Online Competitive Influence Maximization

Jinhang Zuo[1], Xutong Liu[2], Carlee Joe-Wong[1], John C.S. Lui[2], Wei Chen[3]
[1]Carnegie Mellon University, [2]The Chinese University of Hong Kong, [3]Microsoft Research
[1]{jzuo, cjoewong}@andrew.cmu.edu, [2]{liuxt, cslui}@cse.cuhk.edu.hk, [3]weic@microsoft.com

Abstract

Online influence maximization has attracted much attention as a way to maximize influence spread through a social network while learning the values of unknown network parameters. Most previous works focus on single-item diffusion. In this paper, we introduce a new Online Competitive Influence Maximization (OCIM) problem, where two competing items (e.g., products, news stories) propagate in the same network and influence probabilities on edges are unknown. We adapt the combinatorial multi-armed bandit (CMAB) framework for the OCIM problem, but unlike the non-competitive setting, the important monotonicity property (influence spread increases when influence probabilities on edges increase) no longer holds due to the competitive nature of propagation, which brings a significant new challenge to the problem. We prove that the Triggering Probability Modulated (TPM) condition for CMAB still holds, and then utilize the property of competitive diffusion to introduce a new offline oracle, and discuss how to implement this new oracle in various cases. We propose an OCIM-OIFU algorithm with such an oracle that achieves logarithmic regret. We also design an OCIM-ETC algorithm that has worse regret bound but requires less feedback and easier offline computation. Our experimental evaluations demonstrate the effectiveness of our algorithms.

1 Introduction

Influence maximization, motivated by viral marketing applications, has been extensively studied since Kempe et al. [1] formally defined it as a stochastic optimization problem: given a social network G and a budget k, how to select a set of k nodes in G such that the expected number of final activated nodes under a given diffusion model is maximized. They proposed the well-known Independent Cascade (IC) and Linear Threshold (LT) diffusion models, and gave a greedy algorithm that outputs a $(1 - 1/e - \epsilon)$-approximate solution for any $\epsilon > 0$. However, they only considered a single item (e.g., product, idea) propagating in the network. In reality, many different items could propagate concurrently in the same network, interfering with each other and leading to competition during propagation. Several competitive diffusion models [2, 3, 4, 5, 6] have been proposed for this setting. We use a Competitive Independent Cascade (CIC) model [7], which extends the classical IC model to multi-item influence diffusion. We consider the competitive influence maximization problem between two items from the "follower's perspective": given the seed nodes of the competitor's item, the follower's item chooses a set of nodes so as to maximize the expected number of nodes activated by the follower's item, referred to as the influence spread of the item.

We refer to the above problem as the "offline" competitive influence maximization problem, since the influence probabilities on edges, i.e., the probabilities of an item's propagation along edges, are known in advance. It can be solved by a greedy algorithm due to submodularitiy [7]. However, in many real-world applications, the influence probabilities on edges are unknown. We study the competitive influence maximization in this setting, and call it the Online Competitive Influence Maximization (OCIM) problem. In OCIM, the influence probabilities on edges need to be learned through repeated influence maximization trials: in each round, given the seed nodes of the competitor,

Preprint. Under review.

we (i) choose k seed nodes; (ii) observe the resulting diffusion that follows the CIC model to update our knowledge of the edge probabilities; and (iii) obtain a reward, which is the total number of nodes activated by our item. Our goal is to choose the seed nodes in each round based on previous observations so as to maximize the cumulative reward over all rounds.

Most previous studies on the online non-competitive influence maximization problem use a combinatorial multi-armed bandit (CMAB) framework [8, 9, 10], an extension of the classical multi-armed bandit problem that captures the tradeoff between exploration and exploitation [11] in sequential decision making. In CMAB, a player chooses a combinatorial action to play in each round and, observes a set of arms triggered by this action and receives a reward. The player aims to maximize her cumulative reward over multiple rounds, navigating a tradeoff between exploring unknown actions/arms and exploiting the best known action. CMAB algorithms must also deal with an exponential number of possible combinatorial actions, which makes exploring all actions infeasible. CMAB has been applied to non-competitive online influence maximization [8, 9, 10]. To the best of our knowledge, we are the first to study the OCIM problem.

In this paper, we adapt the CMAB framework for the OCIM problem. However, a new challenge arises because the key monotonicity property (influence spread increases when influence probabilities on edges increase) no longer holds due to the competitive nature of propagation, and thus upper confidence bound (UCB) based algorithms [8, 12] cannot be directly applied to OCIM. To meet this challenge, we introduce a new offline oracle that takes ranges of edge probabilities and the competitor's seed set as inputs, and outputs an approximately optimal seed set that maximizes the influence spread of our item. We discuss the algorithms implementing this oracle in various cases. We then prove that the Triggering Probability Modulated (TPM) bounded smoothness condition in [9] still holds for the OCIM problem. The proof is much more involved for OCIM due to the lack of monotonicity property. Based on the TPM condition and the new offline oracle, we follow the principle of Optimism In the Face of Uncertainty (OIFU) to propose the OCIM-OIFU algorithm that achieves the same regret bound for OCIM as that achieved by [9] for the non-competitive setting. We also design an Explore-Then-Commit (OCIM-ETC) algorithm that does not need the new offline oracle and requires fewer observations in each round, but leads to a worse regret bound than OCIM-OIFU, showing the tradeoff between the regret bound and feedback level in the OCIM problem. Our experimental results demonstrate the effectiveness of our proposed algorithms. Due to the space constraint, we move all proofs to the supplementary material.

Related work. Kempe et al. formally defined the influence maximization problem in the seminal work [1]. Since then, the problem has been extensively studied [7, 13]. Borgs et al. [14] presented a breakthrough approximation algorithm that runs in near-linear time, which was improved by a series of algorithms [15, 16, 17] that run in $O((k+l)(m+n)\log n/\epsilon^2)$ expected time for a graph with n nodes and m edges, and returns a $(1-1/e-\epsilon)$-approximate solution with probability at least $1-n^l$.

A number of studies [2, 3, 4, 5, 6] addressed competitive influence maximization problems where multiple competing sources propagate in the same network. Carnes et al. [2] proposed the distance-based and wave propagation models, and considered the influence maximization problem from the follower's perspective. Bharathi et al. [3] extended the single source IC model to the competitive setting and gave a $(1-1/e)$-approximation algorithm for computing the best response to an opponent's strategy. Extensions of the IC and LT models to multi-item diffusion are summarized in [7].

When the influence probabilities of edges are unknown, Chen et al. [8, 18] studied the non-competitive online influence maximization problem under the IC model and edge-level feedback. They proposed a CUCB algorithm and provided its distribution-dependent and -independent regret bounds. Wang & Chen [9] introduced a triggering probability modulated (TPM) bounded smoothness condition to remove an undesired factor in the regret bound in [8]. Wen et al. [10] and Wu et al. [19] further considered edge probabilities represented by latent feature vectors, which is useful for large-scale settings. Vaswani et al. [20] considered a surrogate function as an approximation to the original influence maximization objective, but this heuristic function does not provide a theoretical guarantee.

2 OCIM Formulation

In this section we present the formulation of the Online Competitive Influence Maximization (OCIM) problem. We first introduce the traditional competitive influence maximization problem, and then discuss its online extension where the edge probabilities are initially unknown, so that they need to be learned through repeated runs of the influence maximization task.

We consider a Competitive Independent Cascade (CIC) model, which is an extension of the classical Independent Cascade (IC) model to the multi-item influence diffusion. A network is modeled as a directed graph $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ edges. Every edge $(u, v) \in E$ is associated with a probability $p(u, v)$. There are two types of items, A and B, trying to propagate in G from their own seed sets S_A and S_B. The influence propagation runs as follows: nodes in S_A (resp. S_B) are activated by A (resp. B) at step 0; at each step $s \geq 1$, a node u activated by A (resp. B) in step $s-1$ tries to activate each of its inactive out-neighbors v to be A (resp. B) with an independent probability $p(u, v)$ that is the same for A and B (i.e., we consider a homogeneous CIC model). The homogeneity assumption is reasonable in that typically A and B are two items of the same category (thus competing) so they are likely to have similar propagation characteristics. If two in-neighbors of v activated by A and B respectively both successfully activate v at step s, then a tie-breaking rule is applied at v to determine the final adoption. In this paper, we consider two dominance tie-breaking rules: $A > B$, which means v will always adopt A in a competition, and $B > A$, which means v will always adopt B in a competition. The same tie-breaking rule also applies to the case when a node u is selected both as an A-seed and a B-seed. The dominance tie-breaking rule reflects scenarios such as a novel technology dominating the old technology, or negative information dominating positive information, which is reasonable in practice. The process stops when no nodes activated at a step s have inactive out-neighbors.

We consider the follower's perspective in the optimization task: let A be the follower and B be the competitor. Then given S_B, our goal is to choose at most k seed nodes in G as S_A to maximize the influence spread of A, denoted as $\sigma_A(S_A, S_B)$, which is the expected number of nodes activated by A after the propagation ends. According to the result in [7], the above optimization task under the homogeneous CIC model with the dominance tie-breaking rule has the monotone and submodular property, and thus can be solved by a greedy algorithm with $1 - 1/e$ approximation.

In the online competitive influence maximization (OCIM) problem, the edge probabilities $p(u, v)$'s are unknown and need to be learned: in each round t, given S_B, we choose k seed nodes as $S_A^{(t)}$, observe the whole propagation of A and B that follows the CIC model, and obtain the reward, which is the number of nodes finally activated by A in this round. The propagation feedback observed is used to update the estimates on edge probabilities $p(u, v)$'s, so that we can achieve better influence maximization results in subsequent rounds. Our goal is to accumulate as much reward as possible through this repeated process over multiple rounds. This OCIM model fits the framework of combinatorial multi-armed bandit with probabilistic arm triggering (CMAB-T) [9]: the set of edges E is the set of (base) arms $[m] = \{0, 1, ..., m\}$, and their outcomes follow m independent Bernoulli distributions with expectation $\mu_e = p(u, v)$ for all $e = (u, v) \in E$. We denote the independent samples of arms in round t as $X^{(t)} = (X_1^{(t)}, \ldots, X_m^{(t)}) \in \{0, 1\}^m$, where $X_i^{(t)} = 1$ means the i-th edge is on (or live) and $X_i^{(t)} = 0$ means the i-th edge is off (or blocked) in round t, and thus $X^{(t)}$ corresponds to the *live-edge graph* [1] in round t. The set of actions is the set of all subsets of nodes S_A with at most k nodes. We define the triggered arm set τ_t as the set of edges reached by the propagation from both S_A and S_B. We define $S^{(t)} := \{S_A^{(t)}, S_B\}$ and call it the joint action at round t. Thus, τ_t is the set of edges (u, v) where u can be reached from S_t by passing through only edges $e \in E$ with $X_e^{(t)} = 1$. Notice that although the competition between A and B could occur in the propagation, τ_t is not affected as long as $S_A^{(t)} \cup S_B$ remains the same. We denote the obtained reward at round t as $R(S^{(t)}, X^{(t)})$, which is the number of nodes finally activated by A. The expected reward $\mathbb{E}[R(S^{(t)}, X^{(t)})]$ is a function of the joint action $S^{(t)}$ and the expectation vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$, and is denoted as $r_{S^{(t)}}(\boldsymbol{\mu})$.

The performance of a learning algorithm $\mathcal{A}$ is measured by its expected regret, which is the difference in expected cumulative reward between always playing the best action and playing actions selected by algorithm $\mathcal{A}$. Let $\text{opt}(\boldsymbol{\mu}) = \sup_{|S_A| \leq k, S = \{S_A, S_B\}} r_S(\boldsymbol{\mu})$ denote the expected reward of the optimal action in one round. Since the offline influence maximization under the CIC model is NP-hard [1, 7], we assume that there exists an offline (α, β)-approximation oracle $\mathcal{O}$, which takes S_B and $\boldsymbol{\mu}$ as inputs and outputs a joint action $S^{\mathcal{O}} = \{S_A^{\mathcal{O}}, S_B\}$ such that $\Pr\{r_{S^{\mathcal{O}}}(\boldsymbol{\mu}) \geq \alpha \cdot \text{opt}(\boldsymbol{\mu})\} \geq \beta$, where α is the approximation ratio and β is the success probability. Instead of comparing with the exact optimal reward, we take the $\alpha\beta$ fraction of it and use the following (α, β)-approximation regret for T rounds:

$$Reg^{\mathcal{A}}_{\boldsymbol{\mu},\alpha,\beta}(T) = T \cdot \alpha \cdot \beta \cdot \text{opt}(\boldsymbol{\mu}) - \sum_{t=1}^{T} r_{S^{\mathcal{A},(t)}}(\boldsymbol{\mu}), \tag{1}$$

where $S^{\mathcal{A},(t)} := \{S_A^{\mathcal{A},(t)}, S_B\}$ is the joint action chosen by algorithm $\mathcal{A}$ in round t. Here S_B is given by the environment, and $S_A^{\mathcal{A},(t)}$ is the seed set of item A chosen by algorithm $\mathcal{A}$.

3 Online Algorithms

3.1 Non-monotonicity

The monotonicity condition given in [8, 9] could be stated as follows in the context of OCIM: for any joint action $S = \{S_A, S_B\}$, for any two expectation vectors $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$ and $\boldsymbol{\mu}' = (\mu'_1, \ldots, \mu'_m)$, we have $r_S(\boldsymbol{\mu}) \leq r_S(\boldsymbol{\mu}')$ if $\mu_i \leq \mu'_i$ for all $i \in [m]$. Figure 1 shows a simple example of OCIM that does not satisfy the monotonicity condition. The left and right nodes are the seed nodes of A and B; the numbers below edges are influence probabilities. It is easy to calculate that $r_S(\boldsymbol{\mu}) = \mu_1(1-\mu_2)+2$, for both $A > B$ and $A < B$ tie-breaking rules, so if we increase μ_2, $r_S(\boldsymbol{\mu})$ will decrease, which is contrary to monotonicity. In general, for every edge (u, v), depending on the positions of A-seeds and B-seeds, increasing the influence probability of (u, v) may benefit the propagation of A or may benefit the propagation of B and thus impair the propagation of A. Thus, the influence spread of A has intricate connections with influence probabilities on edges (see Sec. 4.2 for more discussions).

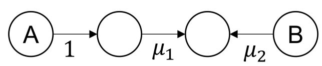


Figure 1: Example of non-monotonicity in OCIM

The lack of monotonicity posts a significant challenge to the OCIM problem. First, we cannot directly use the upper confidence bound type of algorithms [8], since they will not provide an optimistic solution at each round to bound the regret. However, it is still possible to design bandit algorithms following the principle of Optimism In the Face of Uncertainty (OIFU), by introducing a new offline oracle that jointly optimizes for both seed set S^* and the optimal influence probability vector $\boldsymbol{\mu}^*$, where each dimension of $\boldsymbol{\mu}^*$, μ_i^*, is searched within a confidence interval c_i, for all $i \in E$. This is formulated below:

$$\begin{aligned} \underset{S,\,\boldsymbol{\mu}}{\text{maximize}} \quad & r_S(\boldsymbol{\mu}) \\ \text{subject to} \quad & |S_A| \leq k, S = \{S_A, S_B\} \\ & \mu_i \in c_i,\ i = 1, \ldots, m. \end{aligned} \tag{2}$$

We define a new offline (α, β)-approximation oracle $\widetilde{\mathcal{O}}$ that solves the optimization problem in Eq.(2). $\widetilde{\mathcal{O}}$ takes S_B and c_i's as inputs and outputs $\boldsymbol{\mu}^{\widetilde{\mathcal{O}}}$ and action $S^{\widetilde{\mathcal{O}}} = \{S_A^{\widetilde{\mathcal{O}}}, S_B\}$, such that $\Pr\{r_{S^{\widetilde{\mathcal{O}}}}(\boldsymbol{\mu}^{\widetilde{\mathcal{O}}}) \geq \alpha \cdot r_{S^*}(\boldsymbol{\mu}^*)\} \geq \beta$, where $(S^*, \boldsymbol{\mu}^*)$ is the optimal solution for Eq.(2). We defer the design of such an oracle to Sec. 4.

3.2 Triggering Probability Modulated (TPM) Bounded Smoothness

The lack of monotonicity further complicates the analysis of the Triggering Probability Modulated (TPM) condition [9], which is crucial in tightening the regret bound. We use $p_i^S(\boldsymbol{\mu})$ to denote the probability that the joint action $S = \{S_A, S_B\}$ triggers arm i when the expectation vector is $\boldsymbol{\mu}$. The TPM condition in OCIM is given below.

Condition 1. *(1-Norm TPM bounded smoothness). We say that an OCIM problem instance satisfies 1-norm TPM bounded smoothness, if there exists $C \in R^+$ (referred as the bounded smoothness constant) such that, for any two expectation vectors $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, and any joint action $S = \{S_A, S_B\}$, we have $|r_S(\boldsymbol{\mu}) - r_S(\boldsymbol{\mu}')| \leq C \sum_{i\in[m]} p_i^S(\boldsymbol{\mu})|\mu_i - \mu'_i|$.*

Fortunately, with a more intricate analysis, we are able to show the following TPM condition:

Theorem 3.1. *Under both $A > B$ and $B > A$ tie-breaking rules, OCIM instances satisfy the 1-norm TPM bounded smoothness condition with constant $C = \tilde{C}$, where $\tilde{C}$ is the maximum number of nodes that any one node can reach in graph G.*

Algorithm 1 OCIM-OIFU with offline oracle $\widetilde{\mathcal{O}}$

Input: m, Oracle $\widetilde{\mathcal{O}}$.
For each arm $i \in [m]$, $T_i \leftarrow 0$. {maintain the total number of times arm i is played so far.}
For each arm $i \in [m]$, $\hat{\mu}_i \leftarrow 1$. {maintain the empirical mean of X_i.}
for $t = 1, 2, 3, \ldots$ **do**
  For each arm $i \in [m], \rho_i \leftarrow \sqrt{\frac{3 \ln t}{2T_i}}$. {the confidence radius, $\mu_i = +\infty$ if $T_i = 0$.}
  For each arm $i \in [m], c_i \leftarrow \left[(\hat{\mu}_i - \rho_i)^{0+}, (\hat{\mu}_i + \rho_i)^{1-}\right]$. {the estimated ranges of μ_i.}
  $S \leftarrow \text{Oracle}_{\widetilde{\mathcal{O}}}(S_B, c_1, c_2, \ldots, c_m)$.
  Play joint action S, which triggers a set $\tau \subseteq [m]$ of base arms with feedback $X_i^{(t)}$'s, $i \in \tau$.
  For every $i \in \tau$ update T_i and $\hat{\mu}_i$: $T_i = T_i + 1, \hat{\mu}_i = \hat{\mu}_i + (X_i^{(t)} - \hat{\mu}_i)/T_i$.
end for

The proof of the above theorem is one of the key technical contributions of the paper. In the non-competitive setting, an edge coupling method could give a relatively simple proof for the TPM condition.[1] The idea of edge coupling is for every edge $e \in E$, we sample a real number $X_e \in [0, 1]$ uniformly at random, and determine e to be live under $\boldsymbol{\mu}$ if $X_e \le \mu_e$ and blocked if $X_e > \mu_e$, and same for $\boldsymbol{\mu}'$. This couples the live-edge graphs L and L' under $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ respectively. In the non-competitive setting, due to the monotonicity property, we only need to consider the TPM condition when $\boldsymbol{\mu} \ge \boldsymbol{\mu}'$ (coordinate-wise), and this implies that L' is a subgraph of L, which significantly simplifies the analysis. However, in the competitive setting, monotonicity does not hold, and we have to show the TPM condition for every pair of $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, and thus L and L' no longer has the subgraph relationship. In this case, we have to show that for every coupling L and L', for every $v \in V$ that is activated by A in L but not activated by A in L', it is because either (a) some edge $e = (u, w)$ is live in L but blocked in L' while u is A-activated (or equivalently e is A-triggered); or (b) some edge e is live in L' but blocked in L while e is B-triggered. The case (b) is due to the possibility of B blocking A's propagation, a unique scenario in OCIM. The above claim needs a nontrivial inductive proof, and then its correctness ensures the TPM condition.

3.3 OCIM-OIFU Algorithm

With the offline oracle $\widetilde{\mathcal{O}}$, we propose an algorithm following the principle of Optimism In the Face of Uncertainty (OIFU), named Online Competitive Influence Maximization-OIFU (OCIM-OIFU). The algorithm maintains the empirical mean $\hat{\mu}_i$ and confidence radius ρ_i for each edge probability. It uses the lower and upper confidence bounds to determine the range of μ_i: $c_i = \left[(\hat{\mu}_i - \rho_i)^{0+}, (\hat{\mu}_i + \rho_i)^{1-}\right]$, where we use $(x)^{0+}$ and $(x)^{1-}$ to denote $\max\{x, 0\}$ and $\min\{x, 1\}$ for any real number x. It feeds S_B and all current c_i's into the offline oracle $\widetilde{\mathcal{O}}$ to obtain the joint action $S = \{S_A, S_B\}$ to play at each round. The confidence radius ρ_i is large if arm i is not triggered often, which leads to wider search space c_i to find the optimistic estimate of μ_i.

Let $\mathcal{S} = \{S \mid S = \{S_A, S_B\}, |S_A| \le k\}$ be the feasible set of joint actions. We define the reward gap $\Delta_S = \max(0, \alpha \cdot \text{opt}(\boldsymbol{\mu}) - r_S(\boldsymbol{\mu}))$ for all joint actions $S \in \mathcal{S}$. For each arm i, we define $\Delta^i_{\min} = \inf_{S \in \mathcal{S}: p_i^S(\boldsymbol{\mu}) > 0, \Delta_S > 0} \Delta_S$, $\Delta^i_{\max} = \sup_{S \in \mathcal{S}: p_i^S(\boldsymbol{\mu}) > 0, \Delta_S > 0} \Delta_S$. If there is no action S such that $p_i^S(\boldsymbol{\mu}) > 0$ and $\Delta_S > 0$, we define $\Delta^i_{\min} = +\infty$, $\Delta^i_{\max} = 0$. We define $\Delta_{\min} = \min_{i \in [m]} \Delta^i_{\min}$ and $\Delta_{\max} = \max_{i \in [m]} \Delta^i_{\max}$. Let $\widetilde{S} = \{i \in [m] \mid p_i^S(\boldsymbol{\mu}) > 0\}$ be the set of arms that can be triggered by S. We define $K = \max_{S \in \mathcal{S}} |\widetilde{S}|$ as the largest number of arms could be triggered by a feasible joint action. We use $\lceil x \rceil_0$ to denote $\max\{\lceil x \rceil, 0\}$. We provide the regret bound of the CIM-OIFU algorithm.

Theorem 3.2. *For the OCIM-OIFU algorithm on an OCIM problem satisfying 1-norm TPM bounded smoothness (Condition 1) with bounded smoothness constant C, (1) if $\Delta_{\min} > 0$, we have a distribution-dependent bound*

$$Reg_{\boldsymbol{\mu},\alpha,\beta}(T) \le \sum_{i \in [m]} \frac{576 C^2 K \ln T}{\Delta^i_{\min}} + \sum_{i \in [m]} \left(\left\lceil \log_2 \frac{2CK}{\Delta^i_{\min}} \right\rceil_0 + 2 \right) \cdot \frac{\pi^2}{6} \cdot \Delta_{\max} + 4Cm, \tag{3}$$

[1] The original proofs [9, 10] occupy several pages, but we are aware of a shorter proof based on edge coupling.

and (2) we have a distribution-independent bound

$$Reg_{\boldsymbol{\mu},\alpha,\beta}(T) \leq 12C\sqrt{mKT\ln T} + \left(\left\lceil \log_2 \frac{T}{18\ln T} \right\rceil_0 + 2\right) \cdot m \cdot \frac{\pi^2}{6} \cdot \Delta_{\max} + 2Cm. \tag{4}$$

The above regret bounds has the typical form of $O(\sum \frac{1}{\Delta^i_{\min}} \cdot \log T)$ and $\sqrt{T \log T}$, indicating that it is tight on the important time horizon T and gap parameters $\Delta^i_{\min}$'s. In fact, they are the same as in [9], despite that they are for the OCIM problem that does not enjoy monotonicity. This owes to our non-trivial TPM condition analysis (Theorem 3.1) that shows the same condition as in [9] without the monotonicity in the OCIM setting.

3.4 OCIM-ETC Algorithm

In this section, we propose an Online Competitive Influence Maximization Explore-Then-Commit (OCIM-ETC) algorithm. It has two advantages: first, it does not need the new offline oracle discussed in Sec. 3.1; second, it requires less observations of edges than OCIM-OIFU: instead of the observations of all triggered edges, i.e., τ, OCIM-ETC only needs the observations of all direct out-edges of seed nodes. Like other ETC-type algorithms [21], OCIM-ETC divides total T rounds into two phases: exploration phase and exploitation phase. In exploration phase, our goal is to choose each node as the seed node of A for N times, Notice that in each round we can choose k nodes as S_A, so the exploration phase totally takes $\lceil nN/k \rceil$ rounds. In each round, we take k nodes that have not been chosen for N times as S_A and denote their direct out-edges as τ_{direct}; we observe the outcome of edge i for all $i \in \tau_{\text{direct}}$ and update its empirical mean $\hat{\mu}_i$. In exploitation phase, we take $\hat{\mu}_i$'s as inputs of the offline oracle $\mathcal{O}$ mentioned in Sec. 2, get the output action $S^{\mathcal{O}}$ then keep playing it for all remaining rounds. We give its regret bound.

Algorithm 2 OCIM-ETC with offline oracle $\mathcal{O}$

Input: m, N, T, Oracle $\mathcal{O}$.
For each arm i, $T_i \leftarrow 0$. {maintain the total number of times arm i is played so far.}
For each arm i, $\hat{\mu}_i \leftarrow 0$. {maintain the empirical mean of X_i.}
Exploration phase:
for $t = 1, 2, 3, \ldots, \lceil nN/k \rceil$ **do**
    Take k nodes that have not been chosen for N times as S_A.
    Observe the feedback $X_i^{(t)}$ for each direct out-edge of S_A, $i \in \tau_{\text{direct}}$.
    For each arm $i \in \tau_{\text{direct}}$ update T_i and $\hat{\mu}_i$: $T_i = T_i + 1$, $\hat{\mu}_i = \hat{\mu}_i + (X_i^{(t)} - \hat{\mu}_i)/T_i$.
end for
Exploitation phase:
$S^{\mathcal{O}} \leftarrow \text{Oracle}_{\mathcal{O}}(S_B, \hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_m)$.
for $t = \lceil nN/k \rceil + 1, \ldots, T$ **do**
    Play joint action $S^{\mathcal{O}}$.
end for

Theorem 3.3. *For the OCIM-ETC algorithm on an OCIM problem satisfying 1-norm TPM bounded smoothness (Condition 1) with bounded smoothness constant C, (1) if $\Delta_{\min} > 0$, when $N = \max\left\{1, \frac{2C^2m^2}{\Delta^2_{\min}} \ln(\frac{kT\Delta^2_{\min}}{C^3 m})\right\}$, we have a distribution-dependent bound*

$$Reg_{\boldsymbol{\mu},\alpha,\beta}(T) \leq \frac{2C^2m^2n\Delta_{\max}}{k\Delta^2_{\min}} \left(\max\left\{\ln\left(\frac{kT\Delta^2_{\min}}{C^2mn}\right), 0\right\} + 1\right) + \frac{n}{k}\Delta_{\max}, \tag{5}$$

and (2) when $N = (Cmk)^{\frac{2}{3}} n^{-\frac{4}{3}} T^{\frac{2}{3}} (\ln T)^{\frac{1}{3}}$, we have a distribution-independent bound

$$Reg_{\boldsymbol{\mu},\alpha,\beta}(T) \leq O((Cmn)^{\frac{2}{3}} k^{-\frac{1}{3}} T^{\frac{2}{3}} (\ln T)^{\frac{1}{3}}). \tag{6}$$

Although this regret bound is worse than that of the OCIM-OIFU algorithm in Theorem 3.2, as mentioned before, OCIM-ETC requires less feedback and easier offline computation, so it shows the trade-off between regret bound and feedback/computation in the OCIM problem. Notice that for tie-breaking rule $A < B$, in each round, we also need the observations of direct out-edges of S_B, since it is impossible to observe these edges by choosing nodes in S_B as the seed nodes of A.

4 Offline Oracle

In Sec. 3.1, in order to remove the requirement of monotonicity, we introduce a new offline optimization problem that maximizes $r_S(\boldsymbol{\mu})$ over S and $\boldsymbol{\mu}$ at the same time. As mentioned before, the original offline problem, i.e., maximizing $r_S(\boldsymbol{\mu})$ over S when fixing $\boldsymbol{\mu}$, can be solved by several algorithms (e.g., TCIM [6], CELF++ [22]) based on submodularity of $r_S(\boldsymbol{\mu})$ over S. A straightforward attempt on the new oracle is to show the submodularity of $g(S) = \max_{\boldsymbol{\mu}} r_S(\boldsymbol{\mu})$ over S, and then do greedy on g while for each S finding the optimal $\boldsymbol{\mu}$ that maximizes $r_S(\boldsymbol{\mu})$. Unfortunately, we show that $g(S)$ is not submodular, and when given S, even finding the optimal μ_i for one edge i while fixing the values of all others is #P-hard (see supplementary material). This indicates the challenge of implementing the offline oracle. In this section, we first show how to implement the oracle for bipartite graphs, which models the competitive probabilistic maximum coverage problem with applications in online advertising, and then discuss an important property in general networks and show how to design algorithms for other classes of graphs or heuristics for general graphs based on this property.

4.1 Bipartite Graph

We consider a weighted bipartite graph $G = (L, R, E)$ where each edge (u, v) is associated with a probability $p(u, v)$. Given the competitor's seed set $S_B \subseteq L$, we need to choose k nodes from L as S_A that maximizes the expected number of nodes activated by A in R, where a node $v \in R$ can be activated by a node $u \in L$ with an independent probability of $p(u, v)$. As mentioned before, if A and B are attempting to activate a node in L at the same time, the result will depend on the tie-breaking rule. If all edge probabilities are fixed, i.e., $\boldsymbol{\mu}$ is fixed, $r_S(\boldsymbol{\mu})$ is still submodular over S_A, so we can use a greedy algorithm as a $(1 - 1/e, 1)$-approximation oracle $\mathcal{O}_{\text{greedy}}$. Based on it, let us discuss the new offline optimization problem in Eq.(2) under our two tie-breaking rules: (1) $A > B$: since B will never influence nodes in R earlier than A in bipartite graphs, and A will always win the competition, from A's perspective, we can ignore S_B to choose S_A. In this case, all edge probabilities should take the maximum values: for all $i \in E$, μ_i equals to the upper bound of c_i, and we then use the oracle $\mathcal{O}_{\text{greedy}}$ to find S_A. (2) $B > A$: since A will never influence nodes in R earlier than B in bipartite graphs, and B will always win the competition, all out-edges of S_B, denoted as E_{S_B}, should take the minimum probabilities to maximize the influence spread of A. All the other edges in $E \backslash E_{S_B}$ should take the maximum probabilities. Formally, for all $i \in E_{S_B}$, μ_i equals to the lower bound of c_i; for all $i \in E \backslash E_{S_B}$, μ_i equals to the upper bound of c_i. We then use the oracle $\mathcal{O}_{\text{greedy}}$ to find S_A. To sum up, in bipartite graphs, $r_S(\boldsymbol{\mu})$ is optimized by pre-determining $\boldsymbol{\mu}$ based on the tie-breaking rule, and then using the greedy algorithm to get a $(1 - 1/e, 1)$-approximation solution. Since the time complexity of influence computation in the bipartite graph is $O(m)$, the time complexity of the offline algorithm is equal to that of the greedy algorithm, $O(k^2 m)$.

4.2 General Graph

The competitive propagation in the general graph is much more complicated, so it is hard to predetermine all edge probabilities as in the bipartite graph case. However, we have a key observation:

Lemma 4.1. *When fixing the seed set $S = \{S_A, S_B\}$, reward $r_S(\boldsymbol{\mu})$ has a linear relationship with each μ_i (when other μ_j's with $j \neq i$ are fixed). This implies that the optimal solution for the optimization problem in Eq.(2) must occur at the boundaries of the intervals c_i's.*

Lemma 4.1 implies that for any edge e not reachable from B seeds, it is safe to always take its upper bound value since it can only helps the propagation of A. This further suggests that if we only have a small number (e.g. $\log m$) of edges reachable from B, then we can afford enumerating all the boundary value combinations of these edges. For each such boundary setting $\boldsymbol{\mu}$, we can use the IMM algorithm [23] to design a $(1 - 1/e - \epsilon, 1 - n^{-l})$-approximation oracle $\mathcal{O}_{\text{IMM}}$ with time complexity $T_{\text{IMM}} = O((k + l)(m + n) \log n/\epsilon^2)$ due to its submodularity [6, 7]. In the supplementary material, we discuss concrete graphs such as trees that satisfy the above condition.

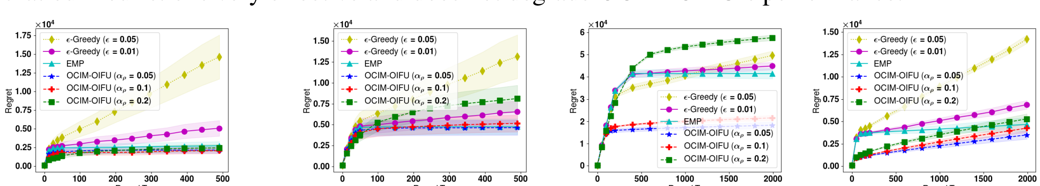
For general graphs, designing an efficient approximation algorithm for the offline problem in Eq. (2) remains a challenging open problem, due to the joint optimization over S and $\boldsymbol{\mu}$ and the complicated function form of $r_S(\boldsymbol{\mu})$. Nevertheless, heuristic algorithms are still possible. In the experiment section, we employee the following heuristic with the $B > A$ tie-breaking rule: For all outgoing edges from B seeds, we set their influence probabilities to their lower bound values, while for the rest, we set them to their upper bound values. This setting guarantees that the first-level edges from the seeds are always set correctly, no matter how we select A seeds. They do not guarantee the correctness of second or higher level edge settings in the cascade, but the impact of those edges to influence spread decays significantly, so the above choice is reasonable as a heuristic.

5 Experiments

Datasets and settings. To validate our theoretical findings, we conduct experiments on two real-world datasets. First, we use the Yahoo! Search Marketing Advertiser Bidding Data (denoted as Yahoo-Ad) [24], which contains a bipartite graph between $1,000$ keywords and $10,475$ advertisers. Each edge represents a bid on a keyword from an advertiser, and our goal is to select a set of keywords that attracts the most advertisers. We then consider the DM network [25] with 679 nodes representing researchers and $3,374$ edges representing collaborations between them. We simulate a researcher asking others (i.e., S_A) to spread her ideas while her competitor (i.e., S_B) promotes a competing proposal. We model non-strategic and strategic competitors by selecting the seed set S_B uniformly at random (denoted as RD) or by running the non-competitive influence maximization algorithm (denoted as IM). Here we focus on the $B > A$ tie-breaking rule. We repeat each experiment 50 times and show the average regret with 95% confidence interval. We provide details of the datasets and other experiment parameters, and results with $A > B$ tie-breaking, in the supplementary material.

Algorithms for comparison. We use the approximation algorithm from Sec. 4.1 and the heuristic from Sec. 4.2 as OCIM-OIFU's offline oracle for Yahoo-Ad and DM respectively. We shrink the confidence interval by α_ρ, i.e., $\rho_i \leftarrow \alpha_\rho \sqrt{3 \ln t / 2T_i}$, to speed up the learning, though our theoretical regret bound requires $\alpha_\rho = 1$. We compare OCIM-OIFU to the ϵ-Greedy algorithm with parameter $\epsilon = 0$ (denoted as the EMP algorithm) and $\epsilon = 0.01, 0.05$, which inputs the empirical mean into the offline oracle with $1 - \epsilon$ probability and otherwise selects S_A uniformly at random. We show results for OCIM-ETC in the supplementary material as it requires many more rounds than OCIM-OIFU.

Experimental results. Figures 2a and 2b show the results for Yahoo-Ad. First, the regret of OCIM-OIFU grows sub-linearly w.r.t round T for all α_ρ, consistent with Theorem 3.2's regret bound. Second, we can observe that OCIM-OIFU is superior to EMP and ϵ-Greedy when $\alpha_\rho = 0.05$. When $\alpha_\rho = 0.2$, OCIM-OIFU may have larger regret due to too much exploration. The results on the DM dataset are shown in Figure 2c and 2d. Generally they are consistent with those on the Yahoo-Ad dataset: OCIM-OIFU also grows sub-linearly w.r.t round T. When $\alpha_\rho = 0.1, 0.05$, OCIM-OIFU has smaller regret than all baselines. Moreover, the difference between the OCIM-OIFU and baselines for the non-strategic competitor (RD) is more significant than that of strategic competitor's (IM's), because the non-strategic competitor is less "dominant" and OCIM-OIFU can carefully trade off exploration and exploitation to maximize A's influence. Note that we use a heuristic to replace the exhaustive process, which trades off the efficiency and the theoretical guarantee. The results show that our heuristic is very effective and does not degrade OCIM-OIFU's performance.



(a) Yahoo-Ad, RD (b) Yahoo-Ad, IM (c) DM, RD (d) DM, IM

Figure 2: Regrets of different algorithms for bipartite and general graphs.

6 Future Work

This paper initiates the first study on OCIM, and it opens up a number of future directions to explore. One is to design efficient offline approximation algorithms in the competitive setting when edge probabilities take a range of values. Another interesting direction is to study other partial feedback models, e.g. we only observe feedback from edges triggered by A but not B. A further direction is to look into distributed online learning, when competitors A and B both are learning from the propagation and deploying seeds for influence maximization.

References

[1] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

[2] Tim Carnes, Chandrashekhar Nagarajan, Stefan M Wild, and Anke Van Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *Proceedings of the ninth international conference on Electronic commerce*, pages 351–360, 2007.

[3] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *International workshop on web and internet economics*, pages 306–311, 2007.

[4] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 665–674, 2011.

[5] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the 2012 siam international conference on data mining*, pages 463–474, 2012.

[6] Yishi Lin and John CS Lui. Analyzing competitive influence maximization problems with partial information: An approximation algorithmic framework. *Performance Evaluation*, 91:187–204, 2015.

[7] Wei Chen, Laks V. S. Lakshmanan, and Carlos Castillo. *Information and Influence Propagation in Social Networks*. Morgan & Claypool Publishers, 2013.

[8] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research*, 17(1):1746–1778, 2016.

[9] Qinshi Wang and Wei Chen. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Advances in Neural Information Processing Systems*, pages 1161–1171, 2017.

[10] Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. Online influence maximization under independent cascade model with semi-bandit feedback. In *Advances in neural information processing systems*, pages 3022–3032, 2017.

[11] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[12] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[13] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. Influence maximization on social graphs: A survey. *IEEE Trans. Knowl. Data Eng.*, 30(10):1852–1872, 2018.

[14] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 946–957, 2014.

[15] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1539–1554, 2015.

[16] Hung T. Nguyen, My T. Thai, and Thang N. Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *SIGMOD*, pages 695–710, 2016.

[17] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. Online processing algorithms for influence maximization. In *SIGMOD*, pages 991–1005, 2018.

[18] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

[19] Qingyun Wu, Zhige Li, Huazheng Wang, Wei Chen, and Hongning Wang. Factorization bandits for online influence maximization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 636–646, 2019.

[20] Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks VS Lakshmanan, and Mark Schmidt. Model-independent online learning for influence maximization. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3530–3539, 2017.

[21] Aurélien Garivier, Tor Lattimore, and Emilie Kaufmann. On explore-then-commit strategies. In *Advances in Neural Information Processing Systems*, pages 784–792, 2016.

[22] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Celf++ optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, pages 47–48, 2011.

[23] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 75–86, 2014.

[24] Yahoo! search marketing advertiser bidding data. `https://webscope.sandbox.yahoo.com`.

[25] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, 2009.

Supplementary Material

A Proof of Theorem 3.1

Proof. Let $r_S^v(\boldsymbol{\mu})$ be the probability that node v is activated by A. From the proof of Lemma 2 in [9], we know that under the $A > B$ or $B > A$ tie-breaking rule, if for every node v and every $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ vectors we have

$$|r_S^v(\boldsymbol{\mu}) - r_S^v(\boldsymbol{\mu'})| \leq \sum_{e \in E} p_e^S(\boldsymbol{\mu}) \, |\mu_e - \mu_e'| \,, \tag{7}$$

then Theorem 3.1 is true. Notice that

$$r_S^v(\boldsymbol{\mu}) = \mathbb{E}_{L \sim \boldsymbol{\mu}} \left[\mathbb{1}\{v \text{ is activated by } A \text{ under } L\} \right] \tag{8}$$

$$r_S^v(\boldsymbol{\mu'}) = \mathbb{E}_{L' \sim \boldsymbol{\mu'}} \left[\mathbb{1}\{v \text{ is activated by } A \text{ under } L'\} \right] \tag{9}$$

where L and L' are two live-edge graphs sampled under $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, respectively. As mentioned in Sec. 3.2, we use an edge coupling method to compute the difference between $r_S^v(\boldsymbol{\mu})$ and $r_S^v(\boldsymbol{\mu'})$. Specifically, for each edge e, suppose we independently draw a uniform random variable X_e over $[0, 1]$, let

$$\begin{aligned}
&L(e) = L'(e) = 1, && \text{if } X_e \leq \min(\mu_e, \mu_e') \\
&L(e) = 1, L'(e) = 0, && \text{if } \mu_e' < X_e < \mu_e \\
&L(e) = 0, L'(e) = 1, && \text{if } \mu_e < X_e < \mu_e' \\
&L(e) = L'(e) = 0, && \text{if } X_e \geq \max(\mu_e, \mu_e')
\end{aligned}$$

where $L(e)$ represents the live/blocked state of edge e in live-edge graph L. Notice that L and L' does not have the subgraph relationship. Let $\boldsymbol{X} := (X_1, \ldots, X_e)$, the difference can be written as:

$$r_S^v(\boldsymbol{\mu}) - r_S^v(\boldsymbol{\mu'}) = \mathbb{E}_{\boldsymbol{X}}[f(S, L, v) - f(S, L', v)], \tag{10}$$

where $f(S, L, v) := \mathbb{1}\{v \text{ is activated by } A \text{ under } L\}$. Since $f(S, L, v) - f(S, L', v)$ could be 0, 1 or -1, we will discuss these cases separately.

1) $f(S, L, v) - f(S, L', v) = 0$.
This will not contribute to the expectation.

2) $f(S, L, v) - f(S, L', v) = 1$.
This will occur only if there exists a path such that: under L, v can be activated by A via this path, while under L', v cannot be activated by A via this path. We denote this event as $\mathcal{E}_1$. We will show that $\mathcal{E}_1$ occurs only if at least one of $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ occurs.

$\mathcal{E}_1^A$: There exists a path $u \to v_1 \to \cdots \to v_d = v$ such that:
1. u is activated by A under both L and L'
2. edge (u, v_1) is live under L but not L'

$\mathcal{E}_1^B$: There exists a path $u' \to v_1' \to \cdots \to v_{d'}' = v$ such that:
1. u' is activated by B under both L and L'
2. edge (u', v_1') is live under L' but not L

Lemma A.1. *$\mathcal{E}_1$ occurs only if at least one of $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ occurs.*

Proof. Let us first discuss the relationship between $\mathcal{E}_1$, $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$. For $\mathcal{E}_1$, if v can be activated by A under L but not L', it is because either: (a) some edge $e = (u, w)$ is live in L but blocked in L' while u is A-activated (or equivalently e is A-triggered); or (b) some edge e is live in L' but blocked in L while e is B-triggered. The former could be relaxed to $\mathcal{E}_1^A$, and the latter could be relaxed to $\mathcal{E}_1^B$. Notice that $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ are not mutually exclusive and we are interested in the upper bound of $\mathbb{P}\{\mathcal{E}_1\}$.

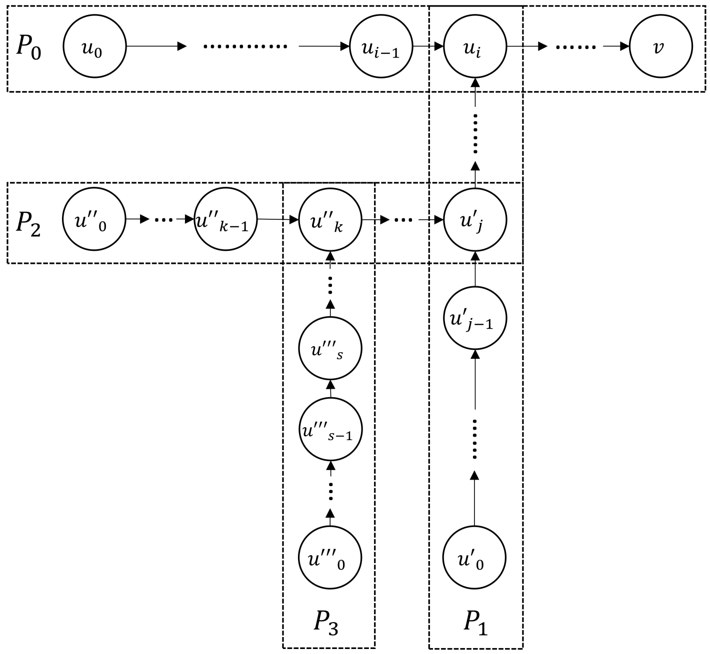


Figure 3: Path P_0, P_1, P_2 and P_3

Assuming $\mathcal{E}_1$ is true, consider the shortest path $P_0 := \{u_0 \to u_1 \to \cdots \to u_{l_0} = v\}$ from one seed node of A, u_0, to node v, such that under L node v is activated by A but under L' it is not. When $\mathcal{E}_1$ is true, there must exist a node that is not activated by A in P_0 under L'. We denote the first node from u_0 to v (i.e., closest to u_0) in P_0 that is not activated by A under L' as u_i.

Next, let us consider the live/blocked state of edge (u_{i-1}, u_i). We already know edge (u_{i-1}, u_i) is live under L. If edge (u_{i-1}, u_i) is blocked under L', since u_{i-1} is activated by A under both L and L', it directly becomes $\mathcal{E}_1^A$. Otherwise, if edge (u_{i-1}, u_i) is live under L', the reason that node u_i is not activated by A could only be that it is activated by B. In this case, there must exist a path $P_1 := \{u'_0 \to u'_1 \to \cdots \to u'_{l_1} = u_i\}$ from one seed node of B, u'_0, to node u_i, such that u_i is activated by B under L' but not L. This can only occur when there exists a node that is not activated by B in P_1 under L. We denote the first node from u'_0 to u'_{l_1} (i.e., closest to u'_0) in P_1 that is not activated by B under L as u'_j. Notice that when the tie-breaking rule is $A > B$, we have $l_1 < i \le l_0$ as B should arrive at u_i earlier than A; when the tie-breaking rule is $B > A$, we have $l_1 \le i \le l_0$ as B should arrive at u_i no later than A.

Then, let us consider the live/blocked state of edge (u'_{j-1}, u'_j). We already know edge (u'_{j-1}, u'_j) is live under L'. If edge (u'_{j-1}, u'_j) is blocked under L, since u'_{j-1} is activated by B under both L and L', it directly becomes $\mathcal{E}_1^B$. Otherwise, if edge (u'_{j-1}, u'_j) is live under L, the reason that node u'_j is not activated by B could only be that it is activated by A. It also means neither $\mathcal{E}_1^A$ nor $\mathcal{E}_1^B$ occurs so far. In this case, there must exist a path $P_2 := \{u''_0 \to u''_1 \to \cdots \to u''_{l_2} = u'_j\}$ from one seed node of A, u''_0, to node u'_j, such that u'_j is activated by A under L but not L'. This can only occur when there exists a node that is not activated by A in P_2 under L'. We denote the first node from u''_0 to u''_{l_2} (i.e., closest to u''_0) in P_2 that is not activated by A under L' as u''_k. Notice that when $A > B$, we have $l_2 \le j \le l_1 < l_0$ as A should arrive at u'_j no later than B; when $B > A$, we have $l_2 < j \le l_1 \le l_0$ as A should arrive at u'_j earlier than B.

Now let us consider the live/blocked state of edge (u''_{k-1}, u''_k). We already know edge (u''_{k-1}, u''_k) is live under L. If edge (u''_{k-1}, u''_k) is blocked under L', since u''_{k-1} is activated by A under both L and L', it directly becomes $\mathcal{E}_1^A$. Otherwise, if edge (u''_{k-1}, u''_k) is live under L', the reason that node u''_k is not activated by A could only be that it is activated by B. In this case, there must exist a path $P_3 := \{u'''_0 \to u'''_1 \to \cdots \to u'''_{l_3} = u''_k\}$ from one seed node of B, u'''_0, to node u''_k, such that u''_k is activated by B under L' but not L. This can only occur when there exists a node that is not activated by B in P_3 under L. We denote the first node from u_0''' to u_{l_3}''' (i.e., closest to u_0''') in P_3 that is not activated by B under L as u_s'''. Notice that when $A > B$, we have $l_3 < k \le l_2 \le l_1$ as B should arrive at u_k'' earlier than A; when $B > A$, we have $l_3 \le k \le l_2 < l_1$ as B should arrive at u_k'' no later than A.

Again, let us consider the live/blocked state of edge (u_{s-1}''', u_s'''). We already know edge (u_{s-1}''', u_s''') is live under L'. If edge (u_{s-1}''', u_s''') is blocked under L, since u_{s-1}''' is activated by B under both L and L', it directly becomes $\mathcal{E}_1^B$. Otherwise, if edge (u_{s-1}''', u_s''') is live under L, similar to the discussion above, we need to consider a new path P_4 with length l_4 and $l_4 < l_2$.

To sum up, if neither $\mathcal{E}_1^A$ nor $\mathcal{E}_1^B$ occurs in path P_0 and P_1, we need to check whether they could occur in a new path P_2 shorter than P_0, and P_3 shorter than P_1. As a result, we only need to check whether $\mathcal{E}_1^A$ or $\mathcal{E}_1^B$ occurs in the path with only one edge. In that case, $\mathcal{E}_1^A$ or $\mathcal{E}_1^B$ occurs for sure. Thus, by induction, we conclude that at least one of $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ occurs when considering any path with more than one edge, so $\mathcal{E}_1$ will occur only if at least one of $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ occurs. ☐

Now, let us consider the two events in $\mathcal{E}_1^A$ for a specific edge $e = (u, v_1)$. We find that the first event $\{u$ is activated by A under both L and $L'\}$, is independent of the second event $\{$edge e is live under L but not $L'\}$, since the live/blocked state of edge e does not affect the activation of its tail node u. Also, for edge $e = (u, v_1)$, the probability of these two events can be written as

$$\mathbb{P}\{u \text{ is activated by } A \text{ under } L \text{ and } L'\} = \mathbb{P}\{e \text{ is triggered by } A \text{ under } L \text{ and } L'\}, \tag{11}$$

$$\mathbb{P}\{e \text{ is live under } L \text{ but not } L'\} = \begin{cases} \mu_e - \mu_e' & \text{if } \mu_e > \mu_e' \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

As a result, we have:

$$\mathbb{P}\{\mathcal{E}_1^A\} \le \sum_{e:\, \mu_e > \mu_e'} \mathbb{P}\{e \text{ is triggered by } A \text{ under } L \text{ and } L'\}(\mu_e - \mu_e') \tag{13}$$

Since $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$ are symmetric, we also have:

$$\mathbb{P}\{\mathcal{E}_1^B\} \le \sum_{e:\, \mu_e' > \mu_e} \mathbb{P}\{e \text{ is triggered by } B \text{ under } L \text{ and } L'\}(\mu_e' - \mu_e) \tag{14}$$

Combining with Lemma. A.1, we have

$$\mathbb{P}\{\mathcal{E}_1\} \le \mathbb{P}\{\mathcal{E}_1^A\} + \mathbb{P}\{\mathcal{E}_1^B\} \tag{15}$$

3) $f(S, \boldsymbol{w_1}, v) - f(S, \boldsymbol{w_2}, v) = -1$.
Similar to the previous case, this will occur only if there exists a path such that: under L', v can be activated by A via this path, while under L, v cannot be activated by A via this path. We denote this event as $\mathcal{E}_{-1}$. We show that $\mathcal{E}_{-1}$ occurs only if at least one of $\mathcal{E}_{-1}^A$ and $\mathcal{E}_{-1}^B$ occurs.

$\mathcal{E}_{-1}^A$: There exists a path $u \to v_1 \to \cdots \to v_d = v$ such that:
1. u is activated by A under both L and L'
2. edge (u, v_1) is live under L' but not L

$\mathcal{E}_{-1}^B$: There exists a path $u' \to v_1' \to \cdots \to v_{d'}' = v$ such that:
1. u' is activated by B under both L and L'
2. edge (u', v_1') is live under L but not L'

Since they are symmetric with $\mathcal{E}_1^A$ and $\mathcal{E}_1^B$, following the same analysis, we can get

$$\mathbb{P}\{\mathcal{E}_{-1}^A\} \le \sum_{e:\, \mu_e' > \mu_e} \mathbb{P}\{e \text{ is triggered by } A \text{ under } L \text{ and } L'\}(\mu_e' - \mu_e) \tag{16}$$

$$\mathbb{P}\{\mathcal{E}_{-1}^B\} \le \sum_{e:\, \mu_e > \mu_e'} \mathbb{P}\{e \text{ is triggered by } B \text{ under } L \text{ and } L'\}(\mu_e - \mu_e') \tag{17}$$

$$\mathbb{P}\{\mathcal{E}_{-1}\} \le \mathbb{P}\{\mathcal{E}_{-1}^A\} + \mathbb{P}\{\mathcal{E}_{-1}^B\} \tag{18}$$

Combining all cases together, we have:

$$\begin{aligned}
|r_S^v(\boldsymbol{\mu}) - r_S^v(\boldsymbol{\mu}')| &= |\mathbb{E}_{\boldsymbol{X}}[f(S,L,v) - f(S,L',v)]| \\
&\leq |1 \cdot \mathbb{P}\{\mathcal{E}_1\} + (-1) \cdot \mathbb{P}\{\mathcal{E}_{-1}\}| \\
&\leq \left|1 \cdot \left(\mathbb{P}\{\mathcal{E}_1^A\} + \mathbb{P}\{\mathcal{E}_1^B\}\right) + (-1) \cdot \left(\mathbb{P}\{\mathcal{E}_{-1}^A\} + \mathbb{P}\{\mathcal{E}_{-1}^B\}\right)\right| \\
&\leq \sum_{e \in E} \mathbb{P}\{e \text{ is triggered by } A \text{ or } B \text{ under } L \text{ and } L'\} \, |\mu_e - \mu'_e| \,. \qquad (19)
\end{aligned}$$

The last inequality above is due to:

$$|\mathbb{P}\{\mathcal{E}_1^A\} - \mathbb{P}\{\mathcal{E}_{-1}^B\}| \leq \sum_{e:\, \mu_e > \mu'_e} \mathbb{P}\{e \text{ is triggered by } A \text{ or } B \text{ under } L \text{ and } L'\}|\mu_e - \mu'_e|$$

$$|\mathbb{P}\{\mathcal{E}_1^B\} - \mathbb{P}\{\mathcal{E}_{-1}^A\}| \leq \sum_{e:\, \mu'_e > \mu_e} \mathbb{P}\{e \text{ is triggered by } A \text{ or } B \text{ under } L \text{ and } L'\}|\mu_e - \mu'_e|$$

Notice that Eq.(19) could be relaxed to:

$$\begin{aligned}
|r_S^v(\boldsymbol{\mu}) - r_S^v(\boldsymbol{\mu}')| &\leq \sum_{e \in E} \mathbb{P}\{e \text{ is triggered by } A \text{ or } B \text{ under } L\} \, |\mu_e - \mu'_e| \\
&\leq \sum_{e \in E} p_e^S(\boldsymbol{\mu}) \, |\mu_e - \mu'_e| \,. \qquad (20)
\end{aligned}$$

□

B Proof of Theorem 3.2

Proof. The main idea is to show that Lemma 5 in [9] still holds for the OCIM-OIFU algorithm in the OCIM setting without monotonicity. Let $\mathcal{N}_t^{\mathrm{s}}$ be the event that at the beginning of round t, for every arm $i \in [m]$, $|\hat{\mu}_{i,t} - \mu_i| \leq 2\rho_{i,t}$. Let $\mathcal{F}_t$ be the event that at round t oracle $\widetilde{\mathcal{O}}$ outputs a solution, $S^{(t)} = \{S_A^{(t)}, S_B\}$ and $\boldsymbol{\mu}^{(t)} = (\mu_1^{(t)}, \ldots, \mu_m^{(t)})$, such that $r_{S^{(t)}}(\boldsymbol{\mu}^{(t)}) < \alpha \cdot r_{S^*}(\boldsymbol{\mu}^*)$, i.e., oracle $\widetilde{\mathcal{O}}$ fails to output an α-approximate solution. In Lemma 5 from [9], it assumes that $\mathcal{N}_t^{\mathrm{s}}$ and $\neg\mathcal{F}_t$ hold. By $\mathcal{N}_t^{\mathrm{s}}$ and $0 \leq \mu_i \leq 1$ for all $i \in [m]$, we have

$$\forall i \in [m], \mu_i \in c_{i,t} = \left[(\hat{\mu}_{i,t} - \rho_{i,t})^{0+}, (\hat{\mu}_{i,t} + \rho_{i,t})^{1-}\right]. \qquad (21)$$

It means that we have the correct estimated range of μ_i for all $i \in [m]$ at round t. Combining with $\neg\mathcal{F}_t$ for the offline oracle $\widetilde{\mathcal{O}}$, we have

$$r_{S^{(t)}}(\boldsymbol{\mu}^{(t)}) \geq \alpha \cdot r_{S^*}(\boldsymbol{\mu}^*) \geq \alpha \cdot \mathrm{opt}(\boldsymbol{\mu}) = r_{S^{(t)}}(\boldsymbol{\mu}) + \Delta_{S^{(t)}}. \qquad (22)$$

By the TPM condition in Theorem. 3.1, we have

$$\Delta_{S^{(t)}} \leq r_{S^{(t)}}(\boldsymbol{\mu}^{(t)}) - r_{S^{(t)}}(\boldsymbol{\mu}) \leq C \sum_{i \in [m]} p_i^{S^{(t)}}(\boldsymbol{\mu})|\mu_i^{(t)} - \mu_i|. \qquad (23)$$

We want to bound $\Delta_{S^{(t)}}$ by bounding $|\mu_i^{(t)} - \mu_i|$. In fact, if $\mathcal{N}_t^{\mathrm{s}}$ holds and $\mu_i^{(t)} \in c_{i,t}$ for all $i \in [m]$,

$$\forall i \in [m], |\mu_i^{(t)} - \mu_i| \leq 2\rho_{i,t}. \qquad (24)$$

All requirements on bounding $\Delta_{S^{(t)}}$ in Lemma 5 from [9] are also satisfied by the OCIM-OIFU algorithm in the OCIM setting. Hence, we can follow the remaining proofs in [9] to derive the distribution-dependent and distribution-independent regret bounds shown in the theorem. □

C Proof of Theorem 3.3

Proof. We utilize the following well-known tail bound in our proof.

Lemma C.1. *(Hoeffding's Inequality) Let $X_1, \ldots, X_n$ be independent and identically distributed random variables with common support $[0,1]$ and mean μ. Let $Y = X_1 + \ldots, +X_n$. Then for all $\delta \geq 0$,*

$$\mathbb{P}\left\{|Y - n\mu| \geq \delta\right\} \leq 2e^{-2\delta^2/n}.$$

Let $\hat{\boldsymbol{\mu}} = (\hat{\mu}_1, \ldots, \hat{\mu}_m)$ be the empirical mean of $\boldsymbol{\mu}$. Recall that oracle $\mathcal{O}$ takes S_B and $\hat{\boldsymbol{\mu}}$ as inputs and outputs a solution $S^{\mathcal{O}}$. Let us define event $\mathcal{F} = \{r_{S^{\mathcal{O}}}(\hat{\boldsymbol{\mu}}) < \alpha \cdot \text{opt}(\hat{\boldsymbol{\mu}})\}$, which represents that oracle $\mathcal{O}$ fails to output an α-approximate solution, and we know $\mathbb{P}(\mathcal{F}) < 1 - \beta$.

We can decompose the regret as:

$$\begin{aligned} Reg_{\boldsymbol{\mu},\alpha,\beta}(T) &\leq \lceil nN/k \rceil \cdot \Delta_{\max} + \Big(T - \lceil nN/k \rceil\Big) \cdot \Big[\alpha\beta \cdot \text{opt}(\boldsymbol{\mu}) - \mathbb{E}\big[r_{S^{\mathcal{O}}}(\hat{\boldsymbol{\mu}})\big]\Big] \\ &\leq \lceil nN/k \rceil \cdot \Delta_{\max} + \Big(T - \lceil nN/k \rceil\Big) \cdot \Big[\alpha\beta \cdot \text{opt}(\boldsymbol{\mu}) - \beta \cdot \mathbb{E}\big[r_{S^{\mathcal{O}}}(\hat{\boldsymbol{\mu}}) \mid \neg\mathcal{F}\big]\Big] \\ &\leq \lceil nN/k \rceil \cdot \Delta_{\max} + \Big(T - \lceil nN/k \rceil\Big) \cdot \Big[\alpha \cdot \text{opt}(\boldsymbol{\mu}) - \mathbb{E}\big[r_{S^{\mathcal{O}}}(\hat{\boldsymbol{\mu}}) \mid \neg\mathcal{F}\big]\Big]. \end{aligned} \tag{25}$$

Next, let us rewrite the TPM condition in Theorem 3.1. For any S, $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, we have

$$\begin{aligned} |r_S(\boldsymbol{\mu}) - r_S(\boldsymbol{\mu}')| &\leq C \sum_{i \in [m]} p_i^S(\boldsymbol{\mu}) |\mu_i - \mu_i'| \\ &\leq C \sum_{i \in [m]} |\mu_i - \mu_i'| \\ &\leq Cm \cdot \max_{i \in [m]} |\mu_i - \mu_i'|, \end{aligned} \tag{26}$$

where C is the maximum number of nodes that any one node can reach in graph G. Let $S^*_{\boldsymbol{\mu}}$ denote the optimal action for $\boldsymbol{\mu}$. Under $\neg\mathcal{F}$, we have

$$\begin{aligned} r_{S^{\mathcal{O}}}(\hat{\boldsymbol{\mu}}) &\geq \alpha \cdot r_{S^*_{\hat{\boldsymbol{\mu}}}}(\hat{\boldsymbol{\mu}}) \\ &\geq \alpha \cdot r_{S^*_{\boldsymbol{\mu}}}(\hat{\boldsymbol{\mu}}) \\ &\geq \alpha \cdot r_{S^*_{\boldsymbol{\mu}}}(\boldsymbol{\mu}) - \alpha \cdot Cm \cdot \max_{i \in [m]} |\mu_i - \hat{\mu}_i| \\ &\geq r_{S^{\mathcal{O}}}(\boldsymbol{\mu}) + \Delta_{S^{\mathcal{O}}} - \alpha \cdot Cm \cdot \max_{i \in [m]} |\mu_i - \hat{\mu}_i|, \end{aligned} \tag{27}$$

where the third inequality is due to Eq.(26). Combining Eq.(26) and Eq.(27) together, we have

$$\begin{aligned} \Delta_{S^{\mathcal{O}}} &\leq r_{S^{\mathcal{O}}}(\hat{\boldsymbol{\mu}}) - r_{S^{\mathcal{O}}}(\boldsymbol{\mu}) + \alpha \cdot Cm \cdot \max_{i \in [m]} |\mu_i - \hat{\mu}_i| \\ &\leq (1 + \alpha) \cdot Cm \cdot \max_{i \in [m]} |\mu_i - \hat{\mu}_i|. \end{aligned} \tag{28}$$

Let us define $\delta_0 := \frac{\Delta_{\min}}{2Cm}$. If $\max_{i \in [m]} |\mu_i - \hat{\mu}_i| < \delta_0$, then we know $S^{\mathcal{O}}$ is at least an α-approximate solution, such that $\Delta_{S^{\mathcal{O}}} = 0$. Then the regret in Eq.(25) can be written as

$$\begin{aligned} Reg_{\boldsymbol{\mu},\alpha,\beta}(T) &\leq \lceil nN/k \rceil \cdot \Delta_{\max} + \Big(T - \lceil nN/k \rceil\Big) \cdot 2m \exp(-2N\delta_0^2) \cdot \Delta_{\max} \\ &\leq \Big(\lceil nN/k \rceil + T \cdot 2m \exp(-2N\delta_0^2)\Big) \cdot \Delta_{\max}. \end{aligned} \tag{29}$$

The first inequality is obtained by applying the Hoeffding's Inequality (Lemma C.1) and union bound to the event $\max_{i \in [m]} |\mu_i - \hat{\mu}_i| \geq \delta_0$. Now we need to choose an optimal N that minimizes Eq.(29). By taking $N = \max\Big\{1, \frac{1}{2\delta_0^2} \ln \frac{4kmT\delta_0^2}{C}\Big\} = \max\Big\{1, \frac{2C^2m^2}{\Delta_{\min}^2} \ln(\frac{kT\Delta_{\min}^2}{C^3m})\Big\}$, when $\Delta_{\min} > 0$, we can get the distribution-dependent bound

$$\text{Reg}_{\boldsymbol{\mu},\alpha,\beta}(T) \leq \frac{2C^2m^2n\Delta_{\max}}{k\Delta_{\min}^2} \left(\max\left\{\ln\left(\frac{kT\Delta_{\min}^2}{C^2mn}\right), 0\right\} + 1\right) + \frac{n}{k}\Delta_{\max}, \tag{30}$$

Next, let us prove the distribution-independent bound. Let $\mathcal{N}$ denote the event that $|\hat{\mu}_i - \mu_i| \leq \sqrt{\frac{2\ln T}{N}}$ for all $i \in [m]$. By the Hoeffding's Inequality and union bound, we have

$$\mathbb{P}\{\neg\mathcal{N}\} \leq m \cdot \frac{2}{T^4} \leq \frac{2}{T^3}. \tag{31}$$

When $\mathcal{N}$ holds, with Eq.(28), we have

$$\Delta_{S^{\mathcal{O}}} \leq 2Cm \cdot \sqrt{\frac{2\ln T}{N}}, \tag{32}$$

and the regret in Eq.(25) can be written as

$$
\begin{aligned}
Reg_{\boldsymbol{\mu},\alpha,\beta}(T) &\leq \lceil nN/k \rceil \cdot n + \Big(T - \lceil nN/k \rceil\Big) \cdot \Delta_{S^{\mathcal{O}}} \\
&\leq \lceil nN/k \rceil \cdot n + O\left(T \cdot Cm \cdot \sqrt{\frac{\ln T}{N}}\right).
\end{aligned} \tag{33}
$$

We can choose N so as to (approximately) minimize the regret. For $N = (Cmk)^{\frac{2}{3}} n^{-\frac{4}{3}} T^{\frac{2}{3}} (\ln T)^{\frac{1}{3}}$, we obtain:

$$
Reg_{\boldsymbol{\mu},\alpha,\beta}(T) \leq O((Cmn)^{\frac{2}{3}} k^{-\frac{1}{3}} T^{\frac{2}{3}} (\ln T)^{\frac{1}{3}}). \tag{34}
$$

To complete the proof, we need to consider both $\mathcal{N}$ and $\neg\mathcal{N}$. As shown in Eq.(31), the probability that $\neg\mathcal{N}$ occurs is very small, and we have:

$$
\begin{aligned}
Reg_{\boldsymbol{\mu},\alpha,\beta}(T) &= \mathbb{E}\left[Reg_{\boldsymbol{\mu},\alpha,\beta}(T) \mid \mathcal{N}\right] \cdot \mathbb{P}\{\mathcal{N}\} + \mathbb{E}\left[Reg_{\boldsymbol{\mu},\alpha,\beta}(T) \mid \neg\mathcal{N}\right] \cdot \mathbb{P}\{\neg\mathcal{N}\} \\
&\leq \mathbb{E}\left[Reg_{\boldsymbol{\mu},\alpha,\beta}(T) \mid \mathcal{N}\right] + T \cdot n \cdot O(T^{-3}) \\
&\leq O((Cmn)^{\frac{2}{3}} k^{-\frac{1}{3}} T^{\frac{2}{3}} (\ln T)^{\frac{1}{3}}).
\end{aligned} \tag{35}
$$

□

D Non-submodularity of $g(S)$ in Section 4

At the beginning of Section 4, we introduce $g(S) = \max_{\boldsymbol{\mu}} r_S(\boldsymbol{\mu})$, which is an upper bound function of $r_S(\boldsymbol{\mu})$ for each S. If $g(S)$ is submodular over S, we can use a greedy algorithm on $g(S)$ to find an approximate solution. However, the following example in Fig. 4 shows that $g(S)$ is not submodular.

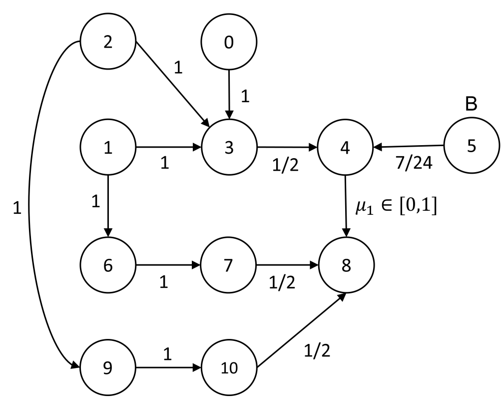


Figure 4: Example showing that $g(S)$ is not submodular

In Fig. 4, the numbers attached to edges are influence probabilities. Only the influence probability of edge $(4, 8)$ is a variable and we denote it as μ_1. We assume $\mu_1 \in [0, 1]$ and $S_B = \{5\}$. Let us consider some choices of S_A. When S_A is chosen as $\{0\}, \{0, 1\}$ or $\{0, 2\}$, the optimal μ_1 that maximizes $r_S(\boldsymbol{\mu})$ is 1; when S_A is chosen as $\{0, 1, 2\}$, the optimal μ_1 that maximizes $r_S(\boldsymbol{\mu})$ is 0. Based on this observation, we can calculate $g(S)$ (assuming $S_B = \{5\}$):

$$
\begin{aligned}
&g(S_A = \{0\}) = 2 + \frac{17}{24}, \\
&g(S_A = \{0, 1\}) = 5 + \frac{17}{24} \times \frac{4}{5}, \\
&g(S_A = \{0, 2\}) = 5 + \frac{17}{24} \times \frac{4}{5}, \\
&g(S_A = \{0, 1, 2\}) = 8 + \frac{17}{24} \times \frac{1}{2} + \frac{3}{4}.
\end{aligned}
$$

Thus we have

$$g(S_A = \{0,1\}) + g(S_A = \{0,2\}) < g(S_A = \{0\}) + g(S_A = \{0,1,2\}), \tag{36}$$

which is contrary to submodularity.

E #P-hardness of finding the optimal μ_i

At the beginning of Section 4, we discuss a special case of the offline optimization in Eq.(2). The following lemma shows it is #P-hard.

Lemma E.1. *Given S and fixing μ_e for all $e \neq i$, finding the optimal $\mu_i \in c_i$ for one edge i that maximizes $r_S(\boldsymbol{\mu})$ is #P-hard.*

Proof. We prove the hardness of this optimization problem via a reduction from the influence computation problem. We first consider a general graph G_0 with n nodes and m edges, where all influence probabilities on edges are set to $1/2$. Given S_A, computing the influence spread of A in such a graph is #P-hard. Notice that there is no seed set of B in G_0. Now let us take one node v in G_0 and denote its activation probability by A as $h_A(G_0, S_A, v)$. Actually, computing $h_A(G_0, S_A, v)$ is also #P-hard and we want to show that it can be reduced to our optimization problem in polynomial time.

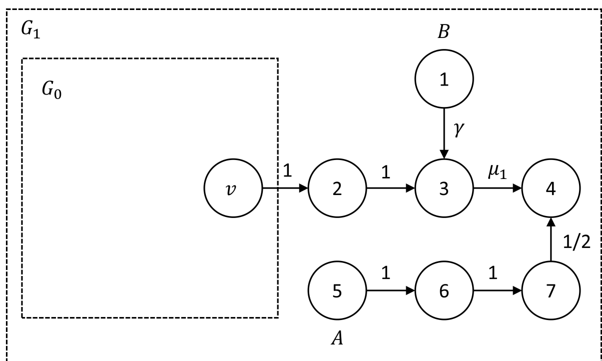


Figure 5: Construction of G_1 based on G_0

We first construct a new graph G_1 based on G_0. For G_1, we keep G_0 and S_A unchanged, then add several nodes and edges as shown in Fig. 5. We add node 1 to the seed set of B and node 5 to the seed set of A, so the joint action $S = \{S_A \cup \{5\}, S_B = \{1\}\}$. In this new graph G_1, we consider the optimization problem of finding the optimal μ_1 (influence probability on edge $(3,4)$) within its range c_1 that maximizes $r_S(\boldsymbol{\mu})$. Notice that the influence probability γ on edge $(1,3)$ is a constant and μ_1 would only affect the activation probability of node 4. We denote the activation probability by A of node 4 as $h_A(G_1, S, 4)$. In order to maximize $r_S(\boldsymbol{\mu})$, we only need to maximize $h_A(G_1, S, 4)$. It can be written as:

$$h_A(G_1, S, 4) = \frac{1}{2}\Big[(1-\gamma)\cdot h_A(G_1, S, v) - \gamma\Big]\cdot \mu_1 + \frac{1}{2}. \tag{37}$$

It is easy to see $h_A(G_1, S, 4)$ has a linear relationship with μ_1, so the optimal μ_1 could only be either the lower or upper bound of its range c_1. Assuming we can solve the optimization problem of finding the optimal μ_1, then we can determine the sign of μ_1's coefficient in Eq.(37): if the optimal μ_1 is the upper bound value in c_1, we have $(1-\gamma)\cdot h_A(G_1, S, v) - \gamma \geq 0$; otherwise, $(1-\gamma)\cdot h_A(G_1, S, v) - \gamma < 0$. It means we can answer the question that whether $h_A(G_1, S, v)$ is larger (or smaller) than $\frac{\gamma}{1-\gamma}$. Notice that $h_A(G_0, S_A, v) = h_A(G_1, S, v)$, so we can manually change the value of γ to check whether $h_A(G_0, S_A, v)$ is larger (or smaller) than $x = \frac{\gamma}{1-\gamma}$ for any $x \in [0,1]$, Recall that all edge probabilities in G_0 are set to $1/2$, so the highest precision of $h_A(G_0, S_A, v)$ should be 2^{-m}. Hence, we can use a binary search algorithm to find the exact value of $h_A(G_0, S_A, v)$ in at most m times. It means computing the activation probability of v in G_0 can be reduced to the optimization problem of finding the optimal μ_1 in G_1, which completes the proof. $\square$

F Proof of Lemma 4.1

Proof. We can expand $r_S(\boldsymbol{\mu})$ based on the live-edge graph model [7]:

$$r_S(\boldsymbol{\mu}) = \sum_L |\Gamma_A(L,S)| \cdot \Pr(L) = \sum_L |\Gamma_A(L,S)| \prod_{e \in E(L)} \mu_e \prod_{e \notin E(L)} (1-\mu_e), \tag{38}$$

where L is one possible live-edge graph (each edge $e \in E$ is in L with probability μ_e and not in L with probability $1-\mu_e$, and this is independent from other edges), $\Gamma_A(L,S)$ is the set of nodes activated by A from seed sets $S = \{S_A, S_B\}$ under live-edge graph L and $E(L)$ is the set of edges that appear in live-edge graph L. Eq.(38) shows that $r_S(\boldsymbol{\mu})$ is linear with each μ_i, so the optimal μ_i must take either the minimum or the maximum value in its range c_i. □

G Discussion of Offline Oracle in Directed Tree

From Lemma 4.1, we know the optimal values of μ_i's must occur at the boundaries of their intervals c_i's. It implies that for any edge e not reachable from S_B, the optimal μ_e is always the upper bound value since edge e can only help the propagation of A. In Section 4.2, we argue that the number of edges reachable from B could be small in some graphs, so that we can afford enumerating all the boundary value combinations of these edges. We discuss such graphs in directed trees. Specifically, we consider the in-arborescence, where all edges point towards the root. For any node u in the in-arborescence, there only exists one path from u to the root; if u is selected as the seed node of B, it could only propagate via this path. Hence, if the depth of the in-arborescence is in the order of $O(\log m)$, the number of edges reachable from S_B would be $O(|S_B| \cdot \log m)$. In this case, we can use the IMM algorithm for $O(m^{|S_B|})$ combinations to obtain an approximate solution with time complexity $O(m^{|S_B|} \cdot T_{\text{IMM}})$. Examples of such in-arborescences with depth $O(\log m)$ could be the complete or full binary trees.

H Additional Experiments

H.1 Dataset Details and Parameter Settings

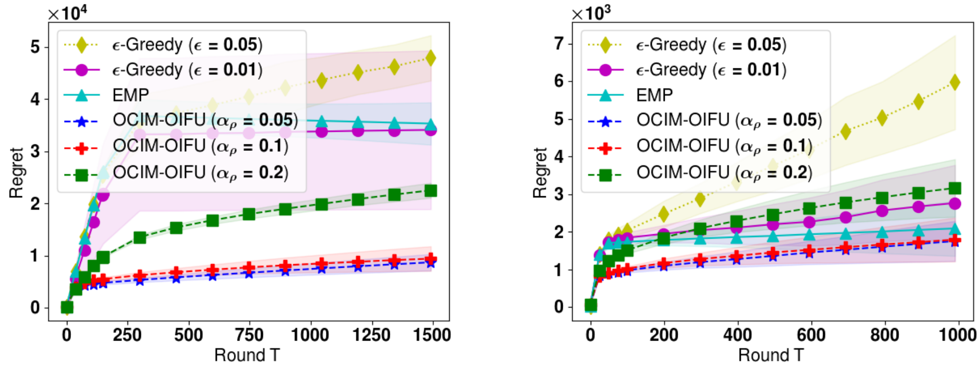
Table 1: Dataset Statistics

Network	n	m	Average Degree
Yahoo-Ad	$11,475$	$52,567$	4.58
DM	679	$3,374$	4.96

We summarize the detailed statistics in Table 1. We set the parameters of our experiments as the following. For the edge weights, Yahoo-Ad uses the weighted cascade method [1], i.e. $p(s,t) = 1/deg_-(s)$, where $deg_-(s)$ is the in-degree of node s, and weights for DM are obtained by the learned edge parameters from [25]. In our experiments, we set $|S_A| = |S_B| = 5$ for Yahoo-Ad and $|S_A| = |S_B| = 10$ for DM dataset. Since the optimal solution given the true edge probabilities cannot be derived in polynomial time, in order to derive the approximate regret, for Yahoo-Ad, we use the greedy solution as the optimal baseline, which is a $(1-1/e, 1)$-approximate solution; for DM dataset, we use the IMM solution as the optimal baseline, which is a $(1-1/e-\epsilon, 1-n^{-l})$-approximate solution.

H.2 Experiments for $A > B$ Tie-breaking Rule

As mentioned in Sec 4.1, we can trivially ignore S_B to choose S_A, and OCIM becomes the online influence maximization problem without competition, so we omit the experiments for bipartite graphs. For general graphs, we use the same DM dataset and parameter settings described in Sec. 5. However, we use a different heuristic, where we set influence probabilities to their upper bound values to optimistically maximize A's influence. We show the results in Figure 6. Overall, the results and the analysis for $A > B$ are consistent with $B > A$.

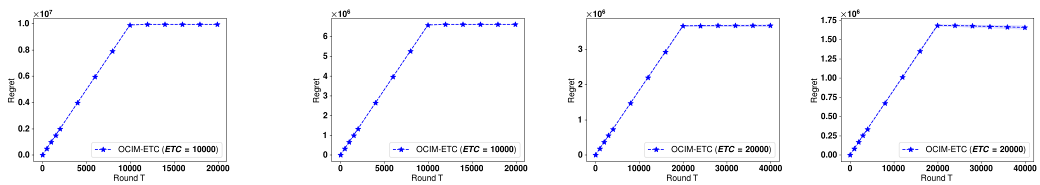


(a) DM, RD

(b) DM, IM

Figure 6: Regrets of different algorithms for general graphs, when $A > B$.

H.3 Experiments for OCIM-ETC



(a) Yahoo-Ad, RD

(b) Yahoo-Ad, IM

(c) DM, RD

(d) DM, IM

Figure 7: Regrets of OCIM-ETC for bipartite and general graphs.

We show the results for the OCIM-ETC algorithm in Figure 7. The dataset and parameter settings are the same, and we set the exploration phase to be $10,000$ and $20,000$ for Yahoo-Ad and DM, respectively. Experiments show that OCIM-ETC has linear regret in the exploration phase and constant regret in the exploitation phase. Compared with OCIM-OIFU, it requires more rounds to learn the unknown influence probabilities and has larger regret than OCIM-OIFU.