Host Profit Maximization: Leveraging Performance Incentives and User Flexibility

Xueqin Chang[♮], Xiangyu Ke[♮], Lu Chen[♮], Congcong Ge[♯], Ziheng Wei[♯], Yunjun Gao[♮]
[♮]Zhejiang University, [♯]Huawei Cloud Computing Technologies Co., Ltd
{changxq,xiangyu.ke,luchen,gaoyj}@zju.edu.cn,{gecongcong1,ziheng.wei}@huawei.com

ABSTRACT

The social network host has knowledge of the network structure and user characteristics and can earn a profit by providing merchants with viral marketing campaigns. We investigate the problem of *host profit maximization* by leveraging performance incentives and user flexibility.

To incentivize the host's performance, we propose setting a desired influence threshold that would allow the host to receive full payment, with the possibility of a small bonus for exceeding the threshold. Unlike existing works that assume a user's choice is frozen once they are activated, we introduce the Dynamic State Switching model to capture *"comparative shopping"* behavior from an economic perspective, in which users have the flexibilities to change their minds about which product to adopt based on the accumulated influence and propaganda strength of each product. In addition, the incentivized cost of a user serving as an influence source is treated as a negative part of the host's profit.

The *host profit maximization* problem is **NP**-hard, submodular, and non-monotone. To address this challenge, we propose an efficient greedy algorithm and devise a scalable version with an approximation guarantee to select the seed sets. In addition, we develop two seed allocation algorithms to balance the distribution of adoptions among merchants with small profit sacrifice. Through extensive experiments on four real-world social networks, we demonstrate that our methods are effective and scalable.

PVLDB Reference Format:
Xueqin Chang, Xiangyu Ke, Lu Chen, Congcong Ge, Ziheng Wei, Yunjun Gao. Host Profit Maximization: Leveraging Performance Incentives and User Flexibility. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

1 INTRODUCTION

Influence maximization (IM) [26] is a crucial task in the analysis of social networks due to its significant commercial value in viral marketing [15], network monitoring [28], social recommendation [60], and so on. Given a social graph and an integer k, the objective of IM is to identify a set of k seed nodes as the source of information propagation such that the expected number of influenced nodes is maximized under a specified diffusion model. The study of IM has attracted significant attention in the fields of data management, leading to a focus on **(1)** designing practical objectives according to real-world application demands [5, 19, 25]; **(2)** modeling information diffusion process based on users' behaviors and inherent properties [7, 32, 56]; and **(3)** devising effective and efficient solutions with quality guarantees [17, 39, 51].

Most existing studies assume that a merchant can determine an optimal set of seed users to initially adopt her/his product[1] by estimating the influence spread in a social network. However, in reality, social networks are often owned by a third-party host like Facebook or TikTok, which keeps the network structure and the user features secret for their own benefit or privacy concerns [27, 32]. Additionally, multiple merchants may compete and launch similar products around the same time in the same marketplace simultaneously [6, 21, 27, 30, 32]. For instance, in 2022, both Apple's iPhone 14 series [1] and Huawei Mate 50 series [23] were introduced in September, and Samsung's Galaxy series [40] was launched in August. Hence, in this work, we consider a scenario where *the social network host conducts the seed selection for multiple competing merchants, while each merchant offers a budget as the quoted price for her/his desired level of influence.* Specifically, we define a more practical host profit maximization problem (§ 2.2) under a novel diffusion model (§ 2.1) which takes into account *"comparative shopping"* behavior [12, 49, 58] from an economic perspective. Finally, effective and efficient solutions are devised with quality guarantee (§ 3).

Host Profit Maximization. The host of a social network platform, who has knowledge about the social graph structure and user characteristics, has the opportunity to earn a profit by providing merchants with influence in marketing campaigns through their platform[2] [3, 19, 32, 42, 52, 61]. The host's profit is computed as the revenue received from merchants minus the cost of paying seed users to incentivize them as the source of information propagation. The revenue is the amount paid by each merchant to the host for a desired number of users adopting their product, while the cost is the payment made by the host to the seed users. Specifically, our revenue computation differs from previous research [3, 4, 19] by introducing a *retail goal or threshold*, i.e., each merchant's desired level of influence spread. In complex market environments, achieving the requested influence level may not always be feasible [42, 61]. Therefore, we formalize that the host earns *partial or even no payment* if they are unable to meet the merchant's requirement, and a *small extra reward* if they exceed the merchant's demand. Moreover, unlike [3, 19, 62], we assume that the cost of incentivizing seed users is a negative part of the host's profit, as only the social network host can evaluate a user's influence ability.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

[1]The term product may also refer to opinions, technologies, innovations, etc.
[2]Influencer marketing has grown from a $1.7 billion in 2016 to a projected $16.4 billion in 2022, reported in https://sproutsocial.com/insights/pr-and-influencer-marketing/.

General Applicability. Our proposed profit formulation has practical applications in various real-world scenarios. For instance, during an election campaign, a candidate needs to secure a specific level of support to win [47]. However, the candidate may not be willing to pay the full price for partial adoption which cannot ensure their victory. On the other hand, once the candidate attains a majority support level like 51%, additional votes can further strengthen their position but are no longer pivotal. Similarly, achieving a higher influence level can help boost brand awareness, user engagement, and lead generation [41, 43, 48]. In this case, the host may need to pay a penalty for failing to achieve the goal.

Dynamic State Switching Model. In the traditional Independent Cascade (IC) model and Linear Threshold (LT) model [26] for influence maximization, a user's choice is fixed after they are activated once. The multiple campaign IC and LT models [7, 9, 27, 32, 56] follow the same approach. Recently, the K-LT [32] and AtI [56] models, both are extensions of the LT model, have been developed from the host perspective. These models split the influence adoption process into two steps: a node is first activated through an LT-like process, and then, they adopt one of the influence sources based on the influence strength of that source within a recent time interval (K-LT) or the similarity of the user's features to those of the products' (AtI). However, the adoptions are frozen upon one-time activation (i.e., users cannot change their minds thereafter), regardless of accumulated influence or the arrival of other even-matched products. In economic and marketing contexts, it is common for users to engage in *"comparative shopping"* behavior [12, 49, 57, 58] where *consumers search for and compare various similar products based on factors such as price, warranty policy, and quality reviews before making a purchase decision*[3]. To capture this behavior, we propose a Dynamic State Switching model (§ 2.1) that allows users to change their minds from product A to product B if **(1)** the accumulated influence of B is greater than that of A, and **(2)** the propaganda strength of B is stronger than that of A. The model converges when no more user is activated and no user changes her mind.

Theoretical Analyses and Solutions. We demonstrate that the host profit maximization problem is *not monotone* but *submodular*. In addition, we prove that the host profit maximization problem is **NP**-hard, and it is even **NP**-hard to approximate with any constant factor by using a reduction from the 3-PARTITION problem (§ 2.3). These results imply that our problem is not tractable in general. However, we develop an effective greedy algorithm with approximation guarantee (§ 3) to allocate seed users for multiple merchants by extending the ROI-Greedy [24]. Notice that it is non-trivial to generalize from a single merchant to multiple merchants due to the requirement of designing a meticulous seed allocation strategy while maintaining theoretical guarantees. We also propose a scalable version of our method with performance bounds by leveraging state-of-the-art approaches for expected influence spread estimation, while a novel unbiased estimation method is specifically tailored for our Dynamic State Switching model. As a side contribution, we consider the practical case that the host would like to maintain long-term business relationships with all merchants and study the problem of fair seed allocation with little profit sacrifice (i.e., up to 10% as shown in § 5) to better balance the distribution of adoptions among merchants. We propose a heuristic search method that selects users with the highest host profit margin for each merchant in a one-by-one manner, and a framework that iteratively selects users to both maximize the host's profit and increase the merchant's influence (§ 4).

Contributions and Roadmap.

- We study the host profit maximization problem where a merchant will make the full payment if a desired influence spread is achieved. The incentivized cost of a user is treated as a negative part of the host's profit (§ 2.2).
- We design the Dynamic State Switching propagation model to capture the *"comparative shopping"* behavior from an economic perspective (§ 2.1).
- We characterize the hardness of solving our problem (§ 2.3), and develop an effective greedy seed selection method to maximize the host's profit with an approximation guarantee (§ 3.1). Moreover, we devise a scalable version of our approximation algorithm (§ 3.2).
- We present a practical scenario, and propose two heuristic methods to balance the distribution of adoptions among products while sacrificing little profit of host (§ 4).
- We conduct thorough experimental evaluations using four real-world social network datasets, and validate that our algorithms are effective and scalable (§ 5).
- We present a thorough literature review about other social advertising variants (§ 6) and conclude our paper (§ 7).

2 PRELIMINARIES

A social network platform, referred to as the *host*, owns a social graph $G = (V, E)$, where V is the set of n users and $E \subseteq V \times V$ represents the set of m social connections. Each edge $e = (u, v)$ is associated with a weight $w_{u,v}$, depicting the influence strength from user u to v. Let $\mathcal{N}^{in}(v)$ be the set of incoming neighbors of node v. $\mathcal{H} = \{h_1, h_2, ..., h_{|\mathcal{H}|}\}$ is a set of $|\mathcal{H}|$ *merchants* who would like to promote their products on a social network. Each merchant h_i submits a campaign proposal to the *host*, which includes a minimum desired influence spread I_i (i.e., a threshold) and the corresponding budget B_i that the merchant is willing to pay. The *host* evaluates the influence diffusion on her social network and selects a set of seed users S_i for merchant h_i, $S_i \cap S_j = \emptyset$ if $i \neq j$. In the following, we present the novel Dynamic State Switching (DSS) information diffusion model (§ 2.1) to capture *"comparative shopping"* behavior and facilitate the influence spread estimation. After that, we formally define our *host profit maximization* problem (§ 2.2) and provide the theoretical characteristics (§ 2.3).

2.1 The DSS Propagation Model

In the classical single-merchant LT model [26], each node is assigned an activation threshold $\theta_v \leq 1$ randomly from the range [0, 1]. The sum of the weights of all incoming edges for each node is normalized to be at most 1. The propagation process begins with a set of seed nodes that are initially active and then progresses in discrete steps. If the sum of the weights of the incoming edges from all active neighbors is equal to or greater than the activation threshold of an inactive node, that node becomes active in the next time stamp. The diffusion process terminates when no more nodes can be activated. Each node can only be activated once and remains active until the end of the propagation process.

[3] 2 in 3 UK online shoppers compare before they buy [12].

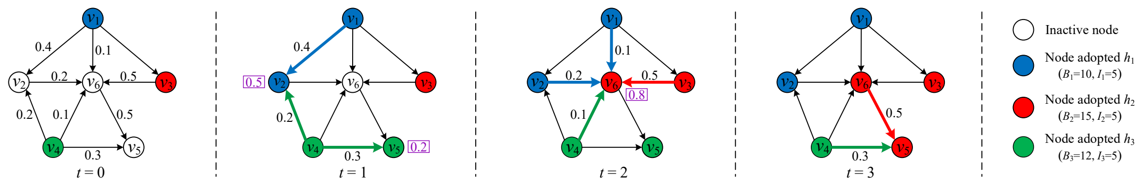


Figure 1: Illustrating propagation of products under *Dynamic State Switching* model

We extend the classical LT model to the multiple-merchant setting, which is referred to as the Dynamic State Switching (DSS) propagation model. This model consists of three phases: activation, adoption, and switching. For each merchant $h_i \in \mathcal{H}$, a set S_i of nodes is selected as its seeds and is initially adopted by product h_i. The influence then propagates as follows:

(1) Activation phase. Similar to K-LT [32] and AtI [56] models, an inactive node in DSS model will be activated in the same way as the LT model. Initially, all nodes are inactive. At time 0, for each merchant h_i, the node $u \in S_i$ become active with its product h_i. At any time $t \geq 1$, an inactive node v becomes active when the sum of incoming weights from its active in-neighbors (regardless of products[4]) is at least v's activation threshold. Once a node becomes active, it remains active until the end of the diffusion process.

(2) Adoption phase. Let $\mathcal{F}_i$ be the set of v's neighbors that have adopted product h_i. When node v is activated, it selects the product that is adopted by most of its active in-neighbors, formally $\arg\max_{h_i \in \mathcal{H}} \sum_{u \in \mathcal{F}i} w_{u,v}$. We assume that each node can only adopt one product due to the competitive nature of the market and consumer's limited budget [27, 32, 56].

(3) Switching phase. After node v is activated, it continuously receives information from neighbors at subsequent time steps. Once node v is aware of new products at any time step, it compares them and will switch its adoption to the product h_j if and only if **(1)** its influence is higher, i.e., $\sum_{u \in \mathcal{F}_j} w_{u,v} > \sum_{u \in \mathcal{F}_i} w_{u,v}$, and **(2)** the host has stronger propaganda strength for it, i.e., $\frac{B_j}{I_j} > \frac{B_i}{I_i}$. The first condition reflects the long-term accumulated influence, while the second condition represents external factors[5] in *"comparative shopping"* behavior from an economic perspective [35, 45].

Comparisons with the existing multi-merchant LT models. Lu et al. [32] proposed the K-LT model, which is an extended LT model that incorporates the most recent effect in the final decision. Specifically, at any time $t \geq 1$, node v decides to adopt a product based only on its neighbors who adopted that product at time $t-1$. However, we consider the accumulative effect of products since the beginning of the information propagation process, rather than the short-term impact at any time step. This is similar to the Weighted-Proportional Competitive (WPCLT) model proposed by Borodin et al. [9]. Another recent model, the AtI model [56], decides on adoption based on the similarity between the user and product features. None of these models consider the well-known *"comparative shopping"* behavior in the economic and marketing context [12, 49, 57, 58]. To the best of our knowledge, we are the first to model the changing of social choices in influence maximization.

Example 1. *Figure 1 shows an example of the* DSS *model. Suppose there are three merchants, each has a seed v_1 (blue), v_3 (red), and v_4 (green), respectively. At time t=1, v_2 becomes active because $w_{v_1,v_2} + w_{v_4,v_2} = 0.4 + 0.2 > \theta_{v_2} = 0.5$. Then, v_2 adopts product h_1 because it carries the largest weight (i.e., $h_1 = 0.4 > h_3 = 0.2$) among v_2's active in-neighbors. v_5 adopts product h_3. However, v_6 remains inactive at time t=1. At time t=2, v_6 becomes active in the activation phase because the total incoming weights from its active in-neighbors is higher than its activation threshold (i.e., $0.9 > 0.8$). In the adoption phase, due to the* DSS *model considering the accumulative effect of products since the beginning of the propagation process, v_6 adopts product h_2 as it carries the largest influence weight (i.e., $h_2 = 0.5 > h_1 = 0.3 > h_3 = 0.1$). However, under the* K-LT *model [32], v_6 will adopt product h_1 because it only considers the effect of v_6's in-neighbors who were activated at the last time step (i.e., only v_2 was activated with h_1 at time t=1). At time t=3, in the switching phase, v_5 switches its adoption to product h_2 since it carries a larger weight (i.e., $h_2 = 0.5 > h_3 = 0.3$) and stronger propaganda strength (i.e., $\frac{B_2}{I_2} = 1.5 > \frac{B_3}{I_3} = 1.2$) than product h_3. The propagation ends at v_5 because there are no more nodes that can be activated and switch adoption further.*

2.2 Problem Definition

We are now ready to define the host profit maximization problem. As mentioned before, $|\mathcal{H}|$ merchants compete in a social network with similar products, each announcing the host with an influence threshold I_i and corresponding budget B_i. The host seeks for an allocation $\mathbb{S}$, which is a set of $|\mathcal{H}|$ disjoint sets $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$, where S_i is the seed set assigned to merchant h_i to conduct the marketing campaign propagation and try to earn maximal profit. We first define the *Revenue function* and *Cost function* for a merchant.

Definition 1. **(Revenue function).** *The revenue that host gains from merchant h_i as $R(S_i)$ for a desired influence level I_i is*

$$R(S_i) = B_i \cdot (1 + \gamma \cdot \frac{\sigma(S_i) - I_i}{I_i}) \tag{1}$$

where $\sigma(S_i)$ is the expected influence of S_i, and γ is a parameter of penalty or reward. When $\sigma(S_i) < I_i$, γ is a penalty parameter (i.e., γ_p) controlling the severity of penalty, and it is a reward parameter (i.e., γ_r) determining the level of reward when $\sigma(S_i) \geq I_i$. Note that, when $\gamma = 1$, the *Revenue function* is reduced to the classical revenue maximization problem with CPE model [3, 4, 19]. Similar to [61], the choice of γ is orthogonal to our problem. More details on the realistic selection of γ (γ_p and γ_r), each merchant's I_i and B_i and other parameters are referred to the experiments in § 5.

Suppose that each node $v \in V$ is associated with an incentive cost $c(v)$ according to its influence ability, we then introduce the notion of *Cost function* for a seed set S_i.

[4]This captures the natural process by which a user becomes familiar with and interested in a category of products through the joint influence of all the products in that category. We assume that similar products share the same set of influence probabilities.

[5]Intuitively, the host would prefer the merchant with a larger quoted price per unit influence and would rank their product higher when the user searches for comparisons. In real applications, this metric could be replaced with any measure of product quality, customer service, price, etc.

Definition 2. **(Cost function).** *The incentive cost that the host needs to pay for selecting S_i as seed set for merchant h_i is*

$$C(S_i) = \sum_{v \in S_i} c(v) \tag{2}$$

It is well known that profit is equal to revenue minus cost. Therefore, the profit that the host earns from merchant h_i is denoted as $P(S_i)$, and $P(S_i) = R(S_i) - C(S_i)$. Finally, we formally define the profit maximization problem from the host's perspective.

Definition 3. **(HOST PROFIT MAXIMIZATION).** *Give a social graph $G = (V, E)$, a merchant set $\mathcal{H}$, and seed user incentive cost $c(v)$, $v \in V$, the goal of our problem is to find a feasible allocation $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$ for all merchants, which can maximize the total profit of the host. Formally:*

$$\arg\max_{\mathbb{S}} P(\mathbb{S}) = \sum_{S_i \in \mathbb{S}} P(S_i), \text{ subject to: } S_i \cap S_j = \emptyset \tag{3}$$

Note that limiting the products adopted by a seed can increase the credibility and persuasion for followers [6, 19, 56].

2.3 Problem Analysis

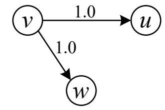
In this section, we first show that the *host profit maximization* problem is *non-monotone* and *submodular*. Then we prove that the problem is **NP**-hard, and is **NP**-hard to approximate within any constant factor.

A possible world $\mathcal{G} = (V, E_{\mathcal{G}})$ is known as one certain instance of an uncertain graph. The influence spread of the seed set $\mathbb{S}$ in $\mathcal{G}$ is denoted by $\sigma_{\mathcal{G}}(\mathbb{S})$, which is the number of users that can be reached from the seed set $\mathbb{S}$ in $\mathcal{G}$. Each world $\mathcal{G}$ exists with a probability $P(\mathcal{G}) = \prod_{(u,v)\in E_{\mathcal{G}}} w_{u,v} \prod_{(u,v)\in E\setminus E_{\mathcal{G}}}(1 - w_{u,v})$, and the influence spread of the seed set is the weighted sum of its influence spread over all possible worlds [52, 56], i.e., $\sigma(\mathbb{S}) = \sum_{\mathcal{G}\sqsubseteq G}(P(\mathcal{G}) \cdot \sigma_{\mathcal{G}}(\mathbb{S}))$.

Notice that under the DSS propagation model, an activated user (except seed user) can only switch its adoption to another product with stronger propaganda strength, that is, if a user that has adopted product h_j switches adoption to the product h_i ($i \neq j$), $\frac{B_i}{I_i} > \frac{B_j}{I_i}$ always holds. Based on the above, we provide the following basic theoretical characteristic of our problem.

Theorem 1. **(Non-monotonicity.)** *The host profit maximization is non-monotone under the DSS propagation model.*

Proof. We illustrate that our problem is *non-monotone* under the DSS propagation model through a counter-example. In Figure 2, we consider that two merchants h_1 and h_2, h_1 proposes influence threshold $I_1 = 5$ and corresponding $B_1 = \$7.5$, as for h_2, $I_2 = 5$ and $B_2 = \$5$, we set penalty parameter $\gamma_p = 1$ and reward parameter $\gamma_r = 0.3$. The costs of users v, u, w are shown in the right table, e.g., host needs to pay \$1.5 to incentivize v as a seed user. We assume that user v was already assigned to S_1 (i.e., seed set of merchant h_1), and under the DSS model, user u and w will be activated by user v with probability 1. The host's profit is $P(\mathbb{S}) = \$7.5(1+1\cdot\frac{3-5}{5}) - \$1.5 = \$3$. If then we assign user u to the seed set S_2 of merchant h_2, the host's profit is reduced to $P(\mathbb{S}') = P(S_1) + P(S_2) = (\$7.5(1 + 1\cdot\frac{2-5}{5}) - \$1.5) + (\$5(1+1\cdot\frac{1-5}{5}) - \$0.5) = \$2$. If we assign the seed sets in the other order (i.e., first S_2 and then S_1), then the profit would increase



User	v	u	w
Cost	\$1.5	\$0.5	\$1

Figure 2: Counter-example of monotonicity

from \$0.5 to \$2. In general, the *host profit maximization* problem is *non-monotone* with respect to the addition of seed sets. □

Theorem 2. **(Submodularity.)** *The host profit maximization is submodular under the DSS propagation model.*

Proof. Let $\mathbb{S} = \{S_1, ..., S_i, ..., S_{|\mathcal{H}|}\}$ and $\mathbb{S}' = \{S'_1, ..., S'_i, ..., S'_{|\mathcal{H}|}\}$ be two seed sets such that $S_i \subseteq S'_i, \forall 1 \le i \le |\mathcal{H}|$. And we denote the marginal profit gain of adding a user v (i.e., $v \in V - \mathbb{S}'$) to S_i in $\mathbb{S}$ as $P(v|\mathbb{S}) = P(v|S_i) = \frac{B_i\cdot\gamma}{I_i}\sigma_{\mathcal{G}}(v|S_i) - c(v)$, and that of adding v to S'_i in $\mathbb{S}'$ is $P(v|\mathbb{S}') = \frac{B_i\cdot\gamma}{I_i}\sigma_{\mathcal{G}}(v|S'_i) - c(v)$, where $\sigma_{\mathcal{G}}(v|S_i)$ ($\sigma_{\mathcal{G}}(v|S'_i)$) denotes the marginal influence gain of adding v to S_i (S'_i).

For any two seed sets $\mathbb{S}$ and $\mathbb{S}'$ (where $S_i \subseteq S'_i$) and any node $v \in V - \mathbb{S}'$. Considering the three phases included in the DSS propagation model, there are three cases when adding v into S_i and S'_i (details can be found in the Appendix of our extended version [11]). Then due to Kempe et al. [26] has proved that influence function $\sigma(\cdot)$ is *submodular* under the LT model, we prove that the marginal gain of adding a user v to seed set $S'_i \in \mathbb{S}'$ is no larger than that of adding v into $S_i \in \mathbb{S}$, i.e., $P(S_1)+...+P(S_i \cup v)+...+P(S_{|\mathcal{H}|}) - P(\mathbb{S}) \ge P(S'_1) + ... + P(S'_i \cup v) + ... + P(S'_{|\mathcal{H}|}) - P(\mathbb{S}')$. We take the weighted sum over all possible worlds, and conclude that our problem is *submodular* under the DSS model. □

Theorem 3. **(Problem hardness.)** *The host profit maximization is NP-hard and is NP-hard to approximate within any factor.*

Proof. We first prove the hardness of our problem using a reduction from the 3-PARTITION problem (*3PM*) [16], and then illustrate it is also **NP**-hard to approximate within any factor. Details can be found in the Appendix of our extended version [11]. □

3 HOST PROFIT MAXIMIZATION

In this section, we first revisit ROI-Greedy [24] algorithm for single merchant profit maximization, then extend it to adapt to our multiple merchants' case, denoted as Fill-Greedy, while non-trivially maintaining its approximation guarantee (§ 3.1). Since the efficient implementation of Fill-Greedy is challenging, we then devise the scalable version by leveraging the notion of *random reverse reachable sets* [8], which also comes with a theoretical guarantee (§ 3.2).

3.1 The Fill-Greedy Algorithm

Revisiting ROI-Greedy. Jin et al. [24] proposed ROI-Greedy algorithm to solve the well-known unconstrained submodular maximization with modular costs (*USM-MC*) [10, 20, 24, 52], whose representative instance is single-merchant profit maximization. ROI-Greedy starts from $S = \emptyset$, iteratively selects the user $v \in V\setminus S$ that maximizes $\frac{\sigma(v|S)}{c(v)}$ and inserts it into S if it satisfies $\sigma(v|S) > c(v)$. ROI-Greedy terminates when no users in $V\setminus S$ can satisfy the condition $\sigma(v|S) > c(v)$. ROI-Greedy ensures a strong approximation guarantee, that is, $f(S_i) - c(S_i) \ge f(S^*_i) - c(S^*_i) - \ln\frac{f(S^*_i)}{c(S^*_i)}\cdot c(S^*_i)$,

Algorithm 1 Fill-Greedy

Input: $\mathcal{H}, V, \gamma_r, \gamma_p$
Output: $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$
1: Initialize $\mathbb{S} = \{\emptyset_1, \emptyset_2, ..., \emptyset_{|\mathcal{H}|}\}$
2: Assign each merchant $\gamma_i = \gamma_p$
3: $\mathcal{M} \leftarrow \{(v, i) : (v, i) \in V \times [|\mathcal{H}|]\}$
4: **while** $\mathcal{M} \neq \emptyset$ **do**
5: $\quad (v^*, i^*) \leftarrow \arg\max_{(v,i)\in\mathcal{M}} \frac{\frac{B_i}{I_i}\gamma_i \cdot \sigma(v|\mathbb{S})}{c(v)}$
6: $\quad \mathcal{M} \leftarrow \mathcal{M} - \{(v^*, i^*)\}$
7: $\quad$ **if** $v^* \in \bigcup_{i\in[|\mathcal{H}|]} S_i$ **then continue;**
8: $\quad$ **if** $\frac{B_{i^*}}{I_{i^*}}\gamma_{i^*}\sigma(v^*|\mathbb{S}) - c(v^*) \leq 0$ **then continue;**
9: $\quad S_{i^*} \leftarrow S_{i^*} \cup \{v^*\}$
10: $\quad$ **if** $\sigma(S_{i^*}) \geq I_{i^*}$ **then**
11: $\quad\quad \gamma_{i^*} = \gamma_r$
12: **Return** $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$

Algorithm 2 CandGeneration

Input: $\mathcal{H}, V, \gamma_p, \gamma_r$
Output: $\boldsymbol{T}$
1: Initialize $\boldsymbol{T} = \emptyset, \boldsymbol{\eta} = \emptyset$
2: Compute each merchant $\eta_i = \frac{B_i}{I_i} \times \max\{\gamma_p, \gamma_r\}$, $\boldsymbol{\eta} \leftarrow \eta_i$
3: $\eta_{\max} \leftarrow \arg\max_{\eta_i\in\boldsymbol{\eta}} \eta_i$
4: **while** $V \neq \emptyset$ **do**
5: $\quad v \leftarrow \arg\max_{u\in V\setminus \boldsymbol{T}} \frac{\sigma(u|T)}{c(u)}$
6: $\quad$ **if** $\eta_{\max} \cdot \sigma(v|T) - c(v) > 0$ **then**
7: $\quad\quad T \leftarrow T \cup \{v\}, V \leftarrow V\setminus\{v\}$
8: $\quad$ **else**
9: $\quad\quad$ **break;**
10: **Return** $\boldsymbol{T}$

where S_i^* is the optimal solution to *USM-MC*. Inspired by the ROI-Greedy, we design Fill-Greedy to allocate seed sets to multiple merchants to maximize the host's overall profit as following. Note that, Fill-Greedy is a non-trivial extension of ROI-Greedy since **(1)** an additional seed allocation strategy among multiple merchants requires meticulous design, and **(2)** more complex proofs of the theoretical guarantee need to be taken into consideration.

Fill-Greedy. Algorithm 1 presents the pseudo-code of Fill-Greedy algorithm. First, we initialize an empty seed set for each merchant (Line 1) together with her proposed γ_p (Line 2). $\mathcal{M} \subseteq V \times [|\mathcal{H}|]$ to denote the set of (user, merchant) candidate pairs (Line 3). In each step, we greedily select the element (v^*, i^*) that increases the profit maximally (i.e., maximizing $\frac{\frac{B_i}{I_i}\gamma_i\cdot\sigma(v|\mathbb{S})}{c(v)}$) (Line 5) and removing it from $\mathcal{M}$ (Line 6), then the picked user v^* is added into S_{i^*} if and only if both conditions are satisfied (Lines 7–8): **(1)** the user v^* has not been assigned to any merchant yet; **(2)** profit marginal gain of (v^*, i^*) is positive. If the influence spread of S_{i^*} after inserting user v^* exceeds h_{i^*}'s threshold I_{i^*}, we set $\gamma_{i^*} = \gamma_r$ (Lines 10–11), which denotes the exceed influence spread will be rewarded with $\frac{B_{i^*}}{I_{i^*}}\gamma_r$ per influenced user. The process terminates when $\mathcal{M}$ is empty (Line 2). The performance of Fill-Greedy is guaranteed by Theorem 4.

Theorem 4. **(Approximation Guarantee).** *For the host profit maximization problem, suppose that Algorithm* Fill-Greedy *returns* $\mathbb{S}$. *Then we have*[6]*:*

$$P(\mathbb{S}) \geq P\left(\mathbb{S}^o\right) - |\mathcal{H}| \cdot \ln\frac{R\left(\mathbb{S}^o\right)}{C\left(\mathbb{S}^o\right)} \cdot C\left(\mathbb{S}^o\right) \tag{4}$$

where $\mathbb{S}^{\boldsymbol{o}} = \{S_1^o, S_2^o, ..., S_{|\mathcal{H}|}^o\}$ is the optimal solution to our problem and S_i^o is the optimal seed set to each merchant. Let $P(\mathbb{S}^o) = R(\mathbb{S}^o) - C(\mathbb{S}^o)$, where $R(\mathbb{S}^o) = \sum_{S_i^o\in\mathbb{S}^o} R(S_i^o)$ and $C(\mathbb{S}^o) = \sum_{S_i^o\in\mathbb{S}^o} C(S_i^o)$ based on Definition 1 and Definition 2.

Intuitively, $\mathcal{M} \subseteq V \times [|\mathcal{H}|]$ can be quite large (i.e., for *NetHEPT* network, $|\mathcal{M}| = 76145$ if $|\mathcal{H}| = 5$), rendering Algorithm 1 from being efficient on large-scale social graphs. Therefore, we propose Algorithm 2 to prune the search space in Algorithm 1, by replacing the whole user set V with the set of candidate user T that are potentially to be selected as seeds (i.e., utilizing T, $|\mathcal{M}|$ is reduced to 47110). Suppose there is a super merchant, we apply ROI-Greedy to select users that satisfy the loosest requirement (Line 6), such that all the potential seed users can be selected into T. The main difference between CandGeneration and ROI-Greedy lies in the metric to decide whether a selected user can be inserted into T: in each iteration, it chooses the user $v \in V\setminus T$ whose maximum revenue marginal gain is larger than its cost, i.e., the maximum profit marginal gain of v is positive ($\eta_{\max} \cdot \sigma(v|T) - c(v) > 0$) (Line 6), where $\eta_{\max} = \arg\max_{i\in|\mathcal{H}|} \frac{B_i}{I_i} \times \max\{\gamma_p, \gamma_r\}$ (Lines 2–3), which ensures the largest possible revenue marginal gain of v.

3.2 Scalable Host Profit Maximization

Algorithm 1 (Fill-Greedy) involves a huge number of influence spread computations to find the user for each merchant that yields the maximum increase in profit $P(S_i)$. However, given any seed set O, computing its exact influence spread $\sigma(O)$ under the LT model is #P-hard [14]. Recent research focuses on sampling-based influence spread estimation, ranging from naive *Monte Carlo (MC)* simulations [26] to advanced *reverse influence sampling* (*RIS*) [8]. Note that the state-of-the-art algorithm [24] for a single merchant profit maximization is also equipped with *reverse influence sampling* technique. Each sampled *reverse reachable (RR)* set from *RIS* is denoted as R, which is a subset of V conceptually generated as follows:

(1) Generate a random graph G' form G by independently removing each edge $(u, v) \in E$ with probability $1 - \sum_{(u,v)\in E} w_{u,v}$.
(2) Select a user $v \in V$ uniformly at random from G'.
(3) R is the set of users reversely reachable from v in G'.

Given any user set O and a random RR set R, we define a random variable $Y(O, R)$ such that $Y(O, R) = 1$ if $O \cap R \neq \emptyset$ and $Y(O, R) = 0$ otherwise. Tang et al. [55] show that $\sigma(O)$ under the LT model equals $n \cdot \mathbb{E}[Y(O, R)]$. Given a set $\mathcal{R} = \{R_1, R_2, ...\}$ of RR sets, $n \cdot \mathbb{E}[Y(O, R)]$ could be unbiasedly estimated by the empirical mean $\sum_{R\in\mathcal{R}} Y(O, R)/|\mathcal{R}|$ based on concentration bounds.

In our problem, we need to design a method to estimate $R(\mathbb{S}) = \sum_{i\in|\mathcal{H}|} B_i(1 + \gamma\frac{\sigma(S_i)-I_i}{I_i})$ for any solution $\mathbb{S} = (S_1, ..., S_{|\mathcal{H}|})$ to our problem. Existing works [3, 4, 19] generate a set $\mathcal{R}_i$ of random RR sets for each merchant $i \in [|\mathcal{H}|]$ with $|\mathcal{R}_1| = |\mathcal{R}_2| = ... = |\mathcal{R}_{|\mathcal{H}|}|$, such that $\sigma(S_i)$ can be estimated using $\mathcal{R}_i$ for each $i \in [|\mathcal{H}|]$, assuming that each user on a social graph can be influenced by multiple products and spread the information to their neighbors. However, according to § 2.1, we take into account that each user could adopt

[6] All the omitted proofs can be found in the Appendix of our extended version [11].

Algorithm 3 Multi-Profit Maximization (MPM)

Input: $\mathcal{H}, V, T, \epsilon, \delta$
Output: $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$

```
1: Initialize 𝕊 = {∅_1, ∅_2, ..., ∅_|H|}, θ_1 ← n, i ← 1
2: while θ_i ≤ θ_max do
3:     Generate two sets of random RR sets, |R_1| = |R_2| = θ_i
4:     𝕊 ← Fill-Oracle(R_1)
5:     β ← (R^{R_1}(𝕊) − C(𝕊))/(R^{R_2}(𝕊) − C(𝕊))
6:     (ε_1 + 1)(ε_1 + 2)/ε_1^2 = R^{R_2}(𝕊)/(5 · i^2/δ) · θ_i/(n · Γ)
7:     (2ε_2 + 2)/ε_2^2 = (R^{R_2}(𝕊) − C(𝕊))/(5 · i^2/δ) · θ_i/(n · Γ)
8:     if (β − 1)/β + ε_1 + ε_2 ≤ ε, ε_1 + ε_2 ≤ ε, β, ε_1, ε_2 > 0 then
9:         break
10:    i ← i + 1, double the sizes of R_1 and R_2 with new random RR sets
11: Return 𝕊 = {S_1, S_2, ..., S_|H|}
```

Algorithm 4 Fill-Oracle

Input: RR sets $\mathcal{R}$
Output: $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$

```
1: Initialize 𝕊 = {∅_1, ∅_2, ..., ∅_|H|}
2: Assign each merchant γ_i = γ_p
3: M ← {(v, i) : (v, i) ∈ T × [|H|]}
4: Let C_R(v) be the number of RR sets covered by v in R
5: while M ≠ ∅ do
6:     (v', i') ← arg max_{(v,i)∈M} ((B_i/I_i) γ_i C_R(v) / c(v)) · n/|R|
7:     M ← M − {(v', i')}
8:     if v' ∈ ⋃_{i∈[|H|]} S_i then continue;
9:     if (B_i'/I_i') γ_i' C_R(v') · n/|R| − c(v') ≤ 0 then continue;
10:    S_i' ← S_i' ∪ {v'}
11:    if n/|R| · C_R(S_i', R) ≥ I_i' then
12:        γ_i' = γ_r
13:    Remove form R all RR sets that are covered by v'
14: Return 𝕊 = {S_1, S_2, ..., S_|H|}
```

and spread at most one product while she can switch adoption in the propagation process, i.e., multiple merchants share with a whole social graph. Based on this, we generate a set of random RR sets $\mathcal{R}$ for all merchants, which is the same as the classical *RIS* approach. Since the current unbiased estimation methods fail to meet the requirements of the switching phase in the propagation of our model, we design a novel method to fill this gap.

Given $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$ and a random RR set R, we define a random variable $C_R(S_i, R)$ such that $C_R(S_i, R) = k_i/|R|$ and $C_R(\mathbb{S}, R) = \sum_{i\in|\mathcal{H}|}(k_i/|R|) = (\sum_{i\in|\mathcal{H}|} k_i)/|R| = |R|/|R| = 1$ if there exists a seed set $S_i \in \mathbb{S}$ intersects R, where k_i denotes the number of users influenced by S_i in R. Otherwise, $C_R(\mathbb{S}, R) = 0$. Given a set $\mathcal{R}$ of random RR sets, we denote $R^{\mathcal{R}}(S_i) = B_i(1 + \gamma\frac{C_{\mathcal{R}}(S_i,\mathcal{R})n/|\mathcal{R}|-I_i}{I_i})$ as an unbiased estimation of $R(S_i)$ for any $i \in |\mathcal{H}|$, where $C_{\mathcal{R}}(S_i, \mathcal{R}) = \sum_{R\in\mathcal{R}} k_i/|R|$. To add them up, we have $P^{\mathcal{R}}(\mathbb{S}) = \sum_{i\in|\mathcal{H}|}(R^{\mathcal{R}}(S_i) - C(S_i))$ as an unbiased estimation of $P(\mathbb{S})$. Note that, considering users may switch adoption after being activated, k_i will be updated once a new seed user is generated.

Based on our Fill-Greedy solution and *RIS* technique, we propose the MPM algorithm, an efficient approach to tackle our problem with approximation guarantee as Theorem 5.

Theorem 5. **(Approximation Guarantee of MPM).** *With probability at least* $1 - \delta$ *for any* $\delta \in (0, 1)$*, MPM returns a solution* $\mathbb{S}$ *satisfies*

$$P(\mathbb{S}) \geq (1 - \epsilon)R\left(\mathbb{S}^o\right) - C\left(\mathbb{S}^o\right) - |\mathcal{H}| \cdot \ln\frac{R\left(\mathbb{S}^o\right)}{C\left(\mathbb{S}^o\right)} \cdot C\left(\mathbb{S}^o\right) \quad (5)$$

where $\mathbb{S}^o$ is the optimal solution and $\delta, \epsilon \in (0, 1)$ are input parameters. Algorithm 3 shows the pseudo-code of MPM, while Algorithm 4 (Fill-Oracle) demonstrates a sub-routine invoked. Algorithm 4 estimates $\sigma(v|\mathbb{S})$ via *RIS*-based method to tackle the challenge that MPM cannot compute the exact value of $\sigma(v|\mathbb{S})$ in polynomial time. Since it is difficult to decide the size of $\mathcal{R}$ for achieving the profit guarantee in Eq. (5) without excessive computational overheads, inspired from [24], we use a trial-and-error method to overcome this hurdle.

Algorithm 3 first generates two collections of RR sets (i.e., $\mathcal{R}_1$ and $\mathcal{R}_2$) with $\mathcal{R}_1 = \mathcal{R}_2 = n$ (Lines 1–3). Then, it uses $\mathcal{R}_1$ as the input to the Fill-Oracle algorithm, which generates a solution $\mathbb{S}$ by employing the greedy method in Fill-Greedy (Line 4). Afterwords, it uses $\mathcal{R}_2$ to verify the quality of solution $\mathbb{S}$ (Lines 6–9) since $\mathcal{R}_2$ is independent of $\mathcal{R}_1$. Intuitively, due to the *Cost Function* is a modular function and $C(\mathbb{S})$ is always the same no matter on $\mathcal{R}_1$ or $\mathcal{R}_2$, we suppose that if the estimation profit from $\mathcal{R}_2$ (i.e., $R^{\mathcal{R}_2}(\mathbb{S}) - c(\mathbb{S})$) is much smaller than the estimation derived from $\mathcal{R}_1$ (i.e., $R^{\mathcal{R}_1}(\mathbb{S}) - c(\mathbb{S})$), it means that $\mathcal{R}_1$ over-estimates $\mathbb{S}$'s profit. In this circumcise, MPM discards solution $\mathbb{S}$, doubles the size of $\mathcal{R}_1$ and $\mathcal{R}_2$ (Line 10) and repeats the above process until a satisfying solution is returned, i.e., **(1)** $\mathcal{R}_2$ agrees the quality of $\mathbb{S}$ generated by $\mathcal{R}_1$ (Lines 8–9), or **(2)** the number of generated RR sets reaches $\theta_{\max}$ (Line 2), where $\theta_{\max} = (8 + 2\epsilon)(1 + \epsilon_1)\, n \frac{\ln\frac{6}{\delta} + \sum_{i\in|\mathcal{H}|}\tau_i \ln\frac{2n}{\tau_i}}{\epsilon^2 \max\{1, R^{\mathcal{R}_2}(\mathbb{S}) - (1+\epsilon_1)C(\mathbb{S})\}}$, τ_i is the maximum number of users that can be selected by merchant h_i. Finally, Algorithm 3 terminates with $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$ (Line 12).

In what follows, we first present key Lemmas used to prove Theorem 5, while tackling two key challenges that existed in Algorithm 3, that is, **(1)** how to set the maximum number of RR sets $\theta_{\max}$ (Line 2) and **(2)** how to set conditions to evaluate whether the current solution satisfies the performance guarantee (Lines 6–9), such that the profit guarantee of MPM can be achieved as fast as possible. After that, we analyze the time complexity of MPM.

Using *Chernoff Inequalities* [7] [37], we prove that in each round of Algorithm 3 (Lines 3–10), the estimations $R^{\mathcal{R}_1}(\mathbb{S})$ and $R^{\mathcal{R}_2}(\mathbb{S})$ are concentration bounds with a high probability.

Lemma 1. *With probability at least 1-*$\frac{2\delta}{3}$*, for each iteration of Algorithm 3, where* $\epsilon_1, \epsilon_2, \beta > 0$*, we have*

$$R^{\mathcal{R}_2}(\mathbb{S}) \leq (1 + \epsilon_1) R(\mathbb{S}) \quad (6)$$

$$R^{\mathcal{R}_1}\left(\mathbb{S}^o\right) \geq (1 - \epsilon_2) R\left(\mathbb{S}^o\right) \quad (7)$$

Based on Lemma 1, we prove that Eq. (5) holds with probability at least $1 - \frac{2\delta}{3}$ when Line 8 in Algorithm 3 is satisfied and $1 - \delta$ when Line 8 is not satisfied. Combining these two cases, the approximation guarantee of MPM is demonstrated. And the theoretical time complexity of MPM can be known as below.

[7] We introduce Chernoff Inequalities in the Appendix of our extended version [11].

Table 1: Merchants' Contracts

$\mathcal{H}$	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}
B_i	9000	9000	7500	6000	7500	12000	9000	6000	7200	6000
I_i	7500	6000	5000	6000	7500	8000	9000	5000	6000	5000
BPI_i	1.2	1.5	1.5	1.0	1.0	1.5	1.0	1.2	1.2	1.2

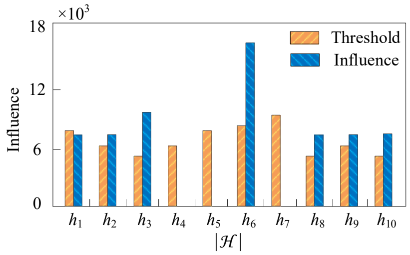


Figure 3: Influence spread comparison using Algorithm 3

Theorem 6. **(Time Complexity of MPM).** *The expected time complexity of* MPM *is* $O(\frac{m\sum_{i\in[|\mathcal{H}|]}\mathbb{E}[P_i(\{v^*\})](\ln\frac{1}{\delta}+n\ln|\mathcal{H}|)}{\epsilon^2})$*, where* v^* *is a random user selected from* G *with probability proportional to its in-degree.*

4 BALANCE-SENSITIVE ALGORITHM

In this section, we present a more practical application based on previous proposed approaches as our side contribution. We first highlight the significance of balancing in practical scenarios (§ 4.1). Then, we introduce two efficient heuristic methods to balance the distribution of adoption results among merchants (§ 4.2 and § 4.3).

4.1 Motivation

The proposed algorithms in § 3 guarantee that the host can gain maximum profit from multiple merchants. However, a host purely pursues profit maximization may fall into a trap that, as the *benefit per influence* of each merchant is significantly different from each other, the supplied influence of some merchants may be far from their required threshold while some of the merchants' influence is far exceeded. That is, the host will dramatically sacrifice some merchants to achieve a larger profit. In Example 2, we give an example via applying Algorithm 3 on a real-world social network *Epinions*. Due to the distribution of influence spread provided by the host being seriously imbalanced, it may harm the reputation of the host and her long-term business cooperation with certain merchants whose requirements are far from being satisfied. In order to improve the algorithms to practical scenarios, we design two heuristic approaches to balance the distribution of adoptions among merchants without largely reducing the host profit.

Example 2. *We consider ten merchants* $\mathcal{H}=\{h_1,h_2,...,h_{10}\}$ *participating in a market campaign, with each requesting demanded influence (threshold)* I_i*, the payment* B_i *it is willing to pay if the demanded influence is satisfied, and benefit per influence* BPI_i *(i.e.,* $BPI_i=B_i/I_i$*) as listed in Table 1. We apply Algorithm 3 to deploy a set of seed users* S_i *to each merchant to satisfy its requirement, while maximizing the profit earned by host. The distribution result of influence are plotted in Figure 3, we can see that host only select seed users for merchants with BPI of 1.5 and 1.2, especially for merchants with BPI* $=1.5$ *(i.e.,* h_2*,* h_3 *and* h_6*), the influence of these merchants are far exceeding their thresholds. However, for those merchants with BPI* $=1.0$ *(i.e.,* h_4*,* h_5 *and* h_7*), host provides them with zero influence, which constitutes a really imbalanced distribution.*

Algorithm 5 Merchant-Driven Multi-Round Search (OBO)

Input: $\mathcal{H}$, V, γ_r, γ_p
Output: $\mathbb{S}=\{S_1,S_2,...,S_{|\mathcal{H}|}\}$
1: Initialize $\mathbb{S}=\{\emptyset_1,\emptyset_2,...,\emptyset_{|\mathcal{H}|}\}$
2: Assign each merchant $\gamma_i=\gamma_p$
3: Order merchants based on decreasing order of $\frac{B_i}{I_i}\gamma_i$
4: **while** $V\neq\emptyset\cup\mathcal{H}\neq\emptyset$ **do**
5:     **for** each $h_i\in\mathcal{H}$ **do**
6:         $v\leftarrow\arg\max_{u\in V\setminus\mathbb{S}}\frac{\frac{B_i}{I_i}\gamma_i\sigma(u|S_i)}{c(u)}$
7:         **if** $\frac{B_i}{I_i}\gamma_i\sigma(v|S_i)-c(v)>0$ **then**
8:             $S_i\leftarrow S_i\cup\{v\}$, $V\leftarrow V\setminus\{v\}$
9:         **else** $\mathcal{H}\leftarrow\mathcal{H}\setminus\{h_i\}$
10:         **if** $\sigma(S_i)\geq I_i$ **then**
11:             $\gamma_i=\gamma_r$
12: **Return** $\mathbb{S}=\{S_1,S_2,...,S_{|\mathcal{H}|}\}$

4.2 Merchant-Driven Multi-Round Search

Our OBO approach is presented in Algorithm 5. We first initialize an empty set of seed sets for each merchant (Line 1) and assign γ_r to each merchant (Line 2) the same as lines 1–2 of Algorithm 1. Subsequently, since it is trivial to see that merchants with larger *benefit per influence* contribute higher profit to host, we preferentially select seed sets for these merchants. Thus, we order merchants based on decreasing order of $\frac{B_i}{I_i}\gamma_i$ (Line 3). For each merchant $h_i\in\mathcal{H}$, we select user v who has not been assigned to any merchant yet and can best increase the profit of h_i (i.e., maximizing $(\frac{B_i/I_i\cdot\gamma_i\cdot\sigma(u|S_i)}{c(u)})$ (Lines 5–6), then add v into S_i if the profit marginal gain of v is positive (Lines 7–8). Otherwise, we discard h_i from $\mathcal{H}$ as there exist no user can yield positive profit marginal gain to h_i (Lines 9). Next, if influence spread of S_i after inserting user v exceeds h_i's threshold I_i, we set $\gamma_i=\gamma_r$ (Lines 10–11) as stated in Algorithm 1. This process terminates when V or $\mathcal{H}$ is empty.

4.3 Profit-Influence Iterative Search

The workflow of our proposed iterative framework is given in Algorithm 6. We first initialize an empty set of seed sets for each merchant and an empty merchant set $\mathcal{H}'$ including merchants whose demands have not been satisfied (Line 1). Next, we assign γ_r to each merchant (Line 2) and construct a set $\mathcal{M}'\subseteq V\times[|\mathcal{H}|]$ of (user, merchant) candidate pairs (Line 3). Then, the framework alternatively and in an iterative manner, selects element (user, merchant) that user can best increase the profit of merchant in Profit Batch (Lines 5–11), and user can increase influence spread of merchant most in Influence Batch (Lines 12–22). Specifically, the Profit Batch is the same as Lines 5–11 of Algorithm 1. In Influence Batch, we first pick merchant h_j whose influence spread has not reached its threshold and insert h_j into $\mathcal{H}'$ (Lines 13–14). If $\mathcal{H}'$ is not empty, we select merchant h_k with minimum influence satisfied ratio (i.e., $\min_{h_j\in\mathcal{H}}(\sigma(S_j)/I_j)$) (Line 16), and pick user w that maximizes influence spread of h_k (Line 17). What's more, if all the merchants' influence spread have been satisfied (i.e., $\mathcal{H}'$ is empty), we set $\mathcal{B}_I=0$ and exit current Influence Batch (Line 18–19). Finally, we add user w into S_k if the profit marginal gain of (w,k) is positive (Lines 21–22). After no user can yield positive profit marginal gain or $\mathcal{M}'$ is empty, the seed sets $\mathbb{S}$ will be returned (Line 23).

Algorithm 6 Profit-Influence Iterative Search (ITER)

Input: $\mathcal{H}$, V, γ_r, γ_p, $\mathcal{B}_P$, $\mathcal{B}_I$
Output: $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$

Initialize $\mathbb{S} = \{\emptyset_1, \emptyset_2, ..., \emptyset_{|\mathcal{H}|}\}$, $\mathcal{H}' = \emptyset$
Assign each merchant $\gamma_i = \gamma_p$
$\mathcal{M}' \leftarrow \{(v,i) : (v,i) \in V \times [|\mathcal{H}|]\}$
while $\mathcal{M}' \neq \emptyset$ **do**
  for $\xi \leftarrow 1$ to $\mathcal{B}_P$ **do**
    $(u,t) \leftarrow \arg\max_{(v,i)\in\mathcal{M}'} \frac{\frac{B_i}{I_i}\gamma_i \cdot \sigma(v|\mathbb{S})}{c(v)}$
    $\mathcal{M}' \leftarrow \mathcal{M}' - \{(u,t)\}$
    if $u \in \bigcup_{i\in[|\mathcal{H}|]} S_i$ **then continue;**
    if $\frac{B_t}{I_t}\gamma_t\sigma(u|\mathbb{S}) - c(u) \leq 0$ **then continue;**
    $S_t \leftarrow S_t \cup \{u\}$
    if $\sigma(S_t) \geq I_t$ **then** $\gamma_t = \gamma_r$
  for $\psi \leftarrow 1$ to $\mathcal{B}_I$ **do**
    for $h_j := h_1$ **to** $h_{|\mathcal{H}|}$ **do**
      if $\sigma(S_j) < I_j$ **then** $\mathcal{H}' \leftarrow \mathcal{H}' \cup \{h_j\}$
    if $\mathcal{H}' \neq \emptyset$ **then**
      $h_k \leftarrow \arg\min_{h_j\in\mathcal{H}}(\sigma(S_j)/I_j)$
      $w \leftarrow \arg\max_{v\in V\setminus\mathbb{S}} \sigma(v|S_k)$
    else
      $\mathcal{B}_I \leftarrow 0$, **break;**
    $\mathcal{M}' \leftarrow \mathcal{M}' - \{(w,k)\}$
    if $\frac{B_k}{I_k}\gamma_k\sigma(w|S_k) - c(w) > 0$ **then**
      $S_k \leftarrow S_k \cup \{w\}$
Return $\mathbb{S} = \{S_1, S_2, ..., S_{|\mathcal{H}|}\}$

Table 2: Datasets

Dataset	n	m	Type
NetHEPT	15.2K	62.8K	undirected
Epinions	75.9K	509K	directed
DBLP	317K	1.05M	undirected
LiveJournal	4.8M	69.0M	directed

5 EXPERIMENTS

We empirically evaluate our proposed algorithms and baseline algorithms on four real-world social networks. All methods are implemented in C++ and run on an Intel i7 2.90GHz CPU and 64GB RAM server. All codes can be found in [11].

5.1 Experimental Settings

Datasets. Table 2 presents the basic statistics of four real-world social networks in our evaluations. **(1)** *NetHEPT* [13] is an academic collaboration network. **(2)** *Epinions* [29] is a who-trust-whom online social network of a general consumer review site. **(3)** *DBLP* [29] is a collaborative network where each node indicates an author and edges indicate co-authorship. **(4)** *LiveJournal* [29] is a free online community where users can explicitly declare their friendship.

Models. We use the *Weighted-Cascade model* [26] to set the propagation probability $p(u,v)$ of each edge in G, i.e., $p(u,v)$ is equal to the reverse of the number of v's in-neighbors. In addition, following prior works [22, 24, 53], we adopt the *Degree-Proportional Cost Model* for cost function. In specific, the cost $c(v)$ of node v in G is proportional to its out-degree $d_{out}(v)$: $c(v) = \mu \cdot d_{out}(v)^{\alpha}$, where μ and α are two input parameters When $d_{out}(v) = 0$, we set $c(v) = 1$.

Table 3: Parameter Settings

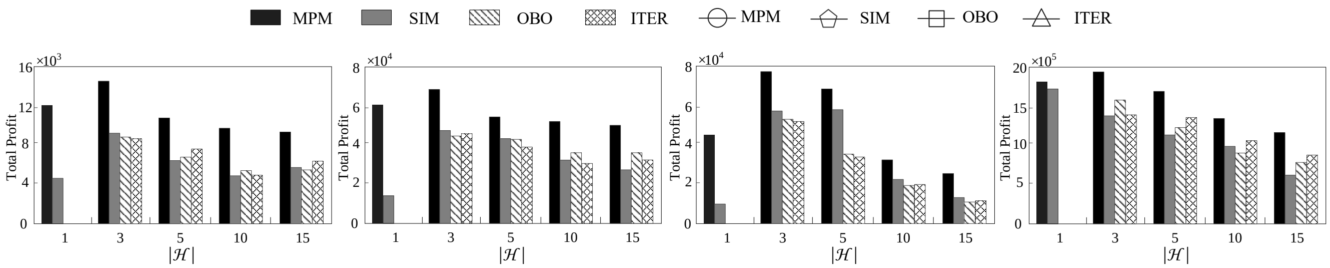
Parameter	Values
$\lvert\mathcal{H}\rvert$	1, 3, **5**, 10, 15
ϵ	0.1, 0.15, **0.2**, 0.25, **0.3**
μ	0.1, **0.2**, 0.3, 0.4, 0.5, 0.6
α	0, 0.2, 0.4, 0.6, 0.8, **1.0**, 1.2
γ_r	0, 0.1, 0.2, **0.3**, 0.4, 0.5
γ_p	0.2, 0.6, **1.0**, 1.4, 1.8

Algorithms. To our best knowledge, this is the first work studying how to maximize the total profit of the host while providing theoretical guarantee in large social graphs. Hence, except for the three methods proposed in this paper, we also extend a state-of-the-art algorithm Simple-Greedy [34, 63] that maximizes profit for a single merchant, such that it could address this problem for multiple merchants. Therefore, we compare four methods listed as follows:
(1) **MPM**: The Multi-Profit Maximization method (§ 3.2).
(2) **SIM**: The extended Simple-Greedy method.
(3) **OBO**: The Merchant-Driven Multi-Round Search method (§ 4.2).
(4) **ITER**: The Profit-Influence Iterative Search method (§ 4.3).
where OBO and ITER can balance the distribution of adoption among merchants. As stated before, MPM can provide an approximation guarantee, while others are heuristics.

Parameters. We summarize the key parameters and their ranges in Table 3. The default values are marked in bold.

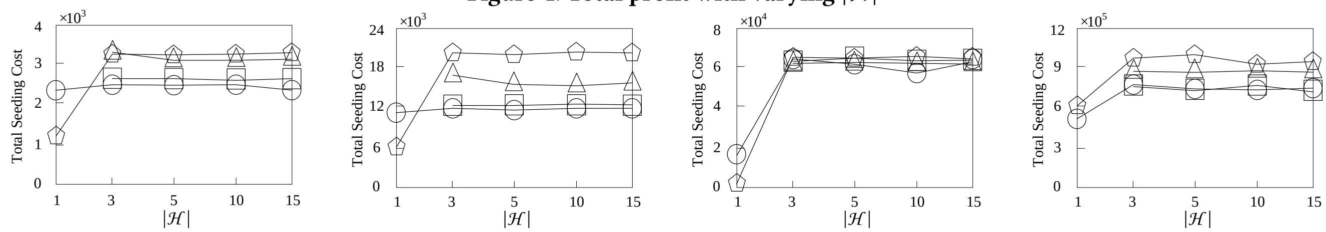
(1) **Failure Probability δ.** We set the failure probability in Algorithm 3 as $\delta = 1/n$, where n denotes the number of nodes in the input graph following prior works [3, 19, 24, 51].
(2) **Sampling Error ϵ.** Following [19, 24], we set $\epsilon = 0.2$ for the *NetHEPT* and *Epinions* datasets, and set $\epsilon = 0.3$ for *DBLP* and *LiveJournal* as default.
(3) **Merchant's Influence Threshold I.** Following the similar setting in [61], the influence threshold of each merchant is generated based on $I_i = \lfloor \omega \cdot \bar{I} \rfloor$, where $\bar{I} = \lfloor n/|\mathcal{H}| \rfloor$ and ω is a factor randomly chosen from 0.5 to 1.5 to simulate different merchant's demand. We assume that the sum of all merchants' influence thresholds does not exceed the number of nodes.
(4) **Merchant's Budget B.** We follow a widely adopted experiment setting in marketing studies [3, 5, 19] that each merchant's budget is proportional to its influence threshold: $B_i = \lfloor \kappa \cdot I_i \rfloor$, where κ is a factor randomly selected from $\{1.0, 1.2, 1.5\}$ to simulate a various budget.
(5) **Values of Profit Batch $\mathcal{B}_P$ and Influence Batch $\mathcal{B}_I$.** We set $\mathcal{B}_P = 10$ and $\mathcal{B}_I = 5$ in Algorithm 6 as default values since we conducted experiments with various values of $\mathcal{B}_P$ and $\mathcal{B}_I$ and observed that the effectiveness results did not vary significantly.
(6) **Reward Ratio γ_r.** We set γ_r no larger than 0.5 since the additional influence spread not always be important. At one extreme (i.e., $\gamma_r = 0$), the host receives no payment reward if the merchant's required influence is satisfied.
(7) **Penalty Ratio γ_p.** we set γ_p to a default value of 1.0, which is the same as CPE model [3, 4, 19]. $\gamma_p < 1$ depicts a minimum revenue clause between merchant and host [31, 46], while $\gamma_p >$ 1 represent harsh earn-out provision.

In all the experiments, we estimate the profit of the algorithms by using $2^4 \times 10^5$ *RR* sets [19, 24], generated independently of the considered algorithms.



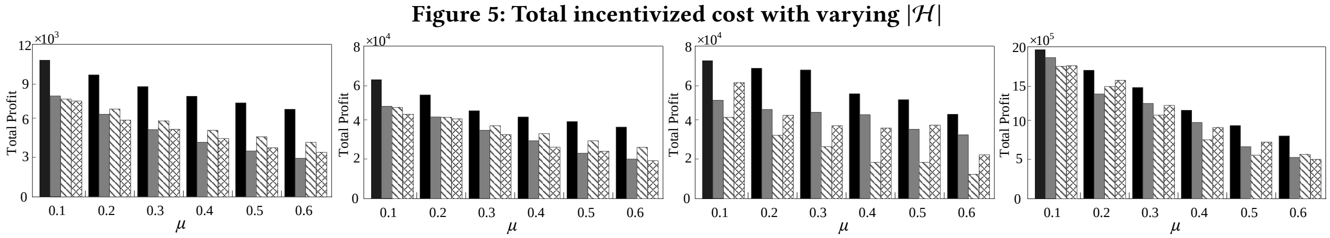
(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

Figure 4: Total profit with varying $|\mathcal{H}|$



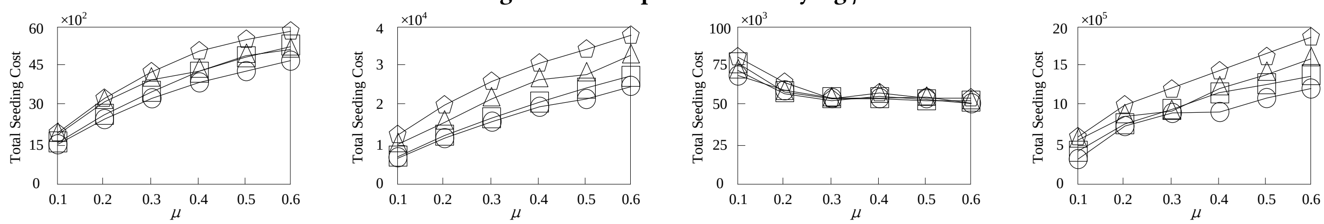
(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

Figure 5: Total incentivized cost with varying $|\mathcal{H}|$



(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

Figure 6: Total profit with varying μ



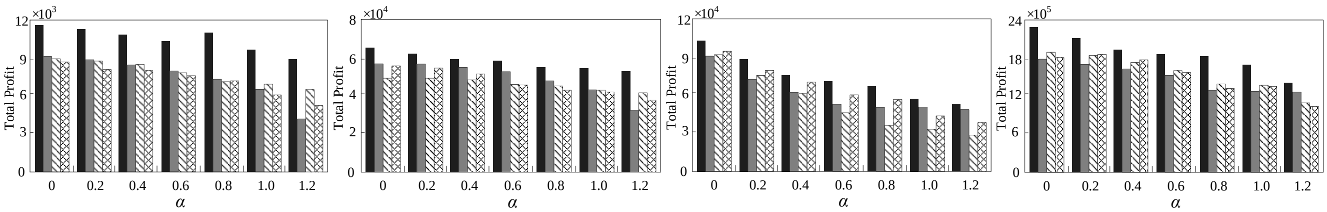
(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

Figure 7: Total incentivized cost with varying μ

5.2 Effectiveness Analyses

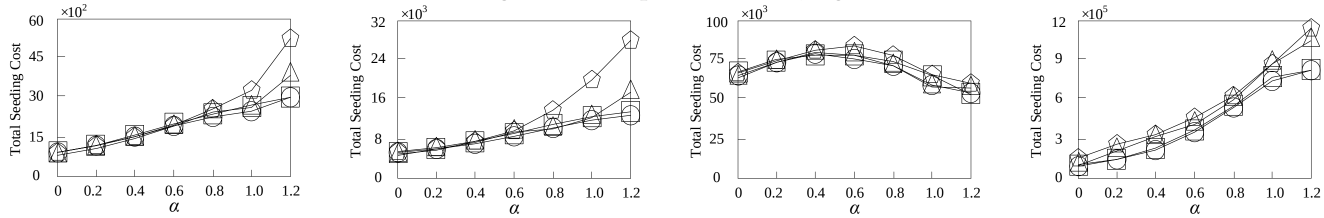
Varying $|\mathcal{H}|$. We vary the number of merchants $|\mathcal{H}|$ from 1 to 15 and report the overall profit and total incentivized cost. Since OBO and ITER are proposed to balance the distribution of influence spread among multiple merchants, we do not consider them in one single merchant case. As shown in Figure 4, MPM attains higher profits than those of all competitors on all datasets. Specifically, for a single merchant ($|\mathcal{H}|=1$), the profit of MPM is much higher than that of SIM, which is consistent with what is reported in [24]. As for multiple merchants ($|\mathcal{H}|>1$), when $|\mathcal{H}|$ increases, the profits of all methods decrease, as the influence demands from merchants become lower, and thus easier to satisfy. Therefore, seed nodes that carry higher marginal profit gain cannot be selected into seed set of merchants with higher *benefit per influence*, due to these merchants have changed γ_p to γ_r ($\gamma_p > \gamma_r$), which results in lower revenue. Figure 5 shows the incentivized cost (i.e., the amount paid by the host for incentivizing seeds) when varying $|\mathcal{H}|$. As expected, the total incentivized cost of MPM is always lower than those of other methods, this is because MPM selects seed nodes by using the return-on-investment selection method in each iteration. The second observation is that when $|\mathcal{H}|>1$ the cost of all methods remains almost stable as $|\mathcal{H}|$ increases since the joint set of seed nodes of all merchants for each method in a graph is almost the same when varying $|\mathcal{H}|$.

Varying μ. We explore the effect of μ, which controls the factor of the cost model. Figure 6 depicts the profits of all methods on four graphs when varying μ from 0.1 to 0.6. MPM gains the highest profit under all settings on four datasets compared to *all* competitors for different μ. When μ increases, the profits of all methods decrease since the cost of every node increase with larger μ, as illustrated in Figure 7. We also observe that the cost of MPM is always lower than those of all competitors over the four datasets. Note that the costs of all methods on *DBLP* do not always increase when μ grows, this is because the costs of nodes in *DBLP* are such large that nodes are



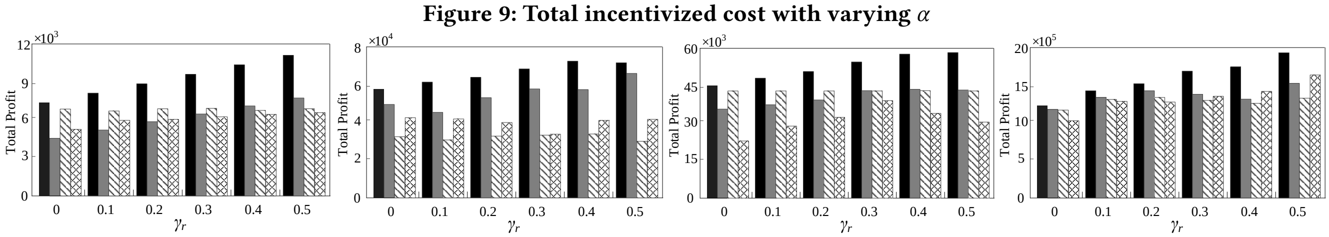
(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

Figure 8: Total profit with varying α



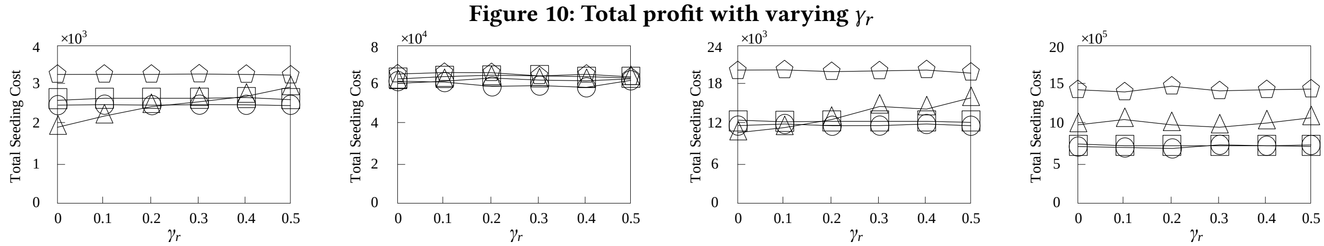
(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

Figure 9: Total incentivized cost with varying α



(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

Figure 10: Total profit with varying γ_r



(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

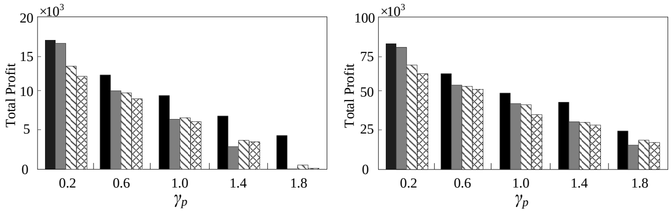
Figure 11: Total incentivized cost with varying γ_r

less likely to satisfy the requirement that marginal revenue gain is larger than the cost (i.e., the marginal profit gain is positive), so fewer seed nodes can be selected into seed sets when μ grows, resulting in fewer total incentivized costs.

Varying α. We investigate the effect of α that controls the index of the degree of the cost model. Figure 8 demonstrates that our MPM always produces the highest profit on *all* datasets under *all* settings, compared to other competitors. The profits of all methods decrease when α grows. The reason is that the costs of all nodes ascend when α increases, as presented in Figure 9. In addition, we observe the incentivized cost of MPM is almost the lowest in Figure 9. Particularly, Figure 9 shows that the cost ascends when $\alpha \leq 0.6$ on *DBLP*, but drops when $\alpha > 0.6$. The reason behind this is that when $\alpha > 0.6$, the costs of nodes are greatly increasing as α grows, and thus more nodes are filtered by the requirement that the marginal profit gain of this node should be positive, which leads to fewer seed nodes contributing lower incentivized costs.

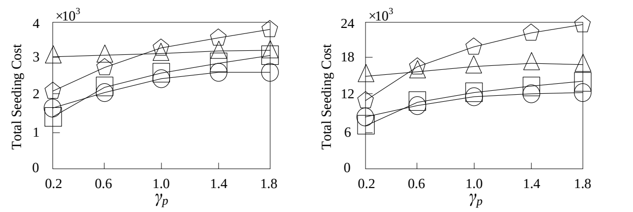
Varying γ_r. Figure 10 presents the profits of all methods when varying γ_r, the reward ratio, from 0 to 0.5. It can be observed that MPM achieves the highest profit over *all* datasets under *all* settings. As γ_r increases, the total profits of all methods ascend. The reason is that when γ_r grows, the host obtains more reward payments from those merchants whose influence spread she provided exceeds their thresholds. In Figure 11, we can see that the cost of all methods grows slightly when γ_r increases. This is because when γ_r is relatively small, the node cannot easily result in positive marginal profit gain due to the low revenue margin, which leads to fewer nodes being inserted into the seed sets. Nevertheless, the total seed cost is not significantly affected as γ_r varies, this is because that the overall seed nodes for all merchants of each method are almost the same when varying γ_r.

Varying γ_p. We demonstrate the impact of the penalty ratio, γ_p in Figure 12 and Figure 13. We only report the results of *NetHEPT* and *Epinions* graphs due to space limits. In Figure 12, the total profit of MPM is consistently higher than that of all competitors, which illustrates the superiority of MPM. Another observation is that the overall profits of all methods descend as γ_p increases, this is because when γ_p grows, the host will be punished more by the merchant



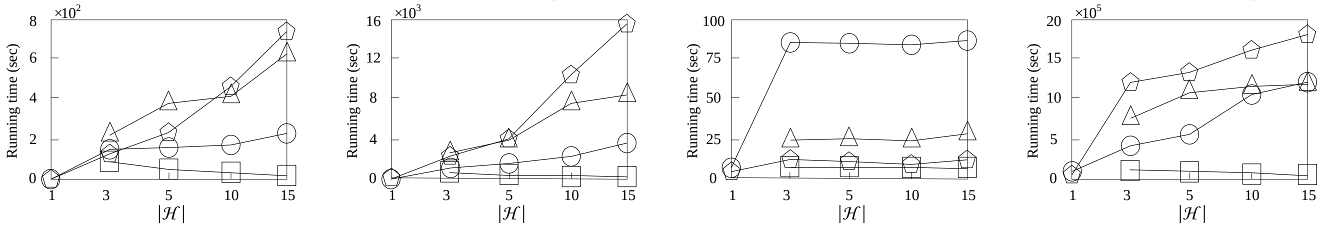
(a) NetHEPT (b) Epinions

Figure 12: Total profit with varying γ_p



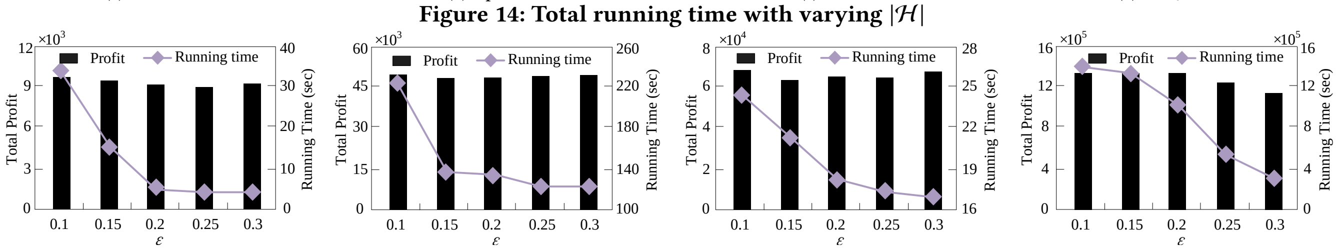
(a) NetHEPT (b) Epinions

Figure 13: Total incentivized cost with varying γ_p



(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

Figure 14: Total running time with varying $|\mathcal{H}|$



(a) NetHEPT (b) Epinions (c) DBLP (d) LiveJournal

Figure 15: The impact of ϵ on total profit and running time

for those partial influence spread that does not reach the threshold when the host cannot satisfy merchant' request, which leads to lower profit. Figure 13 plots that the cost of all algorithms increases as γ_p grows. When γ_p ascends, nodes with higher cost have more chances to be selected into seed sets since they are more likely to satisfy the requirement that marginal profit gain is positive, which results in higher incentivized costs.

5.3 Efficiency Analyses

We present the running time results of $|\mathcal{H}|$ and ϵ on all datasets. Since other parameters do not affect running time significantly, we omit the results due to the space limit.

Varying $|\mathcal{H}|$.We compare the running time of all methods when varying $|\mathcal{H}|$. In Figure 14, it can be observed that the running time of MPM, ITER, and SIM increase with larger $|\mathcal{H}|$. The reason is that when $|\mathcal{H}|$ increases, the number of candidate pairs (i.e., $|\mathcal{M}|$) also grows, leading to more candidate pairs and higher computation overhead. In addition, we observe that MPM runs faster than ITER and SIM in most cases because **(1)** MPM prunes the search space using Algorithm 2 compared to SIM and ITER, and **(2)** does not have Influence Batch which consumes much time compared to ITER. However, OBO runs the fastest of all graphs, since OBO selects seeds for merchants in a one-by-one manner, which reduces to a sequence of simple single merchant seed selection processes. Notice that in *DBLP*, all methods run fast since the costs of nodes in *DBLP* are too large, which filters out numerous nodes and results in less computation time. Also, HPM runs slower than the other three methods since it generates more RR sets to meet the quality requirement, hindering it to be efficient according to Theorem 6.

Varying ϵ. We evaluate the effect of ϵ, the sampling error factor built within the approximation guarantee of MPM (Theorem 5). Since only MPM provides a theoretical guarantee, we compare the total profit (i.e., effectiveness) and running time (i.e., efficiency) of MPM by varying ϵ from 0.1 to 0.3 on four graphs, and for each graph, we use the number of nodes in the graph as the initial number of RR sets in Algorithm 3. Figure 15 presents that the profit does not vary much over the range of values of ϵ. This is because that the theoretical guarantee of MPM depicts the worst-case performance and the actual performance of MPM in real-world cases could be empirically good. Hence, the experiment demonstrates MPM's profit performance is quite stable and robust to the variation of ϵ. In addition, the result shows that the running time decreases when ϵ increases due to the early termination of MPM as ϵ grows (Lines 6–9 of Algorithm 3), which leads to a decrease in the number of generated RR sets. According to Theorem 6, the computation overhead of MPM is dominated by the cost of RR set generation, and hence the running time of MPM descends. Moreover, we observe that running time on *DBLP* is less than that on *Epinions* for all the settings of ϵ. This is because the cost of nodes on *DBLP* is 100 times larger than that of nodes on *Epinions*, so Algorithm 3 filters numerous nodes whose marginal profit gain is hardly assured to be positive, which leads to fewer updates to the RR sets covered by each merchant due to the reduced seed size, and thus accelerating the implementation of MPM.

5.4 Distribution of Influence

We investigate the distribution of influence provided by the host under four methods. Table 4 shows the supplied influence spread of ten merchants of the implemented algorithms on *Epinions* graph. We also list the requests of all merchants, which include influence thresholds, budgets, and corresponding BPIs (i.e., budget/influence).

Table 4: Distribution of Influence on Epinions

Merchant	Budget	Threshold	BPI	Influence				Raio			
				MPM	SIM	OBO	ITER	MPM	SIM	OBO	ITER
h_1	9000	7500	1.20	7066.79	4463.84	3699.03	2655.29	94.22%	59.52%	49.32%	35.40%
h_2	9000	6000	1.50	7112.21	7780.74	5045.46	9271.46	118.54%	129.68%	84.09%	154.52%
h_3	7500	5000	1.50	9269.50	4420.21	4644.57	7798.49	185.39%	88.40%	92.89%	155.97%
h_4	6000	6000	1.00	0.00	0.00	3641.64	274.38	0.00%	0.00%	60.69%	4.57%
h_5	7500	7500	1.00	0.00	0.00	3622.48	344.70	0.00%	0.00%	48.30%	4.60%
h_6	12000	8000	1.50	16070.80	13773.00	3895.77	11295.30	200.89%	172.16%	48.70%	141.19%
h_7	9000	9000	1.00	0.00	0.00	3630.82	407.20	0.00%	0.00%	40.34%	4.52%
h_8	6000	5000	1.20	7038.82	4708.19	3702.83	2608.27	140.78%	94.16%	74.06%	52.17%
h_9	7200	6000	1.20	7065.52	4549.88	3716.03	2611.65	117.76%	75.83%	61.93%	43.53%
h_{10}	6000	5000	1.20	7144.88	4406.71	3717.38	2526.19	142.90%	88.13%	74.35%	50.52%

The result reports that MPM and SIM prefer to satisfy those merchants with higher BPI (i.e., 1.5), and the influence of those merchants with BPI = 1.0 are provided with zero influence. However, OBO and ITER effectively balance the distribution of influence spread among merchants. In particular, OBO shows the best balancing distribution, that is, the influence of merchants is relatively similar to each other. The reason is that OBO selects seed nodes for merchants in a one-by-one manner. ITER presents a similar distribution to MPM, however, ITER optimizes the extreme cases of MPM. For instance, the influence of merchants h_4, h_5 and h_7 are zero under MPM, while under ITER, the influence values of these merchants are 274.38, 344.70, and 407.2, respectively. This is because ITER adds the Influence Batch selection process, in which it selects nodes to best increase the influence of merchants whose requirements are far from being reached. The results on other datasets are qualitatively similar and hence are omitted due to space constraints.

6 RELATED WORK

Influence Maximization. The Influence Maximization (IM) problem was first formulated as a discrete optimization problem by Kempe et al. [26], focusing on two fundamental propagation models (IC and LT model). The IM problem is proved to be **NP**-hard under both models. The $(1-1/e)$-approximation greedy algorithm can be applied to solve IM problem since it is *monotone*, *non-negative*, and *submodular*. Then considerable follow-up research worked on developing more efficient and scalable influence maximization algorithms [17, 18, 22, 39, 51, 54, 55]. Tang et al. [54, 55] utilized RIS with novel heuristics and statistics to reduce the number of RR sets to maintain the $(1-1/e-\epsilon)$-approximation. Tang et al. [51] studied algorithms for online processing of IM efficiently. Guo et al. [17, 18] proposed an efficient random RR set generation algorithm and a solution to reduce the average size of random RR sets. A thorough experimental evaluation of IM algorithms can be found in [2].

Viral Marketing. Viral marketing in online social networks has emerged as an effective way to promote the sales of products and the propagation of information. Recent research studies variants of the IM problem from the perspective of the host (i.e., the owner of the social network), covering both complementary and competitive settings. Complementary viral marketing [5, 33, 38] launches products that tend to be purchased together, while for competitive viral marketing [3, 4, 19, 32, 56], users could adopt at most one product from the collection of similar products. Lu et al. [32] studied the fair seed allocation problem aiming to make each merchant yield a similar influence spread. Han et al. [19] revisited the revenue maximization problem [3, 27] from a fresh perspective and developed novel efficient approximation algorithms with stronger theoretical guarantee. Banerjee et al. studied the complementary [5] and competitive [6] social welfare maximization problem by introducing the concept of utility. [4, 61] investigated the regret minimization problem, which leads to a win-win between the host and the merchants. Other variants with specific constraints are also widely explored [25, 36, 44, 50, 56, 59].

Profit Maximization. Numerous studies tackled revenue maximization assuming that there is a single merchant [20, 24, 34, 52, 53, 63]. Our problem settings have significant differences from prior research: **(1)** Model-wise, [42, 62] simply adopt the K-LT model as we have compared in §2.1, without allowing users to change their mind after activation. To our knowledge, we are the first to model the users' choices changing in influence diffusion to capture the *"comparative shopping"* behavior [12, 49, 57, 58] from an economic perspective. **(2)** Problem-wise, [62] treats the revenue part as profit, without considering the cost of incentivizing seed users as propagation source. In addition, they adopt a fixed seed set size constraint. And in [42], for each merchant, the revenue function is a constant value (i.e., budget). If the influence supplied by the host satisfies the merchant's demand (i.e., threshold), the host will earn the budget, and obtains nothing otherwise. We introduce penalty and reward ratios to simulate a more practical real-world demand.

7 CONCLUSION

In this paper, we study a novel host profit maximization problem for multiple competing products. Each merchant declares her/his campaign proposals including a desired influence demand and corresponding budget, and then the host manages to satisfy the requirements of multiple merchants, aiming to obtain as much profit as possible. A novel information propagation model captures the competing diffusion, and dynamic switch process captures the *"comparative shopping"* behavior [12, 49, 57, 58] from an economic perspective. We prove that our problem is non-monotone, submodular, **NP**-hard, and **NP**-hard to approximate in any constant factor. An effective greedy method and its scalable version, both with approximation guarantees, are devised to tackle our problem. In addition, we propose two heuristics to balance the distribution of influence among merchants without significant loss of overall profit. Extensive experiments on four public datasets demonstrate the superiority of our algorithms in both effectiveness and efficiency.

REFERENCES

[1] Apple. 2022. Retrieved March 27, 2023 from https://www.apple.com/newsroom/2022/09/apple-introduces-iphone-14-and-iphone-14-plus/

[2] Akhil Arora, Sainyam Galhotra, and Sayan Ranu. 2017. Debunking the myths of influence maximization: An in-depth benchmarking study. In *Proceedings of the 2017 ACM SIGMOD International Conference on Management of Data*. 651–666.

[3] Cigdem Aslay, Francesco Bonchi Laks VS Lakshmanan, and Wei Lu. 2017. Revenue Maximization in Incentivized Social Advertising. *Proceedings of the VLDB Endowment* 10, 11 (2017).

[4] Cigdem Aslay, Wei Lu, Francesco Bonchi, Amit Goyal, and Laks VS Lakshmanan. 2015. Viral marketing meets social advertising: ad allocation with minimum regret. *Proceedings of the VLDB Endowment* 8, 7 (2015), 814–825.

[5] Prithu Banerjee, Wei Chen, and Laks VS Lakshmanan. 2019. Maximizing welfare in social networks under a utility driven influence diffusion model. In *Proceedings of the 2019 ACM SIGMOD International Conference on Management of Data*. 1078–1095.

[6] Prithu Banerjee, Wei Chen, and Laks VS Lakshmanan. 2020. Maximizing social welfare in a competitive diffusion model. *Proceedings of the VLDB Endowment* 14, 4 (2020), 613–625.

[7] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. 2012. Topic-Aware Social Influence Propagation Models. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM)*. 81–90.

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

[9] Allan Borodin, Yuval Filmus, and Joel Oren. 2010. Threshold models for competitive influence in social networks. In *International workshop on internet and network economics*. Springer, 539–550.

[10] Niv Buchbinder, Moran Feldman, Joseph Seffi, and Roy Schwartz. 2015. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM J. Comput.* 44, 5 (2015), 1384–1402.

[11] Xueqin Chang, Xiangyu Ke, Chen Lu, Congcong Ge, Ziheng Wei, and Yunjun Gao. 2023. https://github.com/ZJU-DAILY/HPM

[12] Charged. 2022. Google Research: 2 in 3 UK Online Shoppers Compare Before They Buy. Retrieved March 27, 2023 from https://www.chargedretail.co.uk/2022/11/15/google-research-2-in-3-uk-online-shoppers-compare-before-they-buy/

[13] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 199–208.

[14] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *2010 IEEE International Conference on Data Mining (ICDM)*. IEEE, 88–97.

[15] Pedro Domingos and Matt Richardson. 2001. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. 57–66.

[16] Michael R Garey and David S Johnson. 1979. Computers and intractability. *San Francisco: freeman* (1979).

[17] Qintian Guo, Sibo Wang, Zhewei Wei, and Ming Chen. 2020. Influence maximization revisited: Efficient reverse reachable set generation with bound tightened. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2167–2181.

[18] Qintian Guo, Sibo Wang, Zhewei Wei, Wenqing Lin, and Jing Tang. 2022. Influence Maximization Revisited: Efficient Sampling with Bound Tightened. *ACM Transactions on Database Systems (TODS)* (2022).

[19] Kai Han, Benwei Wu, Jing Tang, Shuang Cui, Cigdem Aslay, and Laks VS Lakshmanan. 2021. Efficient and effective algorithms for revenue maximization in social advertising. In *Proceedings of the 2021 ACM SIGMOD International Conference on Management of Data*. 671–684.

[20] Chris Harshaw, Moran Feldman, Justin Ward, and Amin Karbasi. 2019. Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. In *International Conference on Machine Learning*. PMLR, 2634–2643.

[21] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. 2012. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the 2012 International Conference on Data Mining (ICDM)*. 463–474.

[22] Keke Huang, Sibo Wang, Glenn Bevilacqua, Xiaokui Xiao, and Laks VS Lakshmanan. 2017. Revisiting the stop-and-stare algorithms for influence maximization. *Proceedings of the VLDB Endowment* 10, 9 (2017), 913–924.

[23] Huawei. 2022. Retrieved March 27, 2023 from https://consumer.huawei.com/en/phones/mate50-pro/

[24] Tianyuan Jin, Yu Yang, Renchi Yang, Jieming Shi, Keke Huang, and Xiaokui Xiao. 2021. Unconstrained submodular maximization with modular costs: Tight approximation and application to profit maximization. *Proceedings of the VLDB Endowment* 14, 10 (2021), 1756–1768.

[25] Xiangyu Ke, Arijit Khan, and Gao Cong. 2018. Finding seeds and relevant tags jointly: For targeted influence maximization in social networks. In *Proceedings of the 2018 ACM SIGMOD International Conference on Management of Data*. 1097–1111.

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

[27] Arijit Khan, Benjamin Zehnder, and Donald Kossmann. 2016. Revenue maximization by viral marketing: A social network host's perspective. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 37–48.

[28] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 420–429.

[29] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

[30] Hui Li, Sourav S Bhowmick, Jiangtao Cui, Yunjun Gao, and Jianfeng Ma. 2015. Getreal: Towards realistic selection of influence maximization strategies in competitive networks. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 1525–1537.

[31] LinkedIn. 2022. Campaign budgets for Your LinkedIn Ads. Retrieved January 13, 2023 from https://business.linkedin.com/marketing-solutions/success/best-practices/maximize-your-budget

[32] Wei Lu, Francesco Bonchi, Amit Goyal, and Laks VS Lakshmanan. 2013. The bang for the buck: fair competitive viral marketing from the host perspective. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 928–936.

[33] Wei Lu, Wei Chen, and Laks VS Lakshmanan. 2015. From competition to complementarity: comparative influence diffusion and maximization. *Proceedings of the VLDB Endowment* 9, 2 (2015), 60–71.

[34] Wei Lu and Laks VS Lakshmanan. 2012. Profit maximization over social networks. In *2012 IEEE 12th International Conference on Data Mining (ICDM)*. IEEE, 479–488.

[35] Francisco J Martínez-López, Irene Esteban-Millat, Ana Argila, and Francisco Rejón-Guardia. 2015. Consumers' psychological outcomes linked to the use of an online store's recommendation system. *Internet Research* (2015).

[36] Xiaoye Miao, Huanhuan Peng, Kai Chen, Yuchen Peng, Yunjun Gao, and Jianwei Yin. 2022. Maximizing Time-aware Welfare for Mixed Items. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1044–1057.

[37] Michael Mitzenmacher and Eli Upfal. 2017. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press.

[38] Ramasuri Narayanam and Amit A Nanavati. 2012. Viral marketing for product cross-sell through social networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 581–596.

[39] Hung T Nguyen, My T Thai, and Thang N Dinh. 2016. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *Proceedings of the 2016 ACM SIGMOD international conference on management of data*. 695–710.

[40] SamSung. 2022. Retrieved March 27, 2023 from https://www.samsung.com/global/galaxy/galaxy-z-flip4/

[41] SamSung. 2022. Retrieved March 27, 2023 from https://influencermarketinghub.com/influencer-marketing-benchmark-report/

[42] Qihao Shi, Can Wang, Deshi Ye, Jiawei Chen, Sheng Zhou, Yan Feng, Chun Chen, and Yanhao Huang. 2021. Profit maximization for competitive social advertising. *Theoretical Computer Science* 868 (2021), 12–29.

[43] Shopify. 2022. Retrieved March 27, 2023 from https://www.shopify.com/blog/influencer-pricing

[44] Michael Simpson, Farnoosh Hashemi, and Laks VS Lakshmanan. 2022. Misinformation mitigation under differential propagation rates and temporal penalties. *Proceedings of the VLDB Endowment* 15, 10 (2022), 2216–2229.

[45] Vinita Singh, Ranjan Chaudhuri, and Sanjeev Verma. 2018. Psychological antecedents of apparel-buying intention for young Indian online shoppers: Scale development and validation. *Journal of Modelling in Management* (2018).

[46] Snapchat. 2020. Advertising on Snapchat: How pricing works. Retrieved January 13, 2023 from https://forbusiness.snapchat.com/blog/advertising-on-snapchat-how-pricing-works

[47] Statista. 2020. Retrieved March 27, 2023 from https://www.statista.com/topics/5934/2020-presidential-election-and-the-media/#topicOverview

[48] Statista. 2022. Retrieved March 27, 2023 from https://www.statista.com/statistics/666426/goals-influencer-marketing/

[49] Veronika Svatosova. 2020. The importance of online shopping behavior in the strategic management of e-commerce competitiveness. *Journal of Competitiveness* 12, 4 (2020), 143.

[50] Ian P Swift, Sana Ebrahimi, Azade Nova, and Abolfazl Asudeh. 2022. Maximizing Fair Content Spread via Edge Suggestion in Social Networks. *Proceedings of the VLDB Endowment* 15, 11 (2022).

[51] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online processing algorithms for influence maximization. In *Proceedings of the 2018 ACM SIGMOD International Conference on Management of Data*. 991–1005.

[52] Jing Tang, Xueyan Tang, and Junsong Yuan. 2017. Profit maximization for viral marketing in online social networks: Algorithms and analysis. *IEEE Transactions on Knowledge and Data Engineering* 30, 6 (2017), 1095–1108.

[53] Jing Tang, Xueyan Tang, and Junsong Yuan. 2018. Towards profit maximization for online social network providers. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1178–1186.
[54] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 1539–1554.
[55] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. 75–86.
[56] Dimitris Tsaras, George Trimponias, Lefteris Ntaflos, and Dimitris Papadias. 2021. Collective influence maximization for multiple competing products with an awareness-to-influence model. *Proceedings of the VLDB Endowment* 14, 7 (2021), 1124–1136.
[57] Think with Google. 2016. The Rise of Comparison Shopping on Mobile: Which-One's-Best Moments. Retrieved March 27, 2023 from https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/comparison-shopping-mobile/
[58] Wpromote. 2020. The Benefits of Comparison Shopping for Ecommerce Merchants. Retrieved March 27, 2023 from https://www.wpromote.com/blog/amazon-marketing/the-benefits-of-comparison-shopping-for-ecommerce-merchants
[59] Guanhao Wu, Xiaofeng Gao, Ge Yan, and Guihai Chen. 2021. Parallel greedy algorithm to multiple influence maximization in social network. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 3 (2021), 1–21.
[60] Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation: a generative model approach. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 671–680.
[61] Yipeng Zhang, Yuchen Li, Zhifeng Bao, Baihua Zheng, and HV Jagadish. 2021. Minimizing the regret of an influence provider. In *Proceedings of the 2021 ACM SIGMOD International Conference on Management of Data*. 2115–2127.
[62] Yuqing Zhu and Deying Li. 2018. Host profit maximization for competitive viral marketing in billion-scale networks. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1160–1168.
[63] Yuqing Zhu, Zaixin Lu, Yuanjun Bi, Weili Wu, Yiwei Jiang, and Deying Li. 2013. Influence and profit: Two sides of the coin. In *2013 IEEE 13th International Conference on Data Mining (ICDM)*. IEEE, 1301–1306.

APPENDIX

A PROOF OF THEOREM 2

PROOF. Let $\mathbb{S} = \{S_1, ..., S_i, ..., S_{|\mathcal{H}|}\}$ and $\mathbb{S}' = \{S'_1, ..., S'_i, ..., S'_{|\mathcal{H}|}\}$ be two seed sets such that $S_i \subseteq S'_i, \forall 1 \le i \le |\mathcal{H}|$. And we denote the marginal profit gain of adding a user v (i.e., $v \in V - \mathbb{S}'$) to S_i in $\mathbb{S}$ as $P(v|\mathbb{S}) = P(v|S_i) = \frac{B_i \cdot \gamma}{I_i}\sigma_{\mathcal{G}}(v|S_i) - c(v)$, and that of adding v to S'_i in $\mathbb{S}'$ is $P(v|\mathbb{S}') = \frac{B_i \cdot \gamma}{I_i}\sigma_{\mathcal{G}}(v|S'_i) - c(v)$, where $\sigma_{\mathcal{G}}(v|S_i)$ ($\sigma_{\mathcal{G}}(v|S'_i)$) denotes the marginal influence gain of adding v to S_i (S'_i).

For any two seed sets $\mathbb{S}$ and $\mathbb{S}'$ (where $S_i \subseteq S'_i$) and any node $v \in V - \mathbb{S}'$, three cases are as follows when adding v into S_i and S'_i:

(1) Let V_1 and V'_1 be the sets of users that are newly influenced by v on $\mathbb{S}$ (i.e., $|V_1| = \sigma_{\mathcal{G}}(v|S_i)$), and on $\mathbb{S}'$ (i.e., $|V'_1| = \sigma_{\mathcal{G}}(v|S'_i)$). Then the profit marginal gains created by users in V_1 on S_i and V'_1 on S'_i are $P(v|\mathbb{S}) = \frac{B_i \cdot \gamma}{I_i}(|V_1|) - c(v)$ and $P(v|\mathbb{S}') = \frac{B_i \cdot \gamma}{I_i}(|V'_1|) - c(v)$, respectively. It is novel to see that $P(v|\mathbb{S}) \ge P(v|\mathbb{S}')$ since $|V_1| \ge |V'_1|$ due to that Kempe et al. [26] has proved that the influence function $\sigma(\cdot)$ is *submodular* under the LT model.

(2) Let V_2 be the set of users that are newly influenced by v on $\mathbb{S}$ (i.e., $|V_2| = \sigma_{\mathcal{G}}(v|S_i)$), while V'_2 is the set of users that have adopted other product h_j on $\mathbb{S}'$ (i.e., $|V'_2| = \sigma_{\mathcal{G}}(v|S'_i) \cap \sigma_{\mathcal{G}}(S'_j)$). Hence, the profit marginal gain generated by v on $\mathbb{S}$ is $P(v|\mathbb{S}) = \frac{B_i \cdot \gamma}{I_i}(|V_2|) - c(v)$, and that on $\mathbb{S}'$ is $P(v|\mathbb{S}') = (\frac{B_i \cdot \gamma}{I_i} - \frac{B_j \cdot \gamma}{I_j})(|V'_2|) - c(v)$ if the addition of v changes the adoption of users in V'_2 to h_i, otherwise $P(v|\mathbb{S}') = -c(v)$ when nodes in V'_2 stay adopt product h_j. It is clear that $P(v|\mathbb{S}) \ge P(v|\mathbb{S}')$.

Table 5: Frequently used notations

Notation	Description
$G = (V, E)$	A social network with nodes V and edges E
n, m	The numbers of nodes and edges in G, respectively
$\mathcal{H}$	A set of merchants $\{h_1, h_2, ..., h_{\|\mathcal{H}\|}\}$
I_i	The minimum desired influence spread of merchant h_i
B_i	The budget merchant h_i is willing to pay according to I_i
BPI_i	The benefit per influence of merchant h_i, i.e., $BPI_i = \frac{B_i}{I_i}$
S_i	The seed set of h_i
$\sigma(\cdot)$	The influence spread function
$R(\cdot)$	The revenue function, i.e., for merchant h_i, $R(O) = B_i \cdot (1 + \gamma \cdot \frac{\sigma(O) - I_i}{I_i})$, for any $O \subseteq V$
$C(\cdot)$	The cost function, i.e., $C(O) = \sum_{v \in O} c(v)$
$P(\cdot)$	The profit function, i.e., $P(\cdot)$=$R(\cdot)$-$C(\cdot)$
$f(A\|B)$	The marginal gain of A with respect to B for any set function $f(\cdot)$, i.e., $f(A\|B) = f(A \cup B) - f(B)$
$\mathbb{S}$	A collection of sets $\{S_1, S_2, ..., S_{\|\mathcal{H}\|}\}$
$\mathbb{S}^o$	The optimal solution $\mathbb{S}^o = \{S^o_1, S^o_2, ..., S^o_{\|\mathcal{H}\|}\}$
$\mathcal{R}$	A set of RR sets
$C_{\mathcal{R}}(O)$	The number of RR sets covered by O

(3) Let V_3 be the set of v's influenced users that have adopted product h_x (i.e., $|V_3| = \sigma_{\mathcal{G}}(v|S_i) \cap \sigma_{\mathcal{G}}(S_x)$), and V'_3 be the set that have adopted product h_y (i.e., $|V'_3| = \sigma_{\mathcal{G}}(v|S'_i) \cap \sigma_{\mathcal{G}}(S'_y)$) where $h_x \ne h_i, h_y \ne h_i$. It is trivial to prove that $\frac{B_y \cdot \gamma}{I_y} \ge \frac{B_x \cdot \gamma}{I_x}$. Considering the same circumstances as mentioned in (2), We draw the conclusion that $P(v|\mathbb{S}) \ge P(v|\mathbb{S}')$ always holds.

Considering above three cases, it is trivial to prove that the marginal gain of adding a user v to a seed set $S'_i \in \mathbb{S}'$ is no larger than that of adding v into $S_i \in \mathbb{S}$, i.e., $P(S_1) + ... + P(S_i \cup v) + ... + P(S_{|\mathcal{H}|}) - P(\mathbb{S}) \ge P(S'_1) + ... + P(S'_i \cup v) + ... + P(S'_{|\mathcal{H}|}) - P(\mathbb{S}')$. We take the weighted sum over all possible worlds, and conclude that our problem is *submodular* under the DSS model.

□

B PROOF OF THEOREM 3

PROOF. We prove the hardness of our problem using a reduction from the 3-PARTITION problem (*3PM*) [16]. Given a set $X = \{x_1, x_2, ..., x_{3m}\}$ of $3m$ positive integers and the sum of all integers is mT, with $x_i \in (T/4, T/2)$ for $\forall i$. *3PM* requests whether there exists a partition of X into m disjoint 3-element subsets such that the sum of the elements in each partition is equal to T. This problem is known to be strongly **NP**-hard [16], and it implies that the problem remains **NP**-hard even if mT is bounded by a polynomial in m.

Given an instance $\mathcal{P}$ of *3PM*, we reduce it to an instance Q we constructed of our problem with the following steps. We first set the number of companies $|\mathcal{H}| = m$, the budget $B_i = T$, the influence threshold $I_i = T$, for $\forall i$, the cost of each node $c(v) = T/3$, and $\gamma = 0$. And then we construct a directed bipartite graph $G = (U \cup V, E)$: for each integer x_i, G has one node $u_i \in U$ with $x_i - 1$ out-neighbors in V, and all influence probabilities set to 1. Each node $v \in V$ is adjacent to one $u_i \in U$.

Suppose there exists a polynomial time algorithm $\mathcal{A}$ can solve our problem. Run $\mathcal{A}$ on Q to yield an allocation $\mathbb{S} = \{S_1, S_2, ..., S_m\}$.

Then $\mathcal{P}$ is a YES-instance of *3PM* if and only if for all i, $\sigma(S_i) = \sum_{u_j \in S_i} \sigma(u_j) = \sum_{u_j \in S_i} x_j = I_i = T$.

The forward Direction. Suppose the above equation holds for $\forall i$, we show that in this case, each S_i must consist of 3 nodes in U with influence spread value T. From this, the allocation witnesses that the instance $\mathcal{P}$ is a YES-instance. Suppose $|S_i| \neq 3$ for some h_i, $\sigma(S_i) = \sum_{u_j \in S_i} x_j = I_i \neq T$, since for $\forall i$, $x_i \in (T/4, T/2)$, which leads to a paradox.

The reverse Direction. Suppose $S_1, ..., S_m$ are disjoint 3-element subsets with each sum equal to T. We can solve our problem optimally using the allocation $(S_1, ..., S_m)$. It is trivial to see that change any elements in any set will break the satisfactory of I_i.

Approximation hardness. We just proved that our problem is **NP**-hard. To see the hardness of approximation, suppose $\mathcal{B}$ is an algorithm that approximates our problem within a factor of κ. The profit achieved by the algorithm $\mathcal{B}$ on any instance of our problem is $\geq \kappa \cdot \text{OPT}$, where OPT is the optimal (maximum) profit. See the above instance $\mathcal{Q}$ of which the optimal profit is 0. In this instance, the profit achieved by algorithm $\mathcal{B}$ is $\geq \kappa \cdot 0 = 0$, i.e., algorithm $\mathcal{B}$ can solve the our problem optimally in polynomial time, which is impossible unless **P=NP**. Hence, our problem is **NP**-hard to approximate within any factor.

□

C PROOF OF THEOREM 4

Proof. Let $\mathbb{S}_0 = \emptyset$, and $\mathbb{S}_{t-1}$ $(t > 1)$ be the partial solution set constructed by the first $t-1$ iterations of Fill-Greedy. Let $\mathbb{S}'$ be the subset of V that maximizes $R(\mathbb{S}) - C(\mathbb{S}) - |\mathcal{H}| \cdot \ln \frac{R(\mathbb{S})}{C(\mathbb{S})} \cdot C(\mathbb{S})$ (Note that, $P(\mathbb{S}) = R(\mathbb{S}) - C(\mathbb{S})$). To simplify, we use h in the appendix to denote $|\mathcal{H}|$. For $\forall i \in [h]$, We use the following two lemmas to prove Theorem 4.

Lemma 2.

$$\frac{R_i(v_t \mid \mathbb{S}_{t-1})}{c(v_t)} \geq \frac{R(\mathbb{S}') - R(\mathbb{S}_{t-1})}{h \cdot C(\mathbb{S}')} \tag{8}$$

we give the proof of Lemma 2 as follows, where R_i denotes the revenue of merchant h_i, and R_m is the revenue of merchant with maximum *benefit per influence*, i.e., $\frac{B_m}{I_m} = \max_{i \in h}\{\frac{B_i}{I_i}\}$.

$$\begin{aligned}
\frac{R_i(v_t \mid \mathbb{S}_{t-1})}{c(v)} &= \max_{u \in V \setminus \mathbb{S}_{t-1}} \frac{R_i(u \mid \mathbb{S}_{t-1})}{c(u)} \geq \max_{u \in \mathbb{S}' \setminus \mathbb{S}_{t-1}} \frac{R_i(u \mid \mathbb{S}_{t-1})}{c(u)} \\
&= \max_{u \in S'_m \setminus \mathbb{S}_{t-1}} \frac{R_m(u \mid \mathbb{S}_{t-1})}{c(u)} \geq \frac{1}{h} \sum_{i=1}^{h} \max_{u \in S'_i \setminus \mathbb{S}_{t-1}} \frac{R_i(u \mid \mathbb{S}_{t-1})}{c(u)} \\
&\geq \frac{1}{h} \sum_{i=1}^{h} \frac{R_i(S'_i \mid \mathbb{S}_{t-1})}{C(S'_i)} \geq \frac{1}{h} \sum_{i=1}^{h} \frac{R_i(S'_i \mid \mathbb{S}_{t-1})}{C(\mathbb{S}')} \\
&\geq \frac{R(\mathbb{S}' \mid \mathbb{S}_{t-1})}{h \cdot C(\mathbb{S}')} \geq \frac{R(\mathbb{S}') - R(\mathbb{S}_{t-1})}{h \cdot C(\mathbb{S}')}
\end{aligned}$$

Lemma 3.

$$R_i(\mathbb{S}_t) \geq \left(1 - \prod_{k=1}^{t}\left(1 - \frac{c(v_k)}{h \cdot C(\mathbb{S}')}\right)\right) \cdot R_i(\mathbb{S}'). \tag{9}$$

From the Lemma 2, we have $R_i(v_1) \geq \frac{c(v_1) R(\mathbb{S}')}{h \cdot C(\mathbb{S}')}$, which means that Eq.(8) holds for $t = 1$. And then we prove that $R_i(\mathbb{S}_t) \geq \left(1 - \prod_{k=1}^{t}\left(1 - \frac{c(v_k)}{h \cdot C(\mathbb{S}')}\right)\right) \cdot R_i(\mathbb{S}')$ holds by induction.

Based on Lemma 2 and Lemma 3, we prove Theorem 4 by showing that

$$R(\mathbb{S}) - C(\mathbb{S}) \geq R(\mathbb{S}') - C(\mathbb{S}') - h \cdot \ln \frac{R(\mathbb{S}')}{C(\mathbb{S}')} \cdot C(\mathbb{S}') \tag{10}$$

where $P(\mathbb{S}) = R(\mathbb{S}) - C(\mathbb{S})$ $(P(\mathbb{S}') = R(\mathbb{S}') - C(\mathbb{S}'))$. We also consider two cases under multiple merchants based on whether $C(\mathbb{S}) < h \ln \frac{R(\mathbb{S}')}{C(\mathbb{S}')} \cdot C(\mathbb{S}')$.

Case 1: $C(\mathbb{S}) < h \cdot \ln \frac{R(\mathbb{S}')}{C(\mathbb{S}')} \cdot C(\mathbb{S}')$, under Algorithm 1, for $\forall i \in h$, we have

$$\begin{aligned}
0 &> \max_{u \in V \setminus \mathbb{S}} (R_i(u \mid \mathbb{S}) - c(u)) \geq \max_{u \in \mathbb{S}' \setminus \mathbb{S}} (R_i(u \mid \mathbb{S}) - c(u)) \\
&= \max_{u \in S'_m \setminus \mathbb{S}} (R_m(u \mid \mathbb{S}) - c(u)) \geq \frac{1}{h} \sum_{i=1}^{h} \max_{u \in S'_i \setminus \mathbb{S}} (R_i(u \mid \mathbb{S}) - c(u)) \\
&\geq \frac{1}{h}\left(\sum_{i=1}^{h} R_i(S'_i \mid \mathbb{S}) - \sum_{i=1}^{h} C(S'_i)\right) = \frac{1}{h}(R(\mathbb{S}' \mid \mathbb{S}) - C(\mathbb{S}')) \\
&\geq \frac{1}{h}(R(\mathbb{S}') - R(\mathbb{S}) - C(\mathbb{S}'))
\end{aligned}$$

Therefore, $R(\mathbb{S}) \geq R(\mathbb{S}') - C(\mathbb{S}')$. This leads to

$$\begin{aligned}
R(\mathbb{S}) - C(\mathbb{S}) &\geq R(\mathbb{S}') - C(\mathbb{S}') - C(\mathbb{S}) \\
&\geq R(\mathbb{S}') - C(\mathbb{S}') - h \cdot \ln \frac{R(\mathbb{S}')}{C(\mathbb{S}')} \cdot C(\mathbb{S}')
\end{aligned}$$

Case 2: $C(\mathbb{S}) \geq h \cdot \ln \frac{R(\mathbb{S}')}{C(\mathbb{S}')} \cdot C(\mathbb{S}')$. In this case, we demonstrate that $R(\mathbb{S}) - C(\mathbb{S}) \geq R(\mathbb{S}') - C(\mathbb{S}') - h \cdot \ln \frac{R(\mathbb{S}')}{C(\mathbb{S}')} \cdot C(\mathbb{S}')$ trivially holds via Lemma 2 and Lemma 3 stated above, the specific proof is an extension of Case 2 in [24], thus we omit the proof.

□

D PROOF OF LEMMA 1

Proof. Before prove Lemma 1, we first introduce *Chernoff Inequalities* [37] in Lemma 4 as follows.

Lemma 4. **(Chernoff Inequalities [37]).** *Let X be the sum of k i.i.d. random variables sampled from a distribution on $[0, 1]$ and ρ is a mean. Then, for any $\lambda > 0$,*

$$\begin{aligned}
\Pr[X - k\rho \geq \lambda \cdot k\rho] &\leq \exp\left(-\frac{\lambda^2}{2+\lambda} k\rho\right) \\
\Pr[X - k\rho \leq -\lambda \cdot k\rho] &\leq \exp\left(-\frac{\lambda^2}{2} k\rho\right)
\end{aligned} \tag{11}$$

Given any solution $\mathbb{S}$ to the profit maximization problem and any set $\mathcal{R}$ of RR sets, we extend Lemma 4 and introduce the following concentration bounds:

$$\Pr[R^{\mathcal{R}_2}(\mathbb{S}) - R(\mathbb{S}) \geq \epsilon_1 \cdot R(\mathbb{S})] \leq \exp\left(-\frac{\epsilon_1^2}{2+\epsilon_1} \frac{|\mathcal{R}|}{n \cdot \Gamma_1} R(\mathbb{S})\right) \tag{12}$$

$$\Pr[R^{\mathcal{R}_1}(\mathbb{S}^o) - R(\mathbb{S}^o) \leq -\epsilon_2 \cdot R(\mathbb{S}^o)] \geq \exp\left(-\frac{\epsilon_2^2}{2} \frac{|\mathcal{R}|}{n \cdot \Gamma_2} R(\mathbb{S}^o)\right) \tag{13}$$

where $|\mathcal{R}|$ is the number of RR sets, $\Gamma_1 = \sum_{i=1}^{h}(\frac{B_i}{I_i} \cdot \max\{\gamma_r, \gamma_p\})$, and $\Gamma_2 = \sum_{i=1}^{h}(\frac{B_i}{I_i} \cdot \min\{\gamma_r, \gamma_p\})$. Based on this, in the i-th iteration, let Θ_{1i} denote the event that Eq. (6) holds, and Θ_{2i} denote the event that Eq. (7) holds. We set (ϵ^{+}) and (ϵ^{-}) as the solutions to Eq. (12) and Eq. (13), thus we have following equation

$$\exp\left(-\frac{(\epsilon^{+})^2}{2+(\epsilon^{+})}\frac{|\mathcal{R}|}{n\cdot\Gamma_1}R(\mathbb{S})\right)=\frac{\delta}{5i^2}. \tag{14}$$

$$\exp\left(-\frac{(\epsilon^{-})^2}{2}\frac{|\mathcal{R}|}{n\cdot\Gamma_2}R(\mathbb{S}^o)\right)=\frac{\delta}{5i^2}. \tag{15}$$

Then we have $\Pr[\Theta_{i1}]\geq 1-\delta/(5i_2)$, $\Pr[\Theta_{i2}|\Theta_{i1}]\geq 1-\delta/(5i^2)$. Thus, $\Pr[\Theta_{i2}\cap\Theta_{i1}]=\Pr[\Theta_{i2}|\Theta_{i1}]\cdot\Pr[\Theta_{i1}]=1-2\delta/(5i^2)$. For all iterations, we have

$$\begin{aligned}\Pr\left[\bigcap_{i=1}^{\infty}\Theta_{1i}\bigcap_{i=1}^{\infty}\Theta_{2i}\right]&\geq\prod_{i=1}^{\infty}\Pr\left[\Theta_{1i}\cap\Theta_{2i}\right]\geq\prod_{i=1}^{\infty}\left(1-\frac{2\delta}{5i^2}\right)\\&\geq 1-\sum_{i=1}^{\infty}\frac{2\delta}{5i^2}\geq 1-\frac{\pi^2\delta}{15}\geq 1-\frac{2\delta}{3}.\end{aligned} \tag{16}$$

The details of proof are similar in spirit to those in [24]. □

With the above conclusions we further prove Theorem 5.

E PROOF OF THEOREM 5

Proof. We consider two cases that depend on whether Line 8 in Algorithm 3 is satisfied.

Case 1: Line 8 is satisfied. Then in the last iteration, we have

$$(\beta-1)/\beta+\epsilon_1+\epsilon_2\leq\epsilon,\ \epsilon_1+\epsilon_2\leq\epsilon, \tag{17}$$

where $\epsilon_1,\epsilon_2\in(0,1)$ and $\beta>0$. Suppose that both Eq. (6) and Eq. (7) hold. Then,

$$\begin{aligned}R^{\mathcal{R}_1}(\mathbb{S})-C(\mathbb{S})&\geq R^{\mathcal{R}_1}(\mathbb{S}^o)-C(\mathbb{S}^o)-h\cdot\ln\frac{R^{\mathcal{R}_1}(\mathbb{S}^o)}{C(\mathbb{S}^o)}\cdot C(\mathbb{S}^o)\\&\geq(1-\epsilon_2)R(\mathbb{S}^o)-C(\mathbb{S}^o)-h\cdot\ln\frac{R(\mathbb{S}^o)}{C(\mathbb{S}^o)}\cdot C(\mathbb{S}^o),\end{aligned} \tag{18}$$

where the first inequality is due to Theorem 4, and the second inequality is due to Eq. (7). Afterwords, via Eq. (6),

$$\begin{aligned}R^{\mathcal{R}_1}(\mathbb{S})-C(\mathbb{S})&=\beta\left(R^{\mathcal{R}_2}(\mathbb{S})-C(\mathbb{S})\right)\\&\leq\beta(1+\epsilon_1)R(\mathbb{S})-\beta\cdot C(\mathbb{S}).\end{aligned} \tag{19}$$

Finally, we have

$$\begin{aligned}R(\mathbb{S})-C(\mathbb{S})&\geq 1/\beta\left(R^{\mathcal{R}_1}(\mathbb{S})-C(\mathbb{S})\right)-\epsilon_1R(\mathbb{S}^o)\\&\geq 1/\beta\left((1-\epsilon_2)R(\mathbb{S}^o)-C(\mathbb{S}^o)-h\cdot\ln\frac{R(\mathbb{S}^o)}{C(\mathbb{S}^o)}\cdot C(\mathbb{S}^o)\right)-\epsilon_1R(\mathbb{S}^o)\end{aligned} \tag{20}$$

where the first inequality is from Eq. (19) and second inequality is from Eq. (18). And then based on that [24] has proved whether $\beta\leq 1$ there existed

$$P(\mathbb{S})=R(\mathbb{S})-C(\mathbb{S})\geq(1-\epsilon)R(\mathbb{S}^o)-C(\mathbb{S}^o)-h\cdot\ln\frac{R(\mathbb{S}^o)}{C(\mathbb{S}^o)}\cdot C(\mathbb{S}^o).$$

By Lemma 1, when Line 8 in Algorithm 3 is satisfied, with probability at least $1-\frac{2\delta}{3}$, Eq. (5) holds.

Case 2: Line 8 is not satisfied. Then we have

$$\theta_i=(8+2\epsilon)(1+\epsilon_1)n\frac{\ln\frac{6}{\delta}+\sum_{i\in h}\tau_i\ln\frac{2n}{\tau_i}}{\epsilon^2\max\left\{1,R^{\mathcal{R}_2}(\mathbb{S})-(1+\epsilon_1)C(\mathbb{S})\right\}}$$

when Algorithm 3 terminates, and τ_i is the maximum number of users that can be selected by merchant h_i. Note that when $\bigcap_i\Theta_{1i}$ occurs, it implies that $\max\left\{1,R^{\mathcal{R}_2}(\mathbb{S})-(1+\epsilon_1)C(\mathbb{S})\right\}\leq\max\left\{1,(1+\epsilon_1)(R(\mathbb{S})-C(\mathbb{S}))\right\}\leq(1+\epsilon_1)R(\mathbb{S}^o)$. Then we have

$$\theta_i=(8+2\epsilon)n\frac{\ln\frac{6}{\delta}+\sum_{i\in h}\tau_i\ln\frac{2n}{\tau_i}}{\epsilon^2R(\mathbb{S}^o)}$$

When Algorithm 3 terminates. Then by Lemma 4, let $\varrho=\epsilon R(\mathbb{S}^o)/2R(\mathbb{O})$ for any $\mathbb{O}\subseteq V$,

$$\begin{aligned}&\Pr[R^{\mathcal{R}_1}(\mathbb{O})-R(\mathbb{O})\geq\frac{\epsilon}{2}\cdot R(\mathbb{S}^o)]\leq\exp\left(-\frac{\varrho^2}{2+\varrho}\frac{|\mathcal{R}|}{n\cdot\Gamma_1}R(\mathbb{O})\right)\\&\leq\exp\left(-\frac{\epsilon^2}{8+2\epsilon}\frac{|\mathcal{R}|}{n\cdot\Gamma_1}R(\mathbb{S}^o)\right)\leq\exp\left(-\frac{\epsilon^2}{8+2\epsilon}\frac{|\mathcal{R}|}{n}R(\mathbb{S}^o)\right)\leq\frac{\delta}{6\cdot 2^n},\end{aligned}$$

where the second inequality is due to the fact that if $R(\mathbb{O})=R(\mathbb{S}^o)$ the right side of the first inequality achieves its maximum. Similarly, we also have $\Pr[R^{\mathcal{R}_1}(\mathbb{O})-R(\mathbb{O})\leq-\frac{\epsilon}{2}\cdot R(\mathbb{S}^o)]\leq\frac{\delta}{6\cdot 2^n}$. Thus, we have $\Pr[|R^{\mathcal{R}_1}(\mathbb{O})-R(\mathbb{O})|\leq\frac{\epsilon}{2}\cdot R(\mathbb{S}^o),\forall\mathbb{O}\subseteq V]\geq 1-\frac{\delta}{3}$. And then following [24], when $|R^{\mathcal{R}_1}(\mathbb{O})-R(\mathbb{O})|\leq\frac{\epsilon}{2}R(\mathbb{S}^o)$ for all $\mathbb{O}\subset V$, we have

$$R^{\mathcal{R}_1}(\mathbb{S}^o)\geq(1-\frac{\epsilon}{2})R(\mathbb{S}^o), \tag{21}$$

$$R^{\mathcal{R}_1}(\mathbb{S})\leq R(\mathbb{S})+\frac{\epsilon}{2}R(\mathbb{S}^o). \tag{22}$$

Based on the above results, when the event $\bigcap_i\Theta_{1i}$ occurs, we have

$$\begin{aligned}R(\mathbb{S})-C(\mathbb{S})&\geq R^{\mathcal{R}_1}(\mathbb{S})-C(\mathbb{S})-\frac{\epsilon}{2}R(\mathbb{S}^o)\\&\geq R^{\mathcal{R}_1}(\mathbb{S}^o)-C(\mathbb{S}^o)-h\cdot\ln\frac{R^{\mathcal{R}_1}(\mathbb{S}^o)}{C(\mathbb{S}^o)}\cdot C(\mathbb{S}^o)-\frac{\epsilon}{2}R(\mathbb{S}^o)\\&\geq(1-\epsilon)R(\mathbb{S}^o)-C(\mathbb{S}^o)-h\cdot\ln\frac{R(\mathbb{S}^o)}{C(\mathbb{S}^o)}\cdot C(\mathbb{S}^o)\end{aligned}$$

According to Eq. (16), the event $\bigcap_i\Theta_{1i}$ happens with probability at least $1-\frac{2\delta}{3}$. Hence, when Line 8 is not satisfied, with probability at least $1-\frac{2\delta}{3}-\frac{\delta}{3}\geq 1-\delta$, we have

$$P(\mathbb{S})=R(\mathbb{S})-C(\mathbb{S})\geq(1-\epsilon)R(\mathbb{S}^o)-C(\mathbb{S}^o)-h\cdot\ln\frac{R(\mathbb{S}^o)}{C(\mathbb{S}^o)}\cdot C(\mathbb{S}^o).$$

Finally, we combine **Case 1** and **Case 2**, the Theorem 5 is demonstrated.

□

F PROOF OF THEOREM 6

Proof. The time complexity of Algorithm 3 is dominated by the cost of RR set generation, i.e., **(1)** the expected time for generating a random RR set is bounded by $\frac{m\sum_{i\in[|\mathcal{H}|]}\mathbb{E}[P_i(\{v^*\})]}{n}$ [26, 55], and **(2)** the total number of RR sets generated is at most $O(\frac{n\ln\frac{1}{\delta}+n\ln|\mathcal{H}|}{\epsilon^2})$, where $\sum_{i\in|\mathcal{H}|}\tau_i\ln\frac{2n}{\tau_i}$ can be replaced by $n\ln|\mathcal{H}|$, since each user can be picked by at most one merchant [19]. Hence, the expected time complexity of Algorithm 3 is $O(\frac{m\sum_{i\in[|\mathcal{H}|]}\mathbb{E}[P_i(\{v^*\})](\ln\frac{1}{\delta}+n\ln|\mathcal{H}|)}{\epsilon^2})$. □