POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Reinforcement Learning for Digital Advertising Cross-Channel Budget Optimization

Tesi di Laurea Magistrale in
Computer science and engineering

Author: **Fabio Patella**

Student ID: 995601
Advisor: Marcello Restelli
Co-advisors: Marco Mussi, Alessandro Nuara, Alberto Maria Metelli
Academic Year: 2023-2024

Abstract

Digital advertising has become an essential tool for businesses, driven by the increasing use of the internet, cost-effectiveness, precise targeting, and measurable performance metrics. It removes geographic barriers, enabling global reach and integration with other digital marketing strategies. However, as digital advertising grows, efficiently allocating budgets across multiple campaigns has become a critical challenge. Businesses need to determine how much to invest in specific campaigns over others to maximize the overall revenue and conversions generated by their advertisements. Traditional budget allocation methods rely heavily on human expertise, which often struggles to capture the highly dynamic nature of advertising environments. These environments are inherently stochastic, exhibit time-delayed revenue effects, and follow seasonal trends, making the budget optimization increasingly complex. In order to solve this problem, this thesis focuses first on designing a simulator that models the dynamics of an advertising environment, including how user interest evolves over time and leads to conversions. Calibrated with real-world data, the simulator generates synthetic data that follows the structure of a Markov Decision Process (MDP), capturing the sequential and stochastic nature of budget allocation decisions. This structured data serves as the foundation for training a reinforcement learning algorithm. Specifically, the Fitted Natural Actor-Critic (FNAC) method is employed to learn to allocate resources across different advertising channels depending on the current state of the environment. Through extensive experimentation, this research demonstrates that RL-based strategies like FNAC significantly enhance budget efficiency and adaptability compared to traditional optimization approaches. The results show that RL not only improves budget allocation decisions but also adapts dynamically to changing market conditions, offering a promising direction for the future of digital advertising management.

Keywords: digital advertising, budget optimization, reinforcement learning, machine learning

Abstract in lingua italiana

La pubblicità digitale è diventata uno strumento essenziale per le aziende, grazie alla crescente diffusione di Internet, alla sua convenienza, al targeting preciso e alla possibilità di misurare le prestazioni. Essa elimina le barriere geografiche, consentendo un'ampia portata globale e un'integrazione efficace con altre strategie di marketing digitale. Tuttavia, con la crescita della pubblicità digitale, allocare in modo efficiente i budget tra più campagne è diventata una sfida cruciale. Le aziende devono determinare quanto investire in specifiche campagne rispetto ad altre per massimizzare le conversioni e il ricavo complessivo generati dagli annunci pubblicitari. I metodi tradizionali di allocazione del budget si basano fortemente sull'esperienza umana, che spesso fatica a catturare la natura altamente dinamica degli ambienti pubblicitari. Questi ambienti sono intrinsecamente stocastici, presentano effetti ritardati sui ricavi e seguono tendenze stagionali, rendendo l'ottimizzazione del budget sempre più complessa. Per affrontare questo problema, questa tesi si concentra inizialmente sulla progettazione di un simulatore in grado di modellare le dinamiche di un ambiente pubblicitario, includendo l'evoluzione dell'interesse degli utenti e la generazone delle conversioni. Calibrato con dati reali, il simulatore genera dati sintetici che seguono la struttura di un Processo Decisionale di Markov (MDP), catturando la natura sequenziale e stocastica delle decisioni di allocazione del budget. Questi dati strutturati vengono utilizzati per addestrare un algoritmo di apprendimento per rinforzo (RL). In particolare, viene applicato il metodo Fitted Natural Actor-Critic (FNAC) per apprendere come allocare le risorse tra i diversi canali pubblicitari in base allo stato attuale dell'ambiente. Attraverso un'ampia sperimentazione, questa ricerca dimostra che strategie basate sul RL, come FNAC, migliorano significativamente l'efficienza del budget e la capacità di adattamento rispetto ai metodi di ottimizzazione tradizionali. I risultati mostrano che il RL non solo ottimizza le decisioni di allocazione del budget, ma si adatta dinamicamente ai cambiamenti delle condizioni di mercato, offrendo una direzione promettente per la gestione futura della pubblicità digitale.

Parole chiave: pubblicità digitale, ottimizzazione del budget, apprendimento per rinforzo, apprendimento automatico

Contents

1 | Introduction

Digital advertising has grown over the years to become an indispensable tool for businesses in revolutionizing ways of doing marketing and engaging with customers. Several reasons contribute to its growing importance, making it a strong mechanism for driving business growth in contemporary times. Among the primary reasons that digital advertising has gained such prominence is the rapid increase in the usage of the internet. With over 5 and half billion Internet users worldwide by the year 2024 [30], the internet is the network on which the majority of people spend their time socializing, shopping, or consuming content. This, therefore, has driven business to adapt ways in which marketing can reach them. Digital advertising allows firms to reach out directly to this massive group, which previously was unimaginable. The main advantage of digital advertising is that it is much cheaper compared to traditional media. Whereas traditional ways of advertising, such as TV, radio, or print advertisements, require big financial investments, digital advertising opens an economical avenue for businesses of all sizes to effectively compete [9]. Apart from the affordability, digital advertising has the added advantage of precision targeting. While traditional advertising typically relies on broad demographic groups, digital advertising enables businesses to target very specific audiences based on detailed parameters such as age, gender, interests, and even previous online behavior. This targeted approach ensures that businesses can deliver tailored messages to the most relevant consumers, increasing the likelihood of engagement and conversions [37]. Another notable advantage of digital advertising is its measurable results. Unlike traditional advertising, which may be difficult to measure in terms of effectiveness, digital campaigns offer real-time and granular performance insights. In digital campaigns, advertisers can measure impressions, click-through rates, conversions, and return on investment in real time. This ability enables them to make quick changes in strategies and optimize campaigns for better results [33]. In addition, digital advertising has completely opened up avenues from around the world, eliminating geographic barriers that had become stumbling blocks for the expansion of businesses. What was initially reserved for local businesses can be accessed by consumers on the other side of the world with just a few simple clicks. Global accessibility comes with huge opportunities for growth: to enter new markets and attract

new, different kinds of customers. From social networking sites to any other search engines and online marketplaces, companies can reach out with an international audience, thus extending their customer base way beyond what was ever possible with traditional marketing [16]. This is further enhanced by the integration of digital advertising into other digital marketing strategies. Businesses can combine paid digital advertising with organic content, social media outreach, email campaigns, and other tactics to create a cohesive marketing ecosystem. This integrated approach ensures that businesses remain top of mind for consumers at various touchpoints along their purchasing journey. The ability to run cross-channel campaigns increases the likelihood of success and helps build stronger customer relationships [10].

However, despite its many advantages, digital advertising operates in a highly complex market, with several tasks and underlying mechanisms that can benefit from the automation and optimization offered by Artificial Intelligence (AI). Human analysis alone often falls short due to the numerous factors that need to be considered in the decision-making process. For example, seasonality, trends, promotions, and delayed effects of awareness campaigns all contribute to the complexity. Additionally, data provided by platforms such as Google Analytics, especially when it comes to conversion attribution, can be unreliable and may lead to misleading conclusions [27]. Furthermore, cross-channel interactions are often overlooked or difficult to detect through manual analysis, making it challenging to achieve an optimal advertising strategy. From the advertiser's perspective, one of the most significant concerns is improving the Return on Investment (ROI) while planning for future spending. Despite having access to vast amounts of data, accurately analyzing and utilizing this data remains a difficult task. One of the primary methods for increasing ROI is through effective budget allocation across various advertising channels. Improper allocation can lead to wasted resources on channels that are either oversaturated or underperforming. In addition, the timing of investments in certain platforms is critical. For example, specific periods during the year may amplify or diminish the impact of an advertisement campaign. Therefore, effective budget allocation involves understanding not just where to invest, but also when to invest, factoring in seasonal fluctuations, delayed awareness effects, and other such variables [23]. Performance forecasting is another challenge for advertisers, and it is closely tied to budget optimization. Advertisers wish to predict, with a certain degree of confidence, how their advertising expenditures will translate into revenue, conversions, or leads. However, this task becomes increasingly difficult when combined with the need to allocate budgets across different advertising channels. Forecasting performance across multiple platforms adds a layer of uncertainty and complexity to the optimization process. Ultimately, the task of optimally allocating budgets

across various advertising channels and time periods is a highly complex challenge, especially when coupled with the necessity of accurately predicting the campaign outcomes. This highlights the need for advanced tools, such as AI-based algorithms, to analyze vast datasets, account for numerous influencing factors, and optimize both budget allocation and performance forecasting in the digital advertising space.

1.1. Thesis Goal

Traditional budget allocation methods struggle to handle the complexity of advertising environments. They often depend on human knowledge to model key factors, such as fluctuations in cost and impressions over time, variations in how users perceive campaigns, and the cross-channel effects of advertising on user interest. Additionally, conversions do not occur immediately after users see an advertising, there is often a delay, and users may make a purchase decision only after being exposed to multiple advertisements across different channels. Traditional methods frequently simplify the problem, failing to account for all these factors. This thesis aims to explore RL algorithms as an alternative to traditional methods, with the challenge of capturing the full complexity of the advertising model. By doing so, RL can provide a more comprehensive understanding of the problem, ultimately enabling optimal budget allocation over time.

Developing a RL algorithm is a complex task, as it requires a model of the advertising environment and a substantial amount of data for training. Therefore the first objective of this thesis is to design and develop a simulator capable of modeling the dynamics of an advertising environment. These dynamics include the various types of advertising campaigns and how each of them influence the user interest in a product until the user buys the product. This behaviour depends on parameters that are calibrated using real-world data to ensure realistic behavior. The data involved is analyzed to identify seasonal patterns, ensuring that the simulator accurately reflects the identified behavior over time. The simulator is then used to generate synthetic data, structured according to a proposed MDP, which serves as the theoretical foundation for the RL framework. Specifically, the FNAC [24] method is used to optimize budget allocation strategies, with the goal of maximizing total revenue over an extended evaluation period. This allows for capturing the evolving behavior of the learned policy over time. Finally, we aim to evaluate the performance of FNAC and other RL baselines on a real dataset, comparing their effectiveness against a traditional approach. The traditional method learns a fixed budget allocation strategy over time, while RL algorithms attempt to optimize a dynamic policy that adapts to changing conditions. The experimental results demonstrate the performance of each

baseline and highlight why RL methods outperform traditional techniques.

1.2. Structure of the Thesis

This thesis is organized as follows. In Chapter 2, we introduce the foundational concepts of online advertising, exploring how various entities, such as advertisers and publishers, interact within the ecosystem. We discuss how advertisers track user interest and the metrics they use to to evaluate the effectiveness of their campaigns, we examine the different advertising channels available, highlighting their differences, when it is preferable to invest in one over another, and the key metrics to use for evaluating their performance. Additionally, we examine the advertising buying process, focusing on how publishers manage it as an auction and the resulting impact on the cost-impressions curve. In Chapter 3, we introduce the fundamental principles of ML and RL algorithms. We begin with an overview of different types of ML algorithms, followed by a more in-depth discussion of RL algorithms. Specifically, we explore the theory of MDPs as theoretical foundation of the RL framework, review various classifications of RL algorithms, and provide a detailed explanation of the FNAC algorithm. In Chapter 4, we review the existing literature on online advertising optimization, covering both traditional ML techniques and more modern approaches using RL. This includes a discussion of the key results and methodologies that have shaped the current landscape of advertising optimization. In Chapter 5, we provide a detailed description and mathematical formulation of the optimization and learning problem. In Chapter 6 we explain how we modeled the simulator, detailing the reasoning behind each of our choices. We provide the complete algorithm implementing the simulator's behavior, along with the procedure used to tune its parameters. We also explain how the simulator was used to generate and augment data, how we modeled the advertising environment as an MDP, and how the data structures align with the MDP framework. Finally, we outline the implementation and training process of the FNAC algorithm. In Chapter 7, we describe the dataset used for the experiments, detailing the types of campaigns included and the time period during which they were active. A seasonality analysis of the dataset is provided to illustrate the campaigns' behavior over time. We then outline the parameters obtained through the fitting process of the simulator for this dataset. Next, we present the parameters used in the FNAC implementation and training process, as well as the specific parameters employed to model the MDP. We also introduce the other baselines used to assess the model's ability to optimize budgets. Specifically, we compare FNAC with two RL algorithms as Proximal Policy Optimization (PPO) [28] and Soft Actor-Critic (SAC) [13] as well as a traditional machine learning method, Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [15]. Finally, we

present the results from these baselines and provide a comparison of their performance. In conclusion, in Chapter 8, we summarize the findings of the thesis and discuss potential future research directions.

2 | Context

In this chapter, we introduce the key concepts of digital marketing to provide a context for the problem of budget optimization. In Chapter 2.1, we discuss why digital advertising is becoming more and more important and we give the definition of some metrics useful to fully understand the rest of the chapter. In Chapter 2.2, we explain the concept of marketing funnel, in Chapter 2.3 the main advertising channels that can be used to target users of a specific funnel stage and in Chapter 2.4 we explain how auctions for impressions or clicks works for different advertising channel and how this inpacts the cost-impression curve.

2.1. Digital Advertising

Digital advertising has become increasingly important due to the growing reliance on digital platforms and the shift in consumer behavior toward online interactions. With people spending significant time on the internet—browsing websites, engaging on social media, and consuming online content—advertisers have gained unprecedented access to their target audiences. This evolution has made digital advertising a cornerstone of modern marketing strategies, offering businesses the ability to reach specific demographics, measure campaign effectiveness, and engage users in real-time.

In this ecosystem, three main entities interact. Advertisers are businesses or individuals who promote their products or services through advertisements to attract users and encourage actions like purchases or inquiries. Publishers, such as Google, Meta and Amazon provide the platforms that host these advertisements, enabling advertisers to display their content to users across various digital channels. Finally, the users are the target audience of these advertisements, encountering them while using digital platforms. Advertisements can appear in numerous formats, such as banners, videos, or sponsored posts, and users can interact with them either passively, by viewing them , or actively, by clicking on them.

To evaluate the effectiveness of these interactions, Key Performance Indicators (KPIs) are crucial. In the context of digital advertising, KPIs are measurable values that help

advertisers assess whether their campaigns are producing the desired results. They serve as essential tools for goal setting, performance evaluation, and campaign optimization. Some of the most important KPIs are:

- **Impressions**: This metric represents the number of times an advertisement has been shown on a digital platform, regardless of whether it was seen by unique or repeated users. It serves as a fundamental measure for evaluating the reach and exposure of an online advertising campaign.
- **Reach**: This metric refers to the total number of unique users who have seen the advertisement at least once. Unlike impressions, which count every instance an advertisement is displayed (including repeated views by the same user), reach focuses on how many different individuals have been exposed to the campaign. A high reach indicates that the advertisement has successfully engaged a broad audience.
- **Clicks**: A measurement that tracks the number of times users interact with an advertisement by clicking on it, leading them to a designated webpage or other digital asset. This metric helps assess how engaging and relevant the advertisement is to its audience.
- **Conversions**: This indicator reflects whether an advertisement has successfully achieved its intended goal. A conversion typically occurs when a user takes a desired action after viewing an advertisement, such as filling out a form, signing up for a service, or completing an online purchase.
- **Revenue**: This metric is the total income generated by the advertising campaign and can be calculated easily once the number of conversions and the value assigned to each conversion are known.
- **Return on investment (ROI)**: is computed by comparing the revenue generated from the campaign and the total cost of the campaign itself. Because ROI clearly relates advertising expenditure to financial consequences, it is an important indicator for advertisers.
- **Click-Through Rate (CTR)**: Expressed as a percentage, this metric shows how often people click on an advertisement after being exposed to it. It is calculated by dividing the number of advertisement clicks by the total number of impressions, providing insight into an advertisement's effectiveness in generating interest.
- **Conversion Rate (CR)**: This metric indicates the proportion of users who complete a specific action after clicking on an advertisement. It is useful for comparing the performance of different advertising strategies, as a high conversion rate suggests

effective audience targeting and advertisement relevance.

- **Cost-Per-Click (CPC)**: This represents the actual amount an advertiser pays for each click in a pay-per-click (PPC) advertising model. CPC is a key metric for evaluating the cost-efficiency and ROI of digital advertising campaigns.
- **Cost-Per-Impression (CPM)**: This metric shows the cost an advertiser pays for 1000 impressions of an advertisement. It's often used in brand awareness campaigns where the focus is on reach rather than direct actions like clicks or conversions.

2.2. Marketing Funnel

The advertisement funnel is a conceptual model that describes the stages a potential customer goes through from first becoming aware of a product or service to ultimately making a purchase or taking a desired action [1]. This model helps advertisers understand the customer journey and tailor their marketing strategies to guide users through each stage effectively. The advertisement funnel is typically divided into several key stages, each representing a different phase in the buyer's decision-making process. A possible funnel model could be composed by these 4 stages:

- **Awareness**: at this stage the goal is to make potential customers aware of a product or service. This is often the widest part of the funnel and involves reaching a large audience through various channels such as social media advertisements, display banners, video commercials, and influencer partnerships. The purpose here is to create brand recognition and attract as many prospects as possible. This stage is characterized by high impressions and low engagement as users are simply becoming familiar with the brand or product.
- **Interest**: once users are aware of the product, the next step is to cultivate their interest. At this stage, the goal is to engage the audience by providing more in-depth information about the product's features, benefits, and value. Content in this stage might include blog posts, educational videos, webinars, or targeted email campaigns.
- **Intent**: at this stage, users have shown a clear intent to buy and are closer to making a purchase decision. They may have added items to their shopping cart, visited product pages frequently, or signed up for a trial. Advertisers often use strategies such as special offers, limited-time discounts, or targeted advertisements to encourage users to take action.

- **Conversion**: The conversion stage is when the user takes the desired action, such as making a purchase, signing up for a service, or filling out a contact form. This is the most important stage of the funnel, as it represents the culmination of the advertising effort.

Maintaining interest throughout the customer journey can be challenging. Users who enter the funnel at the awareness stage may initially show curiosity, but as they move through stages such as consideration and intent, some will lose interest or become disengaged with the passing of time, especially if they don't see any advertisements for a long period. Figure 2.1 provides a graphical representation of the advertising funnel.

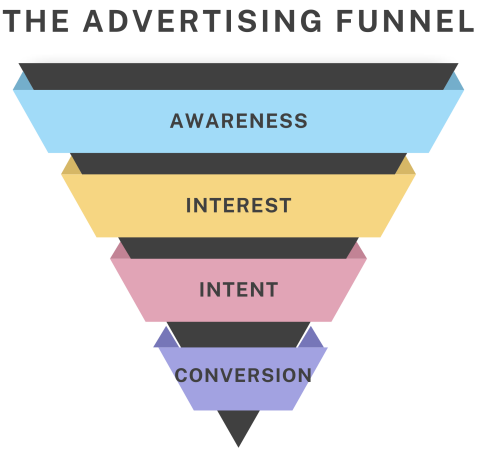


Figure 2.1: Advertising funnel.

2.3. Advertising Channels

As the internet and digital technologies have evolved, the advertising landscape has expanded significantly, offering advertisers an extensive array of channels to choose from. Each channel comes with its unique advantages and limitations, making it better suited for specific objectives and stages of the marketing funnel. For example, search engine advertising is effective for capturing immediate intent, while social media campaigns are well-suited for building brand awareness and fostering engagement [29]. Moreover, the integration of multiple channels, known as multichannel advertising, enables advertisers to maximize reach and effectiveness. This approach involves running the same campaign across different platforms, such as a search engine and a social network, each contributing to different campaign goals and reinforcing the overall message. Given the different types of channels, there will be different users and this leads to an increased size of the possible

engaged audience as well as different and specific kind of target audience. The three main advertising channels are search, display, and social media.

2.3.1. Search

When a user enters a query into a search engine using specific keywords or phrases, targeted advertisements appear on the search engine results page. Since the user is actively looking for information or products, they already have a clear intent. As a result, these advertisements are highly targeted, requiring advertisers to select the keywords they want their advertisements to be associated with. This type of advertising typically follows a pay-per-click (PPC) model, meaning advertisers are charged only when a user clicks on their advertisement. Additionally, advertisers participate in a bidding system, determining how much they are willing to pay for their advertisement to be displayed when a specific keyword is searched. To improve their campaigns and enhance advertisement performance, advertisers analyze key metrics such as click-through rates and conversion rates. Due to its precise targeting, this form of advertising is highly effective at driving conversions, which is why it is commonly used in the final stage of the sales funnel. Table 2.1 reports the market shares of the most commonly used search engines at the beginning of 2025 according to [31]. The most dominant search engine is Google, with a commanding 89.62% market share, significantly outpacing other competitors. Bing follows with 4.04%, while other search engines, such as Yandex (2.62%) and Yahoo (1.34%), have much smaller shares.

Search Engine	Market Share (Jan 2025)
Google	89.62%
Bing	4.04%
Yandex	2.62%
Yahoo	1.34%
Baidu	0.73%
DuckDuckGo	0.68%

Table 2.1: Market Share of Leading Search Engines Worldwide (January 2025).

2.3.2. Display

Display advertising involves showcasing advertisements in the form of banners, images, or videos on third-party websites or mobile apps. This type of advertising is powerful because it leverages visual elements to communicate the campaign's message effectively to the target audience. To achieve success with display advertisements, advertisers must

not only define their campaign objectives and audience but also design visually compelling advertisements that capture users' attention. Unlike search advertisements, the primary purpose of display advertising is to enhance brand awareness rather than drive immediate conversions. Additionally, it is commonly used for retargeting, displaying products that users have previously searched for to re-engage potential customers. KPIs for measuring the success of display advertisements include reach and impressions, while click-through rate and conversion rate are particularly relevant for campaigns focused on driving conversions.

2.3.3. Social Media

Social media advertising involves displaying advertisements on platforms such as Facebook, Instagram, Pinterest, and TikTok. These social networks offer several features that make them highly valuable for marketers. One of their biggest advantages is the vast amount of user data they continuously collect, allowing advertisers to target specific demographics, behaviors, genders, ages, and interests. This level of precision ensures that advertisements are shown to users who are most likely to be interested in the promoted products or services. Additionally, social media enables direct engagement and interaction, helping brands build a sense of community with their audience. Due to its versatility, social media advertising can be utilized at different stages of the marketing funnel, with the KPIs varying based on the campaign's objectives. Figure 2.2 represents the growth of global social media users from 2015 to 2025 according to [6]. It shows a steady increase in the number of active social media users over the past decade, with significant year-over-year growth, especially in the earlier years.

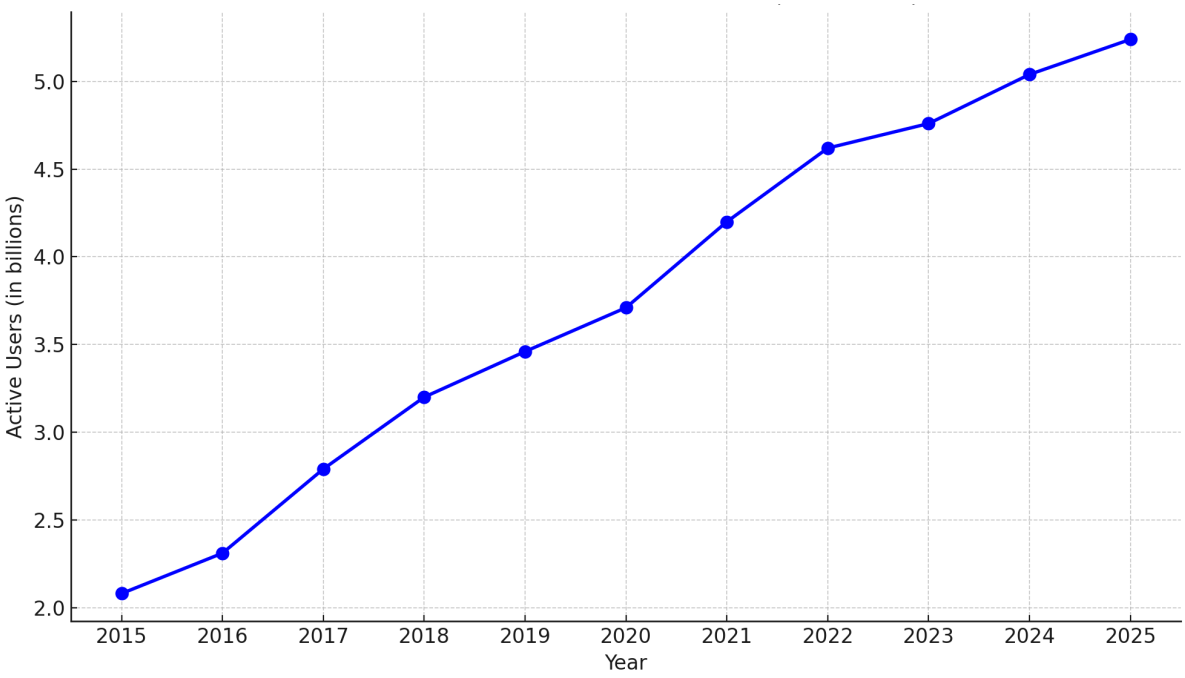


Figure 2.2: Global Social Media Users Over Time (2015-2025) [6].

2.4. Sponsored Auctions

Sponsored auctions are a fundamental mechanism in digital advertising, determining how advertisements are placed on search engines, social media platforms, and display networks. These auctions allow advertisers to bid for advertisement placements based on various factors, such as user relevance, bid amount, and quality of the advertisement. The two primary types of sponsored auctions are search auctions and display auctions, each with distinct characteristics.

Search auctions occur on search engines like Google and Bing, where advertisers bid on specific keywords relevant to user queries. These auctions operate using a PPC model, meaning advertisers only pay when users click on their advertisements. Advertisement placement in search auctions is determined by a combination of bid amount and advertisement quality, as detailed by through techniques as the Second-Price Auction (GSP) [11] and the Vickrey-Clarke-Groves (VCG) mechanism [34].

In contrast, display auctions take place on websites, social media platforms, and advertisement networks where advertisements are shown based on user behavior, interests, and demographic data. Unlike search auctions, display auctions often operate on a CPM or CPC basis rather than PPC. These auctions use Real-Time Bidding (RTB) technology to determine which advertisement gets displayed to a user in milliseconds. This is a much

more complex task that the search auction and it can be catagorized into the single item (a single impression) multi-person auction but still with similar mechanism to GSP and VCG [39].

To better understand how the cost impressions curve of advertisement campaign works, it is useful to see more in detail how FSP and VCG work. In GSP if a user clicks on an advertisement in position k, the search engine charges the advertiser an amount equivalent to what the advertiser in the next lower position, $k + 1$, is paying. If only a single advertisement were displayed per search results page, this method would effectively function as a Vickrey-Clarke-Groves (VCG) auction, also known as a standard second-price auction.

Consider a scenario with N bidders and $K < N$ available slots. Each slot has a click probability denoted as α_i, where higher slots receive more clicks:

$$\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_K. \tag{2.1}$$

To generalize, we can introduce $N - K$ additional virtual slots with a CTR of zero, meaning $\alpha_i = 0$ for $i > K$. Each bidder specifies a valuation v_i representing the perceived value of being shown in a slot and submits a bid b_i, indicating the highest amount they are willing to pay. Importantly, the bid does not necessarily reflect the true valuation v_i. Additionally, each advertisement is assigned a quality score q_i, which represents its relevance and effectiveness for users. Slots are allocated based on the highest product of bid and quality score, $b_i q_i$. The bidder with the highest product secures the first position, the second-highest takes the next slot, and so forth. If a user clicks on an advertisement, the cost incurred by the advertiser follows the formula:

$$p_{(i)} = \alpha_i b_{(i+1)} + \varepsilon, \quad \forall i \in \{1, \ldots, \min(N, K)\}, \tag{2.2}$$

where $b_{(i)}$ denotes the bid of the i-th ranked bidder, and ε represents a minimum increment. If a bidder does not secure a slot or receive clicks, they are not charged. Unlike the VCG mechanism, GSP is not a truthful auction, meaning advertisers may benefit from strategic bidding instead of bidding their true valuation. As analyzed in [11], the worst-case Nash equilibrium in GSP yields revenue equivalent to that of the VCG mechanism under complete information. However, in a Bayesian setting, the worst Bayes-Nash equilibrium in GSP can lead to significantly lower revenue for the auctioneer compared to VCG.

In both GSP and VCG auctions, advertisers submit bids reflecting the maximum amount

they are willing to pay for an impression. However, they do not actually pay their own bid amount. Instead, the second-highest bid (or a variation of it, depending on the auction rules) determines the price paid by the highest bidder. This mechanism ensures that advertisers are constantly influenced by the bidding behavior of their competitors. When only a few bidders participate, the competition remains moderate, and bid adjustments are relatively small. However, as more advertisers enter the auction and competition grows, each new bidder pushes the price upward, forcing existing participants to increase their bids to maintain visibility. This creates a self-reinforcing cycle where every incremental increase in bidding pressure causes a disproportionately large increase in the clearing price, leading to an exponential-like cost impression curve as shown in Figure 2.3.

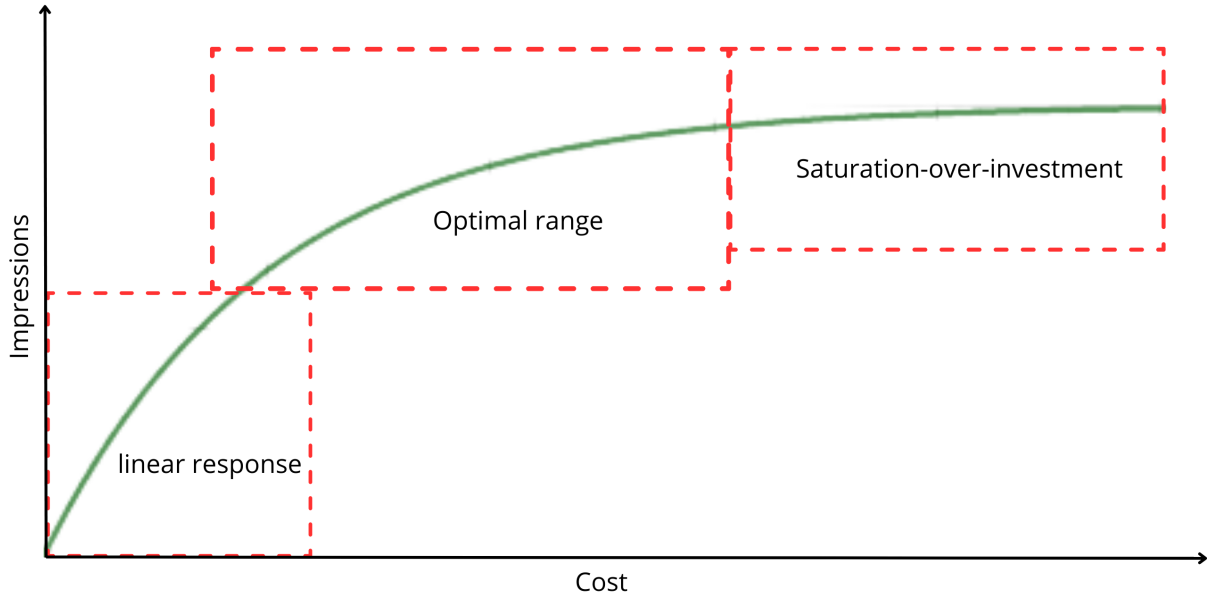


Figure 2.3: Shape of the cost-impressions curve.

3 | Theoretical Background

This section presents the theoretical background necessary to understand the rest of the thesis, specifically in Chapter 3.1 various classifications of Machine Learning (ML) algorithms are discussed while Chapter 3.2 reviews the theory used to develop the reinforcement learning proposed solution to use as a remainder for the rest of the thesis.

3.1. Classification of Machine Learning Algorithms

ML is a subset of AI that enables systems to learn patterns from data and make predictions or decisions without being explicitly programmed. It involves the development of algorithms that improve automatically through experience. ML encompasses several approaches, each tailored to specific types of data and tasks [3]:

- **Supervised Learning**: Supervised learning is characterized by the use of labeled datasets, where each input has an associated output. The goal is for the algorithm to learn the mapping between inputs and outputs to make predictions on unseen data.
- **Unsupervised Learning**: Unsupervised learning operates without labeled outputs. Instead, the algorithm identifies inherent structures and patterns within the data.
- **Reinforcement Learning**: RL presents a unique paradigm where an agent interacts with an environment, learning through a system of rewards and penalties. By continuously refining its strategy, the agent optimizes decision-making over time.

3.2. Reinforcement Learning

Reinforcement Learning is the branch of ML that focuses on training agents to learn the optimal behavior to achieve a specific goal while interacting with an external environment. This interaction consists of a sequence of decisions or actions, and it is typically modeled through the MDP framework which is explained in Chapter 3.2.1. Chapter 3.2.2

presents the different categories of RL algorithms, while Chapter 3.2.3 provides a detailed explanation of the FNAC algorithm.

3.2.1. Markov Decision Processes

Let $X \subset \mathbb{R}^p$ and $A \subset \mathbb{R}^q$ be two compact sets. Consider a controlled Markov chain $\{X_t\}$, where each state X_t belongs to X, and the control parameter A_t lies in A. The transition probabilities of the chain are governed by the kernel:

$$P[X_{t+1} \in U_X \mid X_t = x, A_t = a] = P_a(x, U_X),$$

for any measurable subset $U_X \subset X$. The sequence $\{A_t\}$ represents the control actions, where A_t denotes the action taken at time t.

The objective of the decision-maker is to select the control sequence $\{A_t\}$ to maximize the expected value of the following performance measure:

$$V(\{A_t\}, x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t, A_t) \mid X_0 = x\right],$$

where $0 \leq \gamma < 1$ is a discount factor, and $R(x, a)$ is the reward associated with taking action a in state x. We assume that the reward function is defined and bounded for all (x, a) pairs.

The 5-tuple (X, A, P, r, γ) defines a *Markov Decision Process (MDP)*.

For an MDP $M = (X, A, P, r, \gamma)$, the *optimal value function* V^* for each state $x \in X$ is given by:

$$V^*(x) = \max_{\{A_t\}} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(X_k, A_k) \mid X_0 = x\right],$$

and satisfies the Bellman optimality equation:

$$V^*(x) = \max_{a \in A} \left\{ \int \left[r(x, a, y) + \gamma V^*(y)\right] P_a(x, dy) \right\}. \tag{3.1}$$

The *optimal Q-values* $Q^*(x, a)$ for state-action pairs $(x, a) \in X \times A$ are defined as:

$$Q^*(x, a) = \int \left[r(x, a, y) + \gamma V^*(y)\right] P_a(x, dy).$$

Using Q^*, the optimal policy π^* is determined by the rule:

$$\pi^*(x) = \arg\max_a Q^*(x, a), \quad \forall x \in X.$$

The control sequence defined by $A_t = \pi^*(X_t)$ is optimal as it achieves the value function V^*. The mapping π^* is known as the *optimal policy* for the MDP M.

More generally, a *policy* is a time-dependent mapping π_t defined on $X \times A$, which generates a control sequence $\{A_t\}$ satisfying:

$$P[A_t \in U_A \mid X_t = x] = \int_{U_A} \pi_t(x, a)\, da, \quad \forall t,$$

where $U_A \subset A$ is any measurable subset. The value function $V^{\pi_t}(x)$ denotes $V(\{A_t\}, x)$ when the control sequence $\{A_t\}$ is induced by the policy π_t. A policy π is called *stationary* if it does not depend on the time t.

3.2.2. Classification of Reinforcement Learning Algorithms

Solving an RL task involves determining the optimal policy, that is, the policy that maximizes the expected return. When both the transition model P and the reward function R are known, the MDP can be solved by employing the dynamic programming approach. It decomposes the original problem into two tasks - policy evaluation and policy improvement - and finds the optimal policy by iteratively addressing them. However, the complete knowledge of the MDP is not available in most real-world applications. Moreover, it is common to deal with extremely large or even continuous state-action spaces. Therefore, in these cases, one has to adopt RL techniques that exploit function approximation methods to learn how to generalize the experience gathered from the observed agent-environment interaction. Three are the main approaches in the literature [2]:

- **Policy-based algorithms**: These methods directly build a parameterized policy function and optimize it to maximize the cumulative reward. Specifically, they leverage the Policy Gradient theorem [32] to update the parameters. Policy-based approaches are particularly effective in handling continuous action spaces but tend to be negatively affected by noisy environments and can get stuck in local optima.
- **Value-based algorithms**: These methods approximate value functions and derive the optimal policy indirectly by selecting the actions associated with the highest value estimates. They work well in scenarios where defining an explicit policy is challenging but they struggle with continuous action spaces and it may happen that

good value function approximations may correspond to bad policies.

- **Actor-critic methods**: Representing a hybrid approach, these methods combine the strengths of policy-based and value-based paradigms. They utilize an actor, which models the policy and selects actions, and a critic, which estimates the value function. This allows them to improve learning efficiency while maintaining the ability to handle continuous action spaces. PPO, SAC, and FNAC, which serve as baselines for experiments in Chapter 7, belong to the class of Actor-Critic methods.

Table 3.1 summarizes the key distinctions between these types of methods.

Method	Advantages	Disadvantages
Policy-Based	- Works well with continuous action spaces - Directly optimizes policy	- Negatively affected by noisy environments - Prone to local optima
Value-Based	- Learns a value function and derives a policy from it - Can work with only state-actions pairs without explicit policy.	- Struggles with continuous action spaces - Good value function approximations may correspond to bad policies and vice versa
Actor-Critic	- Combines strengths of policy-based and value-based methods - More stable learning - Uses an actor for policy and a critic for value estimation	- Requires tuning of both actor and critic

Table 3.1: Comparison of Policy-Based, Value-Based, and Actor-Critic Methods [5]

RL algorithm can be classified in different way also depending on how the exploration or data collection is performed:

- **On-Policy:** On-policy methods, learn the policy that they use to make decisions. In on-policy learning, the behavior policy—the policy used to generate experiences—is the same as the learned policy. This means that policy evaluation and improvement happen simultaneously. On-policy algorithms are empirically effective with function approximation and tend to perform well in environments where continuous adaptation is required. However, they require well-designed exploration strategies to balance exploration and exploitation, and may struggle with sample inefficiency as they discard experiences generated under different policies.

- **Off-Policy:** Off-policy methods, learn the optimal policy independently of the behavior policy. This allows for greater flexibility and efficiency, as off-policy algorithms can reuse past experiences and learn from a broader range of data. They can also implement advanced exploration strategies without being constrained by the current policy. Off-policy methods typically learn faster since they maximize over actions at each step. However, they can be susceptible to overestimation biases due to the need to select the best action at each step, and they may require additional mechanisms to mitigate these issues.

Another important distinction in RL is how models learn and update based on data availability. Two primary approaches are:

- **Offline Learning**: involves training a model using a fixed dataset before deployment. The training process requires multiple iterations over the dataset to optimize performance, after which the model remains static unless retrained with new data. This approach is beneficial when working with large, structured datasets. Offline methods are usually off-policy since the dataset is static and does not change based on the agent's learning process. If the agent augments the dataset during training by interacting with the environment using its current learned policy, the algorithm can be considered partially on-policy.
- **Online Learning**: enables models to update continuously as new data arrives. This allows algorithms to adapt dynamically to changing conditions, making them suitable for real-time applications. Online methods can be both on-policy and off-policy.

3.2.3. Fitted Natural Actor Critic

Fitted Natural Actor Critic (FNAC) [24] is an extension of Natural Actor Critic (NAC) algorithm [26]. The main idea of NAC is to use a critic component to estimate the optimal value function (3.1) of the states of a MDP and the actor component which returns the optimal action in a given state. Specifically FNAC modifies the critic of NAC to enable the use of general value function approximators, employing the natural gradient to update the policy of the actor. Natural gradients usually ensures a faster convergence by following the steepest ascent direction in Riemannian space [4, 17] :

$$\nabla_e \theta f(\theta) = G(\theta)^{-1} \nabla_\theta f(\theta), \tag{3.2}$$

where $G(\theta)$ is a positive definite matrix named the metric tensor. A common choice for this matrix in machine learning, particularly in the natural gradient framework, is the Fisher Information Matrix $F(\theta)$:

$$F(\theta) = \mathbb{E}_{\tau \sim p_\theta} \left[\nabla_\theta \log p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)^T \right], \tag{3.3}$$

where p_θ is a probability distribution parameterized by θ.

Using a compatible function approximation:

$$q_w(x_t, a_t) = w^T (\nabla_\theta \log \pi_\theta(x_t, a_t)), \tag{3.4}$$

and recalling Equations (3.2) and (3.3), the natural policy gradient simplifies to:

$$\nabla_e \theta J(\theta) = F(\theta)^{-1} \nabla_\theta J(\theta) = w. \tag{3.5}$$

Consequently, we only need to estimate w instead of $G(\theta)$, leading to a policy improvement step with learning rate α :

$$\theta_{t+1} = \theta_t + \alpha w. \tag{3.6}$$

FNAC employs a dataset D consisting of samples obtained from the environment. During each iteration k, the critic component processes these samples to approximate the value function $\hat{V}_\theta$, which is related to the policy π_θ. This approximation is then utilized to derive an estimated form of the advantage function $\hat{A}_\theta$, using a linear function approximation with compatible basis functions. This advantage function is subsequently used to perform the natural policy gradient update of the actor's parameter θ.

Specifically, $\hat{V}_\theta$ is determined by fitting a regression model and minimizing the Bellman Error with the dataset D, computing v^* such that $\hat{V}_\theta(x) \approx \hat{V}_{v^*}(x)$:

$$v^* = \arg\min_v \sum_t \frac{1}{\hat{\mu}(x_t)} \frac{\pi_\theta(x_t, a_t)}{\pi_0(x_t, a_t)} \left(r_t + \gamma \hat{V}_v(x_{t+1}) - \hat{V}_v(x_t) \right)^2, \tag{3.7}$$

where $\hat{\mu}(x_t)$ is the empirical distribution of state x obtained from the dataset D and $\frac{\pi_\theta(x_t, a_t)}{\pi_0(x_t, a_t)}$ takes into account that we are estimating V^θ with a policy π_θ that is different from the policy π_0 we have used to collect D.

Additionally, the advantage function $\hat{A}_\theta$ is approximated by solving the following regres-

sion problem:

$$w^* = \arg\min_w \sum_t \frac{1}{\hat{\mu}(x_t)} \left(r_t + \gamma \hat{V}_\theta(x_{t+1}) - \hat{V}_\theta(x_t) - \phi^\top(x_t, a_t) w \right)^2, \tag{3.8}$$

where w is a vector of linear coefficients. This vector represents the orthogonal projection of the advantage function $\hat{A}_\theta$ onto the linear space spanned by the compatible basis functions $\phi(x_t, a_t)$:

$$\phi(x_t, a_t) = \frac{\partial \log(\pi_\theta)}{\partial \theta}(x_t, a_t). \tag{3.9}$$

Finally the actor is updated following the update in Equation (3.6).

4 | Related Works

With the proliferation of digital advertising channels, optimizing budget allocation has become a critical challenge for marketers seeking to maximize return on investment while navigating complex and competitive landscapes. Numerous studies have addressed this challenge by leveraging artificial intelligence. This chapter reviews machine learning and reinforcement learning techniques for optimizing advertising campaign budgets.

4.1. Machine Learning Techniques

Yang et al. [38], which introduced a tailored advertising response function. This function was designed to model budget allocation across multiple search advertising markets over a limited time frame. Their approach utilized a dynamic programming solution to optimize the objective function effectively. Han and Gabor [14] developed a contextual bandit framework which uses supervised learning to predict the expected payouts of advertisement campaigns based on contextual features and historical performance data. Next, it extrapolates these predicted payouts to out-of-sample budget levels using a simple functional model to represent the payout distribution. Finally, the framework applies Thompson Sampling to address the explore-exploit trade-off, selecting budgets dynamically by sampling from the predicted payout distributions to maximize campaign performance. The approach proposed by Geyik et al. [12] involves a budget allocation method that distributes funds from the overall campaign to its sub-campaigns based on their performance and spending potential. To evaluate the performance of the sub-campaigns, they consider both last-touch attribution and multi-touch attribution. In last-touch attribution, the user's action is credited to the final advertisement they interact with, whereas in multi-touch attribution, the credit is divided among the advertisements the user encounters. Additionally, the budget allocation is split into two main tasks: determining the spending capability of a sub-campaign and calculating its return on investment. Their experimental results demonstrate that using multi-touch attribution to assess the performance of sub-campaigns results in more effective budget allocation. Nuara et al. [25] proposed algorithm for the online joint bid/budget optimization of pay-per-click multi-channel ad-

vertising campaigns. They formulated the optimization problem as a combinatorial bandit problem, in which they use Gaussian Processes to estimate stochastic functions, Bayesian bandit techniques to address the exploration/exploitation problem, and a dynamic programming technique to solve a variation of the Multiple-Choice Knapsack problem. Kui Zhao et al. [41]: present a data-driven approach to marketing budget allocation, leveraging historical data through a semi-black-box model that combines logit demand curves with neural networks. The authors formulate budget allocation as an optimization problem and design efficient algorithms to solve it, accommodating various business constraints. Chen et al. [8] studied the budget allocation challenge in situations where customer behavior is impacted by the cumulative effects of advertisements from various channels. They approached this by creating an aggregated and deterministic model, referred to as the "fluid" representation, to capture the state dynamics of customer behavior. To determine the dynamic budget allocation policy, they framed the problem as a stochastic dynamic programming issue. Their findings suggested that if the campaign duration is long enough, the marketing agency can achieve nearly optimal performance by using a fixed budget allocation throughout the campaign.

4.2. Reinforcement Learning Techniques

Li et al. [36] addressed the budget allocation problem for multi-channel advertising by proposing the Q-MCKP algorithm, which integrates Reinforcement Learning with the Multi-Choice Knapsack Problem. They enhanced Q-learning by discretizing the cost space into finite sub-intervals, thereby reducing the action space. This approach effectively transforms the budget allocation problem into an MCKP. Liu et al. [20] proposed a bidding strategy for Real Time Bidding in display advertising, which maximizes the number of clicks of an advertisement delivery period by using an optimal bidding factor generation policy. In their bidding strategy, the bidding agent divides an advertisement delivery period into several time slots and adjusts each impression's bidding price based on the optimal bidding factor of each time slot, to adapt to the highly dynamic RTB environment. They utilize the policy-based Twin-Delayed Deep Deterministic (TD3) framework to learn the optimal bidding factor generation policy, to produce an optimal continuous value as the bidding factor for each time slot. Li et al. [18] studied a budget allocation approach that leverages a RL Q-learning framework enriched with an advanced Differential Evolution (DE) algorithm to refine the Q-learning methodology. The RL element makes informed sequential decisions adjusting strategies to favor long-term rewards by assimilating environmental feedback. Complementing this, the DE algorithm introduces a clustering-based mutation technique, exploiting key groupings within the DE popula-

tion to generate novel solutions. Liu et al. [21] design a bidding function that compute the base price of each advertisement impression, and adjusts the base price to fit the real-time RTB environment by introducing a bidding adjustment factor. To this end, they model the adjustment factor decision as an MDP, and then use the stochastic policy SAC to solve the optimal adjustment factor generation policy. Wang et al. [35] propose a hierarchical offline DRL framework HiBid for cross-channel bidding with budget allocation. HiBid add three contributions based on the state-of-the art offline DRL approach mildly conservative q-learning [22]: auxiliary batch loss to alleviate the advertiser competition in high-quality channels, λ-generalization for adaptive constrained bidding strategy in response to changing budget, and CPC-guided action selection scheme for improving cross-channel CPC satisfactory ratio. Badanidiyuru et al. [7] study takes into account how incrementality, which measures the causal effect of showing an advertisement to a potential customer. They investigate the problem of how an advertiser can learn to optimize the bidding sequence in an online manner without knowing the incrementality parameters in advance. They formulate the offline version of this problem as a specially structured episodic MDP and then, for its online learning counterpart, propose a RL algorithm that uses mixed and delayed reward feedback from conversion incrementality. Zhao et al. [40] consider the environment changing problem: the state transition probabilities vary between two days. Since the observation that auction sequences of two days share similar transition patterns at a proper aggregation level, they formulate a robust MDP model at hour-aggregation level of the auction data and propose a controlby-model framework for RTB. Rather than generating bid prices directly, they decide a bidding model for impressions of each hour and perform real-time bidding accordingly. Liao et al. [19] address the problem of optimally displaying a mixed list of advertisements and organic items in e-commerce platform feed. One key problem is to allocate the limited slots in the feed to maximize the overall revenue as well as improve user experience, which requires a good model for user preference. They find that instead of modeling the influence of individual items on user behaviors, the arrangement signal models the influence of the arrangement of items and may lead to a better allocation strategy. They propose Cross Deep Q Network to extract the crucial arrangement signal by crossing the embeddings of different items and modeling the crossed sequence by multichannel attention.

5 | Problem Formulation

In this chapter, we give a description and a mathematical formulation of the objectives of the thesis. In Chapter 5.1, the optimization problem is formulated, while in Chapter 5.2, the formulation of the learning problem is presented.

5.1. Optimization Problem Formulation

An advertiser is provided with a collection of $N \in \mathbb{N}$ advertising campaigns $\mathcal{C} = \{C_1, \ldots, C_N\}$, where C_j is the j-th campaign, and a finite time horizon of $T \in \mathbb{N}$ days and a constant cumulative daily budget $\bar{b}$ which represent the exact total budget that the advertiser want to invest daily on all campaigns. For each day $t \in \{1, \ldots, T\}$ and for each campaign C_j, an advertiser selects a daily budget $b_{j,t}$. By selecting a daily budget $b_{j,t}$ for each campaign C_j, an advertiser obtains an expected revenue $r_{j,t}(b_{1,t}, b_{1,t-1}, \ldots, b_{1,t-\tau}, \ldots, b_{N,t}, \ldots, b_{N,t-\tau})$, where τ represents the number of past time steps over which the allocated budgets are assumed to have an impact on the current revenue. Indeed, the total revenue for campaign C_j at time t, denoted as $r_{j,t}$, is a function of the budgets allocated across all campaigns in $\mathcal{C}$ at time t as well as in previous time periods.

We define the total revenue obtained by a given budget allocation $\mathbf{b}$ as:

$$J(\mathbf{b}) = \sum_{t=1}^{T} \sum_{j=1}^{N} r_{j,t}(b_{1,t}, b_{1,t-1}, \ldots, b_{1,t-\tau}, \ldots, b_{N,t}, \ldots, b_{N,t-\tau}), \tag{5.1}$$

where $\mathbf{b} = (b_{j,t})_{j \in \{1,\ldots,N\}, t \in \{1,\ldots,T\}}$ represents the full budget allocation matrix for all campaigns over time. The function $J(\mathbf{b})$ in Equation (5.1) represents the total revenue obtained from all N advertising campaigns over the entire time horizon T. It is computed as the sum of the revenue $r_{j,t}$ obtained from each campaign C_j at each time step t, where the revenue depends on both the current and past budget allocations up to τ time steps. Our objective is to find the optimal budget allocation $\mathbf{b}^* = \arg\max_{\mathbf{b}}^* J(\mathbf{b})$ that maximizes the total revenue respecting the budget constraint:

$$\sum_{j=1}^{N} b_{j,t} = \bar{b}, \quad \forall t \in \{1, \ldots, T\}. \tag{5.2}$$

5.2. Learning Problem Formulation

We define similarly as done in Chapter 5.1 $\mathbf{b} = (b_{j,t})_{j \in \{1,\ldots,N\}, t \in \{1,\ldots,T\}}$ that represent the full budget allocation matrix for all campaigns over time found by a generic algorithm. Given this and using the total revenue definition (5.1) the expected total revenue can be formulated as:

$$R_T := \mathbb{E}\left[J(\mathbf{b})\right]. \tag{5.3}$$

Our goal is to design an algorithm that maximizes the expected total revenue R_T. Table 5.2 lists the descriptions of the symbols used in this chapter.

Symbol	Description
N	Number of advertising campaigns
$\mathcal{C} = \{C_1, \ldots, C_N\}$	Set of advertising campaigns
T	Finite time horizon of T days
$\bar{b}$	Constant cumulative daily budget
$b_{j,t}$	Daily budget allocated to campaign C_j on day t
τ	Number of past time steps over which budgets impact revenue
$r_{j,t}$	Expected revenue for campaign C_j at time t
$J(\mathbf{b})$	Total revenue obtained from a given budget allocation $\mathbf{b}$
$\mathbf{b}$	Full budget allocation matrix for all campaigns over time
$\mathbf{b}^*$	Optimal budget allocation matrix that maximizes total revenue
R_T	Expected total revenue: $R_T = J(E[J(\mathbf{b})]$

Table 5.2: Symbols and Descriptions for Budget Allocation in Advertising Campaigns.

6 | Proposed Solution

In Chapter 6.1, we describe the design of the simulator model and the methodology used to fit its parameters based on real-world data. In Chapter 6.2, we detail the modeling of the Markov Decision Process. Finally in Chapter 6.3 we outline the implementation and optimization of the FNAC framework.

6.1. Simulator Model

The goal of the simulator is to replicate advertising environments that closely resemble real-world scenarios based on actual data and campaigns. This allows the simulator to generate data that can be used to train RL algorithms as if they were operating on real data. The simulator is designed to approximate real-life user behavior, specifically their interest in products or brands showcased through advertisements. The core concept is that users' interest in a product increases after viewing an advertisement. The higher the interest level, the greater the likelihood of a conversion. Each day, the simulator is provided with a budget allocated for advertising campaigns. Using this budget, it generates impressions that are randomly assigned to some users. After all impressions have been viewed by users, the simulator adjusts their interest levels based on the type of campaign they were exposed to. Awareness campaigns are more effective at increasing interest when initial interest levels are low, whereas conversion campaigns are more impactful at higher interest levels. Both campaign types contribute to increasing user interest, while conversions are generated randomly at the end of each day, with the probability of conversion determined by the user's interest level. This simulation process enables the generation of data, which is then used to train and evaluate the FNAC framework.

6.1.1. Simulator Detailed Logic

A temporal horizon of $T \in \mathbb{N}$ days is fixed, along with $N \in \mathbb{N}$ advertising campaigns denoted as $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$ and their corresponding type between awareness and conversion. The simulator maintains an array of size M, where each element $u_m \in [0, 1]$ represents the interest level of a user m for all m in the range $1 \leq m \leq M$. At the

beginning of each day the interest array is multiplied by a constant δ which represent the forgetting factor i.e how much user lose interest in time with regard to a product or brand. For each day $t \in \{1, \ldots, T\}$, the simulator receives as input a daily budget $b_{j,t}$ for each campaign C_j, where $j \in \{1, \ldots, N\}$. The impressions generated are determined using the following formula:

$$I_{j,t} = \alpha_{j,t} \cdot (1 - \exp(-\beta_{j,t} \cdot b_{j,t})), \tag{6.1}$$

where:

- $I_{j,t}$ represents the number of impressions generated for campaign C_j at time t.
- $b_{j,t}$ is the budget invested in campaign C_j at time t.
- $\alpha_{j,t}$ is the **scaling factor**, which determines the maximum number of impressions that can be generated. It acts as a multiplicative factor, scaling the output of the exponential function to align with a certain range of impressions. It depends on the campaign C_j and may vary over time t.
- $\beta_{j,t}$ is the **sensitivity factor**, which controls the responsiveness of impressions to the input action. A higher β value results in a steeper curve, meaning impressions grow rapidly with small increases in the action. Conversely, a smaller β results in a slower and more gradual growth of impressions. It depends on the campaign C_j and may vary over time t.

The impressions are then assigned to users in a random manner. For every impression a users see, their interest level increases depending on whether they have been exposed to an awareness or a conversion campaign. The interest increment is determined by the following functions:

- **Awareness Campaign:**

$$\Delta_{aw}(u_m) = \lambda_j \cdot (1 - u_m).$$

- **Conversion Campaign:**

$$\Delta_{cv}(u_m) = (1 - u_m) \cdot \lambda_j \cdot u_m,$$

where:

- $\Delta_{aw}(u_m)$ represents the increment in interest level for user m after exposure to an awareness campaign. It is bigger for lower u_m values.
- $\Delta_{cv}(u_m)$ represents the increment in interest level for user m after exposure to a conversion campaign. It is bigger for greater u_m values.
- λ_j a coefficient that describes the quality of the campaign C_j. It modulates the amount of interest increase for each type of campaign, with the functions ensuring that the interest u_m remains bounded between 0 and 1.

Finally, after all impressions have been allocated, conversions are generated based on the following condition:

$$\frac{u_m}{\mu} \geq \mathcal{R},$$

where μ is a constant, and $\mathcal{R}$ represents an array of random values drawn from a uniform distribution.

When a user converts, they are replaced by a new user. The algorithm then returns the total number of conversions for the day. The full algorithm implementation of the simulator is shown in Algorithm 6.1 below.

6.1.2. Simulator Fitting

The simulator fitting process involves estimating the parameters α, β for each campaign and day, and λ for each campaign in the dataset. Additionally, the common values of μ and δ apply across all campaigns. As explained in Chapter 7.2, the parameters α and β for each campaign depend on the specific day of the week (e.g., Monday, etc.) and remain consistent across all occurrences of the same day.

The input data for this fitting process consists of a series of campaign costs and their corresponding impressions. Each campaign fitting is process independently from the others. The parameters α and β in the campaign impression model are fitted using numerical optimization techniques to align the model's predictions with observed data. The model expresses impressions as a function of cost as expressed in Equation (6.1) To fit the model, a loss function $\mathcal{L}(\alpha, \beta)$ is introduced to measure the discrepancy between the actual impressions observed from the data and the impressions predicted by the model. The root mean square error (RMSE) serves this purpose, as it effectively captures the average

Algorithm 6.1 Simulated Advertising Campaign and Conversion Algorithm

Input: Budget for each platform $b_{j,t}$ for day $t \in \{1, \ldots, T\}$
Input: Number of campaigns N, scaling factors α_j for $j \in \{1, \ldots, N\}$, sensitivity factors β_j for $j \in \{1, \ldots, N\}$, quality coefficient λ_j for $j \in \{1, \ldots, N\}$, user array size M, forgetting factor δ
Output: Daily sum of conversions G
for each day $t \in \{1, \ldots, T\}$ **do**
  $G = 0$
  Update user interest: $u_m \leftarrow \delta \cdot u_m$, $\forall m \in \{1, \ldots, M\}$
  for each platform C_j, *where* $j \in \{1, \ldots, N\}$ **do**
    Generate impressions: $I_j = \alpha_{j,t} \cdot (1 - \exp(-\beta_{j,t} \cdot b_{j,t}))$
    for each impression $i \in \{1, \ldots, I_j\}$ **do**
      Randomly pick a user m from $\{1, \ldots, M\}$
      if User sees an awareness campaign **then**
        Increment interest: $u_m \leftarrow u_m + \lambda_j \cdot (1 - u_m)$
      else if User sees a conversion campaign **then**
        Increment interest: $u_m \leftarrow u_m + (1 - u_m) \cdot \lambda_j \cdot u_m$
      end if
    end for
  end for
  Generate conversions:
  for each user $m \in \{1, \ldots, M\}$ **do**
    if Conversion condition $\frac{u_m}{\mu} \geq \mathcal{R}[m]$ **then**
      User converts
      Reset $u_m \leftarrow 0$
      $G = G + 1$
    end if
  end for
  store G for day t
end for

deviation while penalizing larger errors more heavily, defining the loss function as:

$$\mathcal{L}(\alpha, \beta) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(I_{\text{actual},i} - I_{\text{pred},i}\right)^2},$$

where $I_{\text{actual},i}$ represents the real impressions observed in the dataset for instance i in the data, while $I_{\text{pred},i}$ corresponds to the expected number of impressions for instance i predicted by the model following Equation (6.1). All instances are derived from data related to a single campaign, with the number of instances n varying based on the selected seasonality of the dataset. For example, it may depend on the data available for a specific day of the week, week of the month, or entire month, depending on the choosen seasonality

for the dataset as explained in Chapter 7.2. This formulation ensures that the loss function measures the discrepancy between the real impressions data and the model's predicted impressions. The optimization process seeks to find the values of α and β that minimize this loss. Constraints are applied to ensure the parameters remain positive, as negative values for α or β would be nonsensical in this context.

The parameters λ, μ, and δ are fitted simultaneously to align the conversions of the simulator environment with observed real-world conversion data. Using these parameters, the environment simulates user interactions and conversions over a series of days. For each simulation, multiple local iterations are performed to account for randomness, and the results are averaged to produce stable estimates. To evaluate the fit, the daily root mean square error (RMSE) between the simulated and actual conversions from real data is computed. The optimization employs the CMA-ES algorithm, which iteratively adjusts the parameters to minimize the RMSE. This method is particularly effective for non-linear, non-convex optimization problems [15]. The problem admits an infinite number of solutions. Therefore, while the bounds for all parameters are set between [0,1], the values for μ and δ are further rescaled according to the specific dataset, within certain ranges. Among all possible solutions, the one with λ values that best match the real data conversion rates is preferred. Further details about this fitting process will be provided in Chapter 7.

6.2. Proposed MDP

The RL algorithms require data to be structured according to the MDP framework, making it essential to define the MDP formulation for our problem. Let $X \subset \mathbb{R}^p$ represent the set of possible states, where each state $x \in X$ is a vector of length $p = N + 1$, with N being the number of campaigns. Let $A \subset \mathbb{R}^q$ represent the set of possible actions, where each action $a \in A$ is a vector of length $q = N$, corresponding to the budget allocated to each campaign.

The state at time step t, denoted $x_t \in X$, is computed as:

$$x_t = (I_{j,t,\tau} \ \forall j \in \{1, \ldots, N\},\ G_{t,\tau},\ d_t)\,,$$

where:

$$I_{j,t,\tau} = \sum_{s=t-\tau}^{t} I_{j,s},$$

$$G_{t,\tau} = \sum_{s=t-\tau}^{t} G_s,$$

where $I_{j,s}$ represent the impressions generated by campaign C_j and day s, G_s are the number of conversions obtained at day s and τ is the total number of past steps over which the impressions and conversions are assumed to have an impact on the current state.

d_t is an encoded representation of time t. d_t and τ may vary on the specific dataset and implementation choices which are discussed in Chapter 7.

The action vector $a_t \in A$ at time step t is defined as:

$$a_t = [b_{t,1}, b_{t,2}, \ldots, b_{t,N}],$$

where each $b_{t,j}$ represents the budget allocated to campaign C_j at day t. The reward is the total daily conversions obtained on each day, where one step corresponds to one day. The state transition matrix P is not explicitly needed for RL algorithms we used and the discount factor and time horizon are tuned according to the specifics of the dataset as discussed in Chapter 7.

6.3. Design of the Solution and Optimization

The implementation of FNAC follows the algorithm described in [24] and presented in Chapter 3.2.3. This section provides a general overview of the algorithm's design and implementation. For specific parameter values, refer to Chapter 7.

6.3.1. Actor and Critic Component

The Actor and Critic components were implemented as neural networks, both utilizing linear layers and Leaky ReLU activation functions. Specifically, the Critic network takes the state, as defined in Chapter 6.2, as input and passes it through four linear layers, each followed by a Leaky ReLU activation function. The Actor network also takes the state as input and processes it through three linear layers, followed by Leaky ReLU activations. The output is passed through a Softplus activation function to ensure positive values, which serve as the concentration parameters of a Dirichlet distribution used for sampling the action.

6.3.2. Detailed Algorithm

The training process for the model is structured as follows. A total number of iterations, L, and a predefined number of critic training epochs, L_c, are set. In each iteration, a trajectory is randomly sampled from the available dataset. Importance sampling weights are then calculated as the ratio of probabilities assigned to the actions in the trajectory by the current policy and the behavioral policy used during data collection.

For each critic training epoch (L_c), the critic network is optimized by minimizing the loss function specified in Equation (3.7). After the critic is updated, h steps corresponding within the trajectory are selected. For these steps, the compatible features are calculated by evaluating the Jacobian of the log-probabilities of the corresponding actions with respect to the actor network parameters, as described in Equation (3.9). Using the critic, the advantage for the selected weekis estimated and subsequently utilized in a ridge regression procedure to solve the regression problem outlined in Equation (3.8).

The optimal regression weights, w^*, obtained from this process are then used to update the parameters of the actor network according to the update rule provided in Equation (3.6). The pseudocode of the solution is provided in Algorithm 6.2.

Algorithm 6.2 FNAC Training

Input: Dataset D, Total iterations L, Critic epochs L_c, Learning rate α for the actor
for $k = 1$ to L **do**
  Sample a trajectory τ from D
  Compute importance sampling weights: $w_{\text{IS}} = \frac{\pi_\theta(a_t|x_t)}{\pi_0(a_t|x_t)}$
  for $e = 1$ to L_c **do**
    Update critic by minimizing loss: $\min_v \sum_t w_{\text{IS}} \left(r_t + \gamma \hat{V}_v(x_{t+1}) - \hat{V}_v(x_t)\right)^2$
  end for
  Select a subset of steps h in τ
  Compute compatible features: $\phi(x_t, a_t) = \frac{\partial \log \pi_\theta(a_t|x_t)}{\partial \theta}$ for each step $t \in h$
  Estimate advantage: $\hat{A}_\theta(x_t, a_t) = r_t + \gamma \hat{V}_\theta(x_{t+1}) - \hat{V}_\theta(s_t)$ for each step $t \in h$
  Solve regression to compute w^*: $w^* = \arg\min_w \sum_{t \in h} \frac{1}{\hat{\mu}(x_t)} \left(\hat{A}_\theta(x_t, a_t) - \phi^\top(x_t, a_t) w\right)^2$
  Update actor parameters: $\theta_{k+1} = \theta_k + \alpha w^*$
end for

7 | Experimental Setup and Results

This chapter provides a detailed explanation of the dataset used, the implementation details, and the parameter choices made in the solution presented in Chapter 6. Additionally, it describes the baseline algorithms employed for comparison and presents their experimental results. We begin in Chapter 7.1 with a discussion of the dataset, describing its characteristics and relevance to the problem at hand. Following this, in Chapter 7.2, we outline the data fitting procedure, explaining how the dataset was processed and the specific parameter values chosen to ensure an accurate and meaningful representation. Chapter 7.3 then delves into the process of data collection, detailing how the simulator was used to generate the dataset employed in our experiments. Building upon this, in Chapter 7.4, we provide a comprehensive explanation of the FNAC implementation, discussing key configuration choices and parameter settings that influenced the learning process. Next, in Chapter 7.5, we introduce the baseline algorithms selected for comparison, explaining their relevance and how they serve as benchmarks for evaluating the proposed approach. Finally, in Chapter 7.6, we present the experimental results, analyzing the performance of FNAC in relation to the baseline methods and drawing meaningful conclusions from the comparison.

7.1. Experimental Dataset Description

The dataset used for the experimental results focuses on leggings and sports articles, encompassing 58 campaigns conducted over a one-year period (September 2023 to September 2024). From these, we selected only the Italian campaigns running within a shorter timeframe, specifically from July 1, 2024, to September 23, 2024, resulting in a subset of 10 campaigns. These include 5 awareness campaigns and 5 conversion campaigns.

The campaigns are distributed across different platforms: Amazon (3 campaigns), Facebook (6 campaigns), and Google (1 campaign). For each campaign, data on costs, impressions, conversions, and revenue are provided. Notably, the campaigns were active during

varying time intervals, as illustrated in Figure 7.1.

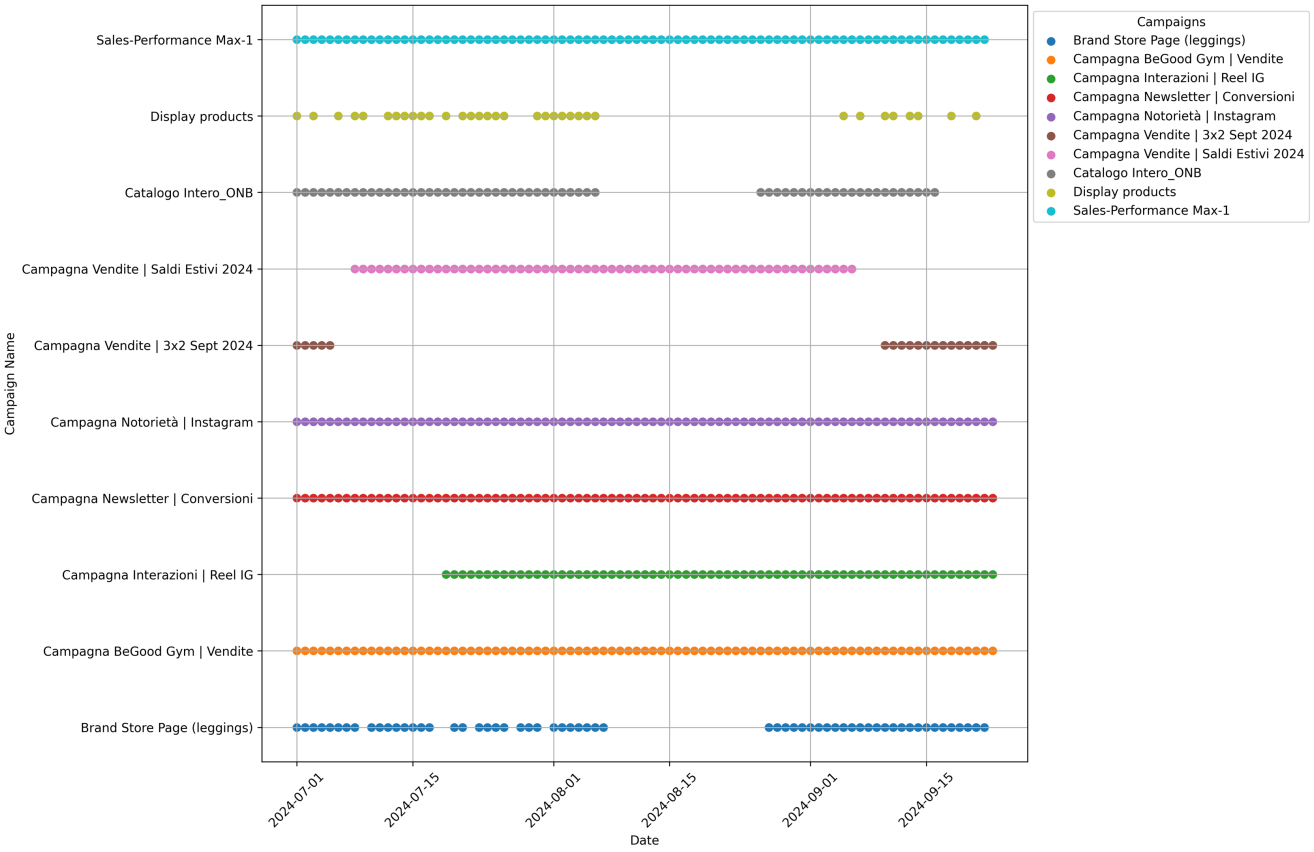


Figure 7.1: Timeline of campaigns activity.

Additionally, we conducted a seasonality analysis of the dataset to examine whether the behavior of each campaign varied significantly over time. Given that the available data covered only a short period, we focused on analyzing variations across different days of the week (e.g., Monday, Tuesday, etc.). The results indicated that while the dataset exhibits a noticeable degree of seasonality, it is not particularly strong. However, the observed patterns remain interesting and worthy of further study. Figure 7.2 presents the seasonality analysis performed on campaign 9 from the dataset as example.

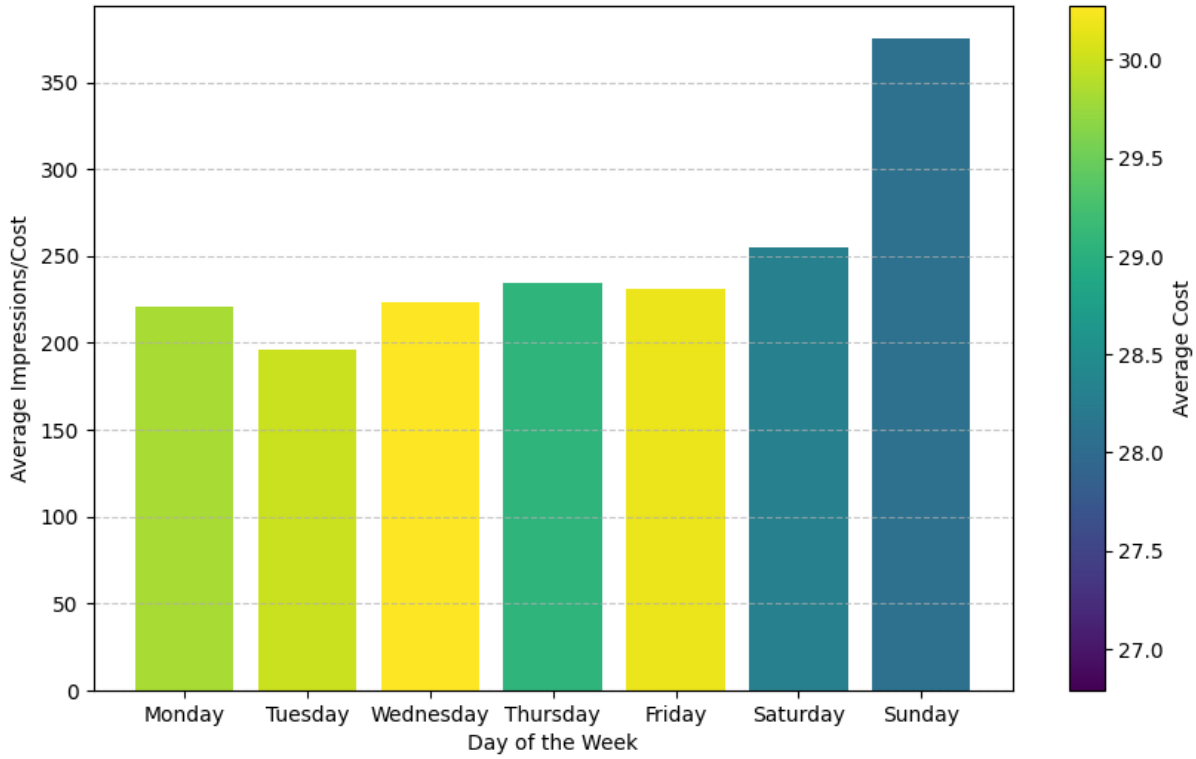


Figure 7.2: The seasonality analysis of a campaign 9 performed for each day of the week, where the columns represent the average impressions-to-cost ratio across all data points. The colors of the columns are determined by the cost.

7.2. Data Fitting

A total of $M = 20000$ users was chosen for the simulator as a balance between practical constraints and performance considerations. This number was selected because most campaigns did not exceed 20000 impressions per day, making it a reasonable choice for maintaining both realism and computational efficiency. The fitting procedure follows the approach outlined in Chapter 6.1.2. For the fitting of the parameters α and β, the data were grouped by the day of the week, and the fitting was performed separately for each group. As a result, each campaign has its own α and β values for each specific day of the week. For the fitting of the parameters λ, μ, and δ, the real data conversions were pre-processed to account for the different tracking methods across platforms. Specifically, the revenue of each campaign was converted into conversions by dividing the revenue by the ratio $\frac{\text{Total Revenue}}{\text{Total Conversions}}$.

Regarding the fitting of μ and δ, these parameters were rescaled to specific ranges: $\delta \in [0.95, 1]$ and $\mu \in [3000, 5000]$. This was done to reduce the issue of infinite solutions that

could arise from arbitrary choices of μ and δ. Different value ranges for these parameters introduce distinct properties to the model. A larger μ corresponds to a higher level of user interest at which conversions occur, thereby giving more weight to conversion-focused campaigns. The chosen range for μ was selected to keep the mean user interest over time within the range of [0.40, 0.50].

The value of δ is also important, as a very high δ can sustain high levels of interest for a prolonged period, reducing the relevance of campaigns aimed at maintaining user engagement. For optimization, the CMA-ES algorithm, as described in Chapter 6.1.2, was used. Among the different runs of the algorithm, the solution with a λ coefficient closer to the real data conversion rate was preferred for conversion campaigns. The daily fitting plot, comparing real conversions with simulated conversions, is shown in Figure 7.3. While the solution does not perfectly match the real data due to inherent differences between the real world and the model, the aggregated sums of conversions over a longer period remain similar—158 simulated conversions compared to 166 real conversions. The discrepancy is mainly attributed to outliers in the original dataset.

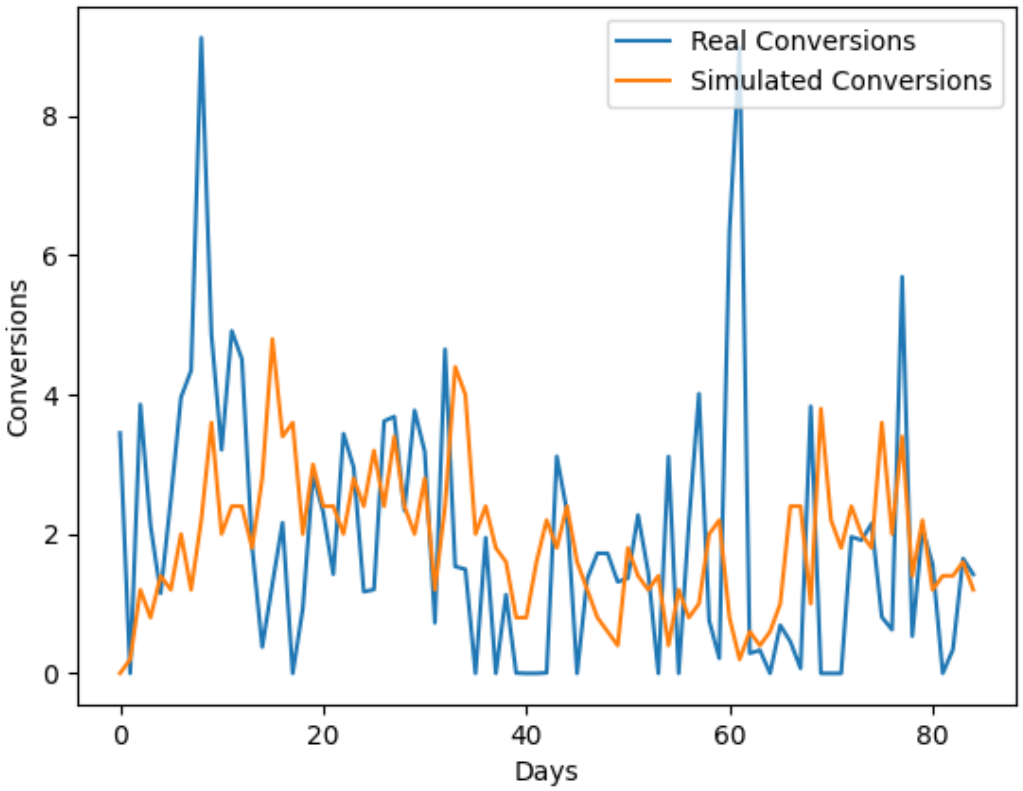


Figure 7.3: Real conversions from data rescaled with revenue in blue, simulated conversions in red.

7.3. Data Generation

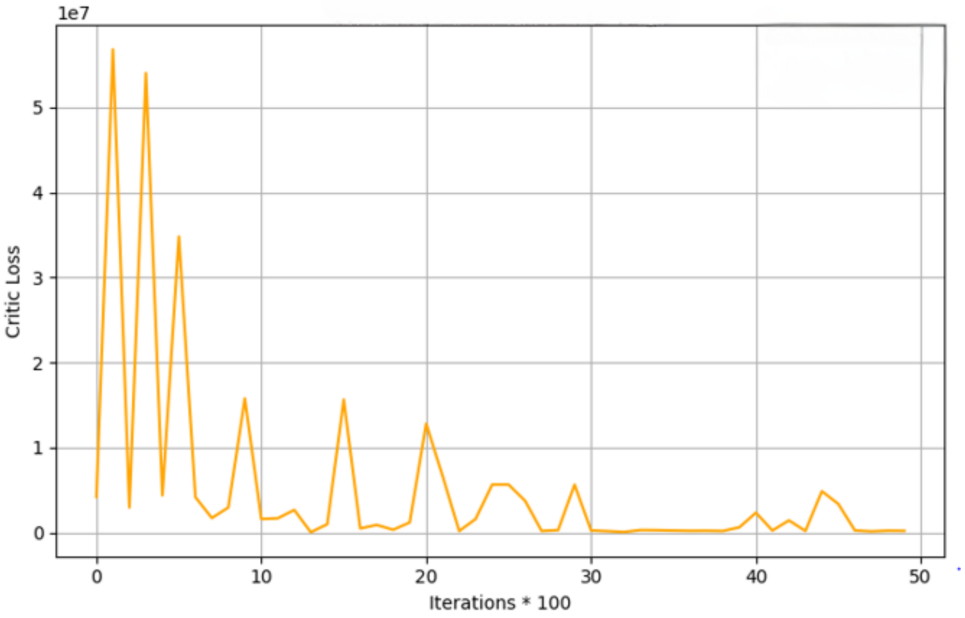
FNAC algorithms require a significant amount of data to be trained effectively. In this work, the simulator described in Chapter 6.1 has been used to generate data following a specific process. The generated data adhere to the structure outlined by the corresponding MDP, as described in Chapter 6.2. Specifically, a value of $\tau = 50$ was chosen, and d_t represents the one-hot encoding for the corresponding day of the week. A total of 4000 trajectories were generated, with each trajectory consisting of 250 steps. Before collecting each trajectory, the simulator is reset, which sets the interest of all users to 0. During the first 250 steps, random actions are performed, ensuring that the environment is explored. After this initial phase, the actual trajectory is collected. For each 50 steps, the same action is used, after which the action is changed. The actions are generated following a Dirichlet distribution with variable concentration parameters taken randomly. These actions are then rescaled using the daily budget $\bar{b}$, which is computed from real data. The

daily budget is obtained by summing the cost for all campaigns and dividing it by the total number of days. For each step, the state and new state are computed based on the past impressions, conversions and on the day. Each step thus contains state,action,new state, reward obtained, and concentration parameters of the action which will be used during the FNAC training. The best trajectories are then augmented by selecting the top 50 actions and generating new trajectories based on similar actions. This process results in the addition of 500 new trajectories, bringing the total number of trajectories in the dataset to 4500.

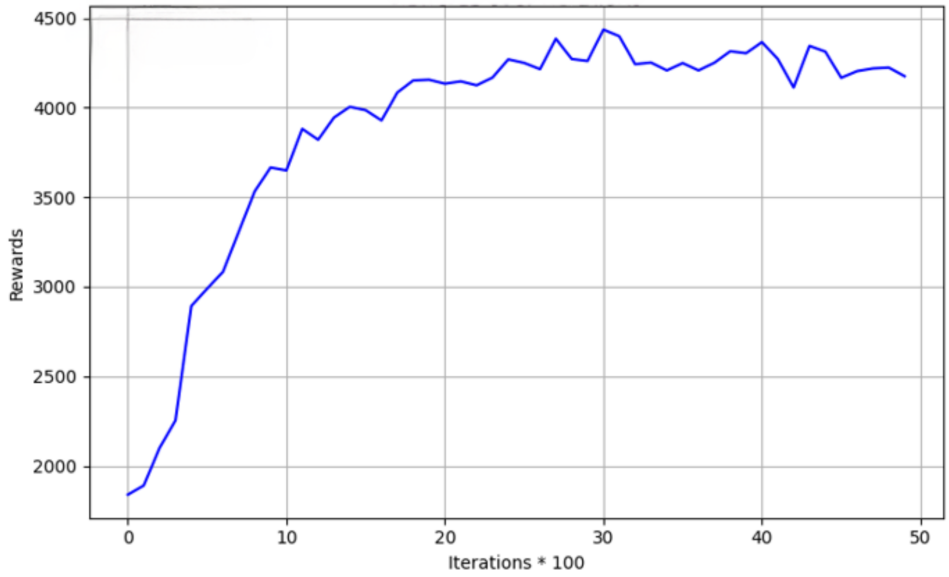
7.4. FNAC Configuration and Implementation Choices

The implementation design adheres to the approach outlined in Chapter 6.3. For the critic network, the architecture consists of two linear layers with 128 units each, followed by a third layer with 64 units, and a final output layer with a single unit. The actor network, on the other hand, has three linear layers: the first with 64 units, the second with 32 units, and the final layer with 10 units, corresponding to the dimensionality of the action space. The learning rate for the critic is set to 0.001, while the actor's learning rate is set to 0.01.

Regarding training, the total number of iterations is $L = 5000$, with the critic being trained for $L_c = 10$ epochs per iteration. Each trajectory has a length of $\tau = 250$ steps, and the actor is updated using $h = 7$ adjacent steps representing a full week. The discount factor adopted is $\gamma = 0.997$. Figure 7.3 illustrates the critic loss and the rewards obtained from the environment over the course of the iterations. Notably, already by the 2000th iteration, the algorithm begins to yield promising results.



(a) Critic loss function over training iterations (the number of iterations is multiplied by 100).



(b) Reward function over training iterations (the number of iterations is multiplied by 100).

Figure 7.4: Critic loss function and reward obtained over training iterations

7.5. Baselines

In addition to the FNAC algorithm described earlier, three other algorithms—CMA-ES, SAC, and PPO were used for the experiments.

PPO [28] is a state-of-the-art RL algorithm designed to optimize the policy of an agent through continuous updates. PPO is an on-policy algorithm that strikes a balance between performance and ease of implementation. It operates by minimizing the clipped objective function, which prevents large, destabilizing policy updates during training. This is achieved by limiting how much the new policy can deviate from the old one, ensuring stable learning. PPO has become widely popular in RL due to its simplicity and effectiveness, particularly in high-dimensional continuous action spaces.

SAC [13] is an off-policy RL algorithm designed for continuous action spaces, combining elements of both value-based and policy-based RL methods. SAC aims to maximize the entropy of the policy along with the expected reward, encouraging exploration and leading to more robust policies. By incorporating entropy regularization, SAC avoids premature convergence to suboptimal policies. It also utilizes two Q-functions (soft Q-values) and a target value function to stabilize training. SAC has shown strong performance in challenging continuous control tasks and is known for its sample efficiency.

CMA-ES [15] is a traditional optimization method rooted in evolutionary strategies, often used for solving non-convex, high-dimensional optimization problems. Unlike RL algorithms, CMA-ES does not involve interaction with an environment but rather evolves a population of candidate solutions through selection, mutation, and recombination. It maintains a covariance matrix to adapt the distribution of candidate solutions based on past successes, making it particularly effective for complex, black-box optimization problems.

7.6. Results and Comparisons

The proposed baselines were trained using different approaches. PPO and SAC were trained online with the simulator described in Chapter 6.1 as the environment, running trajectories of 400 days for a total of 1000000 timesteps. FNAC, on the other hand, was trained offline using the dataset detailed in Chapter 7.3, following the procedure outlined in Chapter 6.3. In contrast, CMA-ES was applied to an optimization function that takes a fixed budget as input, applies it to the simulator for 400 days, and returns the total number of conversions. The number of iterations evaluations performed by CMA-ES was 200 (2000 function calls). A random policy that generates actions using a Dirichlet

distribution with randomly sampled concentration parameters was also included in the experiments.

During testing, the actions taken by different baselines were scaled using the daily budget $\bar{b}$, which is derived from real data as explained in Chapter 7.3. These actions were applied to the simulator as environment over a 400-day period, with 10 evaluation runs. The simulator was reset between each run. Table 7.1 shows that FNAC is the best performing algorithm, followed by SAC. CMA-ES ranks third, maintaining a constant solution over time, which still outperforms the variability observed in PPO. Figures 7.5, 7.6, 7.7 and 7.8 illustrate the learned actions over 400 steps where the first 5 budgets are related to conversions campaigns while the last 5 to awareness campaigns. Except for CMA-ES, the learned actions exhibit a 7-day periodicity, which aligns with the expected seasonality of the problem. The first 50 steps display greater variability, as the state representation includes the sum of impressions and conversions over 50 steps. After this initial phase, only the day of the week changes, leading to a more stable action pattern.

Technique Name	**Mean Conversions**	**Std Conversions**
FNAC	4352.9	51.2
CMA-ES	4302.4	59.8
PPO	4020.9	65.1
SAC	4315.9	47.7
Random policy	1831.7	43.8

Table 7.1: Mean and Standard Deviation of Conversions Generated by Different Algorithms over 10 evaluations on the environment

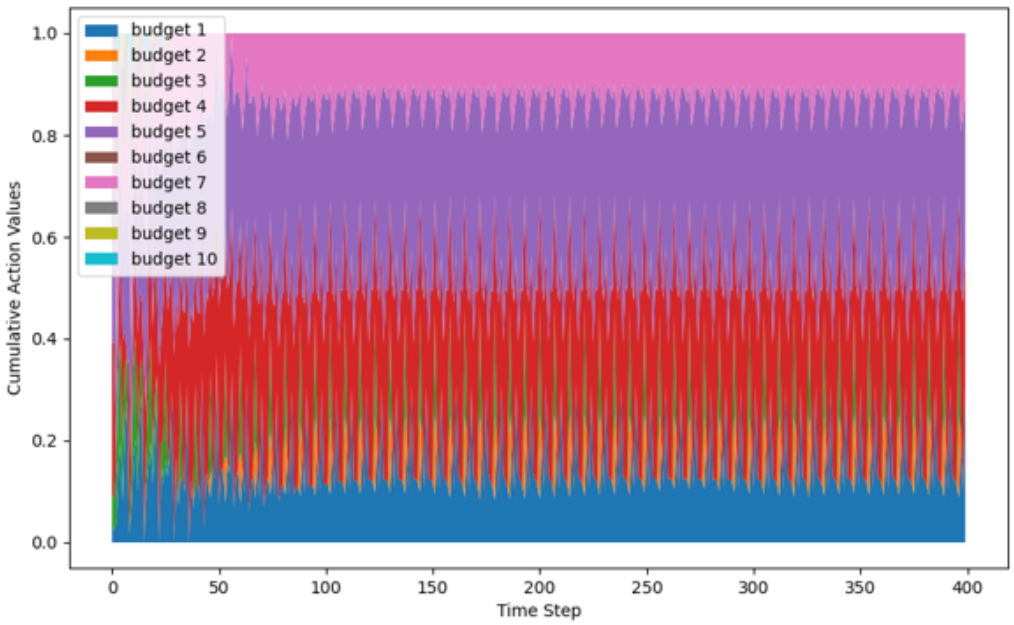


Figure 7.5: PPO.

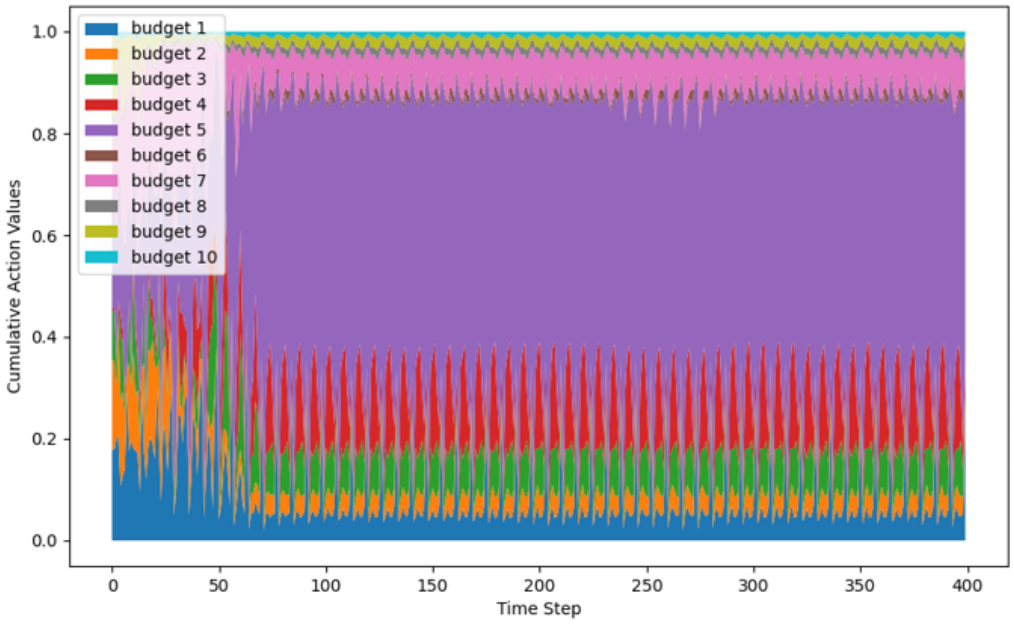


Figure 7.6: SAC.

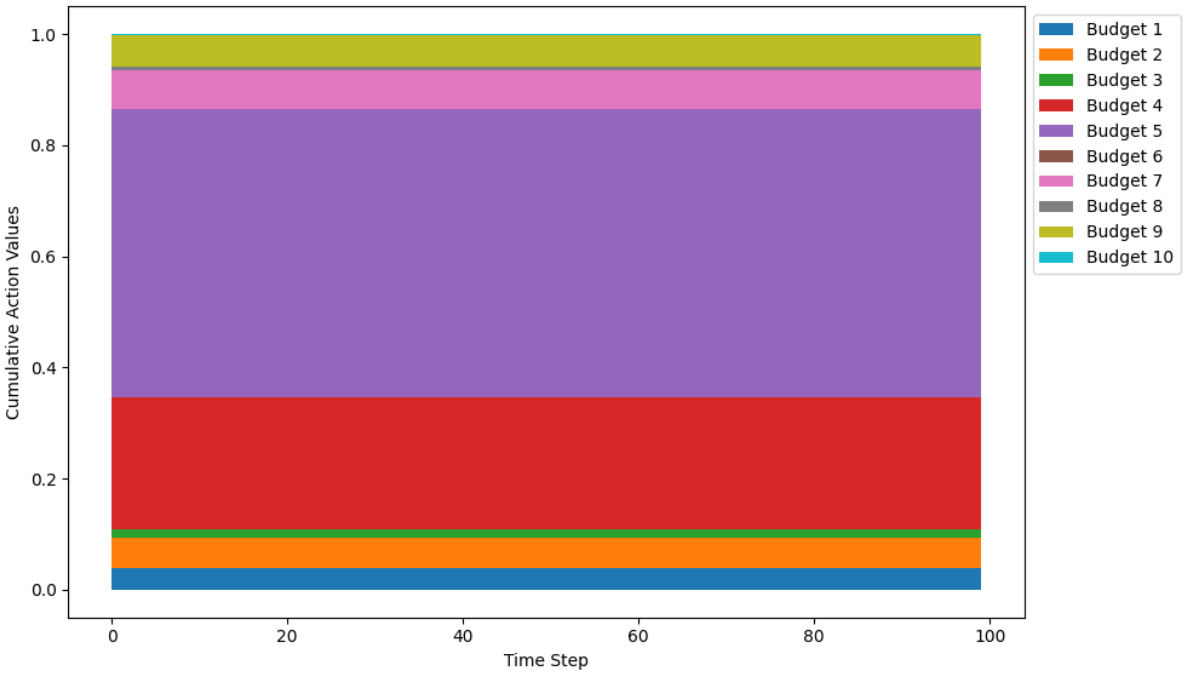


Figure 7.7: CMA-ES.

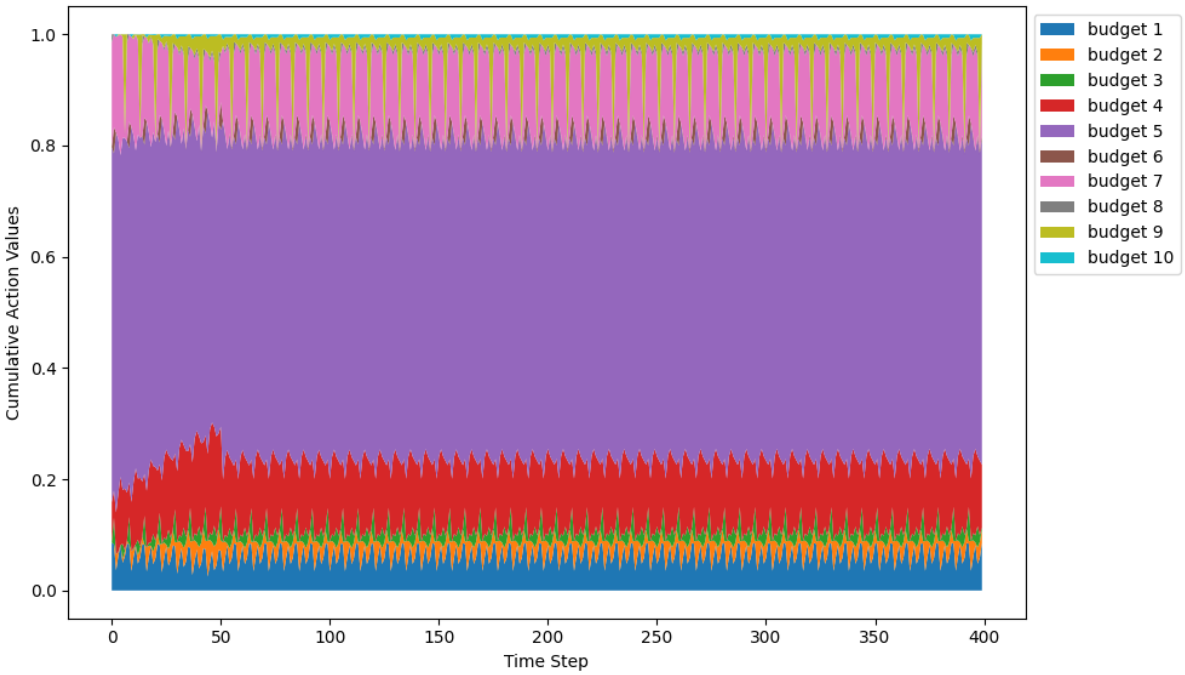


Figure 7.8: FNAC.

Focusing specifically on the FNAC solution, we analyze the learned behavior over a one-week period after the system stabilizes following 50 steps. This is illustrated in Figure 7.7,

which highlights steps 70 to 76 (Monday to Sunday). Some variability is observed, particularly in campaign 9 compared to campaign 6 during the weekend, with both being awareness campaigns. This is consistent with the seasonality analysis done for campaign 9 as shown in Figure 7.2 where the campaign performs better over the weekend. However, overall, the daily behavior remains relatively consistent, aligning with expectations based on the seasonality analysis of the dataset described in Chapter 7.1. In summary, the results indicate that FNAC is the most effective algorithm, achieving the best performance in budget allocation compared to the other methods considered. The learned policy reflects the seasonality of the problem, exhibiting a clear weekly periodicity and greater stability after the first 50 steps. Additionally, the observed variations between different campaigns align with the seasonality analysis, confirming FNAC's ability to adapt dynamically over time.

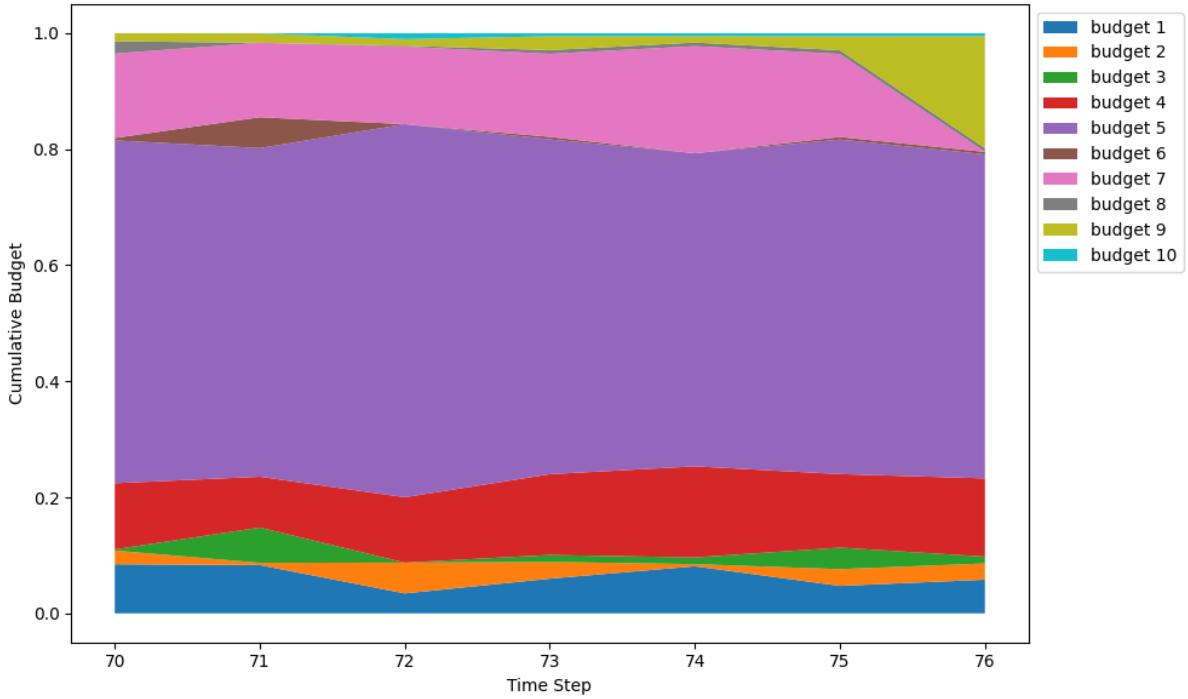


Figure 7.9: FNAC action detailed period.

8 | Conclusions and Future Developments

The rapid evolution of digital advertising demands advanced optimization techniques to efficiently allocate budgets across multiple channels. Traditional budget allocation methods heavily rely on human expertise, which often fails to account for the highly dynamic nature of advertising environments. These environments are inherently stochastic, exhibit time-delayed revenue effects, and follow seasonal trends, making budget optimization increasingly complex. To address this challenge, we focus on RL techniques for optimizing budget allocation across multiple advertising campaigns. The first step in our approach was developing a simulator capable of accurately modeling real-world advertising dynamics, including user interest in products and conversions based on those interests. To ensure realism, the simulator was fitted to a real-world dataset, enabling the generation of a synthetic dataset that closely mirrors real data, an essential step for training RL algorithms, which require large amounts of data. Next, we formulated a MDP as the theoretical foundation for applying RL algorithms to the advertising environment. Using the simulator, we generated a synthetic dataset that adheres to the MDP structure. This dataset was then used to train an offline RL algorithm, FNAC, whose performance was compared against online RL algorithms such as PPO and SAC, both trained using the same simulator as an interactive environment. Additionally, as a benchmark from a more traditional optimization perspective, we applied CMA-ES to an objective function that allocates a fixed budget and returns the resulting conversions through the simulator. Experimental results demonstrated that the proposed RL algorithms achieved promising performance, consistently outperforming CMA-ES. In particular, FNAC effectively captured the dataset's seasonality patterns, dynamically adapting over time to achieve a higher number of conversions compared to a static budget allocation strategy. In summary, this study highlights the effectiveness of RL techniques in optimizing digital advertising budgets within dynamic and uncertain environments. The developed simulator successfully captured the complexities of advertising dynamics, generating realistic data essential for training RL algorithms. These algorithms effectively learned the environment's pat-

terns, enabling the development of adaptive, time-sensitive budget allocation policies that outperformed traditional static methods.

8.1. Future Developments

While the proposed methodology offers clear advantages, further refinements and extensions could enhance its applicability and robustness in real-world scenarios. Future work could explore alternative models for simulating the advertising environment, incorporating varying levels of complexity and realism based on the specific problem and available insights into user behavior. For example, different increment functions for awareness and conversion campaigns could be investigated, alternative conversion generation mechanisms could be considered, or a broader range of campaign types spanning the entire interest funnel could be introduced. Additionally, various approaches for fitting the simulator parameters could be explored, including different levels of temporal aggregation to capture diverse seasonal patterns. Another promising direction is leveraging datasets with a large number of campaigns and developing meaningful aggregation techniques to reduce the total number of campaigns analyzed while preserving essential parameter relationships. This would ensure the model remains both scalable and interpretable.

Bibliography

[1] Ahrefs. The marketing funnel: What it is, how it works, & how to optimize. URL `https://ahrefs.com/blog/marketing-funnels`.

[2] F. AlMahamid and K. Grolinger. Reinforcement learning algorithms: An overview and classification. In *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*.

[3] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 2020.

[4] S.-i. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10 (2):251–276, 1998.

[5] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. A brief survey of deep reinforcement learning. 2017.

[6] Backlinko. Social media users. `https://backlinko.com/social-media-users`, 2025.

[7] A. Badanidiyuru, V. Varadaraja, Z. Feng, T. Li, and H. Xu. Incrementality bidding via reinforcement learning under mixed and delayed feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[8] H. Chen, Y.-J. Chen, S. Park, and D. Shin. Multichannel advertising: Budget allocation in the presence of spillover and carryover effects. *SSRN Electronic Journal*, 2023.

[9] P. Deshwal. Online advertising and its impact on consumer behavior. *International Journal of Applied Research*, 2(2):200–204, 2016.

[10] A. Digital. Cross-channel advertising integration with other digital platforms. 2024. URL `https://www.analyticodigital.com/blog/cross-channel-advertising-integration-with-other-digital-platforms`.

[11] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the general-

ized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*.

[12] S. Geyik, A. Saxena, and A. Dasdan. Multi-touch attribution based budget allocation in online advertising. In *Proceedings of the 2014 IEEE International Conference on Big Data (Big Data)*, pages 8–13, 2014.

[13] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[14] B. Han and J. Gabor. Contextual bandits for advertising budget allocation. In *Proceedings of the ADKDD*, volume 17, 2020.

[15] N. Hansen, S. D. Müller, and P. Koumoutsos. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2019.

[16] Holid.io. Why digital advertising is important. 2025. URL `https://holid.io/why-digital-advertising-is-important/`.

[17] S. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, pages 1531–1538, 2001.

[18] M. Li, J. Zhang, R. Alizadehsani, and P. Pławiak. A multi-channel advertising budget allocation using reinforcement learning and an improved differential evolution algorithm. *IEEE Access*, 12:100559–100570, 2024.

[19] G. Liao, Z. Wang, X. Wu, X. Shi, C. Zhang, Y. Wang, X. Wang, and D. Wang. Title of the paper (replace with actual title). In *Proceedings of the ACM Web Conference 2022*. ACM, 2022.

[20] M. Liu, L. Jiaxing, Z. Hu, J. Liu, and X. Nie. A dynamic bidding strategy based on model-free reinforcement learning in display advertising. *IEEE Access*, 8:213587–213601, 2020.

[21] M. Liu, J. Liu, Z. Hu, Y. Ge, and X. Nie. Bid optimization using maximum entropy reinforcement learning. *Neurocomputing*, 482:15–22, 2022.

[22] J. Lyu, X. Ma, X. Li, and Z. Lu. Mildly conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 1711–1724. Curran Associates, Inc., 2022.

[23] D. Media. What are the key barriers to achieving a high roi in digital marketing campaigns?, 2023. URL `https://demandifymedia.com/`

what-are-the-key-barriers-to-achieving-a-high-roi-in-digital-marketing-campaig
?utm_source=chatgpt.com.

[24] F. S. Melo and M. Lopes. Fitted natural actor-critic: A new algorithm for continuous state-action MDPs. In *ECML PKDD 2008*, 2008.

[25] A. Nuara, F. Trovò, N. Gatti, and M. Restelli. A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 2379–2386, 2018.

[26] J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In *Proc. European Conf. Machine Learning*, pages 280–291, 2005.

[27] M. Sanchez. Why you can't trust google analytics, 2023. URL https://getrecast.com/google-attribution/.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[29] StackAdapt. Advantages of multi-channel advertising. URL https://www.stackadapt.com/resources/blog/advantages-of-multi-channel-advertising.

[30] Statista. Number of internet users worldwide from 2005 to 2022, 2022. URL https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/.

[31] Statista. Worldwide market share of search engines on all devices as of january 2025. https://www.statista.com/statistics/1381664/worldwide-all-devices-market-share-of-search-engines, 2025.

[32] R. S. Sutton, D. A. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.

[33] Tridigiam. Digital advertising vs. traditional advertising: What's the difference? 2025. URL https://tridigiam.com/digital-advertising-vs-traditional-advertising-whats-the-difference/.

[34] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*.

[35] H. Wang, B. Tang, C. Liu, S. Mao, J. Zhou, Z. Dai, Y. Sun, Q. Xie, X. Wang, and

D. Wang. Hibid: A cross-channel constrained bidding system with budget allocation by hierarchical offline deep reinforcement learning. *IEEE Transactions on Computers*, 2023.

[36] X. wang, P. Li, and A. Hawbani. An efficient budget allocation algorithm for multi-channel advertising. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 886–891, 2018.

[37] WordStream. The ultimate guide to digital advertising, 2023. URL `https://www.wordstream.com/blog/ws/2023/02/24/digital-advertising`.

[38] Y. Yang, D. Zeng, Y. Yang, and J. Zhang. Optimal budget allocation across search advertising markets. *INFORMS Journal on Computing*, 27(2):285–300, April 2015.

[39] Y. Yuan and et al. A survey on real-time bidding advertising. In *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*.

[40] J. Zhao, G. Qiu, Z. Guan, W. Zhao, and X. He. Reinforcement learning for bid optimization in online advertising campaigns. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1317–1326. ACM, 2018.

[41] K. Zhao, J. Hua, L. Yan, Q. Zhang, H. Xu, and C. Yang. A unified framework for marketing budget allocation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, page 11. ACM, 2019.

List of Figures

List of Tables