + |C|\eta^{4/5}(1 - \eta^{1/5})$, while if no assignment satisfies a η fraction of the clauses the utility is at most $|V| + (1 - \eta^{1/5})\eta|C|$. To conclude the proof notice that $|C| = \frac{3}{5}|V|$. Hence,

$$\begin{aligned}\frac{|V| + (1 - \eta^{1/5})\eta|C|}{|V| + \eta^{4/5}(1 - \eta^{1/5})|C|} &= \frac{|V| + \frac{3}{5}(1 - \eta^{1/5})\eta|V|}{|V| + \frac{3}{5}\eta^{4/5}(1 - \eta^{1/5})|V|}\\ &= \frac{1 + \frac{3}{5}(1 - \eta^{1/5})\eta}{1 + \frac{3}{5}\eta^{4/5}(1 - \eta^{1/5})},\end{aligned}$$

which is a constant strictly smaller than 1.

Soundness. Consider an assignment L, *i.e.*, a set of literals including v or $\sim v$ for each variable $v \in V$, that satisfies all the clauses. We build an assignment x of ads to scenes as follows. For each variable v, let x assigns the ad a_l to the scene s_v, where $l \in \{v, \sim v\}$ is the literal *not* in the assignment L, *i.e.*, such that $l \in \{v, \sim v\} \setminus L$. Finally, let assign to each scene s_c, where $c \in C$, an ad a_l such that the literal $l \in L$ satisfies the clause and belongs to L. This clause exists since the assignment satisfies all the clauses. Then, for each scene s_v, $v \in V$, the value from the scene is 1. Moreover, for each scene s_c, where $c \in C$, we have that the quality $q_{s_c,x(s_c)} = \epsilon$, while $\Xi(x, s_c)$ is at least $(1 - \epsilon)^4$ since each literal appears in at most five clauses. Hence, the value of the allocation is at least

$$|V| + |C|(1 - \epsilon)^4\epsilon = |V| + |C|\eta^{4/5}(1 - \eta^{1/5}).$$

Completeness. Consider an assignment of ads to nodes x. Let $V^* \subseteq V$ be the set of variables $v \in V$ such that $\tilde{q}(x, s_v) = 1$. Then, notice that the expected value of each scene s_v, $v \in V \setminus V^*$ is 0. Let $C^* \subseteq C$ be the set of clauses c such that an ad is assigned to s_c and $q_{x(s_c),s_c} = \epsilon$. Then, notice that the expected value of each scene s_c, $c \in C \setminus C^*$ is 0. We can split C^* in two subsets. The set $C_2 = \{c \in C^* : x(s_c) \in \{a_v, a_{\sim v}\}_{v \in V^*}\}$, while the set $C_1 = C^* \setminus C_2$. Then, we show that there exists a feasible assignment L that satisfies at least C_2 clauses, implying that $|C_2| \leq \eta|V|$. To see that, consider the assignment $L = \{l : a_l \in \{x(s_c)\}_{c \in C_2}\}$. As a first step, we show that the partial assignment is feasible. Suppose by contradiction that there exist two literals $v, \sim v$ belonging to L. Since $a_v \in L$, then there exists a clause $c \in C_2$ such that $x(s_c) = a_v$. Moreover, since $c \in C^*$, the scene s_c has positive quality and $x(s_v) \neq a_v$. Then, since $v \in V^*$, we have that $x(s_v) = a_{\sim v}$. By the definition of C^*, C_2 does not include any clause c such that $x(s_c) = a_{\sim v}$ since they have 0 utility (the ad has been converted in scene s_v and $\Xi(x, s_c) = 0$). Moreover, it is easy to see that the assignment satisfies all the clauses in C_2 by the definition of C^* and the qualities of the scenes.

Now, we bound the cardinality of C_1. Note that since each variable $v \in V$ appears in 5 clauses (considering v and its negation), for each variable $v \notin V^*$ there exist at most 5 clauses $c \in C$ such that $q_{x(s_c),s_c} = \epsilon$ and $x(s_c) \in \{a_v, a_{\sim v}\}$. Then, for each $c \in C_1$ there is a literal v such that $x(s_c) = a_v$ or $x(s_c) = a_{\sim v}$, $x(s_v) \neq a_v$, and $x(s_v) \neq a_{\sim v}$. Recall that $V \setminus V^*$ is the set of v such that $x(s_v) \neq a_v$ and $x(s_v) \neq a_{\sim v}$. Since each variable appears in at most 5 clauses, we have that $|C_1| \leq 5(|V| - |V^*|)$. Moreover, by the definition of η

it holds $5\epsilon = 5(1-\eta^{1/5}) = 1$. Hence, the total utility is at most

$$\begin{aligned}|V^*| + \epsilon\big[|C_2| + |C_1|\big] &\leq |V^*| + \epsilon[|C_2| + 5(|V| - |V^*|)] \\ &= |V|^* + (|V| - |V^*|) + \epsilon|C_2| \\ &= |V| + \epsilon|C_2| \\ &\leq |V| + \epsilon\eta|C| \\ &= |V| + (1-\eta^{1/5})\eta|C|.\end{aligned}$$

This concludes the proof. □

More interestingly, we can show that the ad allocation problem in the SD-NE setting is APX-Complete by designing a polynomial-time algorithm that works in a greedy fashion providing a constant approximation factor. To provide the algorithm, we need to introduce some preliminary steps. Initially, we establish a relation between ad allocations and matroids. A matroid $\mathcal{M} := (G, \mathcal{I})$ is defined by a ground set G and a collection $\mathcal{I}$ of independent sets, *i.e.*, subsets of G satisfying some characterizing properties (see [19] for a detailed formal definition). We denote with $\mathcal{B}(\mathcal{M})$ the set of the *bases* of $\mathcal{M}$, which are the maximal sets in $\mathcal{I}$. We show that feasible allocations can be represented by the matroid $\mathcal{M} := (G, \mathcal{I})$ such that:

- the ground set is $G := \{(a, s) \mid a \in A \cup \{a_\emptyset\}, s \in S\}$, *i.e*, the set of all the possible assignments of ads to scenes;
- a subset $I \subseteq G$ belongs to $\mathcal{I}$ if and only if I contains *at most one* pair in $\{(a, s) \mid a \in A \cup \{a_\emptyset\}\}$ for each scene $s \in S$, *i.e.*, each scene is assigned at most one ad (while an ad can be allocated to multiple scenes).

Intuitively, an element (a, s) of the ground set G belongs to the independent set I if the ad a is allocated to scene s. However, sets $I \in \mathcal{I}$ do *not* characterize allocations, as they may not specify an ad for each scene. Indeed, allocations are captured by the basis set $\mathcal{B}(\mathcal{M})$ of the matroid $\mathcal{M}$. Let us recall that $\mathcal{B}(\mathcal{M})$ contains all the maximal sets in $\mathcal{I}$ and, thus, a subset $I \subseteq \mathcal{I}$ belongs to $\mathcal{B}(\mathcal{M})$ if and only if I contains *exactly one* pair for each scene $s \in S$. Intuitively, a basis $I \in \mathcal{B}(\mathcal{M})$ defines an allocation such that, each scene $s \in S$ is assigned the ad a such that $(a, s) \in I$. This ad is unique by construction as discussed above.

Then, the utility function f on a subset of G is as follows.

Definition 4.4. Let $f : 2^G \to \mathbb{R}_+$ be the function such that, given a subset $D \in 2^G$, $f(D)$ denotes the expected value of assigning to a scene s the ad such that $(a, s) \in D$ without externalities.[2] Formally:

$$f(D) := \sum_{s \in S} \sum_{a \in A:(a,s) \in D} \Pi^s q_{a,s} \theta_a \prod_{s' \in \sigma^s \setminus \{s\}:(a,s') \in D} (1 - q_{a,s'}).$$

Function f satisfies a crucial property: it provides a *decreasing marginal return*. In particular, we show that the utility function $f : 2^G \to \mathbb{R}_+$ is *monotone submodular*. Formally, a function is monotone if for every pair of subsets D_1, D_2 such that $D_1 \subseteq D_2 \subseteq G$, the property $f(D_1) \leq f(D_2)$ holds. Moreover, we say that f is submodular if, for every pair of subsets D_1, D_2 such that $D_1 \subseteq D_2 \subseteq G$ and $(a, s) \in G$, the following property holds:

$$f(D_1 \cup \{(a, s)\}) - f(D_1) \geq f(D_2 \cup \{(a, s)\}) - f(D_2).$$

Now, we provide a characterization of the function $f(\cdot)$.

[2]Notice that this defines a feasible allocation only if $D \in \mathcal{I}$.

Lemma 4.5. *Given a subset $D \in 2^G$, $f(D)$ can be written as:*

$$f(D) = \sum_{s \in S:\rho(s)=\emptyset} \Pi^s \sum_{a \in A} \theta_a f_{s,a}(D),$$

where

$$f_{s,a}(D) = \sum_{s' \in \sigma^s:(a,s') \in D} q_{a,s'} \prod_{s'' \in \sigma^{s'} \setminus \{s'\}:(a,s'') \in D} (1 - q_{a,s''}).$$

Exploiting this characterization, we can show that function $f(\cdot)$ is monotone submodular. The possibility to allocate an ad multiple times along the same path is necessary for submodularity.

Lemma 4.6. *Function $f(\cdot)$ is monotone submodular.*

Given the submodularity of $f(\cdot)$, we can resort to standard tools of submodular maximization to provide an efficient algorithm to optimize f over $\mathcal{I}$. In particular, we can use the *continuous greedy algorithm* to provide a $(1-1/e)$-approximation [3]. Then, to provide an approximation to the optimal ad allocation, we consider the equivalence between independent sets $I \in \mathcal{I}$ and ad allocations x.

Theorem 4.7. *The ad allocation problem in the* SD-NE *setting admits a polynomial-time algorithm that provides a $(1-1/e)$ approximation.*

The impossibility of designing polynomial-time algorithms finding an optimal allocation in the SD-NE setting (unless P = NP) rules out the possibility of using the Vickrey-Clarke-Groves mechanism and poses the question whether we can design a truthful mechanism in dominant strategies running in polynomial time. Moreover, the analysis of the weak monotonicity of the continuous greedy approach is elusive. An intriguing idea is to use Algorithm 1. Indeed, to maximize monotone submodular functions we can use the simpler greedy approach instead of the more complex continuous greedy with a small loss in the approximation factor. In particular, the greedy algorithm provides a $\frac{1}{2}$-approximation to monotone submodular maximization on a matroid [17]. However, even though Algorithm 1 exhibits weak monotonicity in the SI-NE setting, we show with the following proposition that this property does not hold in the SD-NE setting. Therefore, it cannot be used for designing a truthful mechanism.

Proposition 4.8. *Algorithm 1 does not satisfy Myerson's weak monotonicity in the* SD-NE *setting.*

5 THE SI-E SETTING: DEALING WITH EXTERNALITIES

In this section, we focus on the setting in which there are externalities among ads and the quality of the ads does not depend on the scene. We start our analysis by providing a strong impossibility result. We show that the allocation problem is hard to approximate and that the hardness of the approximation depends on the memory length k. Our reduction is from the following promise problem related to the problem of finding cliques in graphs [12, 21].

Theorem 5.1. *For every $\epsilon > 0$, it is NP-Hard to distinguish whether a graph $G = (V, E)$ with vertexes V and edges E has a clique of size $|V|^{1-\epsilon}$ or all the cliques have a size of at most $|V|^{\epsilon}$.*

Algorithm 2 GREEDY–SI–E

```
Inputs: set of scenes S, set of ads A, memory length k
S_1 ← {s ∈ S : |σ^s| ∈ {1 + j(k+1)}_{j∈ℕ}}
x* ← GREEDY(S_1, A)
return x*
```

We can show that it is NP-Hard to provide an approximation to the ad allocation problem in the SI-E setting which is sublinear in the memory length k. Formally, we can state the following:

THEOREM 5.2. *For any $\epsilon > 0$, it is NP-Hard to approximate the ad allocation problem in the* SI-E *setting to within a factor $|k+1|^{1-\epsilon}$, where k is the memory length.*

We can show that the problem admits a polynomial-time approximation algorithm that provides a $\frac{1}{k+1}$-approximation, matching the lower bound stated above. The pseudocode is provided in Algorithm 2. It extends the greedy algorithm in Section 3 as follows.

Algorithm 2 allocates ads only to scenes at depth $\{1+i(k+1)\}_{i\in\mathbb{N}}$, *i.e.*, it allocates ads only to the scenes $s \in S$ such that $|\boldsymbol{\sigma}^s| \in \{1+i(k+1)\}_{i\in\mathbb{N}}$. In this way, the allocated ads are not subject to any externalities. Moreover, as we show in the following theorem, we allocate ads to a subset of scenes sufficiently large to guarantee a $\frac{1}{k+1}$-approximation of the optimal utility. Then, Algorithm 2 computes the optimal allocation resorting to Algorithm 1. The following theorem formally states the guarantees of the algorithm.

THEOREM 5.3. *Algorithm 2 provides a $\frac{1}{k+1}$-approximation to the ad allocation problem in the* SI-E *setting. Moreover, it runs in polynomial time.*

Finally, we focus on Myerson's weak monotonicity, and we show that Algorithm 2 is monotone.

PROPOSITION 5.4. *Algorithm 2 satisfies Myerson's weak monotonicity property.*

Therefore, the resulting mechanism in which the allocation function is given by Algorithm 2 and the payments are defined as in Myerson's Lemma [16] is truthful in dominant strategies.

6 THE SD-E SETTING: APPROXIMATING THE GENERAL PROBLEM

In this section, we deal with the general ad allocation problem in which there are both externalities among ads and scene-dependent qualities. As Theorems 4.3 and 5.2 show, the problem in the SD-E setting is Poly-APX-Hard. In particular, Theorem 5.2 rules out the possibility of providing an approximation sublinear in k in polynomial time. In this section, we show that the problem in the SD-E setting admits a polynomial-time algorithm that provides a $(\frac{1-1/e}{k+1})$-approximation, thus matching the inapproximability lower bound provided by Theorem 5.2.

Let the matroid $(G, \mathcal{I})$ and the function f be defined as in Section 4. We show that we can apply Algorithm 3 to find a $\frac{1-1/e}{k+1}$-approximation to the problem in the SD-E setting. In particular, the algorithm follows the approach of Section 5, except that it needs to evaluate all the sets of scenes $\{1+j(k+1)\}_{j\in\mathbb{N}}, \{2+j(k+1)\}_{j\in\mathbb{N}}, \dots, \{k+j(k+1)\}_{j\in\mathbb{N}}$ as the qualities depend on the

Algorithm 3 GREEDY–SD–E

```
1: Inputs: set of scenes S, set of ads A, memory length k
2: for i ∈ {1, ..., k} do
3:     S_i ← {s ∈ S : |σ^s| ∈ {i + j(k+1)}_{j∈ℕ}}
4:     x_i ← CONTINUOUSGREEDY(S_i, A, f)
5: i* ← argmax_{i∈{1,...,k}} Σ_{s∈S} (Π^s q̃(x_i, s) θ_{x_i(s)})
6: return x_{i*}
```

scene. Intuitively, the rationale is to enumerate these sets of scenes and, for each of them, to approximate the optimal allocation that employs only those scenes with the continuous greedy algorithm used in the SD-NE setting. This is necessary because the qualities are scene-dependent, and therefore, we have to include each scene in at least one of the considered allocations. We denote with CONTINUOUSGREEDY(S, A, f) the continuous greedy algorithm that works with the matroid defined in Section 4. It considers only scenes in S and ads in A and optimize the monotone submodular function f defined is Section 4. Finally, we take the best allocation among those evaluated by the algorithm. The resulting approximation factor combines the approximation factors retrieved in the SD-NE and in the SI-E settings. The pseudocode is provided in Algorithm 3. The following theorem states the guarantees of the algorithm.

THEOREM 6.1. *Algorithm 3 provides a $\frac{1-1/e}{k+1}$-approximation to the ad allocation problem in the* SD-E *setting. Moreover, it runs in polynomial time.*

We conclude by showing that neither Algorithm 1 nor its extension Algorithm 2 are weakly monotone in the SD-E setting. This result follows from the non-monotonicity of Algorithm 1 in the simpler SD-NE setting (see Proposition 4.8).

PROPOSITION 6.2. *Neither Algorithm 1 nor Algorithm 2 satisfies Myerson's weak monotonicity in the* SD-E *setting.*

REMARK 1. *We can derive an algorithm similar to Algorithm 3 by replacing the continuous greedy algorithm (Line 4) with the greedy Algorithm 1, obtaining a $\frac{1/2}{k+1}$-approximation factor. This algorithm does not satisfy Myerson's weak monotonicity in the* SD-E *setting.*

7 THE ADVANTAGE OF ALGORITHMS LEVERAGING AN ACCURATE USER MODEL

In this section, we underline the importance of adopting an accurate user model. Specifically, under the assumption that our model captures real-world users' behavior, we compare the performance of algorithms disregarding basic user features (*i.e.*, the externalities, the sequential traversal of scenes, or the quality dependence on scenes) with the performance of our approximation algorithms.

PROPOSITION 7.1. *An algorithm disregarding scene-dependent qualities or externalities or sequential traversal of scenes can lead to solutions arbitrarily worse than those returned by* GREEDY–SD–E*, even in simple instances.*

In the following example we show that algorithms disregarding externalities achieve an allocation value arbitrarily smaller than that provided by our approximation algorithms. In particular, we consider a setting in which user's behavior is influenced by externalities and ad qualities are scene dependent. In the Supplemental Material we provide further examples supporting Proposition 7.1.

Example 7.2. Consider a setting with $2n$ scenes $\{s_1, \ldots, s_n\}$, with $n \in \mathbb{N}$, where $s_{i+1} \in \rho(s_i)$ and $\pi_{s_i,s_{i+1}} = 1$ for every $i < n$, while any other $\pi_{s,s'} = 0$. There are $2n$ ads with $q_a = 1$, $\theta_a = 1$, and $\gamma_{a,a'} = 0$ for every other a'. The memory length is $k = 1$. Algorithm 2 guarantees at least $1/2$ of the optimal value. It allocates ads in the odd scenes, providing a value of n, which is the optimal allocation. A greedy algorithm similar to Algorithm 2, that disregards externalities, allocates one ad per scene, providing a value of 1, which corresponds to a $1/n$-ratio of the optimum.

8 EXPERIMENTAL EVALUATION

In the SD-E setting, we compare our algorithms Greedy (Algorithm 1) and Greedy–sd–e (Algorithm 3 in which we replace the continuous greedy algorithm at Line 4 with Algorithm 1) with opt, a baseline algorithm which returns the optimal solution to the ad allocation problem. To compute opt, we formulate the ad allocation problem as an Integer Linear Program (ILP). Being the objective funciton nonlinear, we apply linearinzation techniques and then, we compute the optimal solution using Gurobi solver. Further details on opt are provided in the Supplemental Materials.

Figure 2 shows the allocation expected value and the execution time of the algorithms averaged over 20 instances, which are perfect binary trees. The colored area represents the standard deviation. The dependence on the scene is simulated by sampling the ad qualities from Beta distributions with a spike on a scene, while the externalities among different ads are randomly generated. We set user memory $k = 1$ and we consider the set of ads $A = \{a_\emptyset, a_1, a_2, a_3, a_4\}$ with value per conversion $\theta_a = 1$ for all $a \in A$. The algorithms are tested on trees of varying depth, spanning from 1 to 5.

Figure 2 (top) shows the allocation expected value, which grows in the depth of the tree. The value is determined by summing the contributions of an increasing number of ads as the depth increases. We run algorithm opt until depth 4 because of the exponential growth of its execution time. We observe that the performance of algorithm Greedy is comparable to that of opt, while its execution time scales up to larger instances, as shown in Figure 2 (right). Moreover, Greedy outperforms Greedy–sd–e. This behavior is expected since Greedy–sd–e is robust to worst-case instances in which externalities push ads value towards zero. In particular, each time it selects an ad a_i, with $i \neq \emptyset$, it allocates $a_\emptyset$ to the k consecutive slots to avoid the effect of potentially strong externalities. However, on average instances, Greedy results more profitable, by collecting the value coming from a larger number of allocated ads. In Figure 2 (bottom), we observe that at depth 5, the execution time of Greedy–sd–e is approximately the 10% of that of Greedy. This suggests that Greedy–sd–e could be a valuable choice for large instances.

9 CONCLUDING REMARKS AND FUTURE WORKS

We initiate the investigation of advertising models for self-contained environments, such as augmented-reality platforms. Users experience a continuous experience over time. Hence, it becomes possible for platforms to control which ads show during their traversal

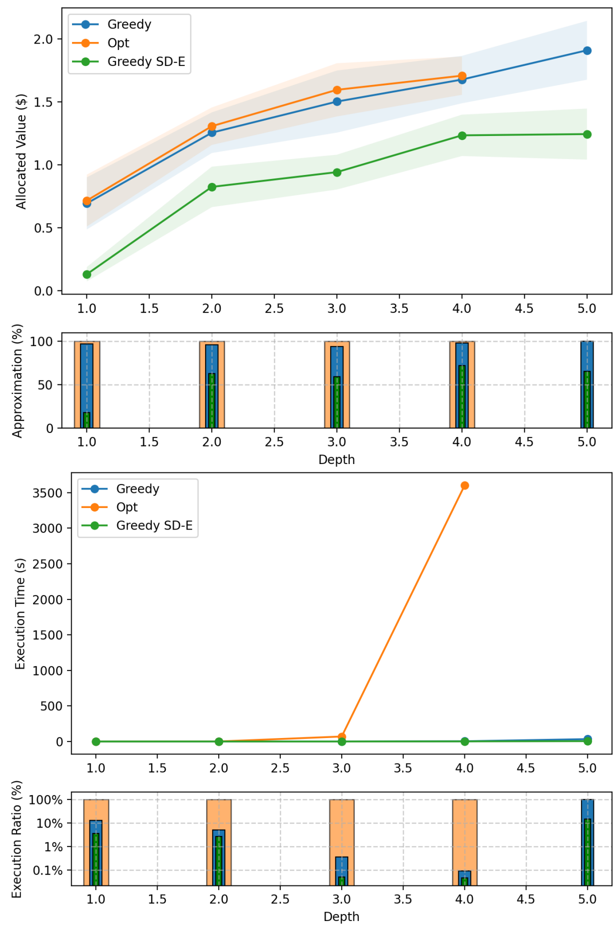

Figure 2: Performance of algorithms Greedy (blue), Greedy–sd–e (green) and opt (orange).

of scenes. We exploit the structure of this environment to maximize ad allocations value through efficient algorithms by taking into account externalities among ads and the dependence of their performance on the visited scene. We provide tight computational complexity results and show that a simple greedy algorithm guarantees a constant approximation when the length of the user memory k is fixed. We also discuss whether the provided allocations satisfy Myerson's weak monotonicity so as to be adopted in truthful mechanisms. Interesting future directions would be to further investigate the design of truthful mechanisms and to develop online learning algorithms to estimate the parameters of our model.

ACKNOWLEDGMENTS

This paper is supported by the Italian MIUR PRIN 2022 Project "Targeted Learning Dynamics: Computing Efficient and Fair Equilibria through No-Regret Algorithms", by the FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence), and by the EU Horizon project ELIAS (European Lighthouse of AI for Sustainability, No. 101120237).

REFERENCES

[1] Kathryn A Braun-LaTour, Michael S LaTour, Jacqueline E Pickrell, Elizabeth F Loftus, and Stanford University) is a Distinguished. 2004. How and when advertising can influence memory for consumer experience. *Journal of Advertising* 33, 4 (2004), 7–25.

[2] Raymond R Burke and Thomas K Srull. 1988. Competitive interference and consumer memory for advertising. *Journal of consumer research* 15, 1 (1988), 55–68.

[3] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. 2011. Maximizing a Monotone Submodular Function Subject to a Matroid Constraint. *SIAM J. Comput.* 40, 6 (2011), 1740–1766. https://doi.org/10.1137/080733991 arXiv:https://doi.org/10.1137/080733991

[4] C. Deng and S. Pekec. 2011. Money for nothing: exploiting negative externalities. In *ACM EC*. 361–370.

[5] N.R. Devanur and S.M. Kakade. 2009. The price of truthfulness for pay-per-click auctions. In *ACM EC*. 99–106.

[6] eMarketer. [n.d.]. Worldwide Digital Ad Spending 2023. https://www.insiderintelligence.com/content/worldwide-digital-ad-spending-2023

[7] G. Farina and N. Gatti. 2017. Adopting the Cascade Model in Ad Auctions: Efficiency Bounds and Truthful Algorithmic Mechanisms. *J. Artif. Intell. Res.* 59 (2017), 265–310.

[8] Uriel Feige. 1998. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)* 45, 4 (1998), 634–652.

[9] D. Fotakis, P. Krysta, and O. Telelis. 2011. Externalities among Advertisers in Sponsored Search. In *Algorithmic Game Theory*. Springer Berlin Heidelberg, 105–116.

[10] N. Gatti, M. Rocco, S. Ceppi, and E. H. Gerding. 2014. Mechanism Design for Mobile Geo-Location Advertising. In *AAAI*. 691–697.

[11] N. Gatti, M. Rocco, P. Serafino, and C. Ventre. 2018. Towards better models of externalities in sponsored search auctions. *Theor. Comput. Sci.* 745 (2018), 150–162.

[12] Johan Håstad. 1999. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica* 182, 1 (1999), 105–142.

[13] IAB. 2021. Interactive Advertising Bureau (IAB) internet advertising revenue report, Full year 2020 results.

[14] D. Kempe and M. Mahdian. 2008. A Cascade Model for Externalities in Sponsored Search. In *Internet and Network Economics*. 585–596.

[15] Mehryar Mohri and Andres Munoz Medina. 2014. Learning theory and algorithms for revenue optimization in second price auctions with reserve. In *International conference on machine learning*. PMLR, 262–270.

[16] Roger B Myerson. 1981. Optimal auction design. *Mathematics of operations research* 6, 1 (1981), 58–73.

[17] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming* 14, 1 (1978), 265–294.

[18] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA.

[19] Alexander Schrijver. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Vol. B.

[20] C.R. Taylor. 2022. Research on advertising in the metaverse: a call to action. *International Journal of Advertising* 41 (2022), 383–384.

[21] David Zuckerman. 2007. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing* 3, 6 (2007), 103–128.