POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Algorithmic Advertising in the Metaverse

Tesi di Laurea Magistrale in
Computer Science Engineering - Ingegneria Informatica

Author: **Alberto Latino**

Student ID: 968943
Advisor: Prof. Nicola Gatti
Co-advisors: Matteo Castiglioni, Alberto Marchesi, Giulia Romano
Academic Year: 2022-23

Abstract

The Metaverse has recently emerged as a promising and revolutionary platform for the world of advertising. With its immersive experience, the metaverse opens the door to a new horizon for the advertisers, allowing the display of eye-catching content with the ability to monitor how users react to the displayed ads. As the user traverses virtual spaces and engage in different activities, an important question arises regarding which ads to display and and where to place them across the virtual platform. In this work, we provide a novel user model for advertising in the metaverse, providing algorithms for an optimal allocation of advertisements. In the provided model, the user traverses virtual locations called *scenes*, in which ads may be allocated. The effectiveness of an ad, referred to as *quality*, may vary among the scenes as the user interests may be influenced by the context in which the ad is displayed. Moreover, we study the effect of inter dependencies among ads, called *negative forward externalities*, which may reduce the quality of subsequent allocations once an ad is displayed. As the externalities can have a significant impact on the quality of an ad, the model allows leaving some scenes unallocated to mitigate this effect. Moreover, an ad may be displayed multiple times, as the user might still click on it in future displays. We assess the time complexity of the problem for various scenarios and we provide approximation greedy algorithms with theoretical guarantees that run in polynomial time for real-world applications. Furthermore, the Myerson's weak monotonicity property is investigated as it ensures the design of truthful auction mechanisms. Finally, we propose a specific class of ads to test the algorithms in challenging scenarios and we formulate the problem as an Integer-Linear-Programming model to compare the performance of the greedy approach against the optimal allocation. In view of this, this work wants to contribute to a first step towards advertising in the Metaverse offering a fresh perspective on this emerging profitable sector.

Keywords: Metaverse, Advertising

Abstract in lingua italiana

Negli ultimi anni, il Metaverso è emerso come una piattaforma promettente e rivoluzionaria per il mondo della pubblicità. Il metaverso infatti, apre le porte a un nuovo orizzonte per gli inserzionisti, consentendo la visualizzazione di contenuti accattivanti con la possibilità di monitorare come gli utenti reagiscono agli annunci visualizzati. Nel metaverso, l'utente naviga attraverso degli spazi virtuali e dunque sorge una domanda importante riguardo a quali annunci allocare e in quale luogo virtuale mostrarli. In questo lavoro, forniamo un nuovo modello dell'utente per la pubblicità nel metaverso, fornendo algoritmi per un'allocazione ottimale delle pubblicità. Nel modello fornito, l'utente attraversa gli spazi virtuali del metaverso, chiamati *scene*, in cui possono essere allocati annunci di vario tipo. L'efficacia di un annuncio, indicata come *qualità*, può variare tra le scene in quanto gli interessi dell'utente possono essere influenzati dal contesto in cui l'annuncio viene visualizzato. Inoltre, studiamo l'effetto delle interdipendenze tra gli annunci, chiamate *esternalità*, che possono ridurre la qualità delle allocazioni successive una volta che un annuncio viene visualizzato. Poiché le esternalità possono avere un impatto fortemente negativo sulla qualità degli annunci futuri, il modello consente di lasciare alcune scene non allocate per attenuare questo effetto. Inoltre, un annuncio può essere visualizzato più volte, in quanto l'utente potrebbe comunque fare click su di esso in visualizzazioni successive. Inoltre, valutiamo la complessità temporale del problema per vari scenari e forniamo algoritmi greedy di approssimazione con garanzie teoriche che funzionano in tempo polinomiale, favorendo dunque un' applicazione in campo reale. Inoltre, viene studiata la proprietà di monotonicità debole di Myerson in quanto garantisce la progettazione di meccanismi di asta veritieri. Infine, proponiamo una classe specifica di annunci per testare gli algoritmi in scenari impegnativi e formuliamo il problema come un modello di Programmazione Lineare Intera per confrontare le prestazioni dell'approccio greedy con l'allocazione ottimale. In considerazione di ciò, questo lavoro vuole contribuire ad un primo passo verso il mondo della pubblicità nel Metaverso, offrendo una nuova prospettiva su questo settore emergente.

Parole chiave: Metaverso, Pubblicità, Advertising

Contents

1 | Introduction

1.1. Overview

The history of advertising can be traced back to the 19th century, when the emergence of newspapers and magazines provided companies with a means to promote their products to a large audience, resulting in a new source of revenue. In the 20th century, advertising expanded rapidly due to the advent of new technologies such as direct mail, radio, television, the internet and mobile devices. In particular, the internet has enabled advertising to be delivered through various channels such as the web or mobile apps and, more significantly, has facilitated a higher degree of ad personalization due to the vast amount of data that users everyday leave in the web. Furthermore, advertising has evolved over the years to become more interactive and engaging, capturing users' attention in a more effective manner than the traditional print ads in newspapers, making advertising one of the most profitable sectors in the global economy.

Only in the U.S, in 2023 the advertising market is expected to grow to a to total of $318 billion, surpassing $300 billion for the first time, with a year-over-year increase of 8% [1]. Furthermore, it is a significant source of income for several tech companies, and in some cases, it accounts for up to 60% of their total revenue. In 2022, Google's advertising revenue generated by Google AdWords, Network Members, and YouTube Ads amounted to 224.47 billion U.S. dollars, representing around 80% of the company's revenue [10]. Over the last five years, the search engine giant saw an average annual growth rate of 17.5% in its advertising revenue. According to [22], Google's advertising revenue is expected to grow by an average of 14.8% per year at least until 2024.

The historical evolution of advertising, not only showed how profitable this sector is , but also highlighted what are the key necessary elements for advertising to occur: a *platform* where users spend their time (e.g., reading news, scrolling social media feeds), and a *technology* capable of displaying ads and collecting user data to deliver more personalized content.

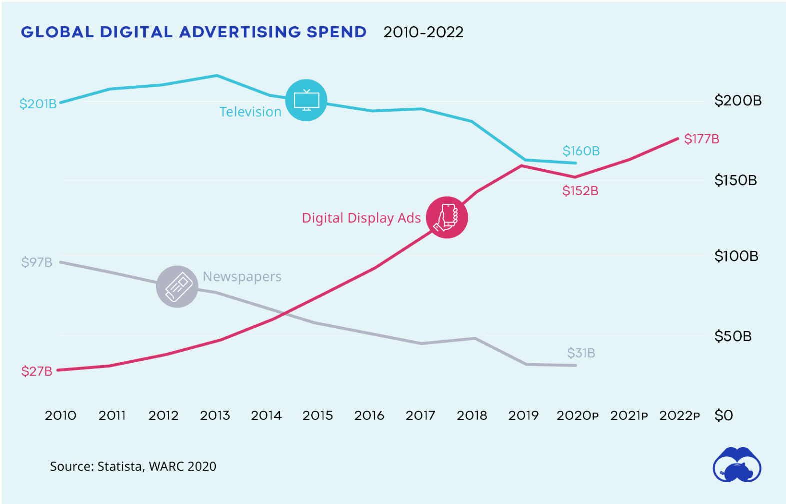


Figure 1.1: Digital Advertising Spend from 2010 to 2022.

The emergence of advanced visual technologies and the availability of more powerful computational resources opened the door to two of the most widely discussed concepts that might potentially revolutionize the world of advertising: *Augmented Reality (AR)* and *Virtual Reality (VR)*. These two emerging technologies enable users to experience digital content in a more immersive and interactive way. The AR overlays visual information onto the user's real-world environment, while the VR provides a completely immersive experience in a simulated environment. AR and VR can enhance advertising by providing immersive and eye-catching content that better capture the user attention, generating a memorable and engaging advertising experience that may potentially result in a higher conversion.

The emergence of AR and VR has led to the creation of a new concept called the Metaverse: a virtual world where users can engage with digital content and interact with each other in real-time. The word 'Metaverse' comes from the combination of the prefix "meta-" which means beyond, and "-verse" from the word universe. In fact, it has been compared to a "parallel universe" that exists alongside our physical world, offering endless possibilities for social interaction, entertainment, and commerce. In order to express their identity in this world, users can also create and customize their avatars, which serve as their digital representation. Similarly to the physical world, in the Metaverse users can purchase virtual objects, own exclusive gadgets or acquire virtual lands, creating a new virtual economy. The ownership of virtual objects is secured through non-fungible tokens (NFT) based on the recent blockchain technology, which provides a secure and shared digital ledger (i.e. a log) that enables users to record transactions and share information. The rising popularity of the Metaverse has attracted investment from venture capitalists and large companies from around the world, encouraging the development of the necessary technologies and the creation of immersive projects that offer a glimpse into what the Metaverse could look like in the coming years. Among the numerous immersive initiatives, Decentraland and SandBox are considered to be two of the most popular ones. The first, is a virtual reality platform based on Ethereum (a decentralized blockchain platform), that enables users to purchase and trade real estate assets within a virtual environment while interacting with each other [18]. The "LAND" non-fungible token (NFT) ensures the land parcels' ownership. This Metaverse platform has experienced remarkable success, as users already exchange virtual land pieces for millions of dollars. Instead, SandBox is a platform that fosters users' creativity, allowing them to create inventive virtual worlds to inhabit [24]. In addition to providing content-creation opportunities, Sandbox permits its community to develop games, assets, and applications on top of their land parcels.

The wide range of possible immersive experiences that Metaverse provide, together with the arise of a new virtual economy, highlight the VR platform's potential as an advertising and marketing destination. Indeed, the primary function of certain augmented reality applications is the advertising itself, wherein users take part in an immersive experience to sample products, anticipating the emotion of possessing such advertised products. As an example, in 2018 Amazon devised the VR Kiosks [13], a virtual reality platform that enables users to navigate through spaces that emulate current Amazon store divisions in a remarkably immersive fashion. As the Amazon's project shows, one of the peculiarities of advertising in the Metaverse is that users are able to experience in advance the emotions triggered by the advertised product, resulting in a higher possibility of engagement. Another uniqueness of Metaverse platforms is that users' preferences can be inferred also by tacit data extracted from their reactions to external stimuli. In fact, augmented and virtual reality technologies, employing techniques like eye-tracking, can measure users' emotional response and observe their points of focus, enabling a more accurate understanding of their desires and emotional triggers [19]. As a consequence, this leads to a higher probability of users engaging with advertisements (i.e. the Click-Through Rate) and purchasing the product upon viewing the ad (i.e. the Conversion Rate). In 2023, the Metaverse Advertising segment is expected to reach \$0.40bn value and it is expected to show an annual growth rate of 24.79%, with a forecasted market volume of US\$1.90bn by 2030 [25]. Indeed, the above mentioned technologies, are able to revolutionize the way advertising is delivered everyday to billions of users in the world, representing a huge opportunity for investors and advertisers. In view of this, the current work wants to unlock the vast potential for the future of advertising, providing a unique user model and allocation algorithms, representing a new step forward for advertising in the Metaverse.

1.2. Related Works

To the best of our knowledge, the field of advertising in the Metaverse is still unexplored and lacks a proposal for a user model. Since we base our model also on the virtual location of the user, it is worth noting a related study in the area of mobile geo-located advertising conducted by Gatti et al. (2014) [8], where the importance of user movements is highlighted as a key ingredient for the success of advertising.

In the field of interactions among advertisements, a model has already been introduced and studied by Fotakis, Krysta, and Telelis (2011) [7] for single-keyword auctions, taking into account the externalities among advertisers. Moreover, Gatti et al. (2018) [9], gives a mathematical formulation of advertisers' externalities while giving computational complexity results. However, the models do not present constant approximation algorithms. Finally, Deng and Pekec (2011) [4] show how negative externalities in ecommerce and internet advertising can result in a no-allocation equilibrium in case of scarce resources.

In the more specific case of Metaverse, Heller et al. (2021) [12] show the potential for new targeting techniques based on emotional and physiological responses, while Bonetti [2] highlights the potential of using product simulation to increase revenues and create an immersive shopping experience while also overcoming operational barriers and saving time and costs. Finally, Taylor (2022) [26] encourages more research on advertising in the Metaverse, highlighting the importance of early academic research in digital advertising. More on State-of-the-Art advertising in the Metaverse in section 2.5.

1.3. Original Contributions

This work initiates the study of a new user model in the Metaverse, providing algorithms that optimally allocate advertisements across the virtual environment. The model extends those currently adopted for search and geo-located mobile advertising as mentioned in the previous paragraph. We imagine that in the Metaverse scenario, the user traverses scenes that represent virtual locations while doing some activities such as sport events, concerts, gaming sessions or conferences, during which they can be targeted by ads. The relevance of ads, referred to as *quality*, may vary among the scenes as user interests may be influenced by external factors that depend on his current location (e.g. an ad may attract the user differently if shown in a concert or in a conference). Furthermore, ads are subject to inter dependencies, called *externalities*, that reduce the ad quality based on their sequential disposal. In fact, based on their content, ads might influence each others as in the case of products that are strategic substitutes, as shown by Deng and Pekec (2011) [4]. However, the existence of externalities is limited as long as the user remembers previously seen ads, hence we introduce a user attention model limited by the length of the user memory. Since the inter dependencies that ads have among each other may heavily affect their quality, sometimes is more convenient to leave some scenes empty to get more user attention in subsequently displayed ads. Thereby, a fictitious empty ad is introduced to tackle this dynamics, allowing to achieve a better allocation value in these strict cases.

The scenes that compose the Metaverse are organized according to a graph structure and another tree-like structure represents the possible paths that the user may traverse starting from a certain node of such a graph.

Hence, it is studied the problem of achieving an optimal allocation of ads among the possible future scenes in the mentioned tree structure, providing approximation algorithms that guarantee theoretical approximation ratio for the allocated value. We will show that having scene-dependent qualities makes the problem Apx-Complete and we provide a greedy algorithm running in polynomial time with an approximation factor of $(1 - 1/e)$. On the other hand, the presence of externalities among ads makes the problem Poly-Apx-Complete and we provide a polynomial-time greedy algorithm with an approximation factor of $1/(k + 1)$.

Having each algorithm a running time that is reasonable for real-world applications, we also investigate their Myerson's weak monotonicity property to understand if it is possible to design of truthful auction mechanism with such algorithms. Finally, we build model the problem as an ILP mathematical problem in order to compare the approximated solutions with the optimal one in both standard scenarios and challenging instances for the greedy approach.

1.4. Structure

The thesis is divided in chapters following the below structure:

- **Chapter 2** gives an overview of the world of Digital Advertising today, analysing several advertising formats, targeting techniques and giving an overview of the State of Art auction algorithms. Finally, it is given an overview of the literature in advertising in the Metaverse and giving a first categorization of metaverse advertising formats. Finally, some technical preliminaries are given to facilitate the reading of the work.
- **Chapter 3** introduces a high level view of the proposed user model, explaining how the virtual location of the Metaverse are organized and how the user moves through

them. Some examples of how the advertisments are organized within the Metaverse structure are given. Lastly, the user's attention model is explained illustrating the negative effects of the externalities.

- **Chapter 4** provides the mathematical formulation of the user model, illustrating the tree-like structure and how transition probabilities among nodes are computed. Furthermore, externalities and user memory are formalised leading to the definition concepts of the *Total Externality*, *Non Convert Probability* and *Adjusted Quality*. Lastly, the Objective Function is provided together with an allocation example.
- **Chapter 5** provides greedy algorithms for the ad allocation problem formulated in Chapter 4, providing their complexity classes, approximation and pseudo-codes. Furthermore, their Myerson's weak monotonicity is investigated.
- **Chapter 6** formalize the problem as an ILP mathematical model for the computation of the optimal solution. The definition of sets, parameters, variables and constraints is provided together with a non-linear version of the objective function. Subsequently, linearization methods are discussed and applied to provide the linearized version to be passed to the Gurobi solver.
- **Chapter 7** describes a high level implementation of the user model describing the generation process of the tested scenarios.In this part, results are reported and discussed in terms of allocated value and execution time. Furthermore, a particular setting is proposed and investigated to show the greedy performance under challenging circumstances.
- **Chapter 8** concludes the thesis summarizing the work done and proposing future works that can be conducted for a user-oriented real world application of advertising in the Metaverse.

2 | Advertising State of the Art

In this chapter are covered the fundamentals of advertising and described the various channels used to reach potential customers. Moreover, it is given an overview of the evolution of the main auction algorithms, explaining their functioning. Finally, the chapter explores the arise of new immersive advertising formats in the Metaverse.

2.1. Digital Advertising

Throughout the history, advertising has been an integral part of communication as various mediums of communication have emerged. For instance, newspapers in the early 1700s and television during the industrial revolution, brought advertising to people's attention. However, the internet has played an even more significant role in changing the way communication and advertising work as it allows to display ads with a higher targeting precision. In the last two decades, the arrival of mobile devices allowed billions of users to connect and share content through various online platforms such websites, mobile apps, and social networks like Facebook and Instagram. Hence, their increased activity on the web resulted in the users being heavily exposed to advertisements every day. This, from the advertiser's perspective, represented an opportunity to sponsor their products to precisely chosen potential customers. In fact, they can show multiple ads to users and segment the population based on their interests, gender, social circle, and geographical position. Such segmentation is made possible due to the vast amounts of information that everyday users leave behind while browsing the web. As a result, the advertising mechanism of today has become significantly more remunerative than in the past and represents one of the most profitable sectors in the tech industry. Despite its numerous benefits, advertising systems has become increasingly complex over the years. With many advertisers competing for the same ad slot, identifying the best way to target potential customers became extremely challenging. In fact, there are usually more advertisers than slots in which their ads can be placed. For this reason an auction mechanism is used to decide which ad will be published in each slot, depending on the amount each advertiser declared is willing to pay, which is determined by their bid.

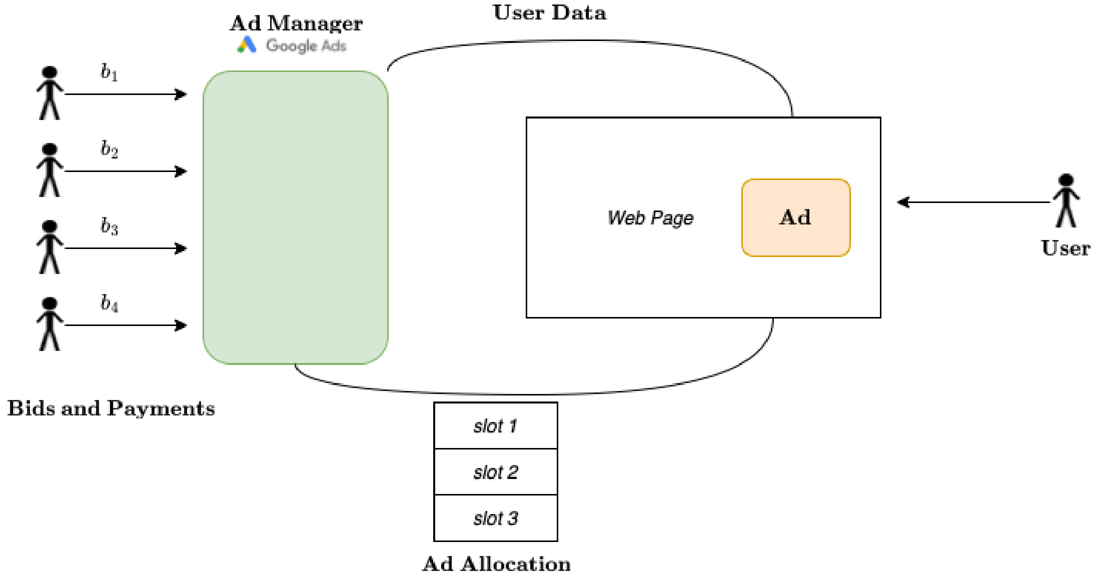


Figure 2.1: Example of Advertising Functioning.

2.2. Formats

The Internet can be browsed through several channels such as search engines, social networks or websites. This results in several ways of doing advertising. Among these, below there are some among the most popular:

- **Search**

  Recent statistics demonstrate that Google's search engine, processes around 99,000 search queries per second, resulting in a staggering 8.5 billion daily searches. As such, Google and other search engines, including Bing and Yahoo!, offer a profitable prospect for advertisers to promote their campaigns. Search queries are associated with specific keywords, which provide insight into the user's search intentions and potential interests. As keywords serve as an indicator of user interests, they are used to identify a suitable match between users and advertisers. Once the match has been found, the auction's winning ads are allocated in the search engine page. When computing the allocation, other than the bid amount also a quality metric influences the ads allocation. The quality metric is a measure of an advertisement's relevance and usefulness to the user based on the searched keywords, hence it represents also the likelihood of the user to click on the ad once it is displayed. Usually, the ads are shown to the user among the first rows of the search results, in the form of sponsored links to the advertiser's web page.

- **Social**

  Social advertising is among one of the fastest-growth formats of advertising. In fact, given the remarkable success that social networks such as Facebook and Instagram have had during the last decade, social advertising became one of the most effective options for advertisers to sponsor their products. In social advertising, contrary to what happens in the search format, ads are not shown based on search keywords. Rather, the advertisers target their potential customers leveraging on user information such as gender, interests or profession. Moreover, the user's preferences may be inferred also based on their activity on the application, such as followed pages, or taking into account their close friends's interests. Finally, the ad content is usually displayed through images and text shown to the user while scrolling the social feeds.

- **Display**

  Display ads represent a prominent category of paid advertising widely adopted by numerous industrial companies. Typically positioned at the top, side, or within the web pages of several websites, these ads exhibit a high level of effectiveness, reaching over 90% of internet users. Display ads usually incorporate visual banners comprising text, images, and audio and are displayed to the audience based on their demographics, interests, and online behaviors.

2.3. Auctions

The advertising mechanism employs auctions to assign ads to the available slots, as the number of advertisers typically exceeds the number of available spaces for which they are competing. Advertisers participate in the auction by submitting bids, representing the highest amount they are willing to pay for a certain slot. The auctioneer then selects which ads to assign and computes the corresponding payments for the advertisers. Various payment algorithms exist, including the Generalized Second Price (GSP) [23], Generalized First Price (GFP), Vickrey-Clarke-Groves (VCG) [27], and Optional Second Price (OSP). Historically, GFP was the first algorithm to be used in online advertising auctions in which each advertiser places a bid for some keywords matching ads. It was developed in 1997 and was based on the concept of a first-price auction, where the highest bidder wins the auction and pays the amount they bid. However the GFP algorithm has been criticized for being unstable because it is vulnerable to bid shading, a strategy used by bidders to reduce the amount they pay for an impression.

In 2002, GSP was developed by Google as a more efficient alternative to GFP. This mechanism was based on the concept of a second-price auction, where the highest bidder wins the auction but the price paid corresponds to the bid amount of the advertiser whose ad is ranked immediately after. The GSP auction mechanism quickly became the dominant bidding algorithm in online advertising and is still widely used today in the context of keyword matching auctions. Although GSP is generally efficient, it can present some problems with regard to truthful bidding, as advertisers may not always bid their true valuation of the competed ad slot. Hence, the VCG algorithm was introduced in online advertising auctions in the early 2000s. The VCG auction mechanism is designed to encourage truthful bidding, meaning that each bidder is incentivized to declare their true value when making a bid. In fact, VCG determines the price each bidder pays by calculating the marginal harm their bids cause to other bidders. Theoretically, VCG is optimal, but its complexity and high computational requirements make it impractical for large-scale auctions. More detailed explanations of these auction algorithms are provided below.

2.3.1. Generalized First-Price

The Generalized First Price (GFP) auction is one of the pioneering mechanisms that has been implemented in sponsored search auctions, due to its simple and intuitive approach. In GFP, a set of n bidders compete for the assignment of k (with $k < n$) slots, each possessing a click-through rate (CTR), which denotes the probability of a slot being clicked. These slots are ranked based on their CTR in descending order. During the auction process, all the advertisers submit their respective bids, and the slot with the highest CTR is allocated to the bidder with the highest bid. Similarly, the slot with the second-highest CTR is assigned to the bidder with the second-highest bid, and so on. Unlike other auction mechanisms, GFP requires the winner to pay the entirety of their bid. Although very successful initially, bidders quickly learned how to tamper the mechanism through overbidding and underbidding manipulations.

2.3.2. Generalized Second-Price

In the Generalized Second Price (GSP), advertisers compete for an ad placements by submitting bids for a specific slot. The auction is designed to maximize the revenue for the publisher, while ensuring fair and efficient allocation of ads to the advertisers.

Formally, let there be n advertisers and $k < n$ ad slots. Each advertiser i has a private valuation $v_{i,j}$ for each ad slot j and makes a bid $b_{i,j}$ to the auctioneer, representing the maximum amount that they are willing to pay for such slot. Note that the bid $b_{i,j}$ is the value declared by the advertiser to the auctioneer and may not necessarily be the same as their true private value $v_{i,j}$.

Subsequently, the ad slots are ranked in order of relevance or importance, and each advertiser communicates a bid for each on of them. Denoting with α_i the click-through probability for each ad slot i, it is reasonable to assume that the top slots have a higher click-through probability:

$$\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k \tag{2.1}$$

Moreover, advertiser is characterized by a quality metric q_i indicating how much the ad sponsored by advertiser i is pertinent for the user. This metric is considered as the auctioneer wants to guarantee the display of meaning ads to the user.

The mechanism awards a single ad slot to each bidder, based on the product of their bid value b_i and quality score q_i. The bidder with the highest product $b_i * q_i$ is allocated the top slot, followed by the bidder with the second-highest product, and so on. Notice that each bidder is awarded with at most one ad slot.

Once the process is completed, if the user clicks on one of the allocated ads, the advertiser pays a price to the auctioneer. The payment to advertiser a is given by the quality q_{a+1} of the ad displayed below a divided by the quality q_a of the ad a, then multiplied by the bid value v_{a+1} of the ad displayed below ad a. This guarantees that the pay-per-click payment of advertiser a is not larger than his bid v_a.

$$p_a = \frac{q_{a+1}}{q_a} v_{a+1} \tag{2.2}$$

Finally, if the bidder does not win any slot or does not receive any click, the payment is equal to zero.

2.3.3. Vickrey-Clarke-Groves

The VCG (Vickrey-Clarke-Groves) advertising auction algorithm is a mechanism for allocating ad slots to advertisers in a way that maximizes the revenue of the auctioneer, while ensuring efficient and truthful bidding by the advertisers. The rationale of the VCG algorithm is based on charging to each advertiser the harm they cause to other bidders.

Formally, let there be n advertisers denoted by $i \in \{1, 2, ..., n\}$, and m ad slots denoted by $j \in \{1, 2, ..., m\}$. Each advertiser i makes a bid $b_{i,j}$ for each ad slot j, which represents the maximum amount the advertiser is willing to pay for that slot.

Under the VCG algorithm, the winning advertiser for each slot is determined by computing the outcome of the auction twice: once with the bidder in question present, and once with the bidder excluded. Hence, the price they pay is not the amount they had bid initially but only the harm their bid has caused to other bidders, which can be at most as high as their original bid.

In the auction, all the possible combinations of bids are considered by the algorithm, and the one maximizing the total sum of bids is kept. The price paid by advertiser i is computed comparing the social welfare generated by the allocation with and without the advertiser i, and then taking the difference between the two:

$$p_i = \max_{\omega \in \Omega} \sum_{j \neq i \in N} b_j(\omega) - \sum_{j \neq i \in N} b_j\left(\omega^*\right). \tag{2.3}$$

where $\omega^* \in \underset{\omega \in \Omega}{\operatorname{argmax}} \sum_{j \in N} b_j\left(\omega^*\right)$ represents the optimal allocation when the advertiser i is present. The formula consists of two terms: the first one represents the cumulative welfare of bidders when advertiser i is not present in the auction, while the second indicates the total welfare of the other bidders when i is present.

The VCG algorithm has the desirable property that truthful bidding is a dominant strategy, meaning that bidders will always achieve the best outcome by submitting their true valuations. However, the algorithm is computationally complex, and also assumes that advertisers have an unlimited daily budget.

2.3.4. Optional Second Price

The Optional Second Price (OSP) algorithm is used in display advertising, when three key actors are involved: Publishers, Ad Networks, and Ad Exchanges. When a user visits a website, the Publisher sends user information to all Ad Networks. Then, each Ad Network conducts an auction with a subset of advertisers and communicates the results to the Ad Exchange, providing the highest and the optional bid (potentially equal to the second-highest bid). Finally, the Ad Exchange considers the highest bids among all Ad Networks and determines the payment price based on the highest optional bid. Therefore, the OSP algorithm enables the Ad Exchange to increase revenue by encouraging truthful bidding and reducing the incentive for strategic behavior.

2.4. Targeting

The quality of an advertisement, which refers to the likelihood of the ad being clicked by a user, depends on various factors, such as the user's preferences, interests, and demographics. For example, in the context of search advertising, factors such as gender, geographic location, time, or language may play a crucial role in identifying the user's interests and optimizing the quality of the ad. On the other hand, in social advertising, the quality of the ad may depend on factors such as the user's interests, social connections, and online activities. By considering these factors, advertisers can create more targeted and effective ad campaigns that are more likely to reach their intended audience, thereby maximizing the click-through-rate of their ads. This represents a benefit not only for the advertisers, but also for consumers since they are exposed to advertisements of products in which they are more interested. Nowadays exist several types of targeting techniques [14] and among the most popular we find:

- **Contextual:** This type of targeting involves placing advertisement close to another content that relates to the advertised product. For instance, if a user is browsing a web page that sells musical instruments, the ads displayed on that page may present music-related products or services to align with the user's interests and needs.
- **Behavioral:** This kind of advertising relates to all those advertisements displayed based on the user's past behavior on the internet or on social media. The information used to target users may relate to previously purchased products, people followed on social media or clicked links. As it will be shown in the next section, behavioral targeting in the Metaverse may rely on top of novel and deeper measurements regarding the user behavior, thereby offering an even more precise behavioral-targeted content.
- **Geo-Targeting:** As the terms says, this targeting is based on the geographical position of the user. In particular, it is based on the rationale that the user's interest toward a certain advertised product, may change based on their location. As an example, when the user is walking around a certain area of the city, advertisements of local shops, restaurants and others might be displayed as the user may be more interested due to his physical proximity. Also this type of targeting is strictly related to the Metaverse context as the user's interests may evolve based on the virtual location of the user.

Advertising campaigns that utilize effective targeting techniques, provide several benefits both for the customers and advertisers perspective. The benefits of targeted advertising may range across several dimensions that also include customer fidelization and brand awareness[17]. Among these we find:

- **High Level of Personalization:** The huge amount of data that users leave in the web navigation or on social applications, represents an extremely useful resource for targeted advertising. In fact, this information is useful to infer both short-term interests, inferred by data such as search keywords, and long-term ones indicated by daily activities. When a brand can show that it really understands what its customers are looking for, people are more likely to respond positively to ads, resulting in an increased Click-Through-Rate (CTR).
- **Increased Brand Perception:** The exposure of users to advertisements represents an opportunity for the advertiser to reinforce its value perception to customers. In fact, through well designed advertising content, the users can develop a favorable impression of the brand, leading to increased brand loyalty and brand awareness.
- **Increased Engagement:** Effective advertising can lead to an increased customer engagement due to its ability to capture the audience's attention. By creating emotional appealing and eye-catching content, advertisers can generate interest in their products motivating customers to take action.

2.5. Advertising in the Metaverse

Recent advancements in augmented and virtual reality technologies have given rise to the concept of the Metaverse, a parallel world that overlaps with reality or offers complete immersion to users. The Metaverse is projected to be the next significant revolution after the Internet, and as a fully immersive world, advertising is expected to play a vital role in this ecosystem. Metaverse advertising will not only revolutionize the way users visualize advertisements, but also the way they interact with it. In fact, given the sensorial characteristic of Metaverse, the users are not passively exposed to ads but they may even be involved into it, interacting with the products that they may potentially buy. As an example, several fashion brands already designed and experimented interactive experiences for users to let them try on some of their new clothes collection in a virtual way. Additionally, users in the Metaverse can not only engage with newly advertised products but can also become part of the advertising themselves. Due to the sensory nature of the Metaverse, users can visually interact with each other by a means of virtual avatars. Hence, when a user's avatar is wearing a branded accessory, they can influence their social circle, acting as a form of advertising that already exists in the real world. Another remarkable revolution regards the availability and the quality of user's data.

In fact, the platform will be able to control what people see, hear and feel, allowing advertisers to gather more precise data regarding how user's react to the displayed ads. Still, to the best of our knowledge, there is no proposed model so far, and several scientific journals invite researchers to focus on this challenge:

- **Heller 2021** et al. [12] in 'The Problems with Immersive Advertising in AR/VR' shows that along the history of advertising, ads have become more interactive. In the Metaverse, advertising is experiential, immersive and personalized. Creators can determine what can be seen, heard and experienced by the user, creating innovative ways of "delivering" advertisements. Moreover, an augmented reality environment can provide a staggering amount of data such as eye-tracking or gesture-based controls, allowing new targeting techniques based on emotional or physiological responses. The availability of this new type of users' data would surely increase the probability of users to interact with the ad. In fact, if ads were to use more personal information and known emotional triggers, they would become irresistibly effective to entice the viewer to watch and absorb the content. As an example, a father would be more likely to click on an ad when this triggers in him family values. For example, Heller reports that a popular Duracell TV commercial featured a military dad missing his daughter and sending her a toy. The ad was selling batteries, but appealed to most parents' love for their kids. On the other hand, young people might be more interested in ads that trigger into them some feeling of belonging or social value. Heller notes that besides the access to more detailed and precise users' data, AR/VR systems offer another unique aspect, namely the potential for advertising to be delivered in a completely new manner. Advertisements could take the form of virtual pop-ups or billboards, or even make use of the users themselves as a promotional tool.

- **Kaur, Gupta 2021** [15] reflects in "Metaverse Technology and the Current Market" how Metaverse is the next big revolution in which every user is replicated by a virtual avatar that resemble their physical real-world characteristics. He also remarks that in the future, cryptocurrencies will likely become the primary payment method used in Metaverse platforms, and every virtual asset ownership will be verified by Non-Fungible Tokens (NFTs). The paper shows the various Metaverse platforms available, such as Decentraland, a virtual gaming platform built on top of the Ethereum blockchain, where users can create, explore, and monetize their content within a virtual environment. In particular, Decentraland provides a Metaverse experience in which users can own virtual lands, interact with each other, and utilize their creativity to create custom experiences. The platform's native cryptocurrency is MANA, which can be used to purchase virtual assets, including lands within the virtual world.

- **Taylor, 2022** [26] highlights the growing use of the term "Metaverse" in households across the world, particularly after the Facebook's rebranding as "Meta." While it is uncertain what the metaverse will look like in the future, it is clear that augmented and virtual reality, NFTs, cryptocurrency, and digital advancements will contribute to a more advanced "virtual world" in which advertising will play an important part. Therefore, the editorial encourages more research on advertising in the metaverse, highlighting the importance of early academic research in digital advertising. The article suggests that existing studies on virtual reality, augmented reality, and 3-D advertising can provide a base for future research, and recent articles on virtual reality can provide a valuable background information. Theories of digital advertising have also advanced considerably over the past several years, providing an excellent base for researchers interested in advertising the metaverse. The editorial calls for submissions on this topic to the International Journal of Advertising.

- **Bonetti** [2] analyses how the rapid emergence of technologies such as AR and VR is enhancing the selling environment, creating a completely renovated shopping experience that potentially increases revenues, while ensuring a fun and immersive shopping time. In fact, the experiencial value is increased thanks to the product simulations offered to the users to directly test items before buying them. On the other hand, the author analyses some critics that say that this technology is more a tool used to gain customers' attention than a viable in-store solution. Nevertheless, in addition to the enhanced user experience, benefits also include overcoming operational barriers, saving time and cutting costs.

- **Duan 2021** [5] in "Metaverse for Social Good: A University Campus Prototype".

  Recently, many tech companies have taken a keen interest in the metaverse and have invested in developing AR/VR technologies. The paper's main contribution is the proposal of a three-layer metaverse architecture: infrastructure, interaction, and ecosystem. The infrastructure layer includes physical technologies, computational power, blockchain, and storage. The interaction layer facilitates the user interaction with the virtual environment, it represents the boundary between the physical and virtual world. Finally, the ecosystem layer forms the core of the metaverse where everything is virtual. To illustrate the proposed architecture, the authors have implemented a metaverse platform prototype that represents the campus of the Chinese University of Hong Kong.

- **Ning 2021** [20] "A Survey on Metaverse: the State-of-the-art, Technologies, Applications, and Challenges".

  This paper provides a comprehensive analysis of the Metaverse. The authors define the Metaverse as a "mirrored image of the real world", and explore the necessary technological requirements and potential applications of the Metaverse. The paper highlights how big companies such as Microsoft, Amazon, and Facebook are investing heavily in VR technologies and have already developed several prototypes. For example, Amazon has been working on a new VR technology since 2018 to enhance the shopping experience for its users, allowing them to interact with digital products in a new virtual shopping mall. As a new type of internet application, the Metaverse is characterized by the integration of various characteristics like hyper spatio-temporality due to its parallel world nature and sociality. The paper also explores potential applications of the Metaverse, including remote offices, virtual gaming platforms and digital entertainment.

2.5.1. Advertising Formats

In the preceding section, we discussed how advertising in the Metaverse engages with users in multiple ways. The previous researches illustrated the numerous and innovative formats for advertising in the Metaverse, which can be categorized as follows:

- **Virtual Billboards**: billboards that can be placed in almost any virtual location including during an events like a football match streamed through the Metaverse. In contrast to the physical world, where only a single billboard can be used, virtual billboards can be modified and personalized for each user.

- **Experience:** a personalized experience can be proposed to the user to advertise a product or a service. For example, a user could try on a new pair of shoes in a virtual showroom or could feel the driving experience of a new BMW model, totally in a virtual manner.

- **Pop-up:** pop-up ads can appear during a Metaverse experience, just like in the traditional web browser navigation. Unlike web pop-ups, which are fixed-size banners, virtual reality pop-ups can be displayed in a fully immersive way.

- **Implicit Advertising:** Given the total control of the metaverse platforms on what the user see, hear or feel, then advertising content can be placed within every moment of the virtual experience. For example, random elements such as a JustEat rider or a Ryanair plane in the sky can serve as implicit forms of advertising.

2.6. Preliminaries

This section aims at providing some notion basis before diving into the more technical aspects of the work. Here, we present some preliminary concepts that are essential for a complete comprehension of the developed model and algorithms.

2.6.1. Truthful Mechanisms

As reported in the previous sections (2.3), auction mechanisms might present non-truthful mechanisms, meaning that bidders may behave dishonestly, leading to a low social welfare. On the other hand, other allocation mechanisms such as the VCG algorithm are truthful (under certain circumstances), meaning that truthful bidding is the dominant strategy for each bidder. Hence it encourages the bidders to make a bid that matches their real value of the slot their are competing for.

In order to understand whether or not a truthful mechanism can be implemented we resort to the use of the Myerson's lemma. Below are reported useful definitions that allow to gradually understand how the Myerson's lemma guarantees truthful mechanisms.

Definition 1.(*Economic mechanism*)

An economic mechanism is a tuple $(A_1, ..., A_n, X, g)$ where:

- A_i is the set of actions of player i,
- X is the set of outcomes,
- $g : A_1 \times ... \times A_n \to X$ is the outcome function.

In a single–item auction, the space of outcomes X corresponds to the space of players N, each outcome corresponding to the player who wins the item (the ad slot).

Definition 2.(*Social Choice Function*)

A social choice function $f : \Theta_1 \times ...\Theta_n \to X$ assigns an outcome x to each possible profile of players' types.

Definition 3.(*Direct (revelation) economic mechanism*)

Given a social choice function $f : \Theta_1 \times ...\Theta_n \to X$, a direct (revelation) economic mechanism is a mechanism $(\Theta_1, ...\Theta_n, X, f)$.

Definition 4.(*Incentive Compatibility*)

A social choice function $f : \Theta_1 \times ... \times \Theta_n \to X$ is incentive compatible (or truthfully implementable) if the Bayesian game induced by the direct revelation economic mechanism $(\Theta_1, ..., \Theta_n, X, f)$ has a pure equilibrium (according to some solution concept) $(s_1^*, ..., s_n^*)$ such that $s_i^*(\theta_i) = \theta_i$ for every player i and type θ_i. (where $s_i^*(\theta_i)$ denotes the optimal strategy of player i when her type is θ_i.)

The type θ_i expresses the valuation for player i in having the item (the ad slot). Hence, in an incentive compatible social function, the encouraged strategy of the players is to bid the real value.

Definition 5.(*Single-parameter linear environment*)

A single-parameter linear environment (a special subclass of quasi-linear environment) is characterized by:

- outcomes space: $X = \{(k, p_1, \ldots, p_n) : k \in K, p_i \in \mathbb{R}\}$.
- utility functions: $U_i(x, \theta_i) = U_i((k, p_1, \ldots, p_n), \theta_i) = \theta_i \cdot \rho_i(k) - p_i$, where $\rho_i : K \to \mathbb{R}$ and $\Theta_i \subset \mathbb{R}$.

The set K represents the set of allocations. The utility of each player i is factorized w.r.t. type θ_i and a function $\rho_i(k)$ defined only on the allocation k (and not on the type.) In principle, function ρ_i can be any (in the case of auctions $\rho_i(k)$ is 0 is $k \neq i$ and 1 otherwise, indicating whether player i won the ad slot). (The name "linear" is due to the form of the utility function, that is linear in the type and in the payment.) The term p_i represents the monetary payment of player i to the mechanism.

Definition 6.(*Weak Monotonicity*)

An allocation function $k(\theta)$ is weakly monotonic if:

$$\theta_i > \theta_i' \Rightarrow \rho_i(k(\theta_i, \theta_{-i})) \geqslant \rho_i(k(\theta_i', \theta_{-i})) \quad \forall i \in N, \theta_i, \theta_i' \in \Theta_i, \theta_{-i} \in \Theta_{-i}$$

where

- θ_i and θ_i' are two different types of player i.
- θ_{-i} represents the types of all the other players

In the case of advertising, this means that if a player i raise its bid for a certain slot, it can only make its position better in winning the auction. Conversely, awarding the good to the second-highest bidder is a non- monotone allocation rule since if you are the winner and raise your bid high enough, you lose.

Theorem(*Myerson's characterization*) A necessary and sufficient condition for a social choice function in single-parameter linear environment to be incentive compatible is that the allocation function is weakly monotonic.

For this reason, when designing allocation algorithms, we investigate whether they satisfy Myerson's weakly monotonicity property. In fact, since the Metaverse model is a one parameter (θ_a) linear environment, then Myerson's weakly monotonicity is *necessary and sufficient* for creating a truthful mechanism in dominant strategies.

2.6.2. Classes of Complexity

The definition of the complexity classes are useful to determine the belonging complexity class of the provided algorithms in section 5. In theoretical computer science, complexity classes are used to group problems with similar computational complexity. The most common measure of computational resources is the amount of time or space required to solve a problem, and complexity classes are defined based on these resources.

Polynomial-time Algorithm. An algorithm is said to be of polynomial time if its running time is upper bounded by a polynomial expression in the size of the input for the algorithm, that is, $T(n) = O(nk)$ for some positive constant k

The main complexity classes are:

- **P**: The set of decision problems that can be solved by a deterministic machine in polynomial time and usually their solution is considered easy to find.
- **NP**: The set of decision problems for which the correctness of the solution can be verified in polynomial time by a deterministic machine, but the solution is hard to find since they are being solved by a non-deterministic machine.
- **NP-hard**: The set of decision problems that are at least as hard as the hardest problems in NP. Furthermore, a problem is classified as NP-Hard when every problem in NP can be reduced in polynomial time to it.
- **NP-complete**: The set of decision problems that are both in NP and NP-hard. It means that NP-Complete problems can be verified in polynomial time and that any NP problem can be reduced to this problem in polynomial time.

As mentioned above, some of the above classes include problems that are hard to solve, hence there might be algorithms that offer an approximated solution of the optimal one. Here are described the main approximation classes of problems i computational theory:

- **APX:** it is defined as the set of problems that belong to NP and can be approximated within a polynomial-time algorithm that can find a solution that is within a constant factor of the optimal solution.

- **APX-Hard:** a problem is said to be APX-hard if there is a PTAS reduction from every problem in APX to that problem.
- **APX-Complete:** a problem is defined as APX-Complete is if it is both APX-hard and APX. Notice that APX-completeness provides an analogue of NP-completeness.
- **Poly-APX (polynomial-time approximation scheme)** is a class of optimization problems for which there exists a polynomial-time algorithm that can find a solution that is within a polynomial factor of the optimal solution.

Hence, an c-approximation algorithm finds a solution that is c times worse than the optimal one, where c is called *approximation ratio*. Notice that for maximization problems $c \leq 1$ the approximation algorithm can achieve a solution that is as good as the optimal one, while for minimization algorithms $c \geq 1$.

2.6.3. Linear Programming

Integer Linear Programming (ILP) is a mathematical optimization technique that can help solve real-world problems in various fields, such as engineering, economics, and management. In simple terms, ILP involves finding the best possible solution to a problem, subject to certain constraints, where the variables in the problem can only take on integer values. In more technical terms, ILP involves the optimization of a linear objective function, subject to linear constraints and integrality constraints. The linear objective function represents the quantity to be maximized or minimized, such as profit or cost, and the linear constraints represent limitations or restrictions on the variables in the problem, such as resource availability or capacity constraints. The integrality constraints, on the other hand, ensure that the variables in the problem take on only integer values. Solving an ILP problem typically involves using algorithms that search through the space of possible solutions to find the optimal solution that satisfies the constraints. These algorithms can be quite complex and require advanced mathematical and computational techniques. Overall, ILP is a powerful tool that can help decision-makers optimize their resources and achieve their goals by providing a rigorous and efficient approach to solving complex problems with integer-valued variables.

3 | User Model in the Metaverse Scenario

This chapter provides a high-level overview of the proposed user model, which includes a representation of the virtual locations in the Metaverse and the possible paths that users may traverse, represented through a tree-like structure. Additionally, the chapter introduces the user's attention model, which explains how ads influence each others when displayed within scenes, as long as the user is able to recall previously seen ads. Furthermore, it is given a representation of how the allocation of advertisements is organized through slates and slots.

3.1. Scenes

In the Metaverse, users can enjoy an augmented reality experience that allows them to explore a variety of virtual places, referred to as "scenes.". Every scene offers a unique space for users to engage in various activities, such as immersive football viewing, fashion events, or even job meetings. As in the real world, users are free to move from one virtual location to another, hence, the Metaverse's scenes are organized in a graph-like structure that represents how virtual places are interconnected with each others. This is modeled with the users able to move among the graph's nodes according to some transition probability that indicates the likelihood of the user to move from one scene to another. As an example, they could move from a work environment to a shopping experience, or from a virtual concert to an immersive video game experience.

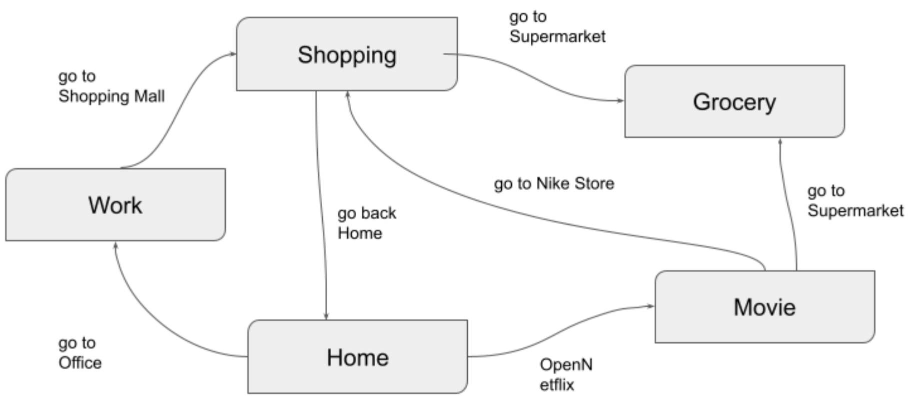


Figure 3.1: Scenes Graph in the Metaverse Model.

3.2. Advertising in the Scenes

As users traverse between scenes and engage in various activities within the Metaverse, they may encounter targeted advertisements for virtual and even real-world products. These advertisements may take the form of immersive pop-ups, virtual billboards, or subtle product placements, as illustrated in section 2.5.1. For instance, a user could look at the virtual sky and come across a virtual representation of a Ryanair flight showing a special promotional offer sponsored by the company. Another example could be the customization of virtual billboards during an immersive football match, which could be personalized based on the user's preferences and interests. To achieve this, each scene within the Metaverse can be allocated with advertisements, organized into slates that may contain several slots for different ad formats. However, to avoid overwhelming users with excessive exposure to advertisements, we suggest leaving some slots empty to mitigate the effects of ad fatigue. This approach allows for a balanced integration of advertisements within the Metaverse, ensuring that users have a positive and engaging experience without feeling bombarded with unwanted marketing messages.

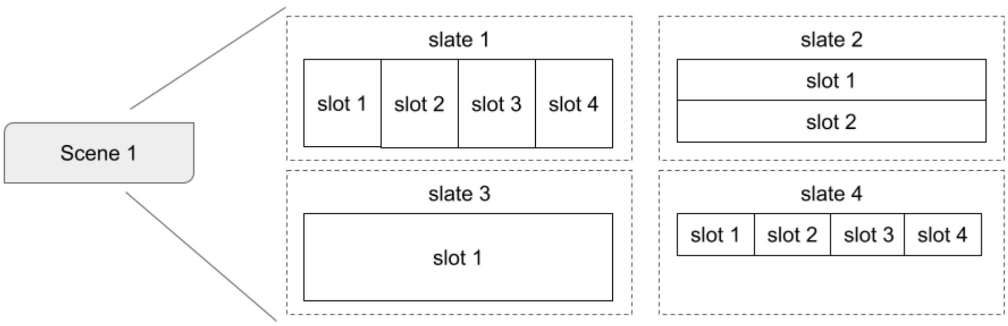


Figure 3.2: Advertising Structure with Slates and Slots.

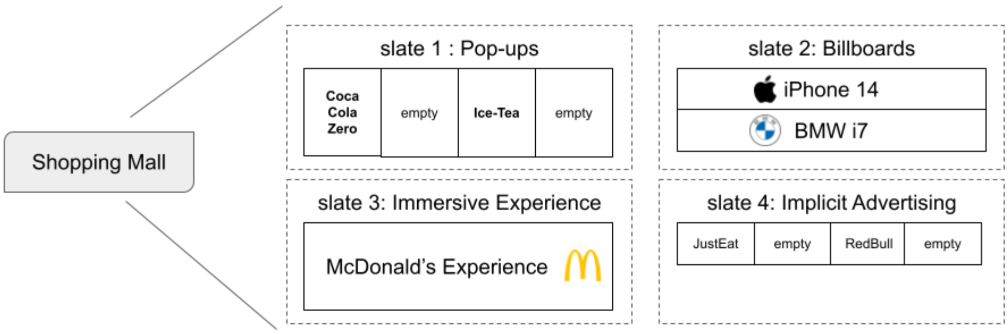


Figure 3.3: Advertisements Allocation in a Virtual Shopping Mall.

Figures 3.2 and 3.3 show respectively the proposed allocation structure and an example of how advertisements could be allocated during a shopping experience in the Metaverse. As the user navigate through a virtual shopping mall, they may encounter advertisements for products such as iPhone or BMW displayed on the walls, like in the physical world.

3.3. User's Attention Model

The impact of advertisements on consumers can vary depending on the context in which they are viewed. Users are continually exposed to thousands of stimuli when targeted by advertising mechanisms. Therefore keeping and understanding the user's attention is a crucial factor in advertising allocation. In addition to the features used in social advertising (e.g., declared interests, age, gender, location, time), the user's attention in the Metaverse scenario may depend on:

- **Current Scene**: The effectiveness of an ad can vary depending on the scene in which it is displayed (same concept of Geo-Located Targeting (section 2.4)). An ad may be effective in one context, such as when a user is visiting a city or experiencing an event, while being ineffective in another context.
- **History of user's actions and scenes**: The actions that a user performed while moving through different scenes (i.e. conversions or tasks), can provide insights into their preferences and inform predictions about their behavior in the future.
- **Other displayed ads**: The attention of a user towards a particular ad can be influenced by the presence of multiple ads displayed simultaneously or by ads displayed in the past. These interdependent effects are referred to as externalities in the field of economics.

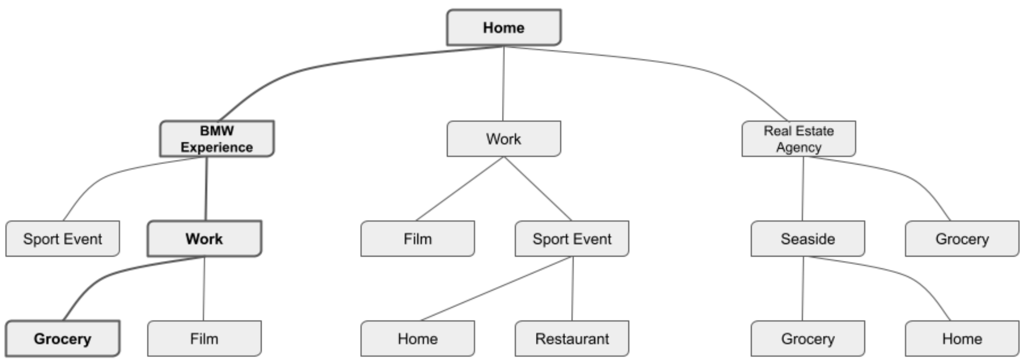


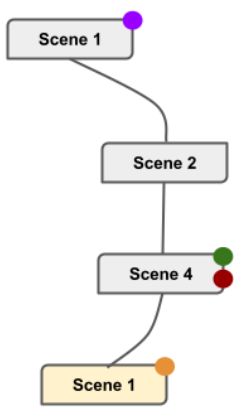
Figure 3.4: Example of Quality History Dependence.

The effectiveness of an advertisement in a given scene may be influenced by the user's recent history. In particular, the quality of an ad in a scene can depend on the last k scenes that the user has visited, where k represents the length of the user's memory.

As long as the user is able to recall the ads visualized in the previous k scenes, externalities may arise among advertisements. Externalities can be classified as positive, negative, or neutral depending on their impact on the user's attention. Positive externalities occur when the presence of other ads enhances the user's attention to the advertised product, while negative externalities occur when other ads diminish the user's attention. Neutral externalities, on the other hand, have no effect on the user's attention towards the ad. The overall quality of an ad allocated in a scene is thereby factored in three terms: the sequence of previously traversed scenes, the externalities due to previous ads, the externality slate/slot.

The example below shows a possible allocation of ads along a path of the tree and highlights the externalities arising among the allocated ads. In particular, we imagine to assign the following advertisements:

- *ad1*:BMW Driving Experience
- *ad2*:BMW i7 Car in the Road
- *ad3*:Mercedes GLC Billboard
- *ad4*:Ryanair Offers Pop-up



	ad 1	ad 2	ad 3	ad 4
ad 1		1.8	0.7	1
ad 2	1.8		0.8	1
ad 3	0.7	0.8		1
ad 4	1	1	1	

$$\underbrace{q_a(S1|S4,S2,S1)}_{\text{scenes sequence}} \cdot \underbrace{\gamma_{a_1,a}\gamma_{a_4,a}\gamma_{a_2,a}}_{\text{ad externalities}} \cdot \underbrace{\Lambda_{S_1,s}}_{\text{slate/slot externality}}$$

Figure 3.5: Example of Allocation with Externalities in a Path.

In the figure 3.5 above, the ads' interdependencies among the ads might be interpreted as follows:

- BMW ads positively affect each other, driving user's attention to the BMW i7.
- BMW-Mercedes ads have negative impact on the user's attention, as its attention is divided among two competitors for the same product.
- Cars advertisements and Ryanair promotions have no significant influence on the attention of the user as they are unrelated.

A more realistic rationale regarding the allocation of advertisements would allow for the possibility of displaying the same ad multiple times during their browsing experience. This is predicated on the belief that if a user encounters an ad and does not engage with it, they may still do so at a later time. Conversely, if a user interacts with an ad, it is improbable they will do so again in subsequent visits. As shown in the problem formulation section (see 4), the quality of an advertisement is reduced when it is displayed multiple times along a path.

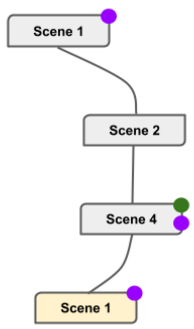


Figure 3.6: Example of Allocation.

Moreover, from the advertiser perspective, the value of allocating an ad in a certain scene of a path, depends on the likelihood of the user to reach that scene following all the path. This probability is calculated by multiplying the transition probabilities between the scenes that make up the path.

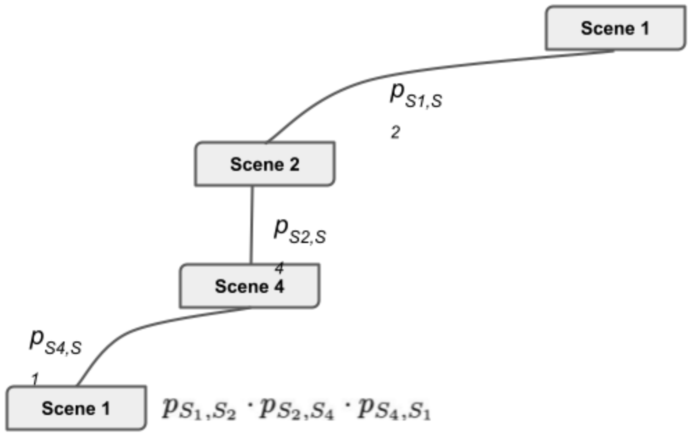


Figure 3.7: Example of Multiple Ad Allocation along a Path.

3.4. Ad Expected Value

As discussed in the previous paragraphs, the value generated by the display of an advertisement in a certain scene is affected by several factors regarding the structure of the Metaverse, the interdependencies among ads and the ability of the user to recall previously allocated ads. These factors collectively influence the conversion probability, which represents the likelihood of a user clicking on an advertisement. The conversion probability is factorized in the following manner:

$$\underbrace{q_a(S1 \mid S4, S2, S1)}_{\text{quality scenes sequence}} \cdot \underbrace{\gamma_{a_1,a}\gamma_{a_4,a}\gamma_{a_2,a}}_{\text{ad externalities}} \cdot \underbrace{\Lambda_{S_1,s}}_{\text{slate/slot externality}} \cdot \underbrace{p_{S_1,S_2}p_{S_2,S_4}p_{S_4,S_1}}_{\text{user transitions}} \tag{3.1}$$

where

- **quality scene sequence** represents the quality of the ad a in the scene $S1$ after the user traversed the path made of scenes $S1, S2, S4$.
- **ad externalities** indicates the total effect of externalities generated by the ads previously allocated in the path.

- **slate/slot externality** indicates the effect of allocating the ad in a certain slot of a particular slate.
- **user transitions** indicates the probability of the user to traverse the path, transitioning among all its scenes.

Finally, once the user has clicked on the displayed ad, the conversion value for the advertiser is represented by θ_a.

4 | Problem Formulation

The user model discussed in the previous chapter is formalized in this chapter, illustrating its mathematical formulation, with a focus on the tree-like structure and the computation of transition probabilities between nodes. In addition, externalities and user memory are formalized to define the "Total Externality" function, "Non-Convert Probability," and "Adjusted Quality." Finally, the study presents the Objective Function representing the allocation value to be maximized.

4.1. Scenes Tree and User Transitions

The user moves through a sequence of possible scenes starting from the root of a tree of scenes and following paths based on the probability of reaching one scene after the other. In other words, a tree of scenes represents a set of all possible paths that a user can make, starting from the root scene. Formally, given a *set of scenes* S, we define $T = (S, \rho)$ a *tree of scenes* and $\rho : S \implies \mathcal{P}(\mathcal{S})$ is the *successor function* that takes as input a scene s and returns the subset $\rho(s)$ made of all the immediate successors of scene s in the tree. We also introduce *terminal scenes* that are all the scenes such that $\rho(s) = \emptyset$. We define also a *transition probability* $\pi_{s,s'} \in [0, 1]$ where $s \in S, s' \in \rho(s)$, the *transition probability* of the user to move from scene s to *s'*.

For every non-terminal scene $s \in S$, the transitions probability are constrained as follows:

$$\sum \pi_{s,s'} = 1 \tag{4.1}$$

We also denote with σ an *ordered sequence* from the root scene to any scene-node in the tree, such that σ_i is the i-th scene in the sequence. In particular we define as σ^s the sequence of scenes from the root to the scene $s \in S$ and $|\sigma^s|$ indicates its length. Each i-th node in the path to scene s is indicated as σ_i^s, where $i \in [|\sigma^s|]$. Therefore, for every $s \in S$ the root scene corresponds to σ_1^s and s to $\sigma_|\sigma^s|$.

We also formally introduce the *reach probability* indicating the probability of the user to reach scene s starting from the node σ_1^s:

$$\Pi^s = \prod_{i=1}^{|\sigma^s|-1} \pi_{\sigma_i^s, \sigma_{i+1}^s} \tag{4.2}$$

4.2. Ads and Qualities

We denote with A the set of ads that need to be allocated in the tree. For simplicity, we assume that at most one ad can be displayed in every scene. We also introduce a fictitious $a_\emptyset$ that indicates that no ad has been allocated to the scene where it is placed. We denote with $x : S \to A \cup a_\emptyset$.

Every ad $a \in A$ allocated in scene $s \in S$ is characterized by $q_{a,s} \in [0,1]$, indicating the user's *conversion probability* when ad a is allocated to scene s when the user has already reached the scene s and no other ad has been displayed in the path to s. In other words, it indicates the conversion probability without considering that the user first needs to reach the scene before clicking on the ads and without considering any interactions previously placed. For simplicity, whenever we focus on scene-independent quality, we use q_a in place of $q_{a,s}$.

4.3. Externalities and User Memory

As previously discussed, the user's attention is influenced by several factors, including the ones of previously allocated ads, called *forward externalities*.

The effect of externalities is 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'. We formally define the externality between ads $a, a' \in A$ with $\gamma_{a,a'} \in [0,1]$.

We assume that $\gamma_{a,a'} \leq 1$ for every $a \neq a' \in A$, while $\gamma_{a,a} = 1$ for every $a \in A$. In other words, the externality among the same ad a placed in different scene-nodes is neutral. We remind that when $\gamma_{a,a'}1$, the externality is negative, meaning that the display of ad a before a' negative affects the quality of ad a', while $\gamma_{a,a'} = 1$ means the externality is neutral, that is, there is no alteration in the quality of a' when a is placed before.

Leaving a scene without any allocated, that is placing the ad $a_\emptyset$, does not introduce any externality to the ads placed in following scenes. Therefore $\gamma_{a_\emptyset,a'} = 1$ for every $a' \in A$

Additionally, we make the assumption that the user may forget the previously viewed ads. Specifically, the user's actions are solely influenced by the ads they have seen in the $k \in N$

preceding scenes, where $k = 0$ denotes complete forgetfulness of all previously viewed ads.

The *Total Externality* to which an ad a in scene s is subject to is defined as:

$$\Gamma(x, s) = \prod_{i=max1,|\sigma^s|-k}^{|\sigma^s|-1} \gamma_{x(\sigma_i^s),x(s)} \tag{4.3}$$

The overall externality considers the externalities of the ads assigned in the scenes preceding s in the path σ^s, taking into account the user's memory k. Specifically, σ^s comprises the $mink, |\sigma^s| - 1$ scenes, which begin from s and move backward to the root, spanning k steps. It is noteworthy that when the user memory is $k = \infty$, the user is capable of perfectly remembering all the ads previously viewed.

4.4. Adjusted Quality and Non Converting Probability

As already mentioned above, the quality of an ad a in scene s depends on its intrinsic quality $q_{a,s}$ and on the total externality $\Gamma(x, s)$ received from previously allocated ads. This holds whenever ad a is not displayed in scenes preceding s.

In the literature on search and social advertising, an ad can be displayed only once in an allocation (see Kempe and Mahdian (2008) [16]). The reason is that the users clicks on an ad only the first time it is displayed. From our understanding, this model does not exactly capture the actual behavior of the user. In fact, when the same ad a is displayed multiple times along the path to scene s the user may behave in different ways. If the user sees an ad and clicks on it, then it is very likely they will never convert again in the future since they might not repeat the conversion or use different channels different from the advertising to repeat the conversion.

In our model we assume that if an ad a has already been clicked by the user, then he will never click again on it. On the other hand, if the ad a has not been clicked, then the user might still click on a in future scenes. Mathematically, this means re-adapting the quality of ad a in scene s when it is displayed in another scene s' preceding s.

In particular, we denote with $H(X, s) \subseteq S$ the subset of scenes s' along the path σ^s in which ad $a = x(s) = x(s')$ has been allocated, excluded scene s. The user's behavior is mathematically represented by the introduction of the *non convert probability*:

$$\Xi(x,s) = \prod_{s' \in H(x,s)} \left(1 - \Gamma(x,s')q_{x(s'),s'}\right) \tag{4.4}$$

The non convert probability indicates the likelihood that the user never clicks on ad $a = x(s)$ when it is allocated in scenes s' strictly before scene s conditioned to the reach of s'.

Finally, given the allocation in the preceding scenes in the path to s, we introduce the *adjusted quality* of the ad allocated in s:

$$\tilde{q}(x,s) = \Gamma(x,s)q_{x(s),s}\Xi(x,s) \tag{4.5}$$

Thus, the *conversion rate* of ad a allocated in scene s is $\Pi^s \tilde{q}(x,s)$.

4.5. Objective Function

We denote with θ_a the *value per conversion* of ad a. Finally, the *allocation expected value* of allocation x is:

$$\sum_{s \in S} \left(\Pi^s \tilde{q}(x,s)\theta_{x(s)}\right) \tag{4.6}$$

The allocation expected value represents the total value of the allocation in the tree of scenes. It represents the function to be maximized in the META allocation problem.

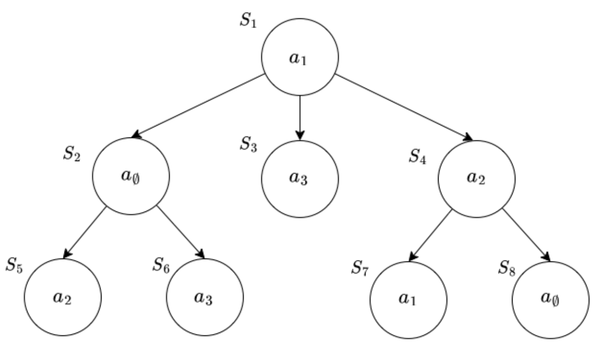


Figure 4.1: Example of the Metaverse Advertising Model.

Example 1.

Figure 4.1 shows an instance of the Metaverse Advertising Model in which the set of scenes is $S = \{s_1, .., s_8\}$ and the set of ads to be allocated is $A = \{a1, a2, a3\} \cup a_\emptyset$. The quality of the ads is $q_{a,s} = 0.1$ for all $a \in A$ and $s \in S$, the externalities are $\gamma_{a_1,a_2} = \gamma_{a_1,a_3} = \gamma_{a_2,a_3} = 0.8$ and the values per conversion are $\theta_{a_1} = 0.5$, $\theta_{a_2} = 0.6$, $\theta_{a_3} = 0.7$. The transition probabilities are $\pi_{s_1,s_2} = \pi_{s_2,s_6} = \pi_{s_4,s_7} = 0.7$, $\pi_{s_1,s_3} = 0.1$, $\pi_{s_1,s_4} = 0.2$, $\pi_{s_2,s_5} = \pi_{s_4,s_8} = 0.3$. The user's memory is set to $k = 2$.

As an example, consider scene s_7 where the total externality come from the ads placed in the path to s_7: $\Gamma(x, s_7) = \gamma_{a_1,a_1}\gamma_{a_2,a_1} = 0.8$. 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 $\Pi^{s_7}\tilde{q}(a_1, s_7)\theta_{a_1} = 0.00504$.

The allocation expected value is $\sum\limits_{s \in S} \Pi^s \tilde{q}(x, s)\theta_{x(s)} = 0.10776$

Notice that in this case the user memory is set to $k = 2$ thereby being the tree of $depth = 2$ the user receives externalities of all the previously allocated ads. If $k = 1$ the allocation expected value increases to 0.11714 as the negative effects of externalities are received only to scenes one level above the current one.

Proposition 4.1. *In the metaverse single-parameter environment, an allocation mechanism M that maps a type profile (θ_a) where $a \in A$ to an allocation x is weakly monotone if for every ad $\hat{a}$ and types $\theta_{a'}$ of the other ads $a' \in A \setminus \hat{a}$ the allocation mechanism M is such that the term $\sum_{s \in S: x^{\theta_{\hat{a}}}(s) = \hat{a}} \prod_{s'} \tilde{q}(x^{\theta_{\hat{a}}}, s')$ is non-decreasing in $\theta_{\hat{a}}$, where $x^{\hat{\theta}_{\hat{a}}} = M((\theta_a)a \in A)$ is the allocation returned by the mechanism with type profile $(\theta_a)a \in A$.*

5 | Finding the Optimal Allocation: Approximation Algorithms

This chapter illustrates greedy allocation algorithms for the proposed user model, showing their theoretical guarantees based on the specific scenario in which are applied. Specifically, the degree of approximation is influenced by factors such as whether the advertisement qualities are dependent on the scenes as well as the existence of any externalities among advertisements.

5.1. Greedy Approach General Idea

To understand the concept of Greedy Algorithms, it is important to first gain an understanding of the problems that these algorithms are designed to solve. Greedy Algorithms are utilized in solving optimization problems where the objective function must be either maximized or minimized. In these types of problems, Greedy Algorithms make the best possible decision at each step by selecting the locally optimal choice. However, it is important to note that Greedy Algorithms do not necessarily guarantee a globally optimal solution, as they do not consider the future consequences of their current decision. Instead, these algorithms select the decision that appears to be the most favorable in the moment. Although the local optima solution may not always be equivalent to the global optimum, Greedy Algorithms can often find a solution in a reasonable amount of time, provided certain conditions are met.

In Figure 5.1, a comparison is made between the Greedy solution and the globally optimal solution for the largest path problem. The Greedy Algorithm initially selects the path on the right-hand side, as it appears to be the most beneficial at the moment. However, this choice ultimately results in a suboptimal solution, as can be observed in the figure.

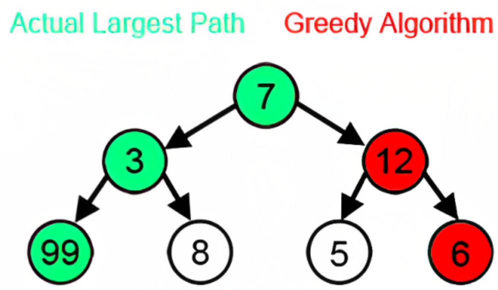


Figure 5.1: Greedy and Exact Solution in the Largest Path Problem.

The Greedy Algorithm's choice leads to a total sum of 25, while the globally optimal solution results in a sum of 109. This example illustrates the potential drawback of Greedy Algorithms, which prioritize local optimization over global optimization.

5.2. Poly-Time Algorithm for META-SI-NE

In the META-SI-NE scenario, there are no externalities among allocated ads and the quality of the ads does not depend on the scene where it is allocated. This case differs from the allocation problem in classical ad auctions because: the allocation might be on a tree instead of a line and an ad can be displayed multiple times, hence introducing externalities.

While the context of this model may not fully replicate a real-world situation, our findings indicate that it is feasible to construct a polynomial-time greedy algorithm for it, that plays a fundamental role in solving others broad and more complex instances. The pseudocode is reported in Algorithm 1.

The algorithm works iteratively, assigning ads to the scene that have empty slots.

We define R, the set represents the subset of scenes that at each iteration, remain unassigned to an ad. In each iteration, the algorithm selects a scene-ad pair $(s^*, a^*) \in S \times A$ that optimizes the expected value of allocating a^* in scene s^*, considering also the externalities introduced allocating a^*. To identify the singular pair chosen during each iteration from all conceivable pairs that maximize the value, we establish a tie-breaking rule.

Proposition 5.1. **Tie-Breaking-Rule**: *Let $\overline{P}$ be the set of pairs $(\overline{s}, \overline{a})$ returned by*

$argmax_{a \in S, a \in A} \Pi^s \tilde{q}(x,s)\theta_a$. Whenever $\overline{P}$ is not a singleton, break ties by assigning to (s^*, a^*) any pair $(\overline{s}', \overline{a}')$ such that $\mid \sigma^{\overline{s}'} \mid$ is the minimum among all $\mid \sigma^{\overline{s}} \mid$ where $(\overline{s}, \overline{a}) \in \overline{P}$ for some $\overline{a} \in A$.

In our context, for the same allocation value the Tie-Breaking-Rule prefers assigning ads to the closest scenes to the root . Once the value-maximizing pair (s^*, a^*) has been found, the algorithm allocates a^* to s^*

Algorithm 5.1 Algorithm 1: GREEDY

```
Inputs: set of scenes S, set of ads A
Initialize R ← S, x(s) ← a_∅ ∀s ∈ S
while R ≠ ∅ do
    (s*, a*) ← argmax_{a∈S,a∈A} Π^s q̃(x, s)θ_a
    Ties are broken according to Definition
    x(s*) ← a*
    R ← R \ s*
end while
return x(.)
```

Theorem 5.1. *Algorithm 1 computes an optimal solution to the META-SI-NE problem.*

Proof. It is important to show that the expected value of an allocation that the expected value of an allocation x can be decomposed into a component for each possible path. Formally, we show that

$$\sum_{s \in S} (\Pi^s \tilde{q}(x,s)\theta_{x(s)}) = \sum_{s \in S : \rho(s) = \emptyset} \Pi^s V_s(x) \tag{5.1}$$

where $V_s(x) = \sum_{s' \in \sigma^s} (\tilde{q}_{x,s'}\theta_{x(s')})$. 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

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

It can be observed that due to the Tie-Breaking rule, the algorithm assigns ads to nodes from top to the bottom of the tree. Let's suppose by contradiction that Algorithm 1 assigns an ad a to node s_1, such that there exist another scene s_2 in σ^{s_1} (with $s_2 \neq s_1$) that has not been assigned with any ad (i.e. $x(s_2) = \emptyset$). Then, the value $\Pi^{s_2}\tilde{q}(x,s_2)\theta_a \geq \Pi^{s_1}\tilde{q}(x,s_1)\theta_a$ since s_2 is in the path to s_1 and in case of same allocated value, the tie-breaking assigns a to s_2.

Furthermore, given an allocation x^* of the Algorithm 1, a node s and another allocation x' such that $x^*(s') = x'(s')$ for all the $s' \in \sigma^s$. Then, it follows that:

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

This is a very immediate consequence of assigning at each step the ad a to scene s that maximize the allocation value. Also, the value of this assignment is not influenced by the partial allocation of previous scenes s'.

Let x^* be the allocation returned by Algorithm 1. The following steps show that its allocation is optimal for each possible path. Given a terminal node $\bar{s}$ and an optimal allocation x_0 for the path that terminates in$\bar{s}$, i.e. $x_o \in argmax_x V_{\bar{s}}(x)$ we show that $V_{\bar{s}}(x^*) \geq V_{\bar{s}}(x_0)$. In particular:

$$\begin{aligned}
\sum_{s\in S}\left(\Pi^s \tilde{q}\left(x^*,s\right)\theta_{x^*(s)}\right) &= \sum_{s\in S:\rho(s)=\emptyset} \Pi^s V_s(x) \\
&\geq \sum_{s\in S:\rho(s)=\emptyset} \Pi^s \max_x V_s(x) \\
&\geq \max_x \sum_{s\in S:\rho(s)=\emptyset} \Pi^s V_s(x) \\
&= \max_x \sum_{s\in C}\left(\Pi^s \tilde{q}(x,s)\theta_{x(s)}\right)
\end{aligned} \tag{5.2}$$

Since the total expected value has been demonstrated to be dependent on the path to the terminal nodes we show the following procedure.

Given a terminal node $\bar{s}$ let x_0 be the optimal allocation for the path terminating in $\bar{s}$.

The approach iteratively modifies the optimal solution x_0 to x^* (Greedy Solution) without decreasing the value of the allocation. In particular, the approach iterates over the scenes $i \in \{1, ..., | \sigma^{\bar{s}} |\}$ and for each scene i, an allocation x_i is built such that the expected value of x_i is at least the expected value of x_{i-1}. This way starting from the optimal solution x_0, the solution is changed with the allocations of the Greedy Algorithm without experiencing any decrease.

In particular, 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 that $x_{|\sigma^{\bar{s}}|} = x^*$. Hence at each step i all the scenes $j \leq i$ have the allocation provided by the Greedy algorithm with a value that is guaranteed to be at least as the one provided by the optimal solution, while all the scenes $j >| \sigma^{\bar{s}} |$ have the allocation provided by the optimal solution x_0.

Let S_i be the set of scene s in $\sigma^{\bar{s}} \setminus \sigma^{s_i}$ such that $x_{i-1}(s) = x^*(s_i)$. Notice that, S_i represents all the scenes after s_i are allocated with the same ad that the Greedy wants to allocate in s_i (i.e. the greedy is inserting a repeated ad).

At each iteration i, since we are adapting the optimal solution to the greedy solution we compare $x_i(\sigma_j^{\bar{s}})$ with $x^*(\sigma_j^{\bar{s}})$

Case 1. If $x_i(\sigma_j^{\bar{s}}) = x^*(\sigma_j^{\bar{s}})$, i.e. the Greedy algorithm allocates the same ad of the optimal solution in the scene i, then setting $x_i = x_{i-1}$ the allocation value does not change and the required conditions are satisfied.

Case 2. Suppose $x_i(\sigma_j^{\bar{s}}) \neq x^*(\sigma_j^{\bar{s}})$ and $S_i = \emptyset$ (i.e. there are no repetitions). Let $x_i(s_i) = x^*(s_i)$ while the allocation x_i is equivalent to x_{i-1} in all the other nodes. Given that the allocation is the same in all the other nodes, the difference between the allocation values of x_i and x_{o-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 $\sigma^{\bar{s}}$ with $x_{i-1}(s') = x_{i-1}(s')$ (it may be $\sigma_i^{\bar{s}}$)). This is due to the fact that if the ad it is replaced had multiple allocations along the path, then their value is increased due to a higher non convert probability. Hence the difference depends on the allocation that had the lowest allocation value.

Furthermore,

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

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

Case 3. Suppose $x_i(\sigma_j^{\bar{s}}) \neq x^*(\sigma_j^{\bar{s}})$ and $S_i \neq \emptyset$ (i.e. the ad that the greedy allocates in s_i, is allocated also in subsequent scenes, since $S \neq \emptyset$. 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 S1. 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 $\sigma^{\bar{s}}$ in x_i and x_{i-1}.

This concludes the proof.

Notice that also the Myerson's weak monotonicity is satisfied since the algorithm returns an optimal solution. Hence, it can be used together with the VCG auction algorithm for the design of a truthful mechanism for real world settings (Nisan et al. 2007 [21]).

5.3. Poly-Time Algorithm for META-SD-NE: Dealing with Scene-dependent Qualities

In this section, we focus on the scenario in which qualities are scene dependent. We show that the allocation problem in this setting is APX-Hard. The reduction is based on the satisfiability of the 3-SAT-5 problem, defined as follows.

Definition(*3-SAT:*) A 3-SAT problem is a decision problem that verifies whether a Boolean formula in conjunctive normal form (CNF) with three variables per clause can be satisfied by some assignment of truth values to its variables.

Definition(*3-SAT-5:*) A 3-SAT-5 instance is a 3-SAT instance in which each variable appears in exactly 5 clauses.

Moreover, for a 3-SAT-5 problem it is valid the following theorem:

Theorem 5.2. *(Feige (1998) [6]). 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.*

Thereby, the reduction to the 3-SAT-5 problem allows to conclude that:

Theorem 5.3. *META-SD-NE is APX-Hard.*

More interestingly, it is shown that META-SD-NE is APX-Complete by designing a polynomial-time algorithm that works in a greedy fashion, providing a constant approximation factor. The *continuous greedy algorithm* (Calinescu et al. 2011 [3]) provides a

$(1-1/e)$-approximation.

Theorem 5.4. *META-SD-NE admits a polynomial-time algorithm that provides a* $(1 - 1/e)$ *approximation.*

The analysis of weak monotonicity of the continuous greedy is elusive. Hence, the greedy Algorithm 1 can be used to maximize the objective function instead of the more complex continuous greedy, with a small loss in the approximation factor, providing a $1/2$ approximation.

However, such an algorithm is not weakly monotone.

Proposition 5.2. *Algorithm 1 is not weakly monotone (in the sense of Myerson) for META-SD-NE.*

5.4. Poly-Time Algorithm for META-SI-E: Dealing with Externalities

In this scenario, we introduce externalities but ads' qualities are scene independent. The allocation problem is hard to approximate and the hardness of the approximation depends on the memory length k. In this case, the reduction comes from the the following promise problem related to the problem of finding cliques in graphs:

Theorem 5.5. *(Håstad (1999), Zuckerman (2007) [28]) 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* $\mid V \mid^{1-\epsilon}$ *or all the cliques have a size of at most* $\mid V \mid^{\epsilon}$

The following theorem shows that is NP-Hard to provide an approximation to META-SI-E sublinear in the memory length k.

Theorem 5.6. *For any* $\epsilon > 0$ *it is NP-Hard to approximate META-SI-E to within a factor* $\mid k+1 \mid^{1-\epsilon}$ *where* k *is the memory length.*

In particular, META-SI-E admits a polynomial-time approximation algorithm that provides a $\dfrac{1}{k+1}$-approximation, matching the lower bound stated in the above theorem.

Theorem 5.7. *Algorithm 2 provides a* $\dfrac{1}{k+1}$*-approximation to META-SI-E. Moreover, it runs in polynomial time.*

The algorithm, whose code is provided in Algorithm 5.2, resorts Algorithm 5.1 as follows.

It allocates ads only to scenes $s \in S$ such that $\mid \sigma^s \mid \in \{1 + i(k+1)\}_{i \in \mathbb{N}}$. This way the allocated ads are not subject to any externality because for each scene s in which an ad a is allocated, k scenes are left empty behind thereby not generating any externality. Although not all the scenes are allocated with an ad, the subset of allocated scenes is sufficient to guarantee a $\frac{1}{1+k}$-approximation of the optimal solution. The optimal allocation is computed resorting to Algorithm 1.

Algorithm 5.2 Algorithm 2: GREEDY-SI-E

1: Inputs: set of scenes S, set of ads A, memory length k
2: Initialize $S_1 \leftarrow \{s \in S :\mid \sigma^s \mid \in \{1 + j(k+1)\}_{i \in \mathbb{N}}\}$
3: $x^* \leftarrow GREEDY(S_1, A)$
4: **return** x^*

Proposition 5.3. *Algorithm 2 is weakly monotone (in the sense of Myerson).*

Proof. It is sufficient to notice that Algorithm 5.2 is equivalent to algorithm 5.1 restricted to the subset of scenes S_1. Then the monotonicity of the algorithm comes from the monotonicity of Algorithm 5.1.

5.5. Poly-Time Algorithm for META-SD-E: Approximating the General Problem

This is the most general scenario, in which ads' qualities are scene dependent and there are externalities among allocated ads. As the theorems 5.3 and 5.6 show, META-SD-E is a Poly-APX-Hard problem, in particular it is possible to provide an approximation sublinear in k in polynomial time. In particular the problem admits a $\frac{1-1/e}{k+1}$-approximation.

The algorithm follows the approach of Algorithm 5.2, except that needs to evaluate all the sets of scenes $\{1 + j(k+1)\}_{j \in \mathbb{N}}, \{2 + j(k+1)\}_{j \in \mathbb{N}}, ..., \{k + j(k+1)_{j \in \mathbb{N}}$ as the qualities depend on the scenes. The rationale is to enumerate these sets of scenes and, for each one of them, to approximate the optimal allocation with the continuous greedy algorithm. Finally, the best allocation is taken among those evaluated by the algorithm.

The resulting aproximation factor comes from the approximation factors of both META-SD-NE and META-SI-E.

Theorem 5.8. *Algorithm 3 provides a $\frac{1-1/e}{k+1}$-approximation to META-SD-E. Moreover it runs in polynomial time.*

Proposition 5.4. *Neither Algorithm 1 nor Algorithm 2 are weakly monotone (in the*

Algorithm 5.3 Algorithm 3: GREEDY-SD-E

1: **Inputs:** set of scenes S, set of ads A, memory length k
2: **for** $i \in \{1, \dots, k\}$ **do**
3: $\quad S_i \leftarrow \{s \in S : |\sigma^s| \in \{1 + j(k+1)\}_{j \in \mathbb{N}}\}$
4: $\quad x^* \leftarrow CONTINUOUSGREEDY(S_i, A, f)$
5: **end for**
6: $i^* \leftarrow argmax_{i \in \{1,\dots,k\}} \sum_{s \in S} (\Pi^s \tilde{q}(x_i, s)\theta_{x_i(s)})$
7: **return** x_i^*

sense of Myerson) for the META-SD-E problem.

Proof. The non monotonicity of Algorithm 1 comes from the Proposition 1, showing that Algorithm 1 is not monotone even for the simpler scenario SD-NE. To conclude, Algorithm 2 is not weakly monotone since it is sufficient to notice that in the case in which $k = 0$ Algorithm 2 is equivalent to Algorithm 1.

Proposition 5.5. *Notice that it is possible to derive a algorithm similar to Algorithm3 in which we replace the continuous greedy algorithm with the greedy Algorithm 1 and obtain a* $\frac{1/2}{k+1}$*-approximation factor. However, this algorithm is not weakly monotone for META-SD-E.*

We conclude by showing that neither the Greedy Algorithm 1 nor its extension Algorithm 2 are weakly monotone for the META-SD-E problem. This result comes from the non-monotonicity of the greedy Algorithm 1 in the simpler setting with no externalities and scene-dependent qualities.

Scenario	Complexity	Best Apx-Ratio	Best Monotone Apx-Ratio
META-SI-NE	Poly	1	1
META-SD-NE	APX-Complete	(1-1/e)	-
META-SI-E	Poly-APX-Complete	1/(k+1)	1/(k+1)
META-SD-E	Poly-APX-Complete	(1-1/e)/(k+1)	-

6 | Finding the Optimal Allocation: Exact Algorithm

This chapter presents the mathematical model for the problem, necessary for computing the optimal solution using Linear Programming techniques. Specifically, it formalizes sets, parameters, and variables, along with their constraints and objective function. Due to the non-linear nature of the objective function, the study discusses linearization techniques to arrive at a final linear version.

6.1. META Problem Statement

In this section we design the META allocation problem as an Integer Linear Programming one. Defining an ILP means denoting its sets, parameters, variables and constraints. The problem is then given to a solver, in this case Gurobi, that will find the optimal solution. The resources for the current advertising allocation problem are the scenes, limited in number, in which we want to allocate the advertisings.

6.1.1. Sets

Sets and Parameters describe the particular instance of the allocation problem we want to solve.

$A = \{a_\emptyset, a_1, a_2, ..., a_m\}$ advertisements to be allocated including the empty ad $a_\emptyset$

$S = \{s_0, s_1, s_2, ..., s_n\}$ scenes of the tree in which allocate the advertisements

In the previously introduced model, the conversion probability of an allocated advertisement depends on several factors, such as the probability of reaching the scene s in which the ad a is allocated. Thus, for a given instance of a tree, the set of pathways that lead to individual scene nodes are produced.

$P = \{\sigma^{s_1}, \sigma^{s_2}, ..., \sigma^{s_n}\}$ where σ^{s_i} is the path to scene s_i, $\forall s \in S$

Furthermore, for a certain scene s, we define $A^{|\sigma^s|}$ as the set of all the possible ads assignments in the path σ^s, including the assignment to scene s.

$A^{|\sigma^s|} = \{\vec{a}_0, \vec{a}_1, \vec{a}_2, ..., \vec{a}_{|A|^{|\sigma^s|}}\}$ all possible ads allocations in $\sigma^s, \forall \sigma^s \in P$

6.1.2. Parameters

$q_{a,s}$ quality of ad a allocated in scene s , $\forall a \in A, \forall s \in S$

$\gamma_{i,j}$ externality of ad i on ad j, $\forall i, j \in A$

$\Pi = \pi_1, \pi_2, ..., \pi_n$ reach probability of scene s, $\forall s \in S$

6.1.3. Variables

The decision variable of the allocation problem indicates what is the allocated ad for each scene. Therefore, we define:

$$\forall s \in S, a \in A \quad X_{s,a} = \begin{cases} 1, & \text{if ad } a \text{ is assigned to scene } s \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

6.1.4. Constraints

Constraints limit the values that the decision variables can assume, thereby limiting the space of feasible solutions. In particular, the modeled contraint below indicates that only one advertisement can be assigned to each scene.

$\forall s \in S, \sum_{a \in A} X_{s,a} = 1$ each scene is assigned exactly with one ad.

6.1.5. Objective Function

As previously discussed in the previous chapter, the total allocation expected value is expressed as:

$$\sum_{s \in S} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)} \right) \tag{6.2}$$

The objective function of the ILP problem is therefore rewritten with the parameter and variables previously introduced as follows:

$$\sum_{s \in S} \sum_{\vec{a} \in A^{|\sigma^s|}} \Pi_s \cdot \prod_{s_i \in \sigma^s - k}^{\sigma^s} \gamma_{a_{s_i},a} \cdot \prod_{s_i \in \sigma^s} x_{s_i,a_{si}} \cdot q_{a,s} \cdot \prod_{\substack{s_h \in \sigma^s \\ |x_{s_h},a}} \left(1 - \sum_{\vec{a}_j \in A^{|\sigma^{s_h}|}} \prod_{s_j \in \sigma^{s_h} - k}^{\sigma^{s_h}} \gamma_{a_{s_j},a} \cdot \prod_{s_j \in \sigma^{s_h}} x_{s_j,a_{sj}} \cdot q_{a,s_h}\right) \tag{6.3}$$

The above objective function sums up the values of the chosen advertisement allocations on the path to each scene in the tree. In fact, for each scene s, there are $n =| A |^{|\sigma^s|}$ possible allocations, and among those, only one is chosen by the optimizer. For a scene s, the factor that indicates which combination of ad is the chosen one is:

$$\prod_{s_i \in \sigma^s} x_{s_i,a_{si}} \tag{6.4}$$

In the objective function above, the index S_h appears in order to compute the *non conversion probability*, given the allocation in scenes preceding s. Since the scenes S_h are strictly before scene s, $\prod_{s_i \in \sigma^s} x_{s_i,a_{si}} = 1$ then $\prod_{s_j \in \sigma^{s_h}} x_{s_j,a_{sj}} = 1$.

This consideration allows us to simplify the objective function presented above in the following:

$$\sum_{s \in S} \sum_{\vec{a} \in A^{|\sigma^s|}} \Pi_s \cdot \prod_{s_i \in \sigma^s - k}^{\sigma^s} \gamma_{a_{s_i},a} \cdot \prod_{s_i \in \sigma^s} x_{s_i,a_{si}} \cdot q_{a,s} \cdot \prod_{\substack{s_h \in \sigma^s \\ |x_{s_h},a}} \left(1 - \prod_{s_j \in \sigma^{s_h} - k}^{\sigma^{s_h}} \gamma_{a_{s_j},a} \cdot q_{a,s_h}\right) \tag{6.5}$$

6.2. Linearization Methods

Non linear objective functions introduce several issues when searching for the optimal solution. For example, non-linear objective functions may have multiple local optima, which can make it difficult for the optimizer to find the global optimum solution. It is noteworthy that in the function 6.5, the factor $\prod_{s_i \in \sigma^s} x_{s_i,a_{si}}$ embodies a product of binary decision variables, thereby introducing non-linearities in the objective function.

We use a linearization method as presented in Fred Glover et al. (1973) [11], which involves adding an auxiliary variable z to linearize the production in the objective function. The introduction of z imposes two additional constraints that ensure its dependence on the original variables. In the general case:

$$z = \prod_{i=0}^{n} x_i \qquad z \leq x_i \text{ for } i = 1, ..., n \qquad z \geq \sum_{i=0}^{n} x_i - (n-1) \tag{6.6}$$

The first constraint states that when at least one of the variables x_i is zero, then z is zero. The second one forces z to be equal to one if the production of the variables is equals to one, which only happens if all of them are equal to one.

Furthermore, given a scene s, we denoted with $\vec{a} \in A^{|\sigma^s|}$ a possible assignment of ads along the path σ^s to scene s. Hence, an auxiliary variable $z_{s,\vec{a}}$ is introduced for every scene s and each assignment $\vec{a}$ on the path σ^s, where $z_{s,\vec{a}}$ equals one if and only if $\vec{a}$ represents the selected assignment along the path σ^s, zero otherwise.

Hence, $z_{s,\vec{a}}$ must observe the following constraints:

$$z_{s,\vec{a}} = \prod_{s_i \in \sigma^s} x_{s_i,a_{s_i}} \qquad \text{and} \qquad z_{s,\vec{a}} \leq x_{s_i,a_{s_i}}, \text{for all } s_i \in \sigma^s$$

Furthermore, we have:

$$z_{s,\vec{a}} \geq \sum_{s_i \in \sigma^s} x_{s_i,a_{s_i}} - (\mid \sigma^s \mid -1)$$

where $\mid \sigma^s \mid$ represents the length of the path to s.

Finally, the objective function can be rewritten as follows:

$$\sum_{s \in S} \sum_{\vec{a} \in A^{|\sigma^s|}} \Pi_s \cdot \prod_{s_i \in \sigma^s - k}^{\sigma^s} \gamma_{a_{s_i},a} \cdot z_{s,\vec{a}} \cdot q_{a,s} \cdot \prod_{\substack{s_h \in \sigma^s \\ | x_{s_h,a}}} \left(1 - \prod_{s_j \in \sigma^{s_h} - k}^{\sigma^{s_h}} \gamma_{a_{s_j},a} \cdot q_{a,s_h}\right) \tag{6.7}$$

Where, for each scene s and each assignment $\vec{a}$ in the path σ^s:

- Π_s is the reach probability of scene s
- $\prod_{s_i \in \sigma^s - k}^{\sigma^s} \gamma_{a_{s_i},a}$ is the total externality generated by all the advertisements displayed before scene s, as long as they appear within k steps before s, where k is the user memory.
- $z_{s,\vec{a}}$ is the auxiliary variable that indicates whether $\vec{a}$ is the chosen assignment for the path σ^s
- $q_{a,s}$ is the quality of advertisement a when assigned to scene s

- $\prod\limits_{\substack{s_h \in \sigma^s \\ | x_{s_h,a}}} \left(1 - \prod\limits_{s_j \in \sigma^{s_h} - k}^{\sigma^{s_h}} \gamma_{a_{s_j},a} \cdot q_{a,s_h}\right)$ is the non convert probability of ad a when allocated in the scenes s_h strictly before scene s. In particular,
- $\prod\limits_{s_j \in \sigma^{s_h} - k}^{\sigma^{s_h}} \gamma_{a_{s_j},a} \cdot q_{a,s_h}$ is the adjusted quality of ad a when allocated in scene s_h, where the production $\prod\limits_{s_j \in \sigma^{s_h} - k}^{\sigma^{s_h}}$ denotes the cumulative externality produced by all ads a_j displayed in scenes prior to s_h, provided that they appear within the user's memory length of k scenes.

7 | Experimental Evaluation

This chapter presents the experimental results of the advertising allocation process. The study compares the performance of the greedy approach to the optimal solution computed by the solver, plotting the total allocated value and the execution time. As the greedy algorithm may be stuck in a local optima, it is expected that the solution found is less effective than the optimal one. However, the greedy approach can find the solution in a considerably reduced number of steps. Finally, the study proposes a particular class of advertisements to test the algorithms in a particularly challenging scenario.

7.1. Implementation

The implementation part describes the generation of the tested instances for each proposed scenario and shows the implementation of the mathematical model for the optimal allocation solution.

7.1.1. Instance Generation

The instance generation involves four main elements, each of which randomly generated:

- **Tree-Structure:** The creation and manipulation of the tree structures is achieved using the *NetworkX* package that provide several functionalities when working with complex networks. The structural dimensions that are of our interest regard the depth of the tree and its branching factor, that represents the number of immediate descendants of each node. Hence, each instance generation wants to provide a random tree with a maximum depth and a maximum branching factor respectively represented by the max_depth and max_branch parameters. The maximum depth of the tree is controlled pruning all the nodes that exceed the max_depth parameter. The branching factor is controlled by removing random descendants from the nodes that exceed the maximum branching factor, until they have a max_branch number of kids.
- **Adjacency Matrix:** Given a certain tree structure, the adjacency matrix is gener-

ated with random probabilities representing the likelihood of moving from one node to another. Notice that, for each non terminal node it needs to satisfy the constraint 4.1 introduced in the problem formulation chapter.

- **Quality Distribution:** As specified in the user model (4.2), the ads' qualities may be scene-dependent (SD) or scene-independent (SI). In the SI scenario, the quality of ads are fixed among the scenes, while in the SD scenario the ads' qualities vary from one scene to another following a Beta distribution. The Beta distribution has been chosen as it well represents the case in which an ad has a quality peak in a node compared to all the others.
- **Externalities:** The user model allows the presence of negative or neutral externalities among advertisements (4.3). They are represented with a matrix γ of dimension $|A| \times |A|$, where $|A|$ is the number of ads, in which $\gamma_{i,j}$ is the influence of ad i on ad j. Notice that the externalities involving the empty ad $a_\emptyset$ as well as the ones of an ad with itself are set to neutral.

7.1.2. Optimal Allocation with Gurobi

The mathematical model is formulated as an Integer-Linear-Programming (ILP) model in section 6 and is needed to compute the optimal allocation solution. To do so, we used the *gurobipy* library of the Gurobi solver. As mentioned in section 6.1 the mathematical model operates on specified sets and parameters that are not directly obtainable from the scenario. In particular, given a scenario, the set of paths to the scenes $P(S)$ is computed from the tree. Then, given the paths and the ads to be allocated, is generated the set $A^{|P(S)|}$ containing all the possible ads allocations for a given path. Once all the required sets are generated, the instance is sent to Gurobi for solving.

The time complexity for the optimizer to achieve the optimal solution is expected to be exponential with respect to the depth of the tree and the number of ads to be allocated, that is $\mid A \mid^{|\sigma^s|}$. Notice also that as the solver may take a long time to compute the optimal solution, a one hour time limit has been set for each instance(*model.setParam('TimeLimit', 3600)*). In these rare cases, the solver might not reach the optimal solution, hence the Greedy might over perform the optimal solution.

7.1.3. Tests

Each scenario is characterized by several parameters that regard the structure of the tree, the dynamics of advertisements or the user behavior. We created a set of experimental tests that measure the performance of the greedy approach in terms of allocation expected value and execution time, comparing its result against the optimal ones achieved by the Gurobi solver. Moreover, in order to achieve more consistent results, each of the tested parameters' values is averaged $n = 20$ times (higher number not feasible due to computation constraints).

Notice that, for each test, a particular parameter is tested while keeping fixed all the others to their default values here reported:

Number of Instances	20
Ads	$[ad_\emptyset, ad_1, ad_2, ad_3, ad_4]$
Convert-Values	[$0, $1,$1,$1,$1]
Default Scenario	Scene Dependent with Externalities
Max Depth	4
Max Branch	2
User Memory	4
Externality Factor	1

The experiments are made of two plots showing the following information:

- On the **X-axis** the range of the tested parameter.
- On the **Y-axis** the allocation expected value achieved by the greedy approach and the optimal solution.
- The standard deviation of the experiments
- An percentage approximation ratio indicating to what extent the greedy approximated the optimal solution.
- In the second plot, on **Y-axis** the execution time of the allocations.
- In the second plot, a percentage approximation ratio of the execution time in logarithmic scale since the greedy is considerably faster compared to Gurobi.

7.2. Single Scenario Experimental Results

In this section we provide the experimental results of the allocation algorithms in four main scenarios, which were created based on the existence or absence of externalities, and whether ad qualities are dependent on or independent of the scenes.

7.2.1. Scene Independent-No Externalities

We evaluated the allocation value and execution time with increasing tree-depth, assuming scene-independent qualities and no externalities. **X-axis:** tree max-depth

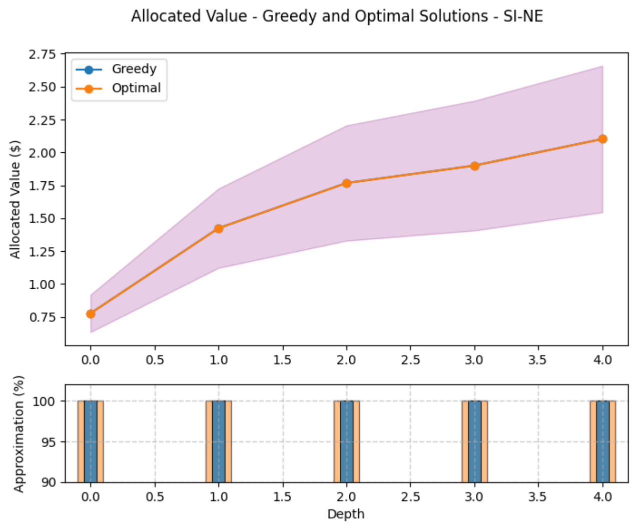


Figure 7.1: Scene Independent without Externalities-Allocation Value.

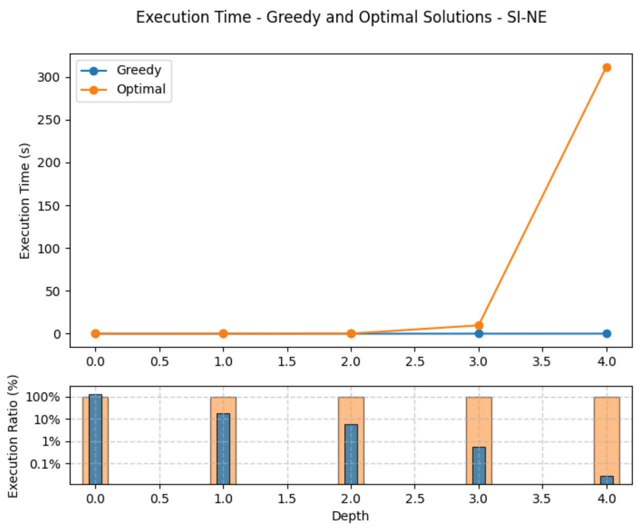


Figure 7.2: Scene Independent without Externalities-Execution Time.

In this scenario, the greedy achieves an optimal solution, as guaranteed by Theorem 5.1. The value increases with the depth of the tree, but deeper nodes contribute less to the total value as they are less likely to be reached. Additionally, its execution time is significantly lower than the one of the solver that is exponential with respect to the depth of the tree.

7.2.2. Scene Dependent-No Externalities

We evaluate the allocation value and execution time with increasing tree-depth, assuming scene-dependent qualities and no externalities.

X-axis: tree-depth.

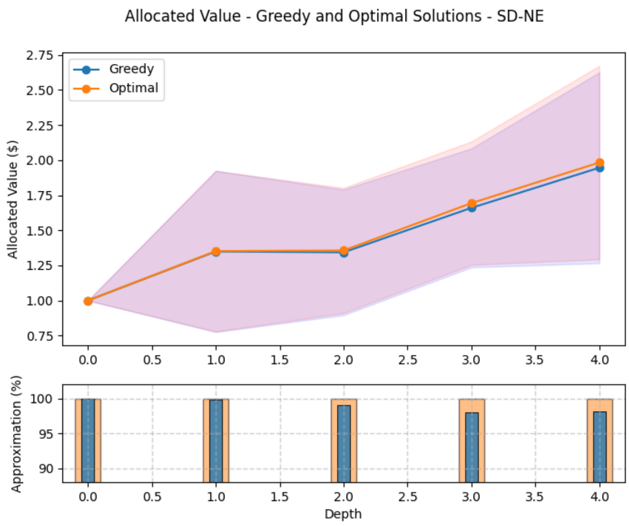


Figure 7.3: Scene Dependent without Externalities-Allocation Value.

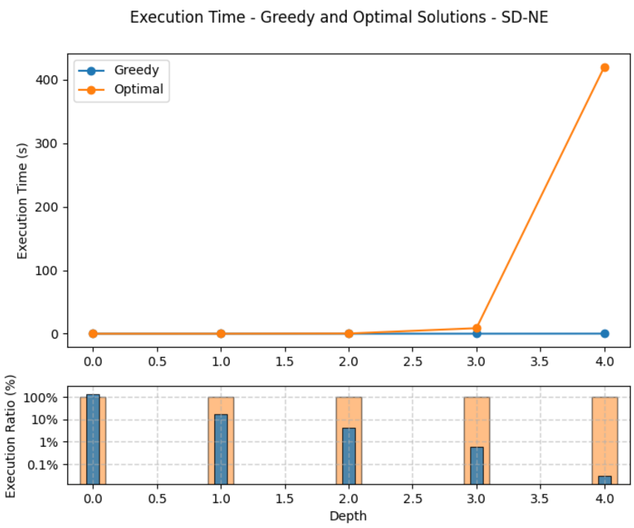
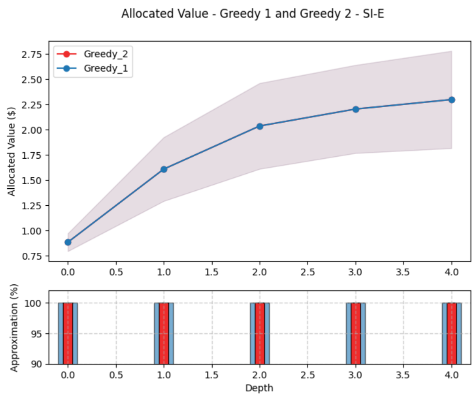


Figure 7.4: Scene Dependent without Externalities-Execution Time.

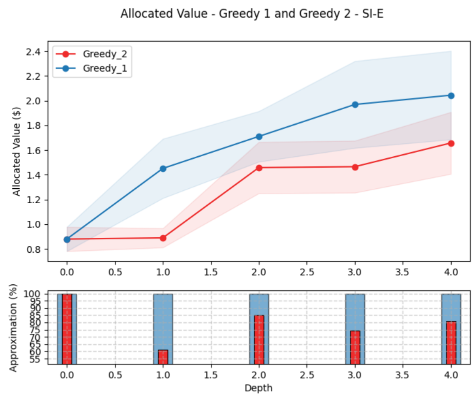
Scenario with scene-dependent qualities and no externalities among ads. The introduction of externalities slightly decreases the greedy performance but still provides a well approximated solution.

7.2.3. Scene Independent-Externalities

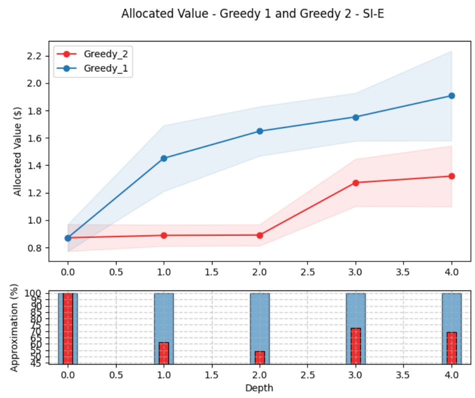
Scenario with Scene-Independent qualities with Externalities. Here are tested the performance of Algorithm 5.1 and Algorithm 5.2 with increasing user memory.



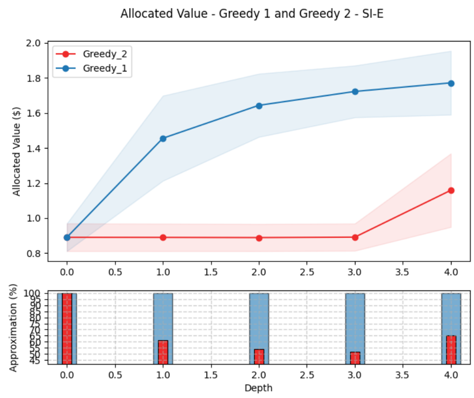
(a) User Memory = 0



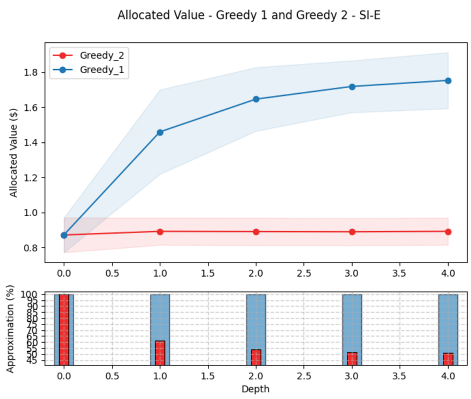
(b) User Memory = 1



(c) User Memory = 2



(d) User Memory = 3



(e) User Memory = 4

Figure 7.5: Algorithm 1 and 2 with changing User Memory

As explained in section 5.4, Algorithm 2 allocates ads along a certain path every k scenes, where k represents the user memory, while leaving the scenes in between unallocated. Although this approach mitigates the effects of externalities, it also leads to a lower expected allocation value, since unallocated scenes, despite having their ad values reduced by externalities, still contribute to the overall value.

It is worth noting that when the user's memory is zero, the two algorithms are identical, whereas a user memory equal to or greater than the maximum depth of the tree, results in the lowest value, since Greedy Algorithm 5.2 only allocates the root scene in this case.

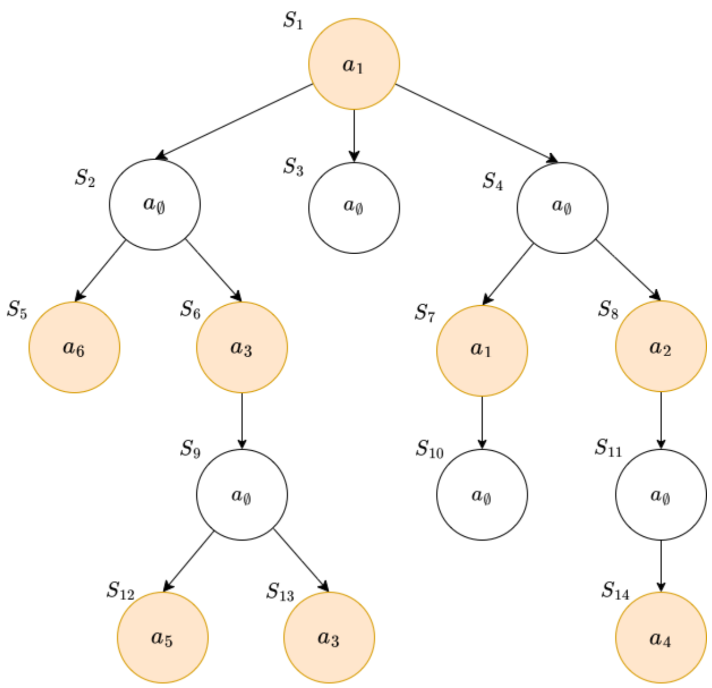


Figure 7.6: Example of Allocation of Algorithm 2 with User Memory = 1.

As figure 7.5 shows, the allocation value of Algorithm 5.1 results to be greater than the one of 5.2, hence the first greedy approach is preferred in the scenario with externalities and its performance is tested and compared against the optimal solution.

The figure below illustrates the allocation value and execution time of the initial greedy approach as the tree depth increases, compared to the optimal solution.

X-axis: tree-depth.

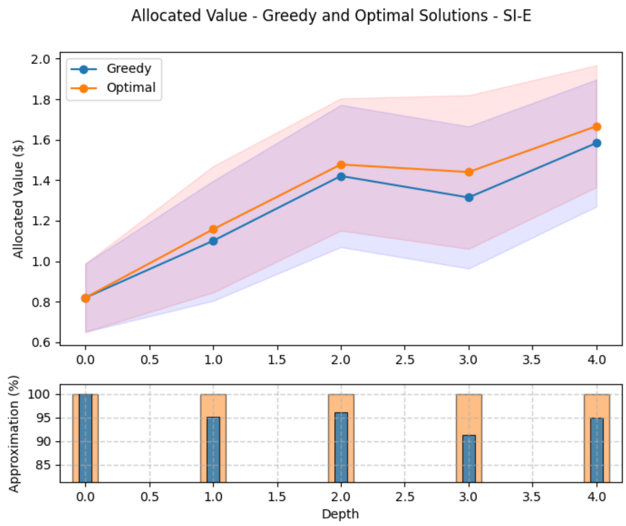


Figure 7.7: Scene Independent with Externalities-Allocation Value.

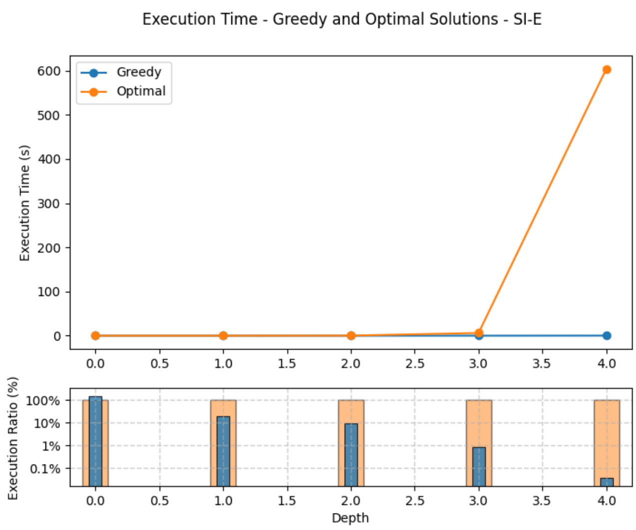


Figure 7.8: Scene Independent with Externalities-Execution Time.

As it can be noticed, the introduction of externalities leads to a slight reduction of the approximation ratio. Moreover, the execution time of the optimizer is exponential with respect to the tree depth, while the greedy provides the solution in a considerably reduced amount of time.

7.2.4. Scene Dependent-Externalities

Assuming scene-dependent qualities with the presence of externalities, we assessed the allocation value and execution time as the tree-depth increased.

X-axis: tree-depth.

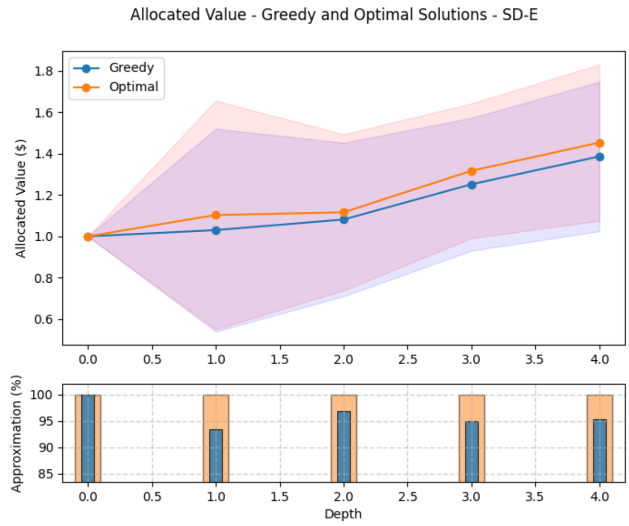


Figure 7.9: Scene Dependent with Externalities-Allocation Value.

.

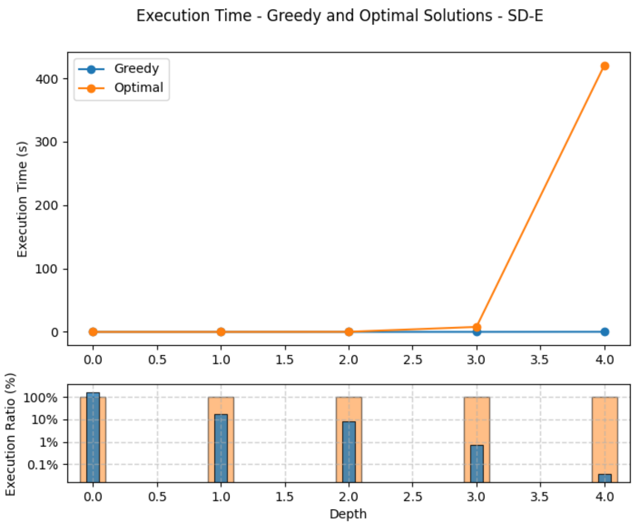


Figure 7.10: Scene Dependent with Externalities-Execution Time.

7.3. Parameters Sensitivity

In the preceding section, we evaluated the performance of the allocation algorithms across four main scenarios, based on the presence or absence of externalities and on the dependence of ads qualities on the scenes. Here, we want to investigate the behavior and responsiveness of the algorithms when some of the model' s parameters change.

Notice that for each of the conducted experiments, we assume the most general of the previously introduced scenarios, which is the Scene Dependent with Externalities (SD-E).

7.3.1. Externality Factor

We introduce a multiplicative factor, called externality factor (ETX_{factor}), that is used to intensify the effect of the externalities among advertisements. This test shows the achieved allocation value as the externality factor increases (i.e. stronger externalities). Notice that the externalities are multiplied by $\frac{1}{ETX_{factor}}$.

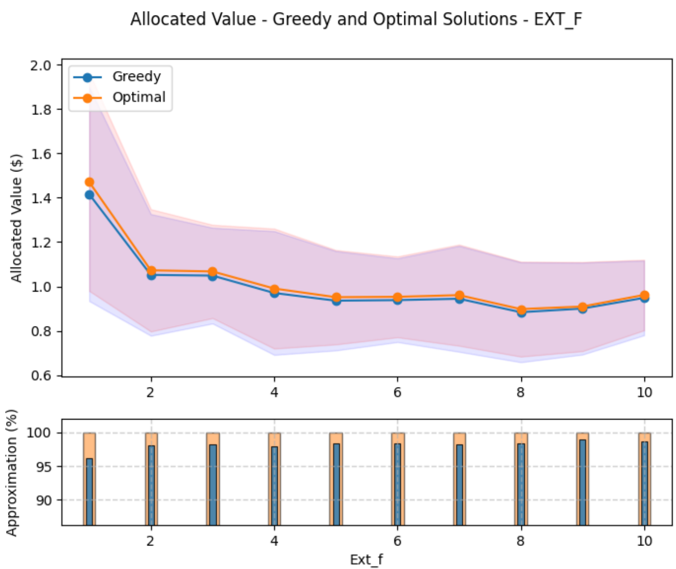


Figure 7.11: Externality Factor-Allocation Value.

The figure above shows that as the externality factor increases ,the allocation value reduces since stronger externalities reduce the quality of subsequent allocated ads. However, the algorithm well approximates the optimal solution even when the ads interactions become stronger.

7.3.2. User Memory

We tested the greedy approach's allocated value as user memory (k) increased and we compared it with the approach that leaves empty scenes to mitigate externalities.

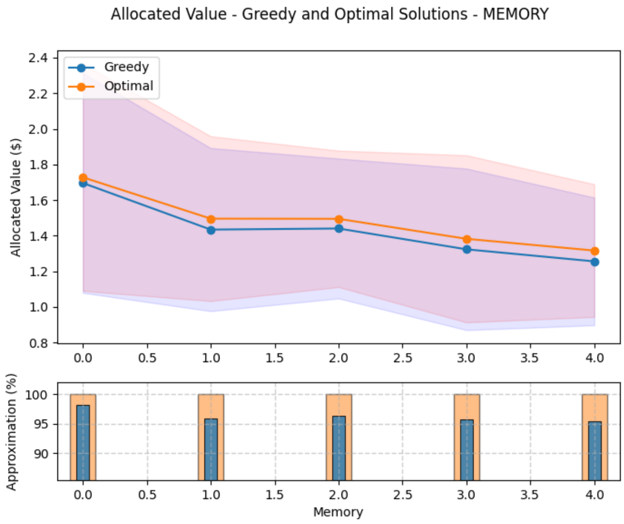


Figure 7.12: User Memory-Allocation Value.

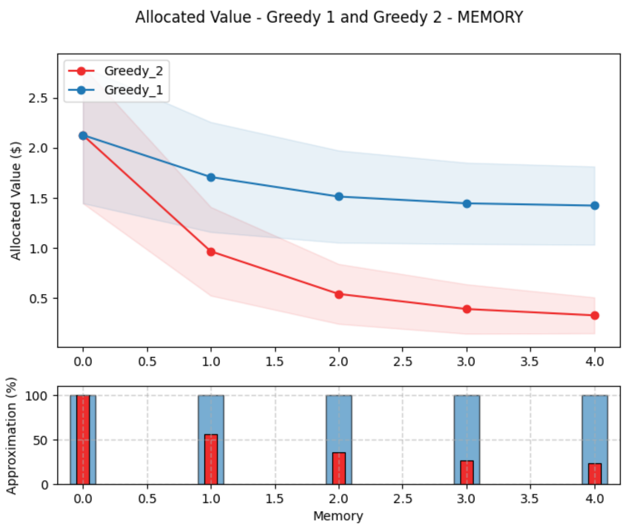


Figure 7.13: Algorithm 1 and Algorithm 2 Allocation Value.

The test above shows the allocation value with increasing user memory k, comparing the performance of the two greedy approaches. As expected, the higher the memory k of the user, the lower the allocated value as the effect of externalities increases with k. Moreover the second greedy achieves more lower values as it leaves k unallocated scenes.

7.3.3. Maximum Branching Factor

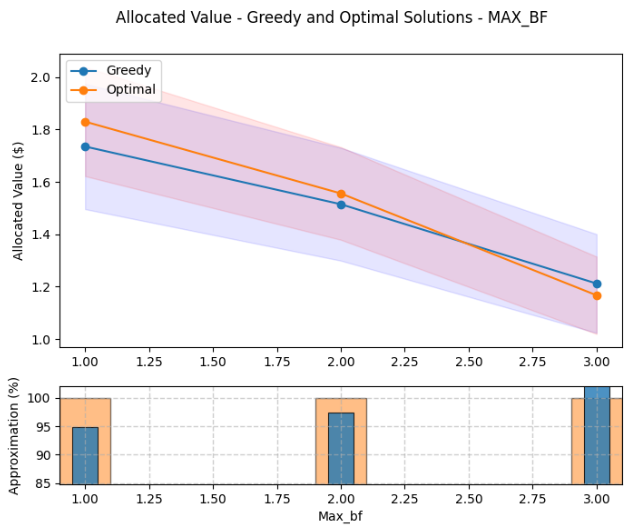


Figure 7.14: Max Branch Factor-Allocation Value.

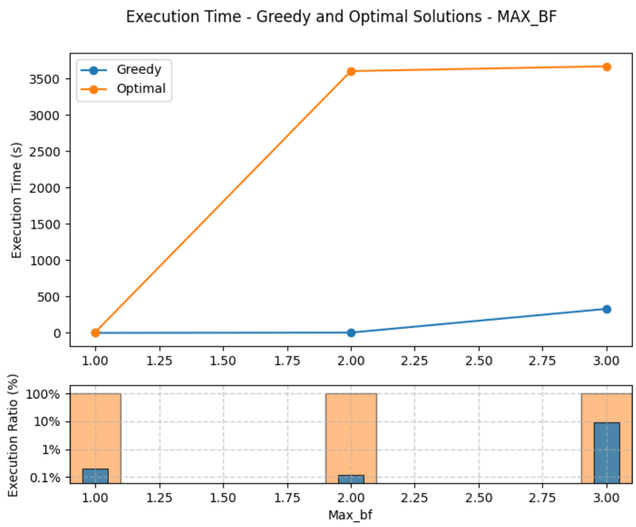


Figure 7.15: Max Branch Factor-Execution Time.

Notice that as the Branching Factor increases, the Allocated value decreases while the execution time increases. The Gurobi execution did not reach the optimal solution as it reached the 1-hour time limit. Consequently, the greedy algorithm achieved a 103.9% of the optimal solution.

7.3.4. Number of Advertisements

Allocation value achieved as the number of ads increases. **X-axis:** number of ads

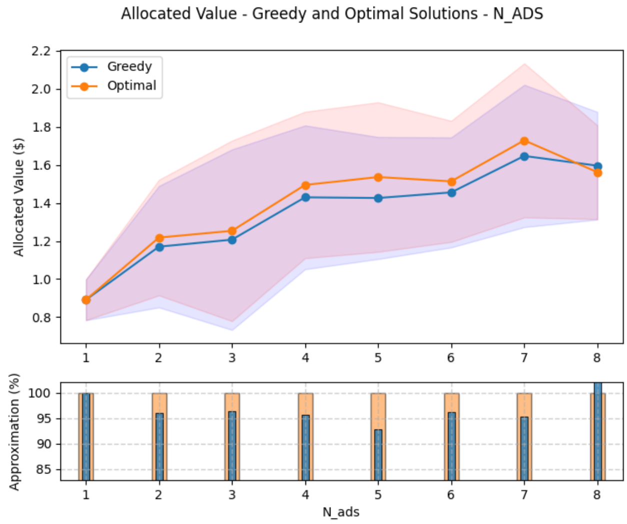


Figure 7.16: Number of Ads-Allocation Value.

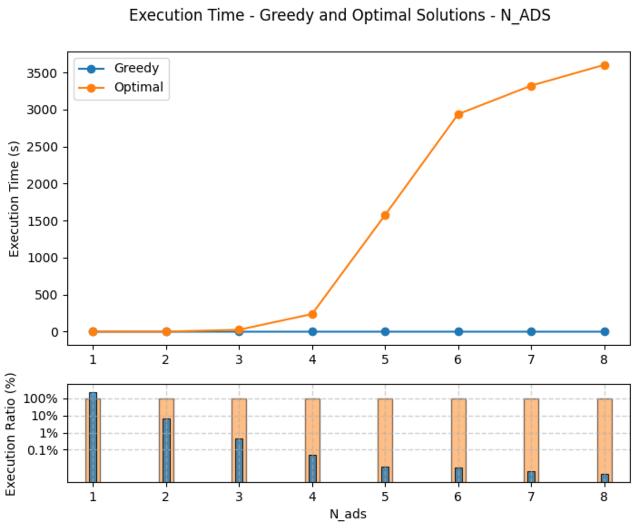


Figure 7.17: Number of Ads-Execution Time.

The test shows that an increasing number of ads, increases the allocation value as well. In fact, having few ads decreases the *non convert probability* (4.4) as the same few ads are displayed many times along the paths, while this effect is mitigated as the number of ads increases. Also time complexity grows as the number of ads are in the base of the exponential.

7.4. Greedy Hard Instances

As discussed in the previous sections, Greedy Algorithms maximize the objective function at each step in a reasonable time, but might be stuck in a local optima. This is due to their characteristic of not taking into account future consequences of the current choice with respect to the total achieved value. In this scenario, the negative consequence of allocating an advertisement is represented by the externalities that it is introducing on other ads. Therefore, the algorithm is deceived by a particular type of advertisements that have a generally high quality but introduce strong negative externalities on other ads. In simple terms, while such ads can attract users, their presence can reduce the adjusted quality of subsequent ads. In each person, attention is a limited resource, in fact if attention is spent on one task, less attention is left for other tasks, thus watching an advertisement causes an attention cost to the user. Thus, ads that strongly attract users may also have negative externalities on other ads as they consume a significant portion of the user's attention, thereby reducing their attention span.

In order to test this particular scenario, we introduce a *quality-externality* scaling factor that modifies advertisements' qualities while also changing their externalities in an inversely proportional manner. As an example, given an ad $a \in A$ and a quality-externality factor q_{ext}, the quality $q_{s,a}$ of ad a, $\forall s \in S$ is scaled by a factor q_{ext}, while its externalities $\gamma_{a,a'}$ $\forall a' \in A \setminus \{a\}$ is scaled by a factor $\frac{1}{q_{ext}}$. Therefore, utilizing the *quality-externality* factor it results in the generation of such advertisements, and adjusting its value allows for the regulation of its strength. Notice that being qualities a representation of the likelihood of the user to click on an advertisement, they are upper-bounded to 1 after being scaled.

The tests presented in the following sections aim at measuring the allocation value of the algorithm when such an advertisements appear among the ones to be allocated. Additionally, an analysis is conducted on the algorithm's response to an increase in the quality-externality factor.

7.4.1. Ads Ratio

This test aims at analysing the algorithm's behavior when the proportion of ads with the previously mentioned characteristic increases. For those ads adopting this characteristic, their quality-externality factor is $q_{ext} = 4$.

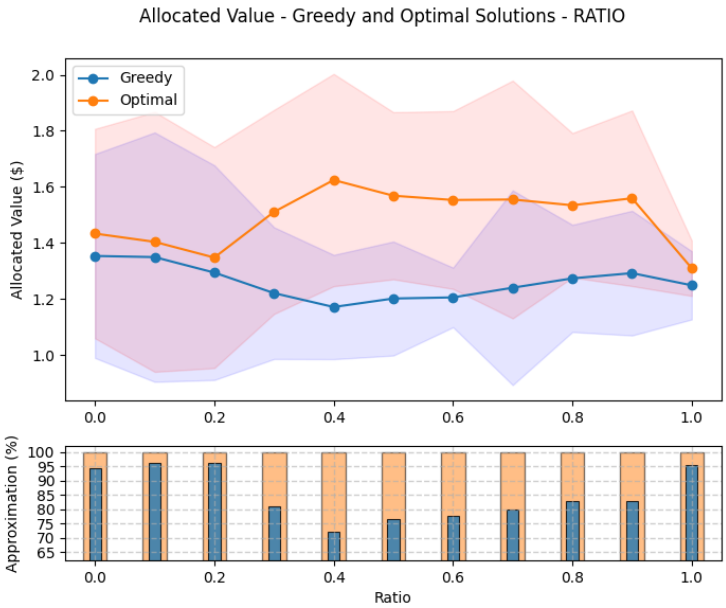


Figure 7.18: Ads Ratio-Allocation Value.

The figure 7.20 can be divided into three main parts:

- **Low Ratio**: When the ratio is significantly low, the greedy algorithm approximates the optimal solution very well.
- **Medium-High Ratio**: As the ratio of the above mentioned ads increases, the algorithm is deceived by them, resulting in lower approximation values. However, the optimal solution still produces higher values as the optimizer can reject local optima choices in favor of superior global ones.
- **High Ratio**: If most ads have this characteristic, there won't be much of a difference between the greedy and optimal solution. This is because the optimizer is compelled to choose ads with negative externalities since there aren't any other options, which leads to a lower value for the optimal solution as well.

7.4.2. Quality-Externality Factor

In this test, only a small fraction ($ratio = 0.35$) of ads exhibit the mentioned characteristic and the algorithm is evaluated by gradually increasing the quality-externality factor for such proportion of advertisements.

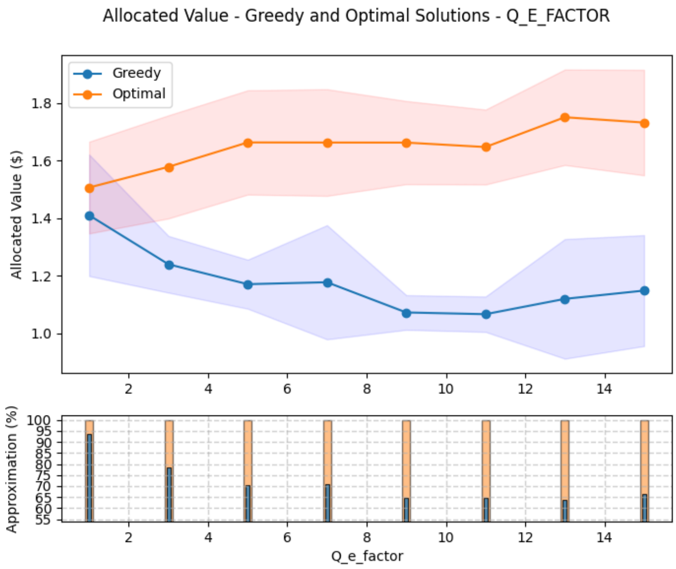


Figure 7.19: q_{ext} Factor-Allocation Value.

When the quality-externality factor is higher, the Greedy algorithm is easily deceived and will result in a lower approximation of the optimal solution.

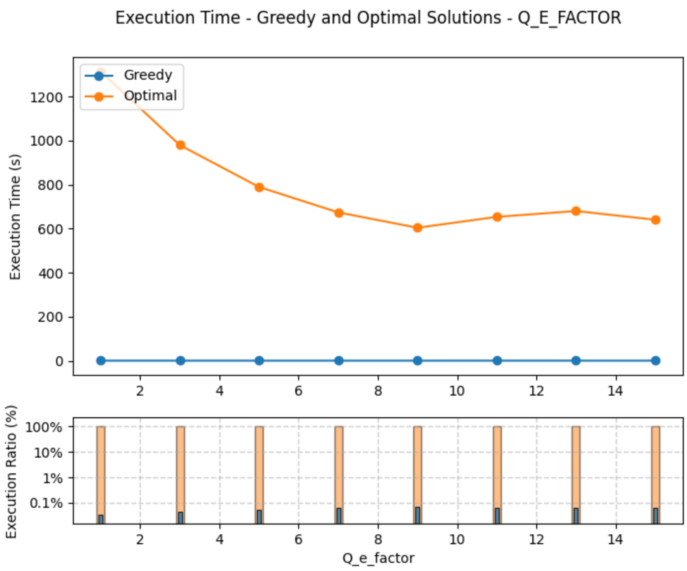


Figure 7.20: q_{ext} Factor-Allocation Value.

8 | Conclusions and Future Works

Amid of the latest advances in Augmented and Virtual reality technologies and the arise of Metaverse platforms, this work proposed and implemented to the best of our knowledge, a first user model suitable for advertising in the Metaverse, providing algorithms for the allocation of advertisements. The reason behind this work is the vast potential that can be unlocked in the world of advertising when users interact with such an immersive technology. In fact, in section 2.5 we have seen that Augmented and Virtual reality provide a controlled environment that allows advertisers to target their potential customers in a completely renewed way, offering a wide range of new advertisements formats such as virtual pop-ups, immersive experiences or implicit advertisements. Furthermore, the imersiveness of the Metaverse, not only provides advertisers with a more effective and eye-catching content, but also permits a more precised targeting due to the granular amount of data that Metaverse platforms could gather. Indeed, techniques such as eye-tracking or emotions prediction would allow to better measure the user's reactions when their are targeted with an ad (2.5). Hence, recording the user's actions along with tracking their location within the virtual environment, would considerably increase the effectiveness of targeting techniques discussed in section 2.4 such as Behavioral and Geographical.

Our model introduced in section 3 organizes the Metaverse in a graph-like structure, where users can navigate its nodes corresponding to virtual locations on the platform, referred to as scenes. The most likely paths that a user might traverse are organized in a tree-like structure, in which the user starts from a root node and then moves according to transition probabilities. Each one of the tree nodes represents a scene, in which the user can be targeted by advertisements. As the quality of an advertisement might be influenced by the scene in which it is displayed, we defined the concept of 'Scene-Dependent' and 'Scene-Independent' advertisements. Besides being dependent on the allocated scene, the quality of an ad is also influenced by the number of times the same ad has been displayed in previous scenes along the path. In customary models, independently from whether the user clicked on an ad or not, they will never click it in the future, meaning that the ad is not displayed more than one time along the path. Conversely, we introduced a more realistic model in which the rationale is that if a user does not click on an ad then they might still click on it in future scenes while interacting with it, reduces the click through rate when it is displayed in future scenes. For this reason, in the problem formulation section 4, we introduced the 'non convert probability', representing for an ad in a given scene the probability that the user didn't convert on the same ad when allocated in previous scenes.

Furthermore, we explored how the quality of an advertisement may change depending on factors such as the inter dependencies with already visualized advertisements in previous scenes. This phenomena, has been formalized with the introduction of the 'negative forward externalities', which represent the negative or neutral effects that ads have on each other, resulting in a reduction in their overall quality. Furthermore, the effects of externalities only arise within the user's viewpoint, hence their effect is perceived as long as the user is able to recall previously seen ads. Thereby, our model was provided with a user memory that indicates how many previous scenes' allocations the user is able to remember. Finally, the dependence of the qualities on the scenes, combined with the presence or absence of externalities, led to the creation of four different scenarios: SI-NE, SD-NE, SI-E, SD-E. In chapter 5, we provided for each of the scenarios, approximated allocation algorithms with theoretical guarantees that run in polynomial time, making them suitable for real-world applications. In particular, each problem has been investigated with respect to its complexity class, and, the Myerson's weak monotonicity has been investigated for the deisgn of truthful auction mechanisms. As shown in the preliminaries (see 2.6.1), the property of weak monotonicity is both a necessary and sufficient condition for the development of truthful mechanisms. When this condition is met, it becomes possible to integrate the algorithm with state-of-the-art truthful auction mechanisms, such as VCG. To compare performance of the greedy approach with the optimal solution, we modeled the problem as an Integer Linear Programming mathematical model and we used the Gurobi optimizer to find the optimal allocation. In section 6 we built the mathematical model and we provided the allocation expected value objective function to maximize. The first proposal of the objective function, was of a non-linear type, hence we applied linearizaton techniques and passed it to the solver as an ILP model.

Finally, in section 7 we analysed the performance of the greedy approaches and we compared the allocated value against the optimal allocation in terms of allocation expected value and execution time. We also found out, that in presence of externalities among ads, the first proposed greedy algorithm provides a better allocation value with respect to the approach that leaves scenes unallocated to mitigate the effects of externalities. In addition, we evaluated the allocated value when each of the model parameters was modified individually. Generally, the analysis showed that the greedy algorithm closely approximates allocated value of the optimal solution, while executing in significantly less time. Nevertheless, the introduction of externalities slightly decreases the approximation ratio, but still keeps considerably good approximations. As the greedy approach generally achieved very good approximation results in the main scenarios, we wanted to investigate a class of instances in which the algorithm is deceived (see 7.4). As we noted, at each step the greedy approach wants to maximize the allocation value by choosing high quality ads, but, does not consider future consequences of such choices, such as the externalities that such ads introduce. Hence, we investigated a specific category of advertisements that exhibit a high quality but also a strongly negative externality effect, which can mislead the algorithm in the allocation process and result in a suboptimal allocation.

To conclude, it is evident that Augmented and Virtual reality technologies are already revolutionizing the future of advertising. Therefore, this work aimed to take an initial step towards advertising in the Metaverse. However, several advances can be made to enrich our model, especially in terms of personalization for individual users. In fact, the parameters of our model are randomly generated and do not take into account user-specific values. As an example, the transition probabilities between scenes are closely related to individual users, since their behavior varies when navigating through the virtual locations of the platform. Similarly, the quality of advertisements and the externalities among them are strongly correlated to the specific user, given that they have varying interests and respond differently to previously displayed ads. Hence, they can be estimated using online learning approaches, where every movement allows to learn transition probabilities, and each action can indicate the user's interest in a particular product category. This would enhance the overall user experience and provide a valuable tool for advertisers, providing an optimal and user-based allocation of advertisements in the Metaverse.

Bibliography

[1] B. Adgate. Forecast: 2023 ad spending will slow down next year but will continue to grow. *Forbes*, December 2022. URL `https://www.forbes.com/sites/bradadgate/2022/12/08/forecast-2023-ad-spending-will-slow-down-next-year-but-will-continue-to-grow/?sh=fed5bad2469d`.

[2] F. Bonetti, G. Warnaby, and L. Quinn. *Augmented Reality and Virtual Reality in Physical and Online Retailing: A Review, Synthesis and Research Agenda*, pages 119–132. 01 2018.

[3] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6): 1740–1766, 2011.

[4] C. Deng and S. Pekec. Money for nothing: Exploiting negative externalities. page 361–370, 2011. doi: 10.1145/1993574.1993634. URL `https://doi.org/10.1145/1993574.1993634`.

[5] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai. Metaverse for social good: A university campus prototype. *CoRR*, abs/2108.08985, 2021.

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

[7] D. Fotakis, P. Krysta, and O. Telelis. Externalities among advertisers in sponsored search. pages 105–116, 2011.

[8] N. Gatti, M. Rocco, S. Ceppi, and E. H. Gerding. Mechanism design for mobile geo-location advertising. *CoRR*, abs/1404.4106, 2014. URL `http://arxiv.org/abs/1404.4106`.

[9] N. Gatti, M. Rocco, P. Serafino, and C. Ventre. Towards better models of externalities in sponsored search auctions. *CoRR*, abs/1604.04095, 2016. URL `http://arxiv.org/abs/1604.04095`.

[10] G. Gavazzi. How much money does google make from advertising?, 2021. URL https://fourweekmba.com/how-much-money-does-google-make-from-advertising/.

[11] F. Glover and E. Woolsey. Further reduction of zero-one polynomial programming problems to zero-one linear programming problems. *Operations Research*, 21(1):156–161, 1973.

[12] B. Heller and A. Bar-Zeev. The problems with immersive advertising: In ar/vr, nobody knows you are an ad. *Journal of Online Trust and Safety*, 1(1), Oct. 2021.

[13] J. Horwitz. Amazon's vr kiosks transform the future of shopping, 2018. URL https://venturebeat.com/business/watch-amazons-vr-kiosks-transform-the-future-of-shopping/.

[14] Indeed. Targeted advertising. https://www.indeed.com/career-advice/career-development/targeted-advertising, 2022.

[15] M. Kaur and B. Gupta. Metaverse technology and the current market. *Insights2Techinfo*, 2021.

[16] D. Kempe and M. Mahdian. A cascade model for externalities in sponsored search. pages 585–596, 12 2008. ISBN 978-3-540-92184-4. doi: 10.1007/978-3-540-92185-1_65.

[17] W. Lau. What is targeted advertising? https://www.adroll.com/blog/what-is-targeted-advertising, 2021.

[18] M. LODGE. Decentraland: A guide to the virtual world built on ethereum, 2021. URL https://www.investopedia.com/what-is-decentraland-6827259.

[19] C. L. Mauro. Metaverse is the future! well, not so fast mr. zuckerberg., 2021.

[20] H. Ning, H. Wang, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand. A survey on metaverse: the state-of-the-art, technologies, applications, and challenges. *CoRR*, abs/2111.09673, 2021.

[21] N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani. Algorithmic game theory. 2007.

[22] Oberlo. Google ad revenue statistics in 2022: How much does google make?, 2022. URL https://www.oberlo.com/statistics/google-ad-revenue.

[23] M. Ostrovsky, B. Edelman, and M. Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97:242–259, 03 2007. doi: 10.2139/ssrn.861164.

[24] T. Sandbox. What is the sandbox?, 2021. URL `https://medium.com/sandbox-game/what-is-the-sandbox-850de68d893e`. Accessed on: April 5, 2023.

[25] Statista. Metaverse advertising in europe. `https://www.statista.com/outlook/amo/metaverse/metaverse-advertising/europe`.

[26] C. R. Taylor. Research on advertising in the metaverse: a call to action. *International Journal of Advertising*, 41(3):383–384, 2022.

[27] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

[28] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Electronic Colloquium on Computational Complexity (ECCC)*, 2006, 01 2005. doi: 10.1145/1132516.1132612.

A | Appendix A

Here are reported the proofs omitted from section 5:

Theorem 5.3 META-SD-NE is APX-Hard.

Proof. let $\eta = \max\left\{c, \left(1 - \frac{1_5}{5}\right)\right\}$, where c is the constant factor approximation in Theorem 2 . Notice that Theorem 2 holds even if we replace the approximation factor c with the weaker constant $\eta \geq c$.

Given an instance of 3-SAT-5 with clauses C and variables V, we build an instance of the META-SD-NE problem with the following structure. The tree of scene 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 $v \in V$. Let $\epsilon = 1 - \eta^{1/5}$, and let l denote a literal, i.e., l is a variable or its negation. The qualities of the 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 $|V| + \left(1 - \eta^{1/5}\right)\eta|C|$. To conclude the proof notice that $|C| = \frac{3}{5}|V|$. Hence

$$\frac{|V| + \left(1 - \eta^{1/5}\right)\eta|C|}{|V| + \eta^{4/5}\left(1 - \eta^{1/5}\right)|C|} = \frac{|V| + \frac{3}{5}\left(1 - \eta^{1/5}\right)\eta|V|}{|V| + \frac{3}{5}\eta^{4/5}\left(1 - \eta^{1/5}\right)|V|} = \frac{1 + \frac{3}{5}\left(1 - \eta^{1/5}\right)\eta}{1 + \frac{3}{5}\eta^{4/5}\left(1 - \eta^{1/5}\right)},$$

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 $l \in \{v, \sim v\} \backslash L$. Finally, let assign to each scene $s_c, c \in C$, an ad a_l such that the literal $l \in L$ satisfies the clause and belongs to L. Notice that this clause exists since the assignment satisfies all the clauses. Then, for each scene $s_v, v \in V$, we have that the expected value from the scene is 1 . Moreover, for each scene s_c, $c \in C$, 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 expected value of the allocation is at least

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

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 $s_v, v \in V \backslash V^*$ 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 $s_c, c \in C \backslash C^*$ is 0 . We can split C^* in two subsets. The set $C_2 = \left\{c \in C^* : x(s_c) \in \{a_v, a_{\sim v}\}_{v \in V^*}\right\}$, while the set $C_1 = C^* \backslash C_2$. 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 $L = \left\{l : a_l \in \{x(s_c)\}_{c \in C_2}\right\}$. 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. Notice that since each variable $v \in V$ appears in 5 clauses (considering v and its negation $\sim v$), 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 exists a literal v such $x(s_c) = a_v$ or $x(s_c) = a_{\sim v}, x(s_v) \neq a_v$, and $x(s_v) \neq a_{\sim v}$. Recall that $V \backslash V^*$ is the set of variable 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 $5\epsilon = 5\left(1-\eta^{1/5}\right) = 1$. Hence, the total utility is at most

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

This concludes the proof.

Theorem 5.4 META-SD-NE admits a polynomial-time algorithm that provides a $(1 - 1/e)$ approximation.

Before proving the theorem, some preliminary steps are needed.

Initially, we establish a relation between ad allocations and matroids. A matroid $M := (G, I)$ is defined by a finite ground set G and a collection I of independent sets, i.e., subsets of G satisfying some characterizing properties (Schrijver 2003). $B(M)$ is denoted as the set of the bases of M, which are the maximal sets in I. We show that feasible allocations can be represented by the matroid M:=(G,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 $\{(a, s)\}_{a \in A \cup \{a_\emptyset\}}$ for each scene $s \in S$, i.e., every scene is assigned to no more than one ad (while an ad can be allocated to multiple scenes).

Then, we define the utility function f on the subset of G as follows.

Definition 4. Let $f : 2^G \to \mathbb{R}_+$be the function such that, given a subset $D \in 2^G$, $f(D)$ denotes the welfare of assigning to a scene s the ad such that $(a, s) \in G$ without externalities. Formally, we write:

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

Function f satisfies a crucial property: it provides a decreasing marginal return. In particular, in the following, we show that the utility function $f : 2^G \to \mathbb{R}_+$is monotone submodular.

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

Lemma 1. 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

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

Exploiting the above characterization, we can show that function $f(\cdot)$ is monotone submodular. **Lemma 2.** Function $f(\cdot)$ is monotone submodular.

Proof. The optimization problem of maximizing the function f over the set I is the maximization of a monotone submodular function over a matroid. Notice that f is monotone submodular by Lemma 2. Hence, applying the continuous greedy algorithm to the problem provides a 1 - 1/e approximation in polynomial-time (Calinescu et al. 2011). The proof is concluded by considering the equivalence between maximizing f over I and the ad allocation problem. In particular, an independent set I is equivalent to an ad allocation in which to each scene $s \in S$ is allocated the ad $a \in A$ such that $(a, s) \in I$ (if any).

Proposition 5.2 Algorithm 1 is not weakly monotone (in the sense of Myerson) for META-SD-NE.

Proof. To prove the statement, we provide an instance in which Algorithm 1 is not weakly monotone. Consider an instance with two ads a_1 and a_2 with value per conversion $\theta_{a_1} = \frac{1}{2} + \epsilon$ and $\theta_{a_2} = \frac{1}{2}$, where $\epsilon > 0$ is an arbitrary small value. There are three scenes s_1, s_2, and s_3 arranged in a line. In particular, s_1 is the root of the tree and $\pi_{s_1,s_2} = \pi_{s_2,s_3} = 1$. The qualities are $q_{a,s_1} = \frac{2}{3}, q_{a,s_2} = \frac{1}{2}$, and $q_{a,s_3} = \frac{1}{2}$ for each ad $a \in \{a_1, a_2\}$

Consider the behavior of the greedy algorithm. As a first step, the greedy algorithm will assign to scene s_1 the ad a_1. Then, the algorithm must choose which ad to assign to scene s_2. Assigning ad a_1, the additional value is given by $q_{a_1,s_2}\theta_{a_1}(1 - q_{a_1,s_1}) = \frac{1}{2}\left(\frac{1}{2} + \epsilon\right)\left(1 - \frac{2}{3}\right) = \frac{1}{12} + \frac{1}{6}\epsilon$, while assigning ad a_2, the additional value of the allocation is $q_{a_2,s_2}\theta_{a_2} = \frac{1}{4}$. Hence, ad a_2 is assigned to scene s_2. Finally, the greedy algorithm assigns scene s_3 to ad a_2 since the additional value assigning ad a_2 is $q_{a_2,s_3}\theta_{a_2}(1 - q_{a_2,s_2}) = \frac{1}{8}$, while the additional value assigning ad a_1 is $q_{a_1,s_3}\theta_{a_1}(1 - q_{a_1,s_1}) = \frac{1}{12} + \frac{1}{6}\epsilon$. Hence, the utility of advertiser a_1 is $q_{a_1,s_1}\theta_{a_1} = \frac{2}{6} + \frac{2}{3}\epsilon = \frac{2}{3}\theta_{a_1}$, while the utility of advertiser a_2 is $q_{a_2,s_2}\theta_{a_2} + q_{a_2,s_3}\theta_{a_2}(1 - q_{a_2,s_2}) = \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$. A similar argument shows that if $\theta_{a_1} = \frac{1}{2} - \epsilon$, ad a_1 is assigned to scenes s_2 and s_3. Thus, the utility of advertiser a_1 is $q_{s_2,a_1}\theta_{a_1} + q_{s_3,a_1}\theta_{a_1}(1 - q_{s_2,a_1}) = \frac{3}{4}\left(\frac{1}{2} - \epsilon\right) = \frac{3}{4}\theta_{a_1}$. Since $\frac{3}{4} > \frac{2}{3}$, this shows that the mechanism is not monotone.

Theorem 5.6 For any $\epsilon > 0$, it is NP-Hard to approximate META-SI-E to within a factor $|k+1|^{1-\epsilon}$, where k is the memory length.

Proof. Given a graph $G = (V, E)$, we build an instance of META-SI-E such that there exists an ad a_v for each $v \in V$. The tree of scenes is composed by a line with a scene s_v for each $v \in V$ in an arbitrary order. All the transition probabilities $\pi_{s,s'}$ are set to 1 , all the qualities $q_{a,s}$ are set to 1 , and all the values per conversion are θ_a are set to 1 . Finally, the externalities $\gamma_{a_v,a_{v'}} = 1$ if $(v, v') \in E$ and 0 otherwise, while the memory length $k = |V| - 1$. We show that if there exists a clique of size $|V|^{1-\epsilon}$, then there exists an ad allocation with value at least $|k+1|^{1-\epsilon}$, while if all the cliques have size at most $|V|^{\epsilon}$, then all the allocations have value at most $|k+1|^{\epsilon}$. Since ϵ can be arbitrary small, this concludes the proof. soundness. Suppose that there exists a clique V^* of size $|V|^{1-\epsilon}$. Consider the allocation x^* in which each ad in V^* is allocated to one of the scene (in an arbitrary order), while the other ads are not allocated. It is easy to see that since there are $|V|^{1-\epsilon}$ different ads allocated and there are no there are no externalities among the allocated ads, *i.e.*, $\gamma_{a_v,a_{v'}} = 1$ for all $v, v' \in V^*$, then the allocation expected value is $|V^*| \geq |V|^{1-\epsilon} = |k+1|^{1-\epsilon}$. completeness. Suppose by contradiction that all the cliques have size at most $|V|^{\epsilon}$ and that there exists an allocation x with value strictly larger than $|k+1|^{\epsilon}$. First, notice that the expected value provided by an ad $a \in A$ is at most 1 . This holds since $\theta_a = 1$ and an ad can be converted ad most one time. Let $\bar{A}$ be the set of allocated ads that do not suffer negative externalities and hence provide a positive utility, i.e., the set of ads a assigned to a scene s such that $\gamma_{x(s'),a} = 1$ for each $s' \in \boldsymbol{\sigma}^s$. Recall that externalities strictly smaller than 1 set the probability of conversion to 0 by construction. Hence all the ads not in $\bar{A}$ provide 0 utility. Then, the set $\bar{A}$ does not include two ads $a_v, a_{v'}$ such that $(v, v') \notin E$. Otherwise, we have that the first visualized ad has negative externalities on the second, contradiction the definition of $\bar{A}$. Let $\bar{V}$ be the node of the graph relative to the ad in $\bar{A}$, *i.e.*, $v \in \bar{V}$ if and only if $a_v \in \bar{A}$.$\bar{V}$ is such that $(v, v') \in E$ for each $v, v' \in \bar{V}$ and hence $\bar{V}$ is a clique. Since the expected value of the allocation is $|\bar{A}| = |\bar{V}| \leq |V|^{\epsilon} = |k+1|^{\epsilon}$ we reach a contradiction,

Theorem 5.7 Algorithm 2 provides a $\frac{1}{k+1}$-approximation to META-SI-E. Moreover, it runs in polynomial time.

Proof. It is easy to see that the algorithm runs in polynomial-time. In the following we prove the approximation guarantees of the algorithm. Given a $i \in \{1, \ldots, k+1\}$, let $S_i = \{s \in S : |\boldsymbol{\sigma}^s| \in \{1 + j(k+1)\}_{j \in \mathbb{N}}\}$. As a first step, we show that $\max_{x(\cdot)} \sum_{s \in S_1} \left(\Pi^s \tilde{q}(x,s)\theta_{x(s)}\right) \geq \max_{x(\cdot)} \sum_{s \in S_i} \left(\Pi^s \tilde{q}(x,s)\theta_{x(s)}\right)$ for each $i \in \{2, \ldots, k+1\}$. To do so, we show that given an $i \in \{2, \ldots, k+1\}$ and an optimal allocation x_i of ads to scenes in S_i, it is possible to design an assignment x_1 to scenes in S_1 with at least the same utility. Notice that an allocation problem restricted to nodes S_i, is equivalent to a problem without externalities since no scene in the set does provide externalities to other scenes in the set. Hence, Algorithm 1 provides an optimal solution to the problem. As shown in the proof of Theorem 1 , Algorithm 1 assigns ads from the top to the bottom of the tree. Moreover, if the ties are broken always in the same way, it is easy to see that an allocation returned by an Algorithm 1 is such that the same ad is allocated to all the scenes at the same depth. Formally, given the allocation x_i returned by the algorithm we can define a function $\bar{x}_i : \mathbb{N} \to A$ such that for each $s \in S_i$ it holds $x_i(s) = \bar{x}_i\left(|\sigma^s|\right)$. Let x_i be an optimal allocation for the set of scenes S_i and let $\bar{x}_i : \mathbb{N} \to A$ be the function that defines the allocation. For each $s \in S_1$, let $\psi(s)$ be the set of nodes s' such that $s \in \sigma^{s'}$ and $S_1 \cap \left(\sigma^{s'} \backslash \sigma^{s}\right) = \emptyset$, i.e., s is the last node in S_1 that precedes s'. Notice that we have shown that given an $s \in S_1$, for each $s', s'' \in \psi(s)$, it holds $x_i\left(s'\right) = \bar{x}_i\left(\left|\sigma^{s'}\right|\right) = \bar{x}_i\left(\left|\sigma^{s''}\right|\right) = x_i\left(s''\right)$. Hence, we can define a new allocation x_1 on scenes in S_1 such that for each $s \in S_1$, it holds $x_1(s) = x_i\left(s'\right)$ for each $s' \in \psi(s)$. Then, it holds

$$\sum_{s \in S_1} \left(\Pi^s \tilde{q}\left(x_1, s\right) \theta_{x_1(s)}\right) \geq \sum_{s \in S_1} \left(\sum_{s' \in \psi(s)} \Pi^{s'}\right) \tilde{q}\left(x_1, s\right) \theta_{x_1(s)} = \sum_{s \in S_1} \sum_{s' \in \psi(s)} \Pi^{s'} \tilde{q}\left(x_i, s'\right) \theta_{x_i(s')} = \sum_{s \in S_i} \Pi^s \tilde{q}\left(x_i, s\right) \theta_{x_i}$$

This proves that $\max_{x(\cdot)} \sum_{s \in S_1} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right) \geq \max_{x(\cdot)} \sum_{s \in S_i} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right)$ for each $i \in \{2, \ldots, k+1\}$. Since there are no externalities among the scenes in S_1, Algorithm 2 returns the optimal allocation x^* to the scenes in S_1 by Theorem 1 .Hence, it holds

$$\sum_{s \in S_1} \left(\Pi^s \tilde{q}\left(x^*, s\right) \theta_{x^*(s)}\right) = \max_{x(\cdot)} \sum_{s \in S_1} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right) \geq \max_{x(\cdot)} \sum_{s \in S_i} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right)$$

for each $i \in \{1, \ldots, k\}$. This implies that

$$\begin{aligned} \sum_{s \in S_1} \left(\Pi^s \tilde{q}\left(x^*, s\right) \theta_{x^*(s)}\right) &\geq \frac{1}{k+1} \sum_{i \in \{1, \ldots, k+1\}} \max_{x(\cdot)} \sum_{s \in S_i} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right) \\ &\geq \frac{1}{k+1} \max_{x(\cdot)} \sum_{i \in \{1, \ldots, k+1\}} \sum_{s \in S_i} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right) \\ &= \frac{1}{k+1} \max_{x(\cdot)} \sum_{s \in S} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right) \end{aligned}$$

where while the second comes from the fact that the externalities are only negative. This concludes the proof.

Theorem 5.8 Algorithm 3 provides a $\frac{1-1/e}{k+1}$-approximation to META-SD-E. Moreover, it runs in polynomial-time.

Proof. As a first step, we show that the given an i allocation x_i provides a good approximation of the optimal allocation value considering only scenes S_i. Formally, we show that $\sum_{s \in S_i} \left(\Pi^s \tilde{q}\left(x_i, s\right) \theta_{x_i(s)}\right) \geq \left(1 - \frac{1}{e}\right) \max_{x(\cdot)} \sum_{s \in S_i} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right)$. To see that, it sufficient to apply Theorem 4 to the problem restricted to scenes S_i. The inequality holds since by considering only the scenes in S_i (that have distance larger than the memory k) the problem is equivalent to a problem without externalities. Let x^* be the allocation returned by Algorithm 3. Then, it holds

$$\sum_{s \in S_i} \left(\Pi^s \tilde{q}\left(x^*, s\right) \theta_{x^*(s)}\right) = \underset{i \in \{1,\dots,k+1\}}{\operatorname{argmax}} \sum_{s \in S} \left(\Pi^s \tilde{q}\left(x_i, s\right) \theta_{x_i(s)}\right)$$

$$\begin{aligned}
&\geq \frac{1}{k+1} \sum_{i \in \{1,\dots,k+1\}} \sum_{s \in S} \left(\Pi^s \tilde{q}\left(x_i, s\right) \theta_{x_i(s)}\right) \\
&\geq \frac{1}{k+1} \sum_{i \in \{1,\dots,k+1\}} \left(1 - \frac{1}{e}\right) \max_{x(\cdot)} \sum_{s \in S_i} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right) \\
&\geq \frac{1}{k+1} \left(1 - \frac{1}{e}\right) \max_{x(\cdot)} \sum_{i \in \{1,\dots,k+1\}} \sum_{s \in S_i} \left(\Pi^s \tilde{q}(x, s) \theta_{x(s)}\right) \\
&= \frac{1}{k+1} \left(1 - \frac{1}{e}\right) \max_{x(\cdot)} \sum_{s \in S} \left(\Pi^s \tilde{q}\left(x_i, s\right) \theta_{x_i(s)}\right),
\end{aligned}$$

where the second inequality comes from the guarantees of the continuous greedy algorithm (Theorem 4), while the third comes from the fact that the externalities are only negative. This concludes the proof.

List of Figures

Acknowledgements

Ringrazio il mio relatore, il Prof. Nicola Gatti, e i correlatori Matteo Castiglioni, Alberto Marchesi e Giulia Romano, con cui ho trascorso gli ultimi mesi del mio percorso di studi nello sviluppare questo progetto di tesi che conclude cinque anni di studio e di bellissime esperienze.

Alla mia famiglia, a mia madre Anna, a mio padre Tonino e a mio fratello Lorenzo, grazie per aver reso possibile questa esperienza e per avermi sempre supportato in qualsiasi scelta nella mia vita.

Un grazie ai miei storici amici della Puglia e ai miei cari amici di Milano, con cui ho condiviso questi fantastici anni. Infine, un grazie speciale ai miei amici di Barcellona e del Lussemburgo che sono stati parte di uno dei momenti più belli di sempre.