Dynamic Knapsack Optimization Towards Efficient Multi-Channel Sequential Advertising

Xiaotian Hao [*1] **Zhaoqing Peng** [*2] **Yi Ma** [*1] **Guan Wang** [3] **Junqi Jin** [2] **Jianye Hao** [1] **Shan Chen** [2] **Rongquan Bai** [2] **Mingzhou Xie** [2] **Miao Xu** [2] **Zhenzhe Zheng** [4] **Chuan Yu** [2] **Han Li** [2] **Jian Xu** [2] **Kun Gai** [2]

Abstract

In E-commerce, advertising is essential for merchants to reach their target users. The typical objective is to maximize the advertiser's cumulative revenue over a period of time under a budget constraint. In real applications, an advertisement (ad) usually needs to be exposed to the same user multiple times until the user finally contributes revenue (e.g., places an order). However, existing advertising systems mainly focus on the immediate revenue with single ad exposures, ignoring the contribution of each exposure to the final conversion, thus usually falls into suboptimal solutions. In this paper, we formulate the sequential advertising strategy optimization as a dynamic knapsack problem. We propose a theoretically guaranteed bilevel optimization framework, which significantly reduces the solution space of the original optimization space while ensuring the solution quality. To improve the exploration efficiency of reinforcement learning, we also devise an effective action space reduction approach. Extensive offline and online experiments show the superior performance of our approaches over state-of-the-art baselines in terms of cumulative revenue.

1. Introduction

In E-commerce, online advertising plays an essential role for merchants to reach their target users, in which Real-time Bidding (RTB) (Zhang et al., 2014; 2016; Zhu et al., 2017) is an important mechanism. In RTB, each advertiser is allowed to bid for every individual ad impression opportunity. Within a period of time, there are a number of impression opportunities (user requests) arriving sequentially. For each impression, each advertiser offers a bid based on the impression **value** (e.g., revenue) and competes with other bidders in real-time. The advertiser with the highest bid wins the auction and thus display ad and enjoys the impression value. Displaying an ad also associates with a **cost**: in Generalized Second-Price (GSP) Auction (Edelman et al., 2007), the winner is charged for fees according to the second highest bid. The typical advertising objective for an advertiser is to maximize its cumulative revenue of winning impressions over a time period under a fixed budget constraint.

In a digital age, to drive conversion, advertisers can reach and influence users across various channels such as display ad, social ad, paid search ad (Ren et al., 2018). As illustrated in Figure 1, the user's decision to convert (purchase a product) is usually driven by multiple interactions with ads. Each ad exposure would influence the user's preferences and interests, and therefore contributes to the final conversion. However, existing advertising systems (Yuan et al., 2013; Zhang et al., 2014; Ren et al., 2017; Zhu et al., 2017; Jin et al., 2018; Ren et al., 2019) mainly focus on maximizing the single-step revenue, while ignoring the contribution of previous exposure to the final conversion, and thus usually falls into suboptimal solutions. The reason is that simply optimizing the total immediate revenue cannot guarantee the maximazation of the long-term cumulative revenue. Besides, there exist some works (Boutilier & Lu, 2016; Du et al., 2017; Cai et al., 2017; Wu et al., 2018) which optimize the overall revenue under an extra-long (billions) request sequence using a single Constrained Markov Decision Process (CMDP) (Altman, 1999). However, the optimization of these methods above is myopic as they ignore the mental evolution of each user and the long-term advertising effects. The learning is particularly inefficient as well.

Apart from the myopic approaches, there exists some literatures considering the long-term effect of each ad exposure. Multi-touch attribution (MTA) (Ji & Wang, 2017; Ren et al., 2018; Du et al., 2019) study the credits assignment to the previous ad displays before conversion. However, these

[*]Equal contribution [1]College of Intelligence and Computing, Tianjin University, Tianjin, China [2]Alimama, Alibaba Group, Beijing, China [3]Department of Automation, Tsinghua University, Beijing, China [4]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. Correspondence to: Junqi Jin <junqi.jjq@alibaba-inc.com>, Jianye Hao <jianye.hao@tju.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

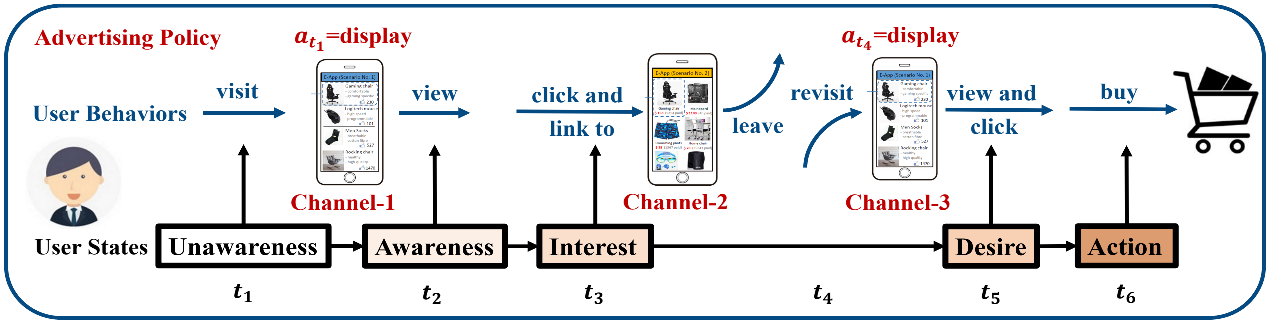

Figure 1. An illustration of the sequential multiple interactions (across different channels) between a user and an ad. Each ad exposure has long-term influence on the user's final purchase decision.

methods only attend to figure out the contribution of each ad exposure, while not providing methods to optimize the strategies. Besides, since all media channels could affect users' conversions, Li et al. (2018); Nuara et al. (2019) propose multi-channel budget allocation algorithms to help advertisers understand how particular channels contribute to user conversions. They optimize the budget allocation among all channels accordingly to maximize the overall revenue. However, the granularity of their optimizations is too coarse. They only optimize the budget allocation in the channel level and do not specifically optimize the advertising sequence for each user, which could lead to suboptimal overall performance.

Considering the shortcomings of existing works, we aim at optimizing the budget allocation of an advertiser among all users such that the cumulative revenue of the advertiser could be maximized, by explicitly taking into consideration the long-term influence of ad exposures to individual users. This problem consists of two levels of coupled optimization: bidding strategy learning for each user and budget allocation among users, which we termed as Dynamic Knapsack Problem. Different from traditional Knapsack problem, a number of challenges arise: 1) Given the estimated long-term value and cost for each user, the optimization space of the budget allocation grows exponentially in the number of users. Besides, since different advertising policies for each user will lead to different long-term values and costs, the overall optimization space is extremely large. 2) The long-term cumulative value and cost for each user are unknown, which are difficult to make accurate estimations.

To address the above challenges, we propose a novel bilevel optimization framework: Multi-channel Sequential Budget Constrained Bidding (**MSBCB**), which transforms the original bilevel optimization problem into an equivalent two-level optimization with significantly reduced searching space. The higher-level only needs to optimize over one dimensional variable and the lower-level learns the optimal bidding policy for each user and computes the corresponding optimal budget allocation solution. For the lower-level, we derive an optimal reward function with theoretical guarantee. Besides, we also propose an action space reduction approach to significantly increase the learning efficiency of the lower-level. Finally, extensive offline analyses and online A/B testing conducted on one of the world's largest E-commerce platforms, Taobao, show the superior performance of our algorithm over state-of-the-art baselines.

2. Formulation: Dynamic Knapsack Problem

Within a time period of k days, we assume that there are N users $\{i=1,...,N\}$ visiting the E-commerce platform. Each user may interact with the app multiple times and trigger multiple advertising requests. During the sequential interactions between an ad and a user, each ad exposure could influence the user's mind and therefore contributes to the final conversion. Given a selected ad, for each individual user i, we build a separate Markov Decision Process (MDP) (Sutton & Barto, 2018) to model their sequential interaction. We denote the advertising policy of the ad towards user i as π_i, which takes user i's state as input and outputs the auction bid. Details of the MDP will be discussed in Section 3.2. For the selected ad, we define $V_G(i|\pi_i)$ and $V_C(i|\pi_i)$ as the expected long-term cumulative value and cost for each user i under policy π_i. Formally,

$$\begin{aligned} V_G(i|\pi_i) &= \mathbb{E}\left[G_i|\pi_i\right] = \mathbb{E}[\sum_{t=0}^{T_i} v_t|\pi_i] \\ V_C(i|\pi_i) &= \mathbb{E}\left[C_i|\pi_i\right] = \mathbb{E}[\sum_{t=0}^{T_i} c_t|\pi_i] \end{aligned} \tag{1}$$

where v_t and c_t represent the value (i.e., the revenue) and cost obtained from each request t according to policy π_i, $G_i = \sum_{t=0}^{T_i} v_t$ and $C_i = \sum_{t=0}^{T_i} c_t$ represent the long-term cumulative value and cumulative cost, T_i is the length of the interaction sequence.

Given the above definitions, for an advertiser, our target is to maximize its long-term cumulative revenue over k days under a budget constraint B, which is formulated as:

$$\begin{aligned} \max_{\Pi} \max_{\mathcal{X}} & \sum_{i=1}^{N} x_i\, V_G(i|\pi_i) \\ s.t. & \sum_{i=1}^{N} x_i\, V_C(i|\pi_i) \leq B \end{aligned} \tag{2}$$

where $\Pi = \{\pi_1, ..., \pi_N\}$, $\mathcal{X} = \{x_1, ..., x_N\}$, and $x_i \in \{0, 1\}$ indicates whether the user i is selected. Since whether displaying an ad to user i does not have any impact on user j's behaviors, $V_G(i|\pi_i)$, $V_C(i|\pi_i)$ and π_i among different users are independent. Thus, given any fixed advertising policy $\Pi = \{\pi_1, ..., \pi_N\}$, $V_G(i|\pi_i)$ and $V_C(i|\pi_i)$ for each user i are fixed and the inner optimization of Equation (2) can be viewed as a classic knapsack problem. The items to be put into the knapsack are the users. However, different advertising policies would lead to different $V_G(i|\pi_i)$s and $V_C(i|\pi_i)$s for each user, thus here we define Equation (2) as a Dynamic Knapsack Problem where the value and cost of each item in the knapsack are dynamic. From the perspective of optimization, Formulation (2) is a typical bilevel optimization, where the optimization of Π is embedded (nested) within the optimization of $\mathcal{X}$. This bilevel optimization is challenging due to the following reasons:

(1) The optimization space of the joint Π is continuous (for the bid space is continuous). The optimization space of $\mathcal{X}$ is discrete, which grows exponentially in the number of users (hundreds of millions). Therefore, the solution space of the combination of Π and $\mathcal{X}$ is enormous and thus is difficult or even impossible to optimize directly.

(2) The value of $V_G(i|\pi_i)$ and $V_C(i|\pi_i)$ are unknown and variable, efficient approaches are required to estimate these values online under limited samples.

3. Methodology: MSBCB Framework

3.1. Bilevel Decomposition and Proof of Correctness

Based on the above analysis, the bilevel optimization (2) is computationally prohibitive and cannot be solved directly. In this paper, we first decompose it into an equivalent two-level sequential optimization process. When taking a fixed policy Π as input, we denote the optimal solution of the degraded and static Knapsack Problem as $K = \mathrm{KP}(\Pi)$. Further, the global optimal solution of Problem (2) could be defined as:

$$K^* = \max_{\pi_1, \pi_2, ..., \pi_N} \mathrm{KP}(\Pi) \tag{3}$$

where $\pi_1, ..., \pi_N$ are independent variables and K^* is the global optimal solution. To obtain K^*, we must firstly specify the form of the function $\mathrm{KP}(\Pi)$.

When taking a fixed policy Π as input, computing $\mathrm{KP}(\Pi)$ is a classic static knapsack problem. However, another challenge in online advertising is that the user requests are arriving sequentially in real time and thus real-time decision makings are required. Complicated algorithms (e.g. dynamic programming) are not applicable due to the incompleteness of all users values and costs.

On the contrary, the Greedy algorithm could compute a greedy solution without completely knowing the whole set of candidate users beforehand. We will discuss this latter. Besides, the Greedy algorithm can achieve nearly optimal solution in the online advertising (Zhang et al., 2014; Wu et al., 2018). As proved by Dantzig (1957), if $\forall i \in 1, ..., N$, $V_C(i|\pi_i) \leq (1-\lambda)B, 0 \leq \lambda \leq 1$, i.e., the cumulative cost for each user is much less than the budget, the Greedy algorithm achieves an approximation ratio of λ, which means the greedy solution is at least λ times of the optimal solution K. The closer the λ gets to 1, the higher the quality of the greedy solution will be. In online advertising, λ is usually greater than 99.9%. Thus, the greedy solution is approximately optimal. We provide the detailed data and proof in Section B.1 of the Appendix. Therefore, in this paper, we refer to the Greedy algorithm, i.e., $\mathrm{KP}(\Pi) \leftarrow \mathrm{Greedy}(\Pi)$.

We define $\mathrm{CPR}_i = \frac{V_G(i|\pi_i)}{V_C(i|\pi_i)}$ as the Cost-Performance Ratio of each user i. The greedy solution is computed by:

(1) Sorting all users according to the Cost-Performance Ratio CPR_i in a descending order;

(2) Pick users from top to bottom until the cumulative cost violates the budget constraint.

$V_G(i\|\pi_i)$	$V_C(i\|\pi_i)$	$\mathrm{CPR}_i = V_G(i\|\pi_i)/V_C(i\|\pi_i)$
20	2	10
18	2	9
16	2	8
14	2	7
12	2	6
10	2	5
8	2	4

Budget Constraint: $B = 8$

Sorting in descending order

CPR_i threshold: $\mathrm{CPR}_{thr} = 7$

Figure 2. The solution computing process of the Greedy algorithm.

An illustration is shown in Figure 2. In this example, the budget constraint $B = 8$. We denote the CPR_i of the last picked user as $\mathrm{CPR}_{\mathrm{thr}}$, the threshold of the cost-performance ratio. In this example, the $\mathrm{CPR}_{\mathrm{thr}} = 7$. The advantage is that the Greedy algorithm only selects users whose $\mathrm{CPR}_i \geq \mathrm{CPR}_{\mathrm{thr}}$. If we could estimate the $\mathrm{CPR}_{\mathrm{thr}}$ beforehand, the Greedy algorithm could compute the solution online, without completely knowing the values and costs of all users.

Now that $\mathrm{KP}(\Pi) \leftarrow \mathrm{Greedy}(\Pi)$ and the Greedy algorithm prefers users with larger CPR_i (only pick users whose $\mathrm{CPR}_i \geq \mathrm{CPR}_{\mathrm{thr}}$), according to Equation 3, to further improve the solution quality, *an intuitive way is to optimize* π_i *for each user* i *such that each* CPR_i *could be maximized, i.e.,* $\pi_i' = argmax_{\pi_i}\ CPR_i$. However, this intuition is incorrect. Maximizing the CPR_i of each user cannot guarantee that the greedy solution $K = \mathrm{Greedy}(\Pi)$ could be maximized. Next, we show that given all users' CPRs are maximized, we can still further improve the solution quality by increasing certain users' allocated budgets and decreasing their CPRs in exchange for greater overall cumulative value. Before we go into the details, we firstly give Lemma 1.

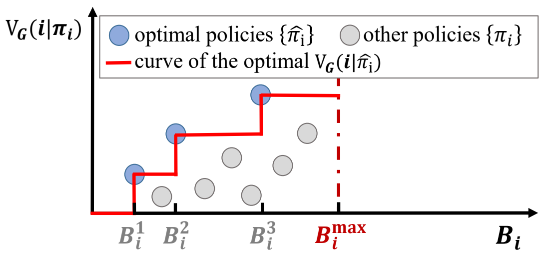

Figure 3. $V_G(i|\hat{\pi}_i)$ is monotonic with $V_C(i|\hat{\pi}_i)$.

Lemma 1. *For each user* i*, the cumulative value* $V_G(i|\hat{\pi}_i)$ *increases monotonically with the increase of cost* $V_C(i|\hat{\pi}_i)$ *within the range of all possible optimal policies* $\{\hat{\pi}_i\}$.

Proof. We assume that the maximum budget allocated to each user i as $B_i \in [0, B_i^{\max}]$, where $B_i^{\max}$ is the maximum cost user i can consume. Then, for each user i, within the current budget constraint B_i, the optimal advertising policy $\hat{\pi}_i$ must be the one which could maximize the cumulative value, i.e., $\hat{\pi}_i = \mathrm{argmax}_{\pi_i} V_G(i|\pi_i)$, s.t. $V_C(i|\pi_i) \leq B_i$. Obviously, as B_i moves from 0 to $B_i^{\max}$, we will get a set of optimal policies $\{\hat{\pi}_i\}$, whose cost $V_C(i|\hat{\pi}_i)$ and value $V_G(i|\hat{\pi}_i)$ are both increasing. An illustration is shown in Figure 3. Thus we complete the proof.

As illustrated in Figure 4, each user's CPR_i (the width of each rectangular slice) is maximized initially. According to Lemma 1, for a user i, if we increase $V_C(i|\pi_i)$ by $\Delta V_C(i)$, i.e., increase the height of user i by $\Delta V_C(i)$, the corresponding $V_G(i|\pi_i)$ will also increase. We denote this increase in value as $\Delta V_G(i)$. Since there is a budget limit, a small increased height $\Delta V_C(i)$ will squeeze out a small area nearby the $\mathrm{CPR}_{\mathrm{thr}}$, whose height is also $\Delta V_C(i)$ and width is $\mathrm{CPR}_{\mathrm{thr}}$ [1]. We denote the increased area by reshaping user i as $\Delta V_G^+ = \Delta V_G(i)$ and the decreased area due to extrusion as $\Delta V_G^- = \mathrm{CPR}_{\mathrm{thr}} * \Delta V_C(i)$. Overall, if $\Delta V_G^+ > \Delta V_G^-$, the total area will be further increased. For

[1]Since $\Delta V_C(j) \ll B$, the area squeezed out could be considered as a tiny and smooth change and the width of the last user is approximately equal to $\mathrm{CPR}_{\mathrm{thr}}$

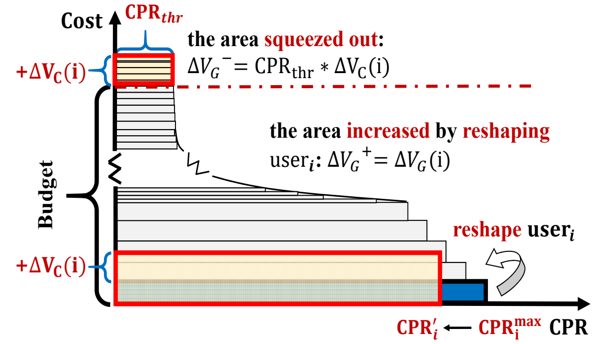

Figure 4. The x-axis denotes each user's CPR_i and y-axis denotes the cumulative cost of the Greedy algorithm. All users are sorted in descending order by their CPRs and arranged from bottom to top. Each rectangular slice's area (in gray) represents $V_G(i|\pi_i) = \mathrm{CPR}_i * V_C(i|\pi_i)$, where CPR_i and $V_C(i|\pi_i)$ are the width and height. Note that, the height of each rectangular slice is much less than the budget constraint, i.e., $V_C(i|\pi_i) \ll B$. The red dashed line marks the position of the budget constraint. The total area of all rectangular slices under the red dashed line constitutes the greedy solution.

any user i, $\Delta V_G^+ > \Delta V_G^-$ yields:

$$\Delta V_G(i) > \mathrm{CPR}_{\mathrm{thr}} * \Delta V_C(i) \tag{4}$$

where $\Delta V_G(i)$ and $\Delta V_C(i)$ are caused by the change of π_i, e.g., from π_i' to π_i''. We denote $\Delta V_G(i)$ as $V_G(i|\pi_i'') - V_G(i|\pi_i')$ and $\Delta V_C(i)$ as $V_C(i|\pi_i'') - V_C(i|\pi_i')$. We conclude that the greedy solution $K = \mathrm{Greedy}(\Pi')$ can be further improved if there exists any user i whose current policy π_i' can be further improved to π_i'' such that $\Delta V_G(i) > \mathrm{CPR}_{\mathrm{thr}} * \Delta V_C(i)$. Otherwise, the current solution is optimal. Finally, we provide the definition of the optimal π_i^* in Theorem 1.

Theorem 1. *Under the Greedy paradigm* ($K = Greedy(\Pi)$)*, for any given* CPR_{thr}*, the optimal advertising policy* π_i^* *for each user* i *is the one which could maximize* $V_G(i|\pi_i) - CPR_{thr} * V_C(i|\pi_i)$*. In other words,* π_i^* *is defined as:*

$$\pi_i^* = \operatorname*{argmax}_{\pi_i} \left[V_G(i|\pi_i) - \mathrm{CPR}_{\mathrm{thr}} * V_C(i|\pi_i)\right] \tag{5}$$

We denote $\Pi^* = \{\pi_1^*, ..., \pi_N^*\}$. The corresponding solution $K_{\mathrm{greedy}}^* = \mathrm{Greedy}(\Pi^*)$ is the optimal Greedy solution of the Dynamic Knapsack Problem defined in Equation (2).

Proof of Theorem 1. We define $\Pi^* = \{\pi_1^*, ..., \pi_N^*\}$, where π_i^* is defined according to Equation (5), $\forall i \in \{1, ..., N\}$. We prove Theorem 1 by contradiction. Given the threshold $\mathrm{CPR}_{\mathrm{thr}}$, we firstly assume that $\mathrm{Greedy}(\Pi^*)$ is not the optimal greedy solution of the Dynamic Knapsack Problem, which means we could at least find a user i, whose policy π_i^* could be further improved to policy π_i'' such that the overall area is increased. This means we could find a better policy π_i'' for user i such that $\Delta V_G(i) > \mathrm{CPR}_{\mathrm{thr}} * \Delta V_C(i)$ according to Equation (4), where $\Delta V_G(i) = V_G(j|\pi_i'') - V_G(i|\pi_i^*)$

and $\Delta V_C(i) = V_C(i|\pi_i'') - V_C(i|\pi_i^*)$ ($V_G(i|\hat{\pi}_i)$ increases monotonically with the increase of $V_C(i|\hat{\pi}_i)$ according to Lemma 1). Further, $\Delta V_G(i) > \text{CPR}_{\text{thr}} * \Delta V_C(i)$ yields:

$$\begin{aligned}&[V_G(i|\pi_i'') - \text{CPR}_{\text{thr}} * V_C(i|\pi_i'')] > \\ &[V_G(i|\pi_i^*) - \text{CPR}_{\text{thr}} * V_C(i|\pi_i^*)]\end{aligned} \tag{6}$$

Equation (6) indicates that

$$\pi_i^* \neq \operatorname*{argmax}_{\pi_i} [V_G(i|\pi_i) - \text{CPR}_{\text{thr}} * V_C(i|\pi_i)]$$

which contradicts the definition of π_i^* in Equation (5). Thus, the theorem statement is obtained.

Algorithm 1 MSBCB Framework.

1: **Input:** an initial CPR_{thr};
2: **Output:** optimal greedy solution of the Dynamic Knapsack Problem;
3: **for** each period until convergence **do**
4: Taking the current estimated CPR_{thr} as input, the agent optimizes the advertising policy π_i for each user i according to Section 3.2 and acquires the optimal $\Pi^* = \{\pi_1^*, ..., \pi_N^*\}$.
5: Based on the current estimated CPR_{thr} and the obtained Π^*, the agent calculates the greedy solution according to Section 3.3 and collects the actual feedback cost and the predefined budget.
6: Update the estimated CPR_{thr} towards $\text{CPR}^*_{\text{thr}}$ by minimizing the gap between the actual feedback cost and the budget according to Section 3.4.
7: **end for**

We present the overall MSBCB framework in Algorithm 1, which involves a two-level sequential optimization process. **(1) Lower-level:** Given any CPR_{thr}, we could obtain the optimal advertising policy Π^* following Equation 5 of Theorem 1, which will be discussed in Section 3.2. Then, based on CPR_{thr} and the optimized Π^*, we could acquire the Greedy solution by selecting users whose $\text{CPR}_i \geq \text{CPR}_{\text{thr}}$, which will be detailed in Section 3.3. **(2) Higher-level:** However, the current CPR_{thr} might $\neq \text{CPR}^*_{\text{thr}}$, which means selecting all users whose $\text{CPR}_i \geq \text{CPR}_{\text{thr}}$ might violate the budget constraint or lead to a substantial budget surplus. Thus, we optimize the current CPR_{thr} towards $\text{CPR}^*_{\text{thr}}$ in Section 3.4. Overall, the optimization space of $\mathcal{X}$ is reduced from 2^N to a one-dimensional continuous variable CPR_{thr}. We conclude that Algorithm 1 could iteratively converge to a unique and approximate optimal solution. We present the proof of convergence in Section B.3 of the Appendix.

3.2. Lower-level Advertising Policy Optimization with Reinforcement Learning

Given a threshold CPR_{thr} as input, we aim to acquire the optimal advertising policy π_i^* defined in Equation (5) of Theorem 1. Combining the definitions of $V_G(i|\pi_i)$ and $V_C(i|\pi_i)$ with Equation (5), we have

$$\begin{aligned}\pi_i^* &= \operatorname*{argmax}_{\pi_i} [V_G(i|\pi_i) - \text{CPR}_{\text{thr}} * V_C(i|\pi_i)] \\ &= \operatorname*{argmax}_{\pi_i} (\mathbb{E}[G_i|\pi_i] - \text{CPR}_{\text{thr}} * \mathbb{E}[C_i|\pi_i]) \\ &= \operatorname*{argmax}_{\pi_i} \mathbb{E}\left[(G_i - \text{CPR}_{\text{thr}} * C_i)\,|\pi_i\right] \\ &= \operatorname*{argmax}_{\pi_i} \mathbb{E}[\sum_{t=0}^{T_i}(v_t - \text{CPR}_{\text{thr}} * c_t)|\pi_i]\end{aligned} \tag{7}$$

Accordingly, we define $r_t = v_t - \text{CPR}_{\text{thr}} * c_t$, i.e., value $-$ $\text{CPR}_{\text{thr}} *$ cost, as the immediate profit acquired at each step t. The objective of Equation (7) is to obtain the optimal advertising policy π_i^* which could maximize the expected long-term cumulative profit. To solve this sequential decision making problem, we formulate it as an MDP and use Reinforcement Learning (RL) (Sutton & Barto, 2018) techniques to acquire the optimal policy π_i^*.

We consider an episodic MDP, where an episode starts with the first interaction between a user and an ad, and ends up with a purchase or exceeding the maximum step T_i as:

- **State** $\mathcal{S}$: The state s_t should in principle reflect the user request status, ad info, user-ad interaction history info and the RTB environment.
- **Action** $\mathcal{A}$: The action each agent can take in the RTB platform is the bid, which is a real number between 0 and the upper bound bid_{max}, i.e., $a_t \in [0, \text{bid}_{\text{max}}]$.
- **Reward** $\mathcal{R}(\mathcal{S} \times \mathcal{A} \to \mathbb{R})$: The immediate reward at step t is defined as $r_t = v_t - \text{CPR}_{\text{thr}} * c_t$.
- **Transition probability** $\mathcal{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1])$: Transition probability is defined as the probability of state transitioning from s_t to s_{t+1} when taking action a_t.
- **Discount factor** γ: The bidding agent aims to maximize the total discounted reward $R_j = \sum_{k=t}^{T_i} \gamma\, r_k$ from step t onwards, where $\gamma \in [0,1]$.

For each user i, we define the state-action value function $Q(s,a) = \mathbb{E}[R_i|s,a,\pi_i]$ as the expected cumulative reward achieved by following the advertising policy π_i. The MDP can be solved using existing Deep Reinforcement Learning (DRL) algorithms such as DQN (Mnih et al., 2013), DDPG (Lillicrap et al., 2015) and PPO (Schulman et al., 2017). After sufficient training, we would acquire the optimized advertising policies $\Pi^* = \{\pi_1^*, ...\pi_N^*\}$ for all users.

3.3. Lower-level User Selection by Greedy Algorithm

Taking the current CPR_{thr} and the optimized advertising policies $\Pi^* = \{\pi_1^*, ...\pi_N^*\}$ as inputs, we aim to obtain the greedy solution of the Dynamic Knapsack Problem. In reality, we cannot know all users' request sequences and their values and costs beforehand because the user requests are arriving sequentially in real time. Thus, many complicated methods depending on the completeness of all users' data, e.g., the dynamic programming approach (Martello et al., 1999), are not applicable. Even the traditional Greedy algorithm cannot be applied either. Fortunately, the greedy solution could be computed online in an easy way: given the threshold CPR_{thr}, the agent only has to select users online whose CPRs are greater than the threshold (an illustration is shown in Figure 2). Therefore, we only have to estimate the $\text{CPR}_i = \frac{V_G(i|\pi_i)}{V_C(i|\pi_i)}$ for each user i. To acquire $V_G(i|\pi_i)$ and $V_C(i|\pi_i)$, besides Q(s,a), we also maintain two other state value functions $V_G(s)$ and $V_C(s)$ according to the Bellman Equation (Sutton & Barto, 2018), where $V_G(s) = \mathbb{E}[G_j|s, \pi_j]$ and $V_C(s) = \mathbb{E}[C_j|s, \pi_j]$.

3.4. Higher-level Optimization by Feedback Control

However, the current estimated threshold CPR_{thr} might have some bias from the optimal $\text{CPR}^*_{\text{thr}}$. Thus, selecting all users whose $\text{CPR}_i \geq \text{CPR}_{\text{thr}}$ might violate the budget constraint or lead to a substantial budget surplus. Only when the estimated CPR_{thr} is exactly the same with the optimal $\text{CPR}^*_{\text{thr}}$, the actual total advertising cost will be equal to the budget. To achieve this, we design a feedback control mechanism, i.e., a PID controller (Åström & Hägglund, 1995), to dynamically adjust the CPR_{thr} towards $\text{CPR}^*_{\text{thr}}$ according to actual feedback of the overall cost. The core formula is:

$$\text{CPR}_{\text{thr}} *= \left[1+\alpha_1(\tfrac{\text{cost}_t}{B}-1)+\alpha_2(\tfrac{\text{cost}_{t-n:t}}{n*B}-1)\right] \quad (8)$$

where cost_t is the actual feedback cost of the current period, B is the budget, $\text{cost}_{t-n:t}$ and $n*B$ are the overall cost and the overall budget of the most recent n periods. α_1 and α_2 are two learning rates. The main idea is when the actual cost exceeds (is less than) the budget, the threshold CPR_{thr} will be increased (decreased) accordingly such that less (more) users will be selected, which will reduce (increase) the cost in turn. The first term $\alpha_1(\frac{\text{cost}_t}{B}-1)$ is designed to keep up with the latest changes. The second term $\alpha_2(\frac{\text{cost}_{t-n:t}}{n*B}-1)$ is designed to stabilize learning.

3.5. Action Space Reduction for RL in Advertising

However, when applying the RL approaches mentioned in Section 3.2 to online advertising, one typical issue is that the sample utilization is inefficient. The main reason is that the action space of the agent is continuous, thus the range of $[0, \text{bid}_{\text{max}}]$ needs to be fully explored in all states. To resolve this problem, we reduce the magnitude of the continuous action space (i.e., $a_t \in [0, \text{bid}_{\text{max}}]$) to a binary one (i.e., $\widehat{a}_t \in \{0, 1\}$) by making full use of the prior knowledge in advertising, which greatly improves the sample utilization of the RL approaches. Specifically, since different bids a_t can only result in two different outcomes $\widehat{a}_t \in \{0, 1\}$, where $\widehat{a}_t = 1$ or 0 indicates whether the ad is displayed to the user, we only have to evaluate the different expected returns resulted by $\widehat{a}_t = 1$ or $\widehat{a}_t = 0$ for $Q(s, a)$. We denote the greedy action $\widehat{a}_t{}^*$ based on the current value estimations as:

$$\widehat{a}_t{}^* = \begin{cases} 1 & \text{if } Q(s, \widehat{a}_t = 1) > Q(s, \widehat{a}_t = 0) \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Then, to obtain an executable bid, for $\widehat{a}_t{}^* = 0$, we could offer a low enough bid, e.g., $a_t = 0$, to make sure that it is impossible to win the auction. For $\widehat{a}_t{}^* = 1$, we propose an optimal bid function which could output a bid greater than the second highest bid while not overbidding.

In detail, we maintain two state-action value functions $Q_G(s, \widehat{a}_t) = \mathbb{E}[G_i|s, \widehat{a}_t, \pi_i]$ and $Q_C(s, \widehat{a}_t) = \mathbb{E}[C_i|s, \widehat{a}_t, \pi_i]$. Since the reward function is defined as $r_t = v_t - \text{CPR}_{\text{thr}} * c_t$, we have $Q(s, \widehat{a}_t) = Q_G(s, \widehat{a}_t) - \text{CPR}_{\text{thr}} * Q_C(s, \widehat{a}_t)$. Then $Q(s, \widehat{a}_t = 1) > Q(s, \widehat{a}_t = 0)$ yields:

$$\begin{aligned} &[Q_G(s, \widehat{a}_t = 1) - \text{CPR}_{\text{thr}} * Q_C(s, \widehat{a}_t = 1)] > \\ &[Q_G(s, \widehat{a}_t = 0) - \text{CPR}_{\text{thr}} * Q_C(s, \widehat{a}_t = 0)] \end{aligned} \quad (10)$$

If $\widehat{a}_t = 0$, the expected immediate cost is 0 (since the ad is not exposed). If $\widehat{a}_t = 1$, we denote the expected immediate cost as $\mathbb{E}[c_t|\widehat{a}_t = 1]$, whose value depends on the pricing model. In online advertising, typical pricing models includes CPM (Cost Per Mille, the advertiser bid for impressions and is charged based on impressions), CPC (Cost Per Click, the advertiser bid for clicks and is charged based on clicks) and CPS (Cost Per Sales, the advertiser bid for conversions and is charged based on conversions). If CPM is used, $\mathbb{E}[c_t|\widehat{a}_t = 1] = \textbf{bid}_t^{\textbf{2nd}}$, where $\textbf{bid}_t^{\textbf{2nd}}$ denotes the second highest bid in the auction. If CPC is used, $\mathbb{E}[c_t|\widehat{a}_t = 1] = \textbf{bid}_t^{\textbf{2nd}} * \text{pCTR}$, where pCTR represents the predicted Click-Through Rate. If CPS is used, $\mathbb{E}[c_t|\widehat{a}_t = 1] = \textbf{bid}_t^{\textbf{2nd}} * \text{pCTR} * \text{pCVR}$, where pCVR represents the predicted Conversion Rate. For ease of presentation, we take CPM for an example. Under CPM,

$$\begin{aligned} Q_C(s, \widehat{a}_t = 1) &= \mathbb{E}[c_t + \sum_{k=t+1}^{T_i} c_k | s, \widehat{a}_t = 1, \pi_i] \\ &= \textbf{bid}_t^{\textbf{2nd}} + \mathbb{E}[\sum_{k=t+1}^{T_i} c_k | s, \widehat{a}_t = 1, \pi_i] \\ Q_C(s, \widehat{a}_t = 0) &= \mathbf{0} + \mathbb{E}[\sum_{k=t+1}^{T_i} c_k | s, \widehat{a}_t = 0, \pi_i] \end{aligned} \quad (11)$$

Notice that the second highest bid $\textbf{bid}_t^{\textbf{2nd}}$ is unknown until the current auction is finished. Substituting Equation (11) into Equation (10), we acquire

$$\begin{aligned}\mathbf{bid}_t^{\mathbf{2nd}} < \Bigg[& \left(\frac{Q_G(s, \widehat{a}_t = 1)}{\text{CPR}_{\text{thr}}} - Q_C^{\text{next}}(s, \widehat{a}_t = 1) \right) \\ & - \left(\frac{Q_G(s, \widehat{a}_t = 0)}{\text{CPR}_{\text{thr}}} - Q_C^{\text{next}}(s, \widehat{a}_t = 0) \right) \Bigg]\end{aligned} \tag{12}$$

where $Q_C^{\text{next}}(s, \widehat{a}_t) = \mathbb{E}[\sum_{k=t+1}^{T_j} c_k | s, \widehat{a}_t, \pi_i]$. We denote the term on the right of the '$<$' in Equation (12) as $\mathbf{b}_t^*$. And we conclude that the bidding agent can always set the bid price $a_t = \mathbf{b}_t^*$ during the online bidding phase, which is the optimal action without any loss of accuracy. Refer to Section B.2 of the Appendix for proof. For CPC or CPS, the optimal bid formula $\mathbf{b}_t^*$ can be easily acquired by substituting the corresponding $\mathbb{E}[c_t | \widehat{a}_t = 1]$ into Equation 11. Here, we reaffirm that our action space reduction technique is a generalized design and is applicable to different pricing models.

4. Empirical Evaluation: Simulations

We start with designing simulation experiments to shed light on the contributions of the proposed framework MSBCB under more controlled settings. Similar to the simulation settings of (Ie et al., 2019), we assume there are a set of users $\{i = 1, ..., N\}$, a set of ads $\mathcal{D}$ and a set of commodity categories $\mathcal{T}$. Each ad $d \in \mathcal{D}$ has an associated category. Each user i has various degrees of interests in commodity categories, which is influenced by the displayed ad. When user i consumes ad d, his interest in category $T(d)$ is nudged stochastically, biased slightly towards increasing his interest, but allows some chance of decreasing his interest. We set $N = 10000$, $|\mathcal{D}| = 2000$ and $|\mathcal{T}| = 20$ in the following experiments. Detailed settings of the simulation environment can be found in Section D.1 of the Appendix.

4.1. Baselines

We compare our MCBCB with following baseline strategies:

- Myopic Approaches: (1) Manual Bid is a strategy that the agent continuously bids at the same price initialized by the advertiser. (2) Contextual Bandit (Zhang et al., 2014) aims at maximizing the accumulated short-term value of each request based on the Greedy framework.
- Greedy with maximized CPR: This approach is similar to our method under the Greedy framework except that each π_i is optimized by maximizing the long-term CPR. In the offline simulation, we enumerate all policies for each user and select the one which could maximize its CPR. This approach is named as Greedy+maxCPR.
- Greedy with state-of-the-art RL approaches: These baselines, i.e., Greedy+DQN, Greedy+DDPG and Greedy+PPO, utilize the same reward function with our MSBCB to optimize the lower-level optimization of Π. The difference is that our MSBCB leverages the action space reduction technique. For DQN and PPO, we discretize the bid action space $[0, \text{bid}_{\max}]$ evenly into 11 real numbers as the valid actions.
- Undecomposed Optimization: These baselines are RL approaches (DQN,DDPG and PPO) based on the Constrained Markov Decision Process (CMDP). They are named as Constrained+DQN, Constrained+DDPG, Constrained+PPO respectively. We follow the CMDP design and settings in (Wu et al., 2018).
- Offline Optimal: The optimal solution of the Dynamic Knapsack Problem can be computed by dynamic programming in offline simulation because we could enumerate all possible policies to get the corresponding long-term values and costs for each user. Note that since users' request sequences are unknown beforehand and there is only one chance for the ad to bid for each request in the online advertising systems, the optimal solution can only be obtained in offline simulation.

4.2. Experimental Results

We conduct extensive analysis of our MSBCB in the following 5 aspects. All approaches aim to maximize the advertiser's cumulative revenue under a fixed budget constraint. All experimental results are averaged over 10 runs. The hyperparameters for each algorithm are set to the best we found after grid-search optimization.

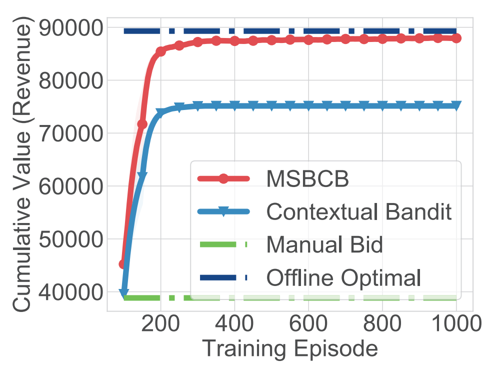

Figure 5. Values comparisons (learning curves) of the myopic approaches with non-myopic approaches and the offline optimal.

Myopic vs Non-myopic. To show the benefits of upgrading the myopic advertising system into a farsighted one, we compare the cumulative revenue achieved by our MSBCB with two other myopic baselines. The learning curves and results are shown in Figure 5 and Table 1. We see that MSBCB outperforms the Manual Bid and the Contextual Bandit by a large margin, which indicates that taking account of the long-term effect of each ad exposure could significantly improve the cumulative advertising results.

MSBCB vs the Offline Optimal. In Figure 5, we also compare our MSBCB with the Offline Optimal, which is computed by a modified dynamic programming algorithm. We see that as the training continues, our MSBCB gradually achieves an approximately optimal solution. Detailed results are summarized in Table 1. Our MSBCB empirically achieves an approximation ratio of 98.53%(±0.36%).

MSBCB vs Greedy with maximized CPR. As discussed in Section 3.1, under the Greedy framework, maximizing each user's CPR_i cannot guarantee that the greedy solution of the Dynamic Knapsack Problem (2) could be maximized. The optimal advertising policy π_i for each user is given by Theorem 1. To experimentally verify the correctness of Theorem 1, we compare the cumulative revenue achieved by MSBCB and the Greedy with maximized CPR. As shown in Figure 6 and Table 1, MSBCB outperforms Greedy with maximized CPR and achieves a $+5.11\%$ improvement.

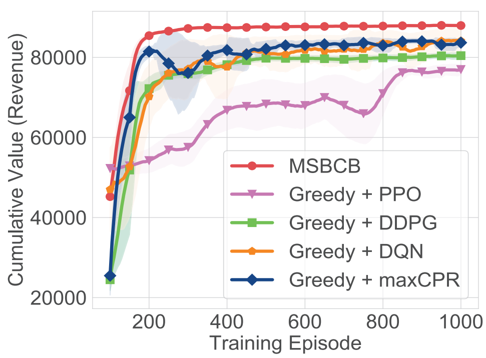

Figure 6. Value comparisons of MSBCB with the Greedy with maximized CPR and the Greedy with state-of-the-art RL.

MSBCB vs Greedy with state-of-the-art RL approaches. Besides, to show the effectiveness of the action-space reduction proposed in Section 3.5, we compare MSBCB with the state-of-the-art DRL approaches under the Greedy framework. As shown in Figure 6 and Table 1, MSBCB outperforms Greedy+DQN, Greedy+DDPG and Greedy+PPO both in the cumulative revenue and the convergence speed, which shows that the action space reduction effectively improves the sample efficiency of RL approaches.

Decomposed MSBCB vs Undecomposed optimization. Similar to (Wu et al., 2018), the undecomposed optimization baselines consider all users requests as a whole and model the budget allocations among all request as a CMDP. As shown in Figure 7 and Table 1, MSBCB outperforms the CMDP based RL approaches by a large margin. The reason of the poor performance in CMDP-based approaches is that these methods model all users' requests as a whole sequence and thus the learning process is particularly inefficient. In contrast, our MSBCB decomposes the whole sequence optimization into an efficient two-level optimization process, thus can achieve better performance more easily.

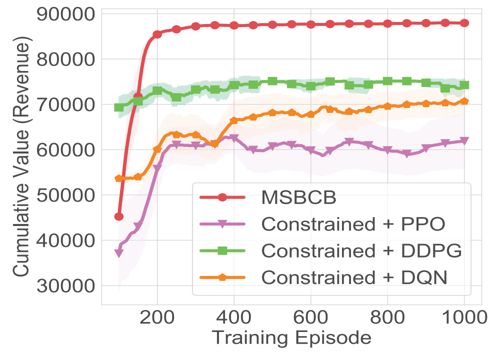

Figure 7. Values comparison (learning curves) of MSBCB and state-of-the-art CMDP based RL approaches.

Table 1. Cumulative values, costs, value improvements (over Contextual Bandit) and the approximation ratio of all approaches.

Method	Revenue	Cost	Revenue Impro	Approximation Ratio
Manual Bid	38838.28	11995.10	-48.31%	43.5%
Contextual Bandit	75137.30	11995.46	0%	84.15%
Constrained + PPO	61890.92	11954.07	-17.63±16.11%	69.31±13.56%
Constrained + DDPG	74259.12	11996.12	-1.19±3.66%	83.17±3.08%
Constrained + DQN	70662.65	11881.12	-5.96±7.83%	79.14±6.59%
Greedy + maxCPR	83668.70	11914.12	11.35±2.84%	93.70±2.36%
Greedy + PPO	76970.35	11825.59	2.44±3.52%	86.20±2.93%
Greedy + DDPG	80424.69	11841.28	7.04±1.13%	90.07±0.92%
Greedy + DQN	84117.09	11794.24	11.95±4.96%	94.21±4.14%
MSBCB	**87947.99**	**11957.57**	**17.95±0.42%**	**98.50±0.33%**
MSBCB (enum)	**89251.77**	**11988.36**	**18.78%**	**99.96%**
Offline Optimal	89291.11	11999.23	18.84%	100.00%

The complete comparisons of all approaches are shown in Table 1. The budget constraint B is set to 12000 for all experiments. In Table 1,we also add an MSBCB (enum), which is the theoretical upper bound of our MSBCB. The difference between MSBCB (enum) and MSBCB is that: the MSBCB (enum) computes the optimal advertising policy π_i^* for each user i by enumerating all possible policies. Instead of utilizing the RL approach, MSBCB (enum) could find the one which maximizes $V_G(i|\pi_i) - \text{CPR}_{\text{thr}} * V_C(i|\pi_i)$. We see MSBCB (enum) is very close to the optimal solution and reaches an approximation ratio of 99.96%.

4.3. Effectiveness of Action Space Reduction

As shown in Table 2, MSBCB achieves a revenue of 75000 in only 61 epochs, reducing more than 60% samples compared with the state-of-the-art RL baselines without using the action-space reduction technique. As for learning process, our MSBCB achieves the same revenue (80000) more than 10 times faster than the baselines, reducing more than 90% samples and finally reaches the highest revenue. Thus, with the action space reduction technique, our MSBCB could reach a higher performance with a faster speed and significantly improve the sample efficiency. More analysis of our MSBCB, e.g., the convergence of Π^* and $\mathrm{CPR}^*_{\mathrm{thr}}$, and the hyperparameter settings of the offline experiments are shown in Section D. of the Appendix.

Table 2. The training epochs and the number of samples needed by different approaches when achieving the same revenue level.

Revenue	75000		80000		85000	
Method	#Epoch	#Samples	#Epoch	#Samples	#Epoch	#Samples
Greedy+PPO	817	4183040	-	-	-	-
Greedy+DDPG	154	788480	853	4362240	-	-
Greedy+DQN	373	1909760	754	3855360	-	-
MSBCB	**61**	**312320**	**71**	**363520**	**104**	**532480**

5. Empirical Evaluation: Online A/B Testing

We deployed MSBCB on one of the world's largest E-commerce platforms, Taobao. Our platform is authorized by the advertisers to dynamically adjust their bid prices for each user request according its value in the real-time auction. In the online experiments, we compare MSBCB with two models widely used in the industry.

- Cross Entropy Method (CEM), which is a deployed production model, whose target is to optimize the immediate rewards. We consider CEM as the control group in the following evaluations.
- Contextual Bandit, which has been explained in previous section and is reserved as a contrast test.

The experiment involves 135,858,118 users and 72,147 ad items from 186 advertisers. For fair comparison, we control the consumers and the advertisers involved in the A/B testing to be homogeneous. In detail, the 135,858,118 users are randomly and evenly divided into 3 groups. For users in group #1, all 186 advertisers adopt the CEM algorithm. For users in group #2, all 186 advertisers adopt the Contextual Bandit algorithm. For users in group #3, all 186 advertisers adopt our MSBCB. Table 3 summarises the effects of the Contextual Bandit and our MSBCB compared to the Cross Entropy Method from Dec.10 to Dec.20 in 2019. From Table 3, we see that our MSBCB achieves a +10.08% improvement in revenue and a +10.31% improvement in ROI with almost the same cost (-0.20%). The results indicate that upgrading the myopic advertising strategy into a farsighted one could significantly improves the cumulative revenue. Besides, as shown in Figure 8, the daily ROI improvement also demonstrates the effectiveness of our MSBCB compared with the Contextual Bandit.

Given that there are only 186 advertisers take part in our online experiment, one frequently asked question is"**How does the MSBCB work across all ads?**" Since 186 is relatively small compared with the total number of advertisers, their policy updates would not cause dramatic changes to the RTB environment. In other words, the RTB environment is still approximately stationary from a single-ad perspective. This setting also works well with our practical business model-providing better service for VIP advertisers (about 0.2% of all the advertisers). In the case that the majority of the advertisers adopt MSBCB, the system cannot be estimated as being stationary from any single-ads perspective and explicit multi-agent modeling and coordination should be incorporated. Detailed analysis of the improvement in revenue for each advertiser is presented in Table 7 and Figure 19 of the Appendix. More details about the deployment and experimental results (e.g., the online model architecture) can also be found in Section C. and E. of the Appendix.

Table 3. The overall performance comparisons of the A/B testing. CVR represents the Conversion Rate of the users. #PV represents the number of page views. ROI $= \frac{\text{Revenue}}{\text{Cost}}$ means Return On Investment. (Notice that CEM is the control group and the improvements of Contextual Bandit and MSBCB are compared over CEM.)

Method	Revenue	Cost	CVR	#PV	ROI
Contextual Bandit	+0.91%	-3.26%	+4.78%	+4.62%	+4.31%
MSBCB	**+10.08%**	**-0.20%**	**+6.04%**	**+15.37%**	**+10.31%**

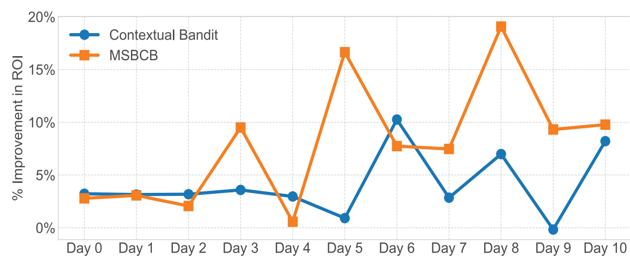

Figure 8. Daily ROI improvement comparisons of Contextual Bandit and MSBCB over Cross Entropy Method.

6. Conclusion

We formulate the multi-channel sequential advertising problem as a Dynamic Knapsack Problem, whose target is to maximize the long-term cumulative revenue over a period of time under a budget constraint. We decompose the original problem into an easier bilevel optimization, which significantly reduces the solution space. For the lower-level optimization, we derive an optimal reward function with theoretical guarantees and design an action space reduction technique to improve the sample efficiency. Extensive offline experimental analysis and online A/B testing demonstrate the superior performance of our MSBCB over the state-of-the-art baselines in terms of cumulative revenue.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (Grant Nos.: 61702362, U1836214), the Special Program of Artificial Intelligence and the Special Program of Artificial Intelligence of Tianjin Municipal Science and Technology Commission (No.: 569 17ZXRGGX00150) and the Alibaba Group through Alibaba Innovative Research Program. We deeply appreciate all teammates from Alibaba group for the significant supports for the online experiments.

References

Altman, E. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

Åström, K. J. and Hägglund, T. *PID controllers: theory, design, and tuning*, volume 2. Instrument society of America Research Triangle Park, NC, 1995.

Boutilier, C. and Lu, T. Budget allocation using weakly coupled, constrained markov decision processes. 2016.

Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y., and Guo, D. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 661–670. ACM, 2017.

Dantzig, G. B. Discrete-variable extremum problems. *Operations research*, 5(2):266–288, 1957.

Du, M., Sassioui, R., Varisteas, G., Brorsson, M., Cherkaoui, O., et al. Improving real-time bidding using a constrained markov decision process. In *International Conference on Advanced Data Mining and Applications*, pp. 711–726. Springer, 2017.

Du, R., Zhong, Y., Nair, H., Cui, B., and Shou, R. Causally driven incremental multi touch attribution using a recurrent neural network. *arXiv preprint arXiv:1902.00215*, 2019.

Edelman, B., Ostrovsky, M., and Schwarz, M. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review*, 97(1):242–259, 2007.

Ie, E., Jain, V., Wang, J., Navrekar, S., Agarwal, R., Wu, R., Cheng, H.-T., Lustman, M., Gatto, V., Covington, P., et al. Reinforcement learning for slate-based recommender systems: A tractable decomposition and practical methodology. *arXiv preprint arXiv:1905.12767*, 2019.

Ji, W. and Wang, X. Additional multi-touch attribution for online advertising. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Jin, J., Song, C., Li, H., Gai, K., Wang, J., and Zhang, W. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 2193–2201. ACM, 2018.

Li, P., Hawbani, A., et al. An efficient budget allocation algorithm for multi-channel advertising. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 886–891. IEEE, 2018.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Martello, S., Pisinger, D., and Toth, P. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, 1999.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Nuara, A., Sosio, N., TrovÃ, F., Zaccardi, M. C., Gatti, N., and Restelli, M. Dealing with interdependencies and uncertainty in multi-channel advertising campaigns optimization. In *The World Wide Web Conference*, pp. 1376–1386. ACM, 2019.

Ren, K., Zhang, W., Chang, K., Rong, Y., Yu, Y., and Wang, J. Bidding machine: Learning to bid for directly optimizing profits in display advertising. *IEEE Transactions on Knowledge and Data Engineering*, 30(4):645–659, 2017.

Ren, K., Fang, Y., Zhang, W., Liu, S., Li, J., Zhang, Y., Yu, Y., and Wang, J. Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1433–1442. ACM, 2018.

Ren, K., Qin, J., Zheng, L., Yang, Z., Zhang, W., and Yu, Y. Deep landscape forecasting for real-time bidding advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 363–372. ACM, 2019.

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

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Wu, D., Chen, X., Yang, X., Wang, H., Tan, Q., Zhang, X., Xu, J., and Gai, K. Budget constrained bidding by model-free reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1443–1451. ACM, 2018.

Yuan, S., Wang, J., and Zhao, X. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, pp. 1–8, 2013.

Zhang, W., Yuan, S., and Wang, J. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1077–1086. ACM, 2014.

Zhang, W., Ren, K., and Wang, J. Optimal real-time bidding frameworks discussion. *arXiv preprint arXiv:1602.01007*, 2016.

Zhu, H., Jin, J., Tan, C., Pan, F., Zeng, Y., Li, H., and Gai, K. Optimized cost per click in taobao display advertising. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2191–2200. ACM, 2017.