Maximizing the Spread of Influence through a Social Network Using Partial Incentives

ABHISHEK K. UMRAWAL, University of Illinois Urbana-Champaign, USA
ELIOT W. ROBSON, Narmi Inc., USA
VANEET AGGARWAL*, Purdue University, USA
CHRISTOPHER J. QUINN, Iowa State University, USA

We study a generalization of the widely studied discrete influence maximization problem. Instead of marketers using a budget to send free products to a few influencers, we consider allowing discounts to partly incentivize a larger set of influencers under the same budget. We show that this problem is an instance of maximizing the multilinear extension of a monotone submodular set function subject to an L_1 budget constraint and characterize its optimal solution in terms of solutions to the discrete influence maximization problem. Using this characterization, we propose and analyze an efficient $(1 - 1/e)$-approximation algorithm. We further show that, with negligible additional computation, this algorithm enables marketers to evaluate cost–benefit trade-offs over a range of budgets. We conduct small-scale experiments on synthetic and real-world social networks to demonstrate the structure of the optimal solution and the behavior of the greedy approximation. We also perform large-scale experiments on real-world networks to evaluate the performance and scalability of our method in contrast to existing approaches. Finally, we illustrate the practicality of our method for budget selection targeting desired influence and for profit maximization.

JAIR Associate Editor: Gilles Pesant

JAIR Reference Format:
Abhishek K. Umrawal, Eliot W. Robson, Vaneet Aggarwal, and Christopher J. Quinn. 2026. Maximizing the Spread of Influence through a Social Network Using Partial Incentives. *Journal of Artificial Intelligence Research* 86, Article 15 (June 2026), 25 pages.
DOI: 10.1613/jair.1.20410

1 Introduction

With an increasing number of users spending time on social media platforms and engaging with family, friends, and influencers within communities of interest (such as in fashion, cooking, gaming, etc.), there are significant opportunities for marketing firms to leverage word-of-mouth advertising on these platforms. In particular, marketing firms can select sets of influencers within a relevant community to sponsor by providing free samples, so they will discuss and promote the product on their social media accounts.

The question of which set of influencers to sponsor in order to maximize the expected cascade size is known as *influence maximization* (IM) [Domingos and Richardson 2001]. Under standard diffusion models, Kempe et al. [Kempe et al. 2003] showed that this discrete optimization problem involves maximizing a submodular set function under a cardinality constraint, which is NP-hard. We refer to this combinatorial optimization problem

---

*Corresponding Author.

Authors' Contact Information: Abhishek K. Umrawal, ORCID: 0000-0002-2037-3694, aumrawal@illinois.edu, University of Illinois Urbana-Champaign, Urbana, Illinois, USA; Eliot W. Robson, ORCID: 0000-0002-1476-6715, eliot.robson24@gmail.com, Narmi Inc., New York, New York, USA; Vaneet Aggarwal, ORCID: 0000-0001-9131-4723, vaneet@purdue.edu, Purdue University, West Lafayette, Indiana, USA; Christopher J. Quinn, ORCID: 0000-0002-9053-1504, cjquinn@iastate.edu, Iowa State University, Ames, Iowa, USA.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).
DOI: 10.1613/jair.1.20410

as *discrete influence maximization* (DIM). Those authors proposed using a greedy algorithm from [Nemhauser et al. 1978] which achieves a $(1 - 1/e)$-approximation ratio.

An important limitation of DIM as a model for viral marketing is the binary nature of influencer sponsorships, i.e., an influencer is either provided a free product or not. Especially for expensive products, this can be restrictive, as the marketer may only be able to provide a few influencers with the product. Influencers have an incentive to try products in order to post regularly and engage their audiences. Thus, a marketer with a limited budget could provide discounts to a much larger set of influencers than they could send free products to. Influencers who receive a small discount may be less likely to try and promote the product than influencers who receive a large discount. Nonetheless, this gives the marketer more options and, for the same budget, could potentially improve sales beyond what would have been possible with free products alone.

We consider a continuous optimization problem for influence maximization, where the marketer can provide partial incentives. We refer to this problem as *fractional influence maximization* (FIM). We show that the objective of FIM is a continuous analog of the submodular objective of DIM. We characterize this problem and show that it is no more challenging to approximate than the widely studied discrete problem. In addition, we show that near-optimal approximations to the continuous problem can be constructed from near-optimal approximations for the discrete problem, for which there are many existing methods available. Furthermore, we identify a continuous path of near-optimal solutions (over an interval of budgets), allowing the marketer to evaluate cost–benefit trade-offs over a range of budgets. In particular, we show that this enables budget selection for a desired level of influence and the maximization of profit, instead of only maximizing expected spread fully using the fixed budget.

The method we propose uses an efficient, dedicated subroutine to solve the discrete problem, using only minimal additional overhead, and thus can be substantially more efficient than methods for general classes of continuous submodular optimization problems.

1.1 Our Contributions

The main contributions of the paper are as follows.

1. We first characterize the optimal solution for the FIM problem with a budget t by showing that it is a convex combination of two (not necessarily optimal) solutions to the FIM problem for integer budgets, one for budget $\lfloor t \rfloor$ and one for budget $\lfloor t \rfloor + 1$ (Section 4.1).
2. Using the above characterization, we propose and analyze a procedurally simple and computationally efficient $(1 - 1/e)$ greedy approximation algorithm for maximizing the multilinear extension of a monotone submodular set function subject to an L_1 constraint. That class of problems includes an important generalization of discrete influence maximization. Our algorithm only requires access to a value oracle for the discrete problem and achieves similar influence as standard methods for more general classes of problems, while using orders of magnitude less computation (Section 4.2).
3. With negligible additional work, our method can identify a continuous path of near-optimal solutions up to the specified budget, allowing the marketer to evaluate cost–benefit trade-offs over a range of budgets (Section 4.3).
4. We perform both small- and large-scale experiments to support our claims. Using a small synthetic network and a small real-world social network, we empirically demonstrate the characterization of the optimal solution for the FIM problem. Using large real-world social networks, we demonstrate the performance and computational efficiency of our approximation algorithm, and we construct the continuous path of near-optimal solutions to illustrate budget selection for desired influence and cost–benefit trade-offs across budgets (Section 5).

1.2 Literature Review

There is a large literature on maximizing submodular set functions and continuous analogs such as DR-submodular functions. We discuss the most relevant works.

1.2.1 Optimizing a Submodular Set Function using its Multilinear Extension. Our algorithm is designed to (approximately) optimize a continuous function using an oracle for the corresponding set function. There are several works with the roles reversed—the authors proposed algorithms to optimize a set function using its multilinear extension as a surrogate [Calinescu, Chekuri, Pal, et al. 2011; Calinescu, Chekuri, Pál, et al. 2007; Chekuri and Quanrud 2019; Chekuri, Vondrák, et al. 2014; Vondrák 2008]. [Calinescu, Chekuri, Pal, et al. 2011] proposed a continuous greedy algorithm to maximize the multilinear extension and then used the pipage rounding framework [Ageev and Sviridenko 2004] to provide a near-optimal solution maximizing a submodular set function. Chekuri et al. [Chekuri, Vondrák, et al. 2014] use a similar approach using contention resolution schemes. Chekuri et al. [Chekuri and Quanrud 2019] achieve low adaptivity complexity by using parallelization.

1.2.2 Optimizing DR-Submodular Functions. There have been a number of works proposing methods to maximize continuous functions with properties analogous to diminishing returns of submodular set functions, such as DR-submodular functions [Bian et al. 2017; L. Chen et al. 2020; Hassani et al. 2017; Soma and Yoshida 2015]. Multilinear extensions are special cases of DR-submodular functions [Calinescu, Chekuri, Pal, et al. 2011]. Bian et al. [Bian et al. 2017] proposed a variant of the Frank–Wolfe algorithm [Frank, Wolfe, et al. 1956] that provides a $(1 - 1/e)$-approximation of the optimal solution. Later, Hassani et al. [Hassani et al. 2017] proposed projected gradient ascent for (stochastic) continuous submodular maximization. Recently, Chen et al. [L. Chen et al. 2020] studied the same problem in an online setting with similar approximation guarantees using Frank–Wolfe and projected gradient descent variants.

In contrast to existing methods, our greedy algorithm does not involve a variant of the continuous greedy process proposed in [Calinescu, Chekuri, Pál, et al. 2007] or a variant of the Frank–Wolfe algorithm [Frank, Wolfe, et al. 1956]. Instead, our algorithm queries the underlying set function, rather than the multilinear extension (as in existing methods). Evaluating the multilinear extension requires an exponential number of queries to the underlying set function by definition; while exact evaluations of a multilinear extension are prohibitive, approximate evaluations are possible through (polynomial time) random sampling [Calinescu, Chekuri, Pál, et al. 2007].

The runtime of the existing algorithms is a function of the step size used in the continuous greedy process. The step size also affects the approximation guarantees. A high value of the step size makes the algorithm faster but weakens the approximation; similarly, a low value of the step size strengthens the approximation but makes the algorithm slower. In contrast, our proposed greedy algorithm does not involve continuous greedy hill climbing and consequently does not require a step size parameter. For details on the computational complexity of our algorithm, refer to Section 4.2.1.

1.2.3 Influence Maximization. We next discuss works in the influence maximization literature that consider continuous optimization problems. Kempe et al. [Kempe et al. 2003] famously proposed using a greedy algorithm from [Nemhauser et al. 1978] for DIM, where the decision maker chooses a subset of users to seed the diffusion. Kempe et al. [Kempe et al. 2003] also considered a more general setting we refer to as general influence maximization (GIM) where, instead of selecting users to seed the diffusion, the decision maker selects how much to invest in different *marketing strategies* that in turn influence users to different extents.

Kempe et al. [Kempe et al. 2003] proposed a continuous greedy algorithm for solving GIM. Yang et al. [Y. Yang et al. 2016] considered a special case of GIM where each marketing strategy only directly affects a single user and the relationship between the expenditures (from the budget) and the effects on the users could vary. Yang et al. called this problem continuous influence maximization (CIM) and proposed a heuristic procedure. Chen et al.

[W. Chen et al. 2020] considered another special case of Kempe et al.'s GIM problem, known as lattice influence maximization (LIM), where they focused on discretized marketing strategies with a granularity parameter. They proposed an efficient continuous greedy algorithm using reverse influence sampling [Borgs et al. 2014; Tang, Shi, et al. 2015; Tang, Xiao, et al. 2014]. It is an adaptation of gradient methods for DR-submodular maximization for influence maximization.

Moreover, in general, the LIM objective function is not a multilinear function. Demaine et al. [Demaine et al. 2014] also proposed an influence model with partial incentives. Unlike the aforementioned marketing strategies in [W. Chen et al. 2020; Kempe et al. 2003; Y. Yang et al. 2016] and in the present work, Demaine et al. [Demaine et al. 2014] considered strategies that affect the peer-to-peer influences (i.e., edge weights in the independent cascade model). They proposed efficient greedy approximations under such strategies. Kempe et al. [Kempe et al. 2003], Chen et al. [W. Chen et al. 2020], and Demaine et al. [Demaine et al. 2014] provide approximation guarantees of $(1 - 1/e - \epsilon)$ for some $\epsilon > 0$. Relatedly, Bhimaraju et al. [Bhimaraju et al. 2024] consider a fractional-budget model with an affine discount–adoption mapping and obtain a $(1 - 1/e)$ approximation.

Many additional aspects of the influence maximization problem have also been studied, including fairness in influence maximization [Becker et al. 2022; Lin et al. 2023; Nguyen et al. 2022], and the use of deep learning [Kamarthi et al. 2020; Kumar et al. 2022; Ling et al. 2023]. See the survey [Li et al. 2023] for more details. In this paper, we focus on another special case of Kempe et al.'s generalization, where each marketing strategy affects a single user, and the budget for marketing directly corresponds to adoption probability, a natural assumption in marketing situations that has been incorporated by models introduced by prior work [Domingos and Richardson 2001]. Specifically, we consider the most natural extension of the widely studied discrete problem (DIM), for which there is also a uniform effect across users per unit budget expended.

1.3 Organization

The rest of the paper is organized as follows. Section 2 reviews basic definitions of submodular optimization and influence maximization. Section 3 formalizes the problem we consider. Section 4 provides an efficient greedy algorithm for FIM with approximation guarantees and its applications to budget selection. Section 5 provides experimental evaluations using synthetic and real-world social networks. Section 6 concludes the paper and discusses future directions. Furthermore, a table of notations is provided in Section A.

2 Background

In this section, we provide definitions and formulate the problem of interest. In Section 2.1, we review basic definitions and properties of submodular functions. In Section 2.2, we review influence maximization.

2.1 Submodularity

We first review key terms and properties in submodular optimization; see [Kempe et al. 2003; Vondrák 2008] for more details. Let Ω denote the ground set of n elements and let $\mathcal{S} = 2^{\Omega}$ be the set of all subsets of Ω.

A set function $f : \mathcal{S} \to \mathbb{R}$ is *submodular* if it satisfies the natural "diminishing returns" property: the marginal gain from adding an element v to a set $S \in \mathcal{S}$ is at least as high as the marginal gain from adding the same element v to a superset $T \in \mathcal{S}$ of S. Formally, for any sets $S, T \in \mathcal{S}$ with $S \subseteq T$, and any $v \notin T$,

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T).$$

A set function f is *monotone* (non-decreasing) if for any $S \subseteq T$ we have $f(S) \leq f(T)$.

For a vector $\mathbf{x} = (x_1, \dots, x_n)$ with $x_i \in [0,1]$, the *multilinear extension* $F : [0,1]^n \to \mathbb{R}$ of the set function f is

$$F(\mathbf{x}) = \sum_{S \subseteq \Omega} f(S) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i).$$

At the vertices $\mathbf{x} \in \{0,1\}^n$ of the hypercube $[0,1]^n$, the multilinear extension satisfies $F(\mathbf{x}) = f(S)$ for $S = \{i \in \{1,\dots,n\} : x_i = 1\}$. An equivalent definition [Vondrák 2008] is $F(\mathbf{x}) = \mathbb{E}[f(\hat{\mathbf{x}})]$, where $\hat{\mathbf{x}}$ is a random set obtained by including each i independently with probability x_i. This relationship underlies strategies that estimate a multilinear extension via Monte Carlo sampling (querying a value oracle for the set function f); our procedure does not rely on such estimates.

2.2 Influence Maximization Review

Diffusion models describe how information and content spread over a social network. We focus on the *independent cascade* (IC) model [Goldenberg et al. 2001a,b], which is widely used in the influence maximization literature [W. Chen et al. 2020; Demaine et al. 2014; Goyal, Bonchi, and Lakshmanan 2011; Goyal, Lu, et al. 2011; Kempe et al. 2003; Tang, Shi, et al. 2015; Y. Yang et al. 2016]. Other models include linear threshold [Granovetter 1978; Schelling 2006] and pressure threshold [Stutsman et al. 2026].

In the IC model, given a graph $G = (V, E)$, the process starts at time 0 with an initial set of active nodes S (the *seed set*). When a node $v \in S$ first becomes active at time u, it is given a single chance to activate each currently inactive neighbor w, succeeding with probability $p_{v,w}$, independently of the history thus far. If w has multiple newly active neighbors, their attempts are sequenced in an arbitrary order. If v succeeds, then w becomes active at time $u + 1$; regardless of success or failure, v cannot make further activation attempts to w in subsequent rounds.

Define indicators

$$Y_u^{(v)} = \begin{cases} 1, & \text{if node } v \text{ is active at time } u, \\ 0, & \text{otherwise.} \end{cases}$$

The diffusion is *progressive*: nodes can switch from inactive to active, but not back. Thus, $Y_u^{(v)} \le Y_{u+1}^{(v)}$.

The *influence* $\sigma(S)$ of S is the expected number of active nodes at the end of the cascade (denoted by U), given that the users in S become active at time 0:

$$\sigma(S) = \mathbb{E}\left[\sum_{v \in V} Y_U^{(v)} \,\middle|\, Y_0^{(v)} = 1 \ \forall v \in S,\ Y_0^{(v)} = 0 \ \forall v \notin S \right]. \tag{1}$$

Kempe et al. [Kempe et al. 2003] showed that under IC, $\sigma(S)$ is a monotone submodular set function. They proposed the *discrete influence maximization* (DIM) problem: for an integer budget k, find a subset S with $|S| \le k$ maximizing $\sigma(S)$.

Problem 1 (Discrete Influence Maximization (DIM)). *Given a set V, a budget k, and an expected influence function σ, solve*

$$\begin{aligned} &\arg\max_{S \subseteq V} \ \sigma(S), \\ &\quad s.t. \ \ |S| \le k. \end{aligned}$$

DIM is NP-hard [Kempe et al. 2003]. A classical greedy algorithm [Nemhauser et al. 1978] yields a $(1 - 1/e)$-approximation for monotone, submodular σ (such as under the IC model).

3 Problem Formulation

In this section, we define the problems we study in this work.

3.1 Social Influence Maximization under Partial Incentives

We study a generalization of Problem 1 (DIM) that we refer to as *fractional influence maximization* (FIM), where the decision maker can *partially incentivize* users to seed the diffusion.

This is in contrast to DIM, where influencers selected as seeds are either fully incentivized or not incentivized at all.

To define this generalization, we modify the IC model as follows: instead of activating all initially chosen nodes at time 0, the probability that a node is initially active equals its discount/incentive level, with 0 meaning no discount and 1 meaning a free product. The rest of the diffusion process is unchanged.

For each user $i = 1, \dots, n$, let $d_i \in [0, 1]$ be the discount/incentive the user receives, and let $\mathbf{d} = (d_1, \dots, d_n)$. Then user i becomes influenced at time 0 with probability d_i, i.e.,

$$Y_0^{(v)} = \begin{cases} 1, & \text{with probability } d_v, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Analogous to $\sigma(S)$ in (1), define the **influence** $\tilde{\sigma}(\mathbf{d})$ of a discount vector $\mathbf{d} \in [0, 1]^n$ as the expected number of active nodes at the end of the cascade (time U) when node i is initially active with probability d_i:

$$\tilde{\sigma}(\mathbf{d}) = \mathbb{E}\left[\sum_{v \in V} Y_U^{(v)} \,\middle|\, Y_0^{(v)} = 1 \text{ with probability } d_v \text{ for each } v \in V\right].$$

3.2 Problem Statement

Let G be a weighted directed social network and let $k \in \mathbb{R}_+$ be a budget. In FIM, the goal is to select $\mathbf{d} = (d_1, \dots, d_n)$ maximizing the spread of influence in G, where d_i denotes the discount allocated to v_i, subject to an L_1 budget constraint $\langle \mathbf{1}, \mathbf{d} \rangle \leq k$ and bounds $d_i \in [0, 1]$ for all i.

Problem 2 (Fractional Influence Maximization (FIM)). *Given a set V, a real-valued budget k, and an expected influence function $\tilde{\sigma}$, solve*

$$\begin{aligned} &\operatorname*{arg\,max}_{\mathbf{d} \in [0,1]^n} \ \tilde{\sigma}(\mathbf{d}), \\ &\quad s.t. \ \langle \mathbf{1}, \mathbf{d} \rangle \leq k. \end{aligned}$$

Lemma 1. *The expected influence function $\tilde{\sigma}$ for FIM is the multilinear extension of the DIM influence function σ,*

$$\tilde{\sigma}(\mathbf{d}) = \sum_{S \subseteq V} \sigma(S) \prod_{i : v_i \in S} d_i \prod_{i : v_i \notin S} (1 - d_i).$$

Proof. Define $\mathbf{Y}_u = \left(Y_u^{(1)}, \dots, Y_u^{(n)}\right)$ and $\mathbf{1} = (1, \dots, 1)$. The IC model can be written as

$$P_{\mathbf{Y}_0, \dots, \mathbf{Y}_U | \mathbf{d}} = P_{\mathbf{Y}_0 | \mathbf{d}} \, P_{\mathbf{Y}_1 | \mathbf{Y}_0} \prod_{u=2}^{U} P_{\mathbf{Y}_u | \mathbf{Y}_{u-1}, \mathbf{Y}_{u-2}},$$

so $\mathbf{d}$ only directly affects $\mathbf{Y}_0$. Let S_0 denote the random subset of indices active at time 0. Then

$$\begin{aligned} \tilde{\sigma}(\mathbf{d}) &= \mathbb{E}_{P_{\mathbf{Y}_0, \dots, \mathbf{Y}_U | \mathbf{d}}}[\langle \mathbf{1}, \mathbf{Y}_U \rangle] = \mathbb{E}_{P_{\mathbf{Y}_0 | \mathbf{d}}}\Big[\mathrm{E}_{P_{\mathbf{Y}_1, \dots, \mathbf{Y}_U | \mathbf{Y}_0}}[\langle \mathbf{1}, \mathbf{Y}_U \rangle \mid \mathbf{Y}_0]\Big] \\ &= \mathbb{E}_{P_{\mathbf{Y}_0 | \mathbf{d}}}[\sigma(S_0)] = \sum_{S \subseteq V} \sigma(S) \prod_{i : v_i \in S} d_i \prod_{i : v_i \notin S} (1 - d_i), \end{aligned}$$

where the last equality follows because d_i is the probability that v_i is active at time 0 as in (2). Since σ is a monotone submodular set function for DIM, $\tilde{\sigma}$ is its multilinear extension. □

FIM is a subclass of maximizing the multilinear extension of a monotone submodular set function under an L_1 budget constraint, and it is also a special case of CIM [Y. Yang et al. 2016]. CIM permits the probability that node i is a seed to be a non-linear function of d_i; in general, the CIM objective is not multilinear and can be more challenging to optimize. FIM generalizes DIM by allowing fractional incentives, while remaining more tractable than the broader model classes discussed in Section 1.2.3.

4 Main Results

In this section, we begin by proving important properties of FIM. We then propose *Multilinear Extension Greedy* (MLE-Greedy), a greedy $(1 - 1/e)$-approximation algorithm, for FIM. Unlike other methods, our method does not require either a value oracle or a gradient oracle for the multilinear function. Our method also does not require discretization or a step size to be chosen by the user. We also extend the proposed method to produce a continuous path of solutions that are near-optimal for all budgets from 0 up to the specified k, with negligible additional work needed.

4.1 Characterization of the Solution to FIM

We now discuss a series of important properties about FIM that will motivate our proposed greedy approximation algorithm. Since our proposed approximation algorithm is applicable not just for the FIM problem, but more generally for optimizing a multilinear extension $F(\mathbf{x})$ (of a monotone submodular set function $f(S)$), in the following we will use the more generic notation $F(\mathbf{x})$ instead of the FIM-specific $\tilde{\sigma}(\mathbf{d})$. Recall that we let $F : [0,1]^n \to \mathbb{R}$ denote the multilinear extension of a monotone non-decreasing submodular set function $f : 2^n \to \mathbb{R}$. Let $\mathbf{e}_i$ denote the basis vector for the ith dimension of $\mathbb{R}^n$. For a vector $\mathbf{z}$, the inequality $\mathbf{z} \geq 0$ denotes element-wise non-negativity.

Theorem 1 ([Calinescu, Chekuri, Pál, et al. 2007; Vondrák 2008]). *The multilinear extension F of a monotone non-decreasing submodular set function f satisfies the following properties everywhere in $[0,1]^n$:*

- *F is **non-decreasing** along any line of direction $\mathbf{z} \geq 0$,*
- *F is **concave** along any line of direction $\mathbf{z} \geq 0$, and*
- *F is **convex** along any line of direction $\mathbf{e}_i - \mathbf{e}_j$, $\forall\, i, j \in \{1, \dots, n\}$.*

Corollary 1.1. *There is an optimal solution $\mathbf{x}^*$ to FIM satisfying $\langle \mathbf{1}, \mathbf{x}^* \rangle = k$.*

Proof. This follows directly from Theorem 1. For any optimal solution $\mathbf{x}^*$ not on the hyperplane, meaning it satisfies $\langle \mathbf{1}, \mathbf{x}^* \rangle < k$, we can add to it any positive vector $\mathbf{z} \in \{\mathbf{z}' \mid 0 \leq z'_i \leq 1 - x^*_i,\ \langle \mathbf{1}, \mathbf{z}' \rangle = k - \langle \mathbf{1}, \mathbf{x}^* \rangle\}$ for which $\mathbf{x}^* + \mathbf{z}$ is on the hyperplane $\langle \mathbf{1}, \mathbf{x}^* + \mathbf{z} \rangle = k$ and remains feasible (i.e., in the hypercube $[0,1]^n$). By the non-decreasing property of Theorem 1, $F(\mathbf{x}^* + \mathbf{z}) \geq F(\mathbf{x}^*)$ so $\mathbf{x}^* + \mathbf{z}$ must also be optimal. □

Theorem 2. *For FIM, there is an optimal solution $\mathbf{x}^*$ with at most one coordinate that is not integer-valued.*

Proof. Let $\mathscr{P}$ denote the (n−1-dimensional) polytope formed by the intersection of the hypercube $[0,1]^n$ and the hyperplane $\langle \mathbf{1}, \mathbf{x} \rangle = k$,

$$\mathscr{P} = \left\{ \mathbf{x} \mid 0 \leq x_i \leq 1 \text{ for } i = 1, \dots, n,\ \sum_{i=1}^{n} x_i = k \right\}. \tag{3}$$

Let $\mathbf{x}^*$ denote an optimal solution to FIM in $\mathscr{P}$. By Corollary 1.1, there is such an optimal solution. Suppose $\mathbf{x}^*$ has at least two coordinates with non-integer values. We will show that we can iteratively identify a new optimal solution with one less non-integer-valued coordinate.

Without loss of generality, suppose $0 < x^*_1 < 1$ and $0 < x^*_2 < 1$. Let $\mathbf{e}_1$ and $\mathbf{e}_2$ denote the unit vectors for those dimensions, respectively. Construct a function $\psi : \mathbb{R} \to \mathbb{R}$ of F moving from $\mathbf{x}^*$ along the direction $\mathbf{e}_1 - \mathbf{e}_2$,

$$\psi(t) = F(\mathbf{x}^* + t(\mathbf{e}_1 - \mathbf{e}_2)).$$

By Theorem 1, ψ is convex. Let $\mathbf{x}(t) = \mathbf{x}^* + t(\mathbf{e}_1 - \mathbf{e}_2)$ denote the points in $\mathbb{R}^n$ along the line induced by t. We next argue $\mathbf{x}(t) \in \mathscr{P}$ for some t.

First, $\mathbf{x}(t)$ remains on the hyperplane induced by the budget constraint. For all $t \in \mathbb{R}$, $\langle \mathbf{1}, \mathbf{x}(t) \rangle = k$ by linearity. Second, we characterize the line segment $\mathbf{x}(t)$ that remains in $[0,1]^n$. Let $t_{\min}$ and $t_{\max}$ denote the values of t corresponding to the endpoints of that line segment,

$$t_{\min} = -\min\{x_1^*, 1 - x_2^*\}, \qquad t_{\max} = \min\{1 - x_1^*, x_2^*\}.$$

The endpoint $t_{\max} = \min\{1 - x_1^*, x_2^*\}$ corresponds to setting $x_1 \leftarrow 1$ or $x_2 \leftarrow 0$, whichever event happens first as t increases from 0. The endpoint $t_{\min} = -\min\{x_1^*, 1 - x_2^*\}$ likewise corresponds to $x_1 \leftarrow 0$ or $x_2 \leftarrow 1$. Denote the interval as $\mathscr{I} = [t_{\min}, t_{\max}]$. For every $t \in \mathscr{I}$, $\mathbf{x}(t) \in [0,1]^n$.

Thus, for $t \in \mathscr{I}$, $\mathbf{x}(t) \in \mathscr{P}$ and so remains feasible for FIM. Since ψ is convex on $\mathbb{R}$, it is convex on $\mathscr{I}$. By convexity, $F(\mathbf{x}^*) \leq \max\{F(\mathbf{x}(t_{\min})),\ F(\mathbf{x}(t_{\max}))\}$. Both $\mathbf{x}(t_{\min})$ and $\mathbf{x}(t_{\max})$ have one more integer-valued coordinate than $\mathbf{x}^*$, and so we can replace $\mathbf{x}^*$ with whichever one has the larger function value. Repeating yields an optimal solution with at most one non-integer coordinate. ☐

Corollary 2.1. *For FIM with an integer-valued budget k, there is an optimal solution* $\mathbf{x}^* \in \{0,1\}^n$.

Proof. By Theorem 2, there is an optimal solution $\mathbf{x}^*$ with at least $n-1$ integer-valued coordinates. By the constraint $\langle \mathbf{1}, \mathbf{x}^* \rangle = k$, the remaining coordinate must also be integer-valued when k is an integer. ☐

Since the multilinear extension F matches the set function f for integer-valued solutions, that is for $\mathbf{x} \in \{0,1\}^n$, with $S = \{i \mid 1 \leq i \leq n,\ x_i = 1\}$, we have $F(\mathbf{x}) = f(S)$, so Corollary 2.1 implies the optimal solutions also match for integer k.

Corollary 2.2. *For an integer-valued budget k, there is an optimal solution to DIM that is also an optimal solution to FIM.*

Proof. The proof follows directly from Corollary 2.1 using the optimal solution $\mathbf{x}^* \in \{0,1\}^n$. ☐

The following theorem refines Theorem 2.

Theorem 3. *For FIM with budget* $0 < t < n$, *there is an optimal solution* $\mathbf{x}^*$ *that is the convex combination of two (not necessarily optimal) solutions to FIM for integer budgets, one for budget* $\lfloor t \rfloor$ *and one for budget* $\lfloor t \rfloor + 1$.

Proof. The claim holds trivially for integer-valued t. Following Theorem 2, we first discuss the sets of extreme points for different budgets t with $0 \leq t \leq n$. We extend the notation of the polytope $\mathscr{P}$ (3) for different budgets. For $t \in [0,n]$, let $\mathscr{P}_t$ denote the polytope (3) for budget t. Let $\mathscr{E}_t$ denote the set of extreme points of $\mathscr{P}_t$. For integer values of t, $t \in \{0, 1, \ldots, n\}$,

$$\mathscr{E}_t = \{\mathbf{x} \mid \mathbf{x} \in \{0,1\}^n,\ \langle \mathbf{1}, \mathbf{x} \rangle = t\},$$

which are just the corners of the hypercube $[0,1]^n$ with exactly t non-zero coordinates.

For a non-integer budget t, the extreme points are those with exactly one non-integer coordinate, which necessarily has value $t - \lfloor t \rfloor$:

$$\mathscr{E}_t = \{\mathbf{x} \in [0,1]^n \mid \langle \mathbf{1}, \mathbf{x} \rangle = t,\ |\{i \mid x_i \notin \{0,1\}\}| \leq 1\}.$$

This also holds for integer t (then all coordinates are integer-valued).

By Theorem 2, there is an optimal solution $\mathbf{x}^* \in \mathscr{E}_t$. Fix any $\mathbf{x} \in \mathscr{E}_t$ and let i denote the coordinate with a non-integer value ($x_i = t - \lfloor t \rfloor$). Let $\mathbf{x}'$ be the point with the same coordinates as $\mathbf{x}$ except $x_i' = 0$; then $\mathbf{x}' \in \mathscr{E}_{\lfloor t \rfloor}$. Likewise, let $\mathbf{x}''$ be the point with the same coordinates as $\mathbf{x}$ except $x_i'' = 1$; then $\mathbf{x}'' \in \mathscr{E}_{\lfloor t \rfloor + 1}$. Thus $\mathbf{x}$, $\mathbf{x}'$, and $\mathbf{x}''$ are identical except for coordinate i. Letting $\theta = t - \lfloor t \rfloor$, we can express

$$\mathbf{x} = \mathbf{x}' + \theta(\mathbf{x}'' - \mathbf{x}').$$

Algorithm 1 MLE-Greedy

```
Input $k$.                                                              ▷ Budget (real-valued).
$\{\ldots, \mathbf{x}(\lfloor k \rfloor), \mathbf{x}(\lfloor k \rfloor + 1)\} \leftarrow$ DIM-Oracle $(\lfloor k \rfloor + 1)$    ▷ Calling the DIM-Oracle.
return $(1 - (k))\mathbf{x}(\lfloor k \rfloor) + (k)\mathbf{x}(\lfloor k \rfloor + 1)$             ▷ Calculating the FIM solution.
```

□

Corollary 3.1. *The optimal value of FIM is a piecewise-linear and convex function of the budget for* $0 < t < n$.

Proof. As shown in the proof of Theorem 3, for a non-integer t, any extreme point $\mathbf{x} \in \mathscr{E}_t$ can be written as $\mathbf{x}' + \theta(\mathbf{x}'' - \mathbf{x}')$ for a pair $(\mathbf{x}', \mathbf{x}'')$ with $\mathbf{x}' \in \mathscr{E}_{\lfloor t \rfloor}$ and $\mathbf{x}'' \in \mathscr{E}_{\lfloor t \rfloor + 1}$ that differ in exactly one coordinate, with $\theta = t - \lfloor t \rfloor \in [0, 1]$. Consider $\psi_{(\mathbf{x}', \mathbf{x}'')}(\theta) = F(\mathbf{x}' + \theta(\mathbf{x}'' - \mathbf{x}'))$. By multilinearity, $\psi_{(\mathbf{x}', \mathbf{x}'')}(\theta) = F(\mathbf{x}') + (F(\mathbf{x}'') - F(\mathbf{x}'))\theta$ is linear in θ. The optimal value at budget t is therefore the pointwise maximum (over a finite set) of linear functions of θ, hence piecewise-linear and convex. □

Using synthetic experiments, in Section 5.1 we explore the shape of F as a function of increasing budget, as well as a corresponding path of solutions.

We show an example where the path of optimal solutions $\mathbf{x}^*(t)$ is discontinuous. We next propose a simple greedy algorithm that exploits the combination of properties in Corollary 2.2 and Corollary 3.1.

4.2 Greedy Approximation Algorithm for Solving FIM

We now propose a greedy approximation algorithm for solving FIM, referred to as Multilinear Extension (MLE)-Greedy (Algorithm 1), using the results discussed in Section 4.1. As we will see in Section 4.2.1 and Section 4.2.2, MLE-Greedy is both simple algorithmically and near-optimal; we show near-optimality below in Theorem 4.

For a budget $k \in \mathbb{R}_+$, write $k = \lfloor k \rfloor + (k)$ where $\lfloor \cdot \rfloor$ and $(\cdot)$ denote the integral and fractional parts, respectively. Let $\mathbf{x}(\lfloor k \rfloor)$ and $\mathbf{x}(\lfloor k \rfloor + 1)$ denote *nested* approximations to the DIM problem for integer budgets $\lfloor k \rfloor$ and $\lfloor k \rfloor + 1$, respectively, with $\mathbf{x}(\lfloor k \rfloor) \leq \mathbf{x}(\lfloor k \rfloor + 1)$ element-wise, meaning they differ in a single coordinate (corresponding to nested subsets). Differing in a single coordinate is precisely the nested property and is key to the efficiency of our methods. We note that in general, optimal solutions for DIM are not nested (e.g. $\mathbf{x}^*(\lfloor k \rfloor) \not\leq \mathbf{x}^*(\lfloor k \rfloor + 1)$).

MLE-Greedy calls a subroutine DIM-Oracle to obtain nested solutions for budgets $\lfloor k \rfloor$ and $\lfloor k \rfloor + 1$. We then construct the solution to FIM by returning the convex combination

$$\mathbf{x}(k) := (1 - (k))\mathbf{x}(\lfloor k \rfloor) + (k)\mathbf{x}(\lfloor k \rfloor + 1)\,. \tag{4}$$

We now discuss the subroutine DIM-Oracle.

In [Nemhauser et al. 1978], the authors proposed a well-known greedy $(1 - 1/e)$-approximation algorithm for optimizing a monotone and submodular objective under a cardinality constraint. For DIM, more efficient greedy methods have been proposed, such as CELF [Leskovec, Krause, et al. 2007], CELF++ [Goyal, Lu, et al. 2011], community-based methods [Robson and Umrawal 2025; Umrawal and Aggarwal 2023; Umrawal, Quinn, et al. 2023], sketch-based methods [Cohen et al. 2014], and reverse influence sampling (RIS) based methods [Borgs et al. 2014; Guo et al. 2020; Tang, Shi, et al. 2015; Tang, Xiao, et al. 2014]. We note that greedy methods for maximizing a (generic) monotone submodular function f, such as CELF and CELF++, directly query f, while methods specially designed for parameterized problems like influence maximization, such as RIS, can achieve efficiency gains by avoiding direct evaluation. While we focus on influence maximization as a key application, our algorithms can be applied more generally for multilinear optimization.

4.2.1 Computational Complexity. For a given input k, MLE-Greedy first calls the subroutine DIM-Oracle exactly once to calculate the nested solutions to the DIM problem up to budget $\lfloor k \rfloor + 1$ (line 2). After that, MLE-Greedy calculates the convex combination of the solutions for budgets $\lfloor k \rfloor$ and $\lfloor k \rfloor + 1$ (line 3). Hence, the runtime of our method for budget k is the same as that of the subroutine used for the DIM-Oracle for budget $\lfloor k \rfloor + 1$. This runtime depends on the exact subroutine used, and improving this has been the subject of substantial prior work. The state-of-the-art asymptotic runtime for RIS methods with budget k is $O(nk \log n/\epsilon_1^2)$ (where ϵ_1 is a user-chosen parameter trading off error and computational complexity), achieved by [Guo et al. 2020], leading to the runtime for our method, MLE-Greedy, of $O(nk \log n/\epsilon_1^2)$. We used the code provided by the authors as the DIM-Oracle for all of our experiments in Section 5.2.

4.2.2 Approximation Guarantee. We next discuss the $(1-1/e)$-approximation guarantee for the solution obtained using the MLE-Greedy algorithm.

Theorem 4. *MLE-Greedy is a $(1 - 1/e)$-factor approximation algorithm for FIM. Equivalently,*

$$F(\mathbf{x}(k)) \geq (1 - 1/e)\, F(\mathbf{x}^*(k))\,.$$

We defer the proof. Theorem 4 will follow immediately from a more general result, Theorem 5, that we later present.

4.3 Near-Optimal Path

We extend (4), the convex combination MLE-Greedy forms from nested solutions for budgets $\lfloor k \rfloor$ and $\lfloor k \rfloor + 1$, into a continuous, piecewise-linear path corresponding to near-optimal solutions for any budget $t \in [0, n]$. To simplify the presentation, we discuss constructing a path for all budgets $t \in [0, n]$, but the construction can be terminated to obtain a path just for budgets $t \in [0, \lfloor k \rfloor + 1]$.

We begin by fixing $\mathbf{x}(t)$ for integer values of t, $t \in \{0, 1, \ldots, n\}$. Let $S_t \subseteq \Omega$ denote a subset of size t such that the csubsets are nested,

$$\emptyset = S_0 \subset S_1 \subset S_2 \subset \cdots \subset S_n = \Omega,$$

and are near-optimal with respect to the optimal solution with the same cardinality. In [Nemhauser et al. 1978], the authors proposed a well-known greedy algorithm for DIM that returns such a nested sequence with an approximation ratio $(1 - 1/e)$. Letting S_t^* denote an optimal solution to DIM for an (integer) budget t, we have

$$f(S_t) \geq (1 - 1/e)\, f(S_t^*) \quad \text{for all } t \in \{0, \ldots, n\}. \tag{5}$$

Construct the path $\mathbf{x}(t)$ as follows. For integer $t \in \{0, 1, \ldots, n\}$, for each coordinate $i \in \{1, \ldots, n\}$, set

$$x_i(t) = \begin{cases} 1, & \text{if } i \in S_t, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Next, for non-integer $t \in [0, n]$, set $\mathbf{x}(t)$ as a convex combination of $\mathbf{x}(\lfloor t \rfloor)$ and $\mathbf{x}(\lfloor t \rfloor + 1)$,

$$\mathbf{x}(t) = \mathbf{x}(\lfloor t \rfloor) + (t - \lfloor t \rfloor)\big(\mathbf{x}(\lfloor t \rfloor + 1) - \mathbf{x}(\lfloor t \rfloor)\big). \tag{7}$$

By construction, the coordinates of $\mathbf{x}(\lfloor t \rfloor)$ are binary-valued and differ from those of $\mathbf{x}(\lfloor t \rfloor + 1)$ in a single coordinate, namely the element j that gets added to $S_{\lfloor t \rfloor}$ to form $S_{\lfloor t \rfloor + 1}$.

4.3.1 Proof of Approximation Guarantee.

Theorem 5. *For all $t \in [0, n]$, the path $\mathbf{x}(t)$ is feasible for FIM for budget t and near-optimal, with*

$$F(\mathbf{x}(t)) \geq (1 - 1/e)\, F(\mathbf{x}^*(t)).$$

Proof of Theorem 5. We first note that continuity and piecewise-linearity follow by the path's construction from nested subsets used to define $\mathbf{x}(t)$ for integer values of t and then using convex combinations of those coordinates for non-integer values of t.

Feasibility. Feasibility follows by construction (6), as $0 \leq \mathbf{x}(t) \leq 1$ and thus remains in the hypercube $[0,1]^n$. For integer t, $\mathbf{x}(t)$ is binary-valued with exactly t elements and thus $\langle \mathbf{1}, \mathbf{x}(t) \rangle = t$. For non-integer t, by (7), $0 \leq \mathbf{x}(t) \leq 1$ and $\langle \mathbf{1}, \mathbf{x}(t) \rangle = t$.

Near-optimality.

$$
\begin{aligned}
F(\mathbf{x}(t)) &= F\big(\mathbf{x}(\lfloor t \rfloor) + (t - \lfloor t \rfloor)\big(\mathbf{x}(\lfloor t \rfloor + 1) - \mathbf{x}(\lfloor t \rfloor)\big)\big) && (8)\\
&= F(\mathbf{x}(\lfloor t \rfloor)) + (t - \lfloor t \rfloor)\Big(F(\mathbf{x}(\lfloor t \rfloor + 1)) - F(\mathbf{x}(\lfloor t \rfloor))\Big) && (9)\\
&= f(S_{\lfloor t \rfloor}) + (t - \lfloor t \rfloor)\big(f(S_{\lfloor t \rfloor + 1}) - f(S_{\lfloor t \rfloor})\big) && (10)\\
&= (1 - (t - \lfloor t \rfloor)) f(S_{\lfloor t \rfloor}) + (t - \lfloor t \rfloor) f(S_{\lfloor t \rfloor + 1}) && (11)\\
&\geq (1 - (t - \lfloor t \rfloor))\,(1 - 1/e)\, f(S^*_{\lfloor t \rfloor}) + (t - \lfloor t \rfloor)\,(1 - 1/e)\, f(S^*_{\lfloor t \rfloor + 1}) && (12)\\
&= (1 - 1/e)\left[f(S^*_{\lfloor t \rfloor}) + (t - \lfloor t \rfloor)\big(f(S^*_{\lfloor t \rfloor + 1}) - f(S^*_{\lfloor t \rfloor})\big)\right], && (13)\\
&= (1 - 1/e)\left[F(\mathbf{x}^*(\lfloor t \rfloor)) + (t - \lfloor t \rfloor)\big(F(\mathbf{x}^*(\lfloor t \rfloor + 1)) - F(\mathbf{x}^*(\lfloor t \rfloor))\big)\right] && (14)\\
&\geq (1 - 1/e)\, F(\mathbf{x}^*(t)). && (15)
\end{aligned}
$$

Here, (8) uses (7); (9) uses linearity of F along a unit direction (only one coordinate changes between $\lfloor t \rfloor$ and $\lfloor t \rfloor + 1$); (10) follows from (6); (11) groups terms (note that the coefficients $(1 - (t - \lfloor t \rfloor))$ and $(t - \lfloor t \rfloor)$ are both non-negative); (12) follows from (5) and near-optimality of the greedy DIM sequence [Nemhauser et al. 1978] ; (13) rearranges terms; and (15) follows from the piecewise linear upper envelope for the optimal value as a function of t established by Corollary 3.1. □

With effectively the same computational expense of identifying a near-optimal solution for the (maximum) budget k, the decision maker can identify near-optimal solutions for any budget $t \in [0, k]$. This gives the decision maker greater flexibility in evaluating whether they would want to expend the maximum budget or a smaller budget.

We note that once near-optimal solutions for integer budgets $t \in \{1, \ldots, \lfloor k \rfloor + 1\}$ are identified (line 2 of MLE-Greedy), the path segments between two pairs of adjacent integer budgets can be computed independently. This is in contrast to several other algorithms that optimize continuous analogs of submodular set functions [Bian et al. 2017; Calinescu, Chekuri, Pal, et al. 2011; Chekuri and Quanrud 2019; Chekuri, Vondrák, et al. 2014; L. Chen et al. 2020; W. Chen et al. 2020; Soma and Yoshida 2015; Vondrák 2008], which iteratively construct a path inside $[0,1]^n$ in steps beginning at $\mathbf{x} = 0$.

In the following sections, we discuss how this continuous path of near-optimal solutions allows the marketer to address cost–benefit trade-offs over a range of budgets, specifically budget selection for a desired influence and profit maximization.

4.3.2 *Budget Selection for Desired Influence.*

We first discuss how this continuous path of near-optimal solutions is used for budget selection for a desired influence $\tilde{\sigma}_{\text{desired}}$. Observe that the influence on the continuous path $\tilde{\sigma}(\mathbf{x}(t))$ is a monotonically increasing function in t. Furthermore, given a nested solution $\mathbf{x}$ to DIM up to a maximum integral budget k, we can efficiently find solutions to FIM $\mathbf{x}(t)$ for all real $t \in [0, k]$. Thus, given a desired influence $\tilde{\sigma}_{\text{desired}}$, we may use binary search to obtain a real $\tau \in [0, k]$ such that $\tilde{\sigma}_{\text{desired}} \approx \tilde{\sigma}(\mathbf{x}(\tau))$. Importantly, to perform this, the only substantial computation is the evaluation of $\tilde{\sigma}(\cdot)$, since the fractional solution $\mathbf{x}(t)$ can be computed in constant time given the DIM solution $\mathbf{x}$.

Algorithm 2 Budget-Influence

```
1:  Input $\mathbf{x}, \tilde{\sigma}_{\text{desired}}, \varepsilon_2, \tilde{k}_2$.    ▷ DIM solution, desired influence, precision, and max budget with
    $\tilde{\sigma}(\mathbf{x}(\tilde{k}_2)) \geq \tilde{\sigma}_{\text{desired}}$.
2:  $\text{low} \leftarrow 0, \text{high} \leftarrow \tilde{k}_2$    ▷ Variables for binary search.
3:  while $\text{low} + \varepsilon_2 < \text{high}$ do    ▷ Finding the desired budget using binary search.
4:      $\text{mid} \leftarrow (\text{low} + \text{high})/2$
5:      if $\tilde{\sigma}(\mathbf{x}(\text{mid})) < \tilde{\sigma}_{\text{desired}}$ then
6:          $\text{low} \leftarrow \text{mid}$
7:      else
8:          $\text{high} \leftarrow \text{mid}$
9:      end if
10: end while
11: return high
```

Remark 1. *This problem is similar to the submodular cover problem [Wolsey 1982], which seeks to identify a minimum-cost set that satisfies a submodular constraint. The submodular cover problem is NP-hard and does not have constant-factor approximation guarantees in general. There are approximation algorithms with problem-dependent guarantees [Wolsey 1982] and bi-criteria guarantees [Goyal, Bonchi, Lakshmanan, and Venkatasubramanian 2013].*

The pseudocode for our proposed procedure, Budget-Influence, is shown in Algorithm 2. The inputs on line 1 are the desired fractional influence $\tilde{\sigma}_{\text{desired}}$, the precomputed solution to the analogous DIM problem $\mathbf{x}$, the precision parameter ε_2, and the maximum budget $\tilde{k}_2$. The parameter $\tilde{k}_2$ is only used as an initial upper bound on the binary search; hence we require that $\tilde{\sigma}(\mathbf{x}(\tilde{k}_2)) \geq \tilde{\sigma}_{\text{desired}}$. This can be satisfied during the computation of the original DIM solution $\mathbf{x}$. Lines 3–12 are a standard application of continuous binary search to the function $\tilde{\sigma}(\mathbf{x}(\cdot))$. We return the upper bound on the final interval (line 15), as this is an overestimation of the budget required to give the final desired influence.

Observe that this algorithm uses $O(\log(\tilde{k}_2/\varepsilon_2))$ evaluations of the fractional influence function $\tilde{\sigma}(\cdot)$, since it terminates once the binary search interval is smaller than ε_2. Note that ε_2 controls the precision of the *budget* approximation along the near-optimal path computed from the DIM solution $\mathbf{x}$ (not the precision of the final output influence).

4.3.3 Budget Selection for Profit Maximization. We next discuss how this continuous path of near-optimal solutions is used for budget selection for profit maximization. Consider a marketing scenario where a marketer wishes to promote a product by sponsoring units of the product to influential individuals, where individuals are represented by nodes in a network and influence is represented by edges. We assume that each person will purchase at most a single unit of the product, and people are influenced by other nodes in this network to purchase units of the product at a fixed price.

Let $p \geq 0$ be the price per unit of the product and $c \geq 0$ be the cost of sponsoring one unit of the product. The overall profit $\pi(t)$ that the marketer makes by sponsoring t units is

$$\begin{aligned}\text{total profit} &= \text{total revenue} - \text{total cost of sponsorship},\\ &= \text{price per unit} \times \text{influence} - \text{cost per unit} \times \text{budget},\\ \pi(t) &= p\,\tilde{\sigma}(\mathbf{x}(t)) - ct. \end{aligned} \tag{16}$$

The total revenue $p\,\tilde{\sigma}(\mathbf{x}(t))$ is non-decreasing and, as a function of t along our path, concave; the total cost ct is linear. Hence, the total profit $\pi(t)$ is a concave function of t.

Algorithm 3 BUDGET-PROFIT

```
1:  **Input** $\pi, \varepsilon_3, \tilde{k}_3$.    ▷ Profit function (Equation (16)), precision, max budget, such that $[0, \tilde{k}_3]$ contains the
    maximum value of $\pi$ and $\mathbf{x}$ evaluated to a budget of at least $\tilde{k}_3$.
2:  low ← 0, high ← $\tilde{k}_3$                                ▷ Variables for ternary search.
3:  **while** low + $\varepsilon_3$ < high **do**                ▷ Finding the desired budget using ternary search.
4:      $m_1$ ← low + (high − low)/3
5:      $m_2$ ← high − (high − low)/3
6:      $\pi_1$ ← $\pi(m_1)$
7:      $\pi_2$ ← $\pi(m_2)$
8:      **if** $\pi_1 < \pi_2$ **then**
9:          low ← $m_1$
10:     **else**
11:         high ← $m_2$
12:     **end if**
13: **end while**
14: **return** high
```

We are interested in maximizing $\pi(t)$ over t. Because π is concave (but not monotone), binary search is not appropriate; we instead compute the budget t^* maximizing π using ternary search [CP-Algorithms 2025]. We propose BUDGET-PROFIT for this problem. See Algorithm 3 for the pseudocode. It evaluates π (Equation (16)) as a subroutine. The inputs are the profit function π, the error parameter ε_3, and the maximum budget $\tilde{k}_3$. We return the upper bound on the final search interval (line 15). Similar to BUDGET-INFLUENCE, $\tilde{k}_3$ is used as an upper bound for the ternary search, and ε_3 is the precision of the approximation to the optimal budget t^*.

BUDGET-PROFIT uses $O(\log(\tilde{k}_3/\varepsilon_3))$ evaluations of π, since it terminates once the ternary search interval is smaller than ε_3. Note that ε_3 controls the precision of the budget approximation (not the precision of the final output influence).

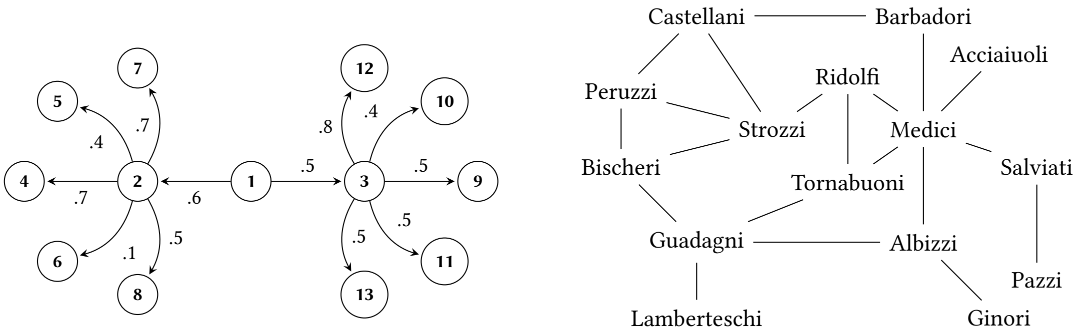
5 Experiments

In this section, we provide three sets of experiments: 1) calculating the optimal solution on small networks to demonstrate the discontinuity in the path for the solution and to show how well nested subsets selected by a greedy search can approximate the former, 2) evaluating our greedy approximation algorithm on large real-world networks, and 3) computing the near-optimal path of solutions to address the cost-benefit trade-offs over a range of budgets. The data and source code for this paper are available at https://github.com/abhishekumrawal/Fractional-IM-JAIR.

5.1 Optimal Solution

We ran experiments on two small networks to empirically explore the structure of optimal solutions to FIM, focusing on the piecewise linearity and convexity of the optimal value function F between integer-valued budgets, as well as the behavior of the corresponding solution path as the budget varies.

By Corollary 2.2, optimal solutions for integer-valued budgets may be obtained by solving the discrete influence maximization (DIM) problem. By Corollary 3.1, when optimal solutions for successive integer budgets are nested, the optimal solution for any intermediate budget is given by a convex combination of those integer-budget solutions, yielding a continuous path. When optimal solutions for integer budgets are not nested, however, the



(a) Synthetic network.

(b) Florentine families network [Breiger and Pattison 1986].

Fig. 1. Synthetic network and the Florentine families network [Breiger and Pattison 1986].

path of optimal solutions for non-integer budgets need not be continuous, even though the optimal value as a function of the budget remains piecewise linear and convex.

5.1.1 *Network Data.*

We used two networks. The first is a synthetic directed network with 13 nodes and 12 edges, shown in Figure 1a. Edge activation probabilities for this network are indicated next to the edges and were chosen to induce asymmetric downstream influence.

The second network is the Florentine families network [Breiger and Pattison 1986], consisting of 15 nodes and 40 edges, shown in Figure 1b. As this network is undirected, each edge was replaced by two directed edges. For edge weights, we used the weighted cascade model [Kempe et al. 2003], assigning each incoming edge to a node v a weight of $1/\text{in-degree}(v)$.

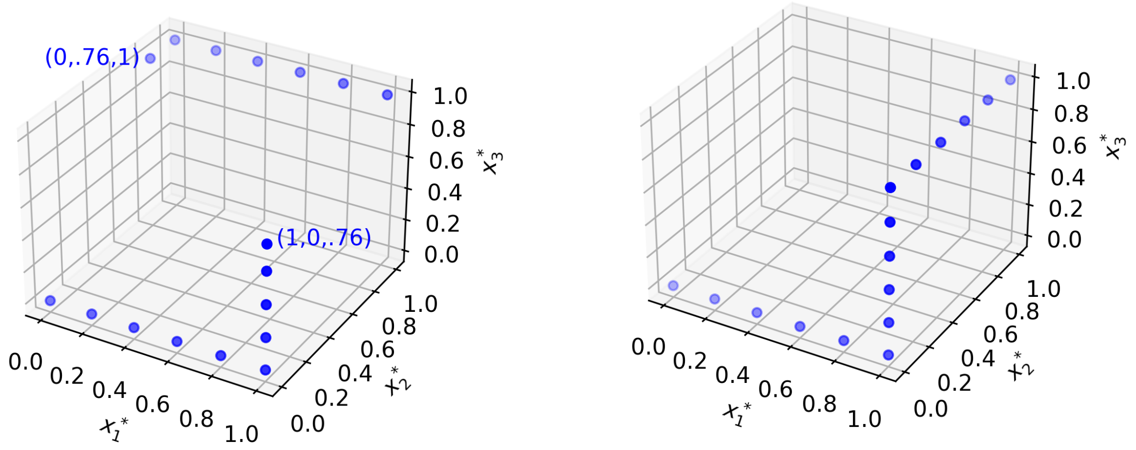
5.1.2 *Experiment Details.*

For both networks, we evaluated the influence function $F(\mathbf{x})$ over a discretized feasible region. Each coordinate x_i was restricted to the grid $\{0, 0.01, 0.02, \ldots, 1\}$, inducing a uniform discretization of the hypercube $[0,1]^n$. For binary allocations $\mathbf{x} \in \{0,1\}^n$ satisfying $\langle \mathbf{1}, \mathbf{x} \rangle \leq 3$, the value of $F(\mathbf{x})$ was estimated using 1,000 Monte Carlo simulations of the independent cascade process, and the empirical mean diffusion size was used as an estimate of $F(\mathbf{x})$. For all remaining (fractional) allocations on the discretized feasible region, $F(\mathbf{x})$ was computed using its multilinear extension, following Lemma 1. By construction, this multilinear extension coincides with the DIM influence function on all binary allocations and extends it to fractional incentives.

For the synthetic network, after computing optimal solutions to DIM for budgets $k = 1, 2, 3$, we focused on fractional allocations supported on the three nodes selected at budget $k = 3$. This restriction yields a three-dimensional subcube $[0,1]^3$ of the original feasible region; for any fixed budget k, the constraint $\langle \mathbf{1}, \mathbf{x} \rangle = k$ defines a two-dimensional hyperplane slice within this subcube. Over this region, the values of F at the eight binary corners of $\{0,1\}^3$ were estimated via Monte Carlo simulation, and the value of F at every fractional allocation on the discretized grid was computed analytically via multilinear interpolation.

Incentive levels were discretized with a step size of 0.01 in each coordinate. For each fixed budget k, we exhaustively maximized $F(\mathbf{x})$ over all discretized allocations satisfying $\langle \mathbf{1}, \mathbf{x} \rangle = k$, yielding an approximate maximizer on the corresponding hyperplane. The budget values shown in the figures correspond to selected values used for visualization; the optimization itself is carried out over the full discretized feasible set.

5.1.3 *Results and Discussion.*



(a) General optimal solution for the synthetic network.

(b) Nested optimal solution for Florentine families network.

Fig. 2. Path of $\mathbf{x}^*$ for different values of k. $k = \langle \mathbf{1}, \mathbf{x}^* \rangle$.

Synthetic network. For the synthetic network, we found that the optimal solutions to DIM for integer budgets $k \in \{1, 2, 3\}$ were not nested. Up to relabeling for clarity, the optimal seed sets were

$$S_1 = \{1\}, \qquad S_2 = \{2, 3\}, \qquad S_3 = \{1, 2, 3\}.$$

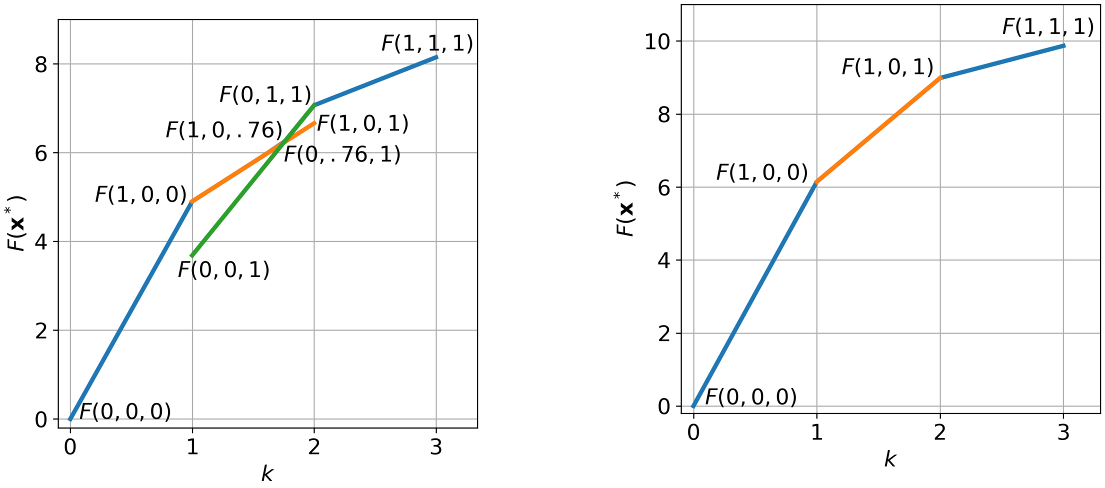
Within the discretized fractional domain described above, the maximizing allocation changes abruptly as the budget varies. As a result, the solution path $\mathbf{x}^*(k)$ exhibits a discontinuity for budgets in the interval $[2, 3]$. In contrast, the optimal value function $F(\mathbf{x}^*(k))$ remains piecewise linear and convex, corresponding to the upper envelope of linear segments associated with different pairs of integer-budget solutions. The resulting solution path and value function are shown in Figure 2a and Figure 3a, respectively.

Florentine families network. For the Florentine families network, the optimal solutions to DIM for budgets up to $k = 3$ were nested. By Corollary 3.1, this implies that optimal solutions for all budgets in $[0, 3]$ form a continuous path obtained via convex combinations of successive integer-budget optima. The solution path and corresponding optimal values are shown in Figure 2b and Figure 3b. In these figures, only the three nonzero coordinates of $\mathbf{x}^*$ are reported.

5.2 Greedy Approximation and Near-Optimal Path

In this section, we discuss two sets of experiments: 1) evaluating the performance of our greedy approximation in terms of solution quality and computational efficiency, and 2) constructing the continuous path of near-optimal solutions to address the cost/benefit trade-offs over a range of budgets.

5.2.1 Experimental Setup. We first describe the network data and experimental setup shared between these two sets of experiments.



(a) General optimal solution for the synthetic network.

(b) Nested optimal solution for Florentine families network.

Fig. 3. Path of $F(\mathbf{x}^*)$ for different values of k. $k = \langle \mathbf{1}, \mathbf{x}^* \rangle$.

Network Data. We ran experiments using several real-world social network graphs. The graph data is available from the Stanford Large Network Dataset Collection [Leskovec and Krevl 2014]. The number of nodes, number of edges, and type of each network are provided in Table 1.

The Amazon [J. Yang and Leskovec 2015] network is a graph of products purchased together on Amazon. The DBLP [J. Yang and Leskovec 2015] network is a graph of co-authorship for research publications in computer science. The Deezer [Rozemberczki and Sarkar 2020] network is a graph of mutual follower relationships among Deezer users from Europe. The Facebook [Leskovec and Mcauley 2012] network is a graph representing circles (or friends lists) from Facebook. The Wikipedia [Leskovec, Huttenlocher, et al. 2010a,b] network is a who-votes-on-whom graph of Wikipedia users to become an administrator. The YouTube [J. Yang and Leskovec 2015] network is a friendship graph among users of the platform.

Each edge in the undirected networks is replaced by two directed edges. For edge weights, we use the *weighted cascade model* [Kempe et al. 2003] where for each node $v \in V$, the weight of each edge entering v was set to $1/\text{in-degree}(v)$.

Table 1. Basic information of the networks used.

Network	Nodes	Edges	Type
Amazon	334,863	925,872	Undirected
DBLP	317,080	1,049,866	Undirected
Deezer	28,281	92,752	Undirected
Facebook	4,039	88,234	Undirected
Wikipedia	7,115	103,689	Directed
YouTube	1,134,890	2,987,624	Undirected

Experimental Details (Overall). Our experimental benchmarks mainly used Python, using the Pooch library [Uieda et al. 2020] to automate downloading graph data files and the NetworkX library [Hagberg et al. 2008] to process these data. To compute the influence of given seed sets, we evaluated their influence with 10,000 Monte Carlo simulations each, using the CyNetDiff library [Robson, Reddy, et al. 2024] for efficient computation of the diffusion process. All experiments were run on a workstation with a 3.00 GHz, 8-core Intel Xeon E5-1660v3 CPU and 64GB of memory.

5.2.2 *Greedy Approximation.*

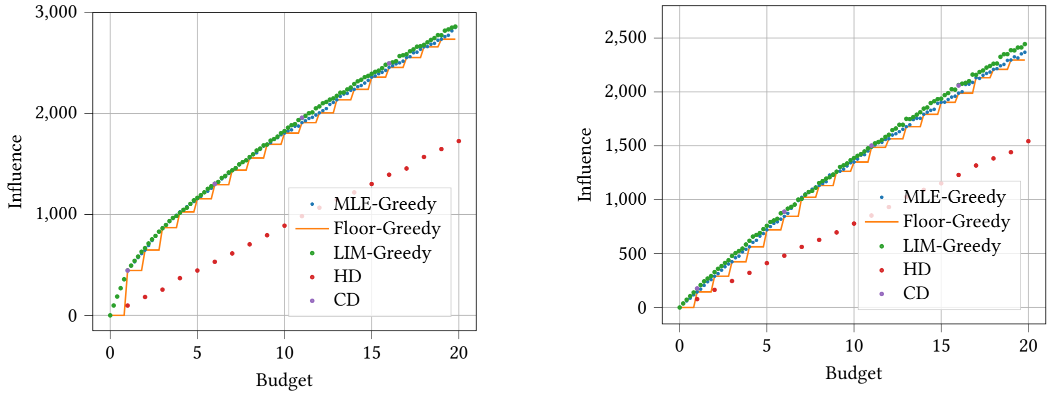
Algorithms. As the proposed algorithm MLE-Greedy employs a subroutine for solving DIM, we use the state-of-the-art RIS-based greedy algorithm [Guo et al. 2020] as a black-box DIM solver subroutine, with C++ code provided by the authors. For the purpose of assessing the performance of the proposed greedy algorithm, we use the following baselines. All implementations besides Floor-Greedy were provided by [W. Chen et al. 2020] in C++.

1. **Floor-Greedy** — for any budget k, calculate $\textsc{Oracle}(F, \lfloor k \rfloor)$, i.e., ignore the fractional budget.
2. **LIM-Greedy** [W. Chen et al. 2020] — a state-of-the-art, RIS-based adaptation of gradient methods for DR-submodular maximization for the problem of IM.
3. **HD** [W. Chen et al. 2020] — a heuristic baseline: choose top M nodes with the highest degrees from V and then distribute the budget to those M nodes proportional to their degrees. We set $M = 200$ in our experiments.
4. **CD** [Y. Yang et al. 2016] — an algorithm for personalized marketing. For each discount $c \in \{0.1, 0.2, \ldots, 1.0\}$, the algorithm returns a vector $\mathbf{x}$ such that $\mathbf{x}_i \in \{0, c\}$ for every i. Then an exhaustive search is run for the best value of c. Finally, it runs coordinate descent to achieve a better result.

Experimental Details (Additional). For LIM-Greedy [W. Chen et al. 2020], we set the discretization parameter $\delta = 0.1$ and error parameter $\epsilon = 0.5$ as suggested by the authors. For the RIS-based greedy, we set the error parameter $\epsilon_1 = 0.01$, as suggested by the authors. For our experiments, we compared our method and the baselines in terms of the influence of the output partial incentive vector and empirical run times. We ran the methods for all budgets $k \in \{0.2, 0.4, \ldots, 20\}$. We used Python to implement MLE-Greedy. The source code of the RIS-based greedy algorithm [Guo et al. 2020] and LIM-Greedy algorithm [W. Chen et al. 2020] provided by their authors are written in C++.

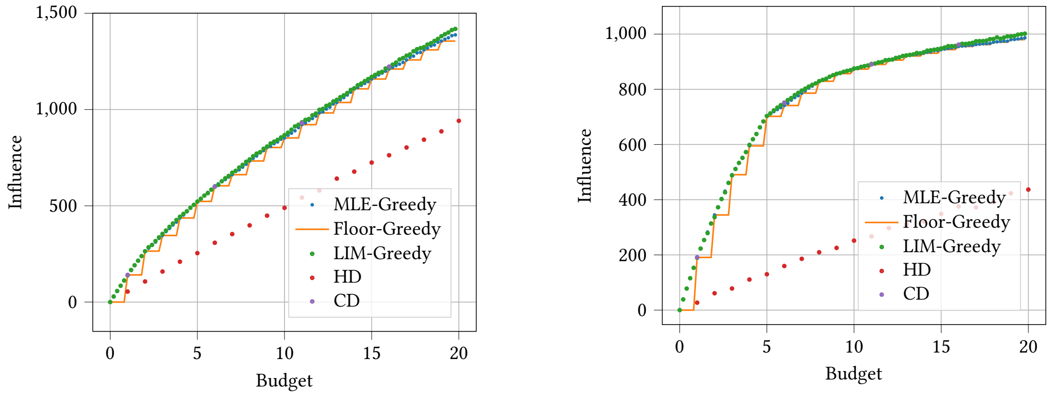
Results. Figure 4 provides the influence (on y-axis) using different algorithms for different values of the budget (on x-axis) for different networks. For each algorithm, the approximate solutions are obtained using a single run for a budget of 20. Approximate solutions for budgets lower than 20 are then obtained as subsets of the approximate solution for budget 20, as our method and the baseline obtain sequences of solutions up to the input budget. Figure 5 depicts the runtime in seconds (y-axis) using different algorithms for different values of the budget (x-axis), for different networks. We use a log scale for plotting, as actual runtime values for MLE-Greedy are much smaller (fractions of a second) than those for LIM-Greedy.

Discussion. From Figure 4, we note that in terms of influence, for all the networks, our proposed algorithm MLE-Greedy outperforms the Floor-Greedy algorithm for non-integer budgets. Furthermore, for all the networks, the influence values are almost the same for MLE-Greedy and LIM-Greedy algorithms except for high (> 15) values of the budget, where LIM-Greedy performs slightly better. From Figure 5, we note that the proposed MLE-Greedy takes orders of magnitude less time than LIM-Greedy across all networks. Moreover, for all the networks, the runtime savings (i.e., the gap in runtimes) due to MLE-Greedy compared to LIM-Greedy increase both as the budget and the network size increase. Furthermore, by comparing across Figures 4c to 4e, we note



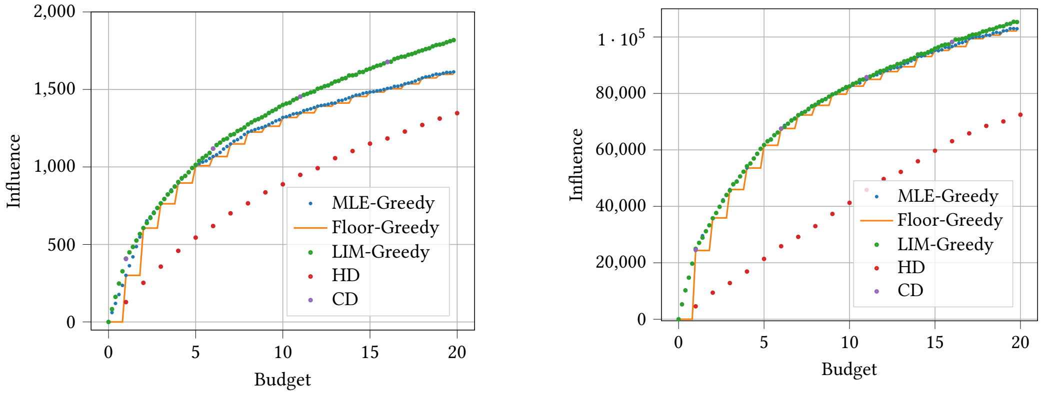
(a) Amazon network

(b) DBLP network



(c) Deezer network

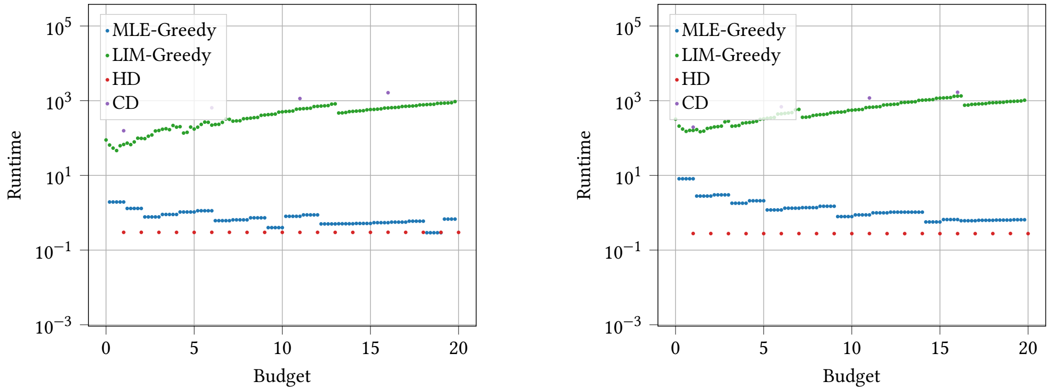
(d) Facebook network



(e) Wikipedia network

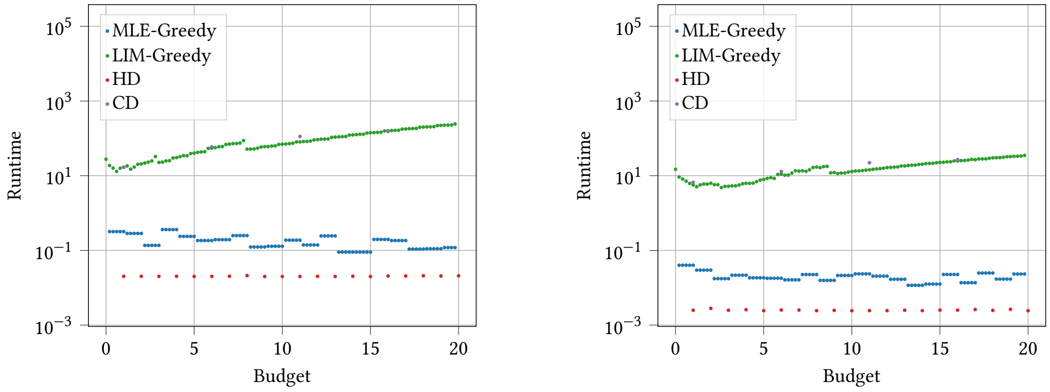
(f) YouTube network

Fig. 4. Influence vs. Budget (k) for different networks.



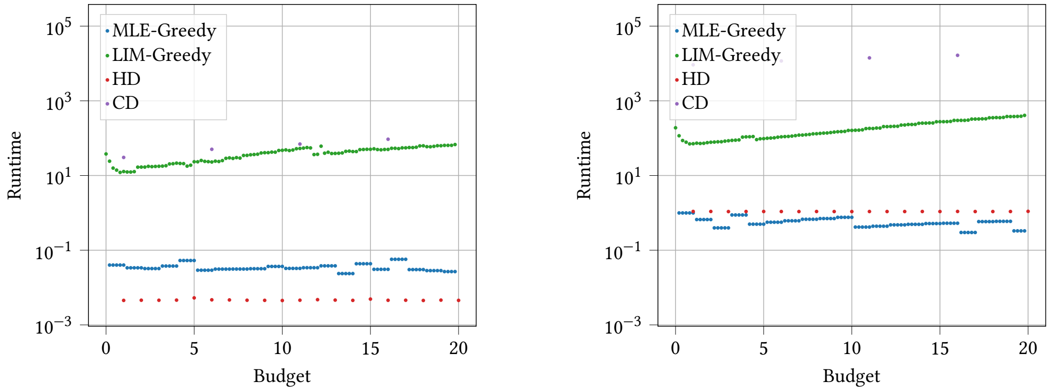
(a) Amazon network

(b) DBLP network



(c) Deezer network

(d) Facebook network



(e) Wikipedia network

(f) YouTube network

Fig. 5. Runtime (in seconds) vs. Budget (k) for different networks.

that this increase in runtime savings is increasing in the size of the network (n). This suggests that MLE-Greedy is highly scalable, even for large networks.

5.2.3 Near-Optimal Path. We ran experiments to construct the continuous path of near-optimal solutions to illustrate: 1) budget selection for desired influence and 2) budget selection for profit maximization. This demonstrates the utility of our method in enabling the marketer to address the cost-benefit trade-offs across a range of budgets.

Algorithms. We implemented Budget-Influence for budget selection for desired influence and Budget-Profit for budget selection for profit maximization. In these experiments, we did not include other baseline methods, as it was unclear whether other algorithms could be used as a subroutine for the high-level search methods outlined here.

Experimental Details. For all of our experiments, for both Algorithms 2 and 3, we used error parameters $\epsilon_2 = \epsilon_3 = 0.1$, and the maximum budgets $\tilde{k}_2 = \tilde{k}_3 = 20$. The values of cost per unit and price per unit for the profit function π (an input to Budget-Profit) were set at $c = 100$ and $p = 1$, respectively. Furthermore, we used the RIS-based greedy algorithm [Guo et al. 2020] to generate the initial DIM solution vector $\mathbf{x}$ which is an input to Budget-Influence and the profit function Equation (16), used as a subroutine in Budget-Profit. That time to compute this solution vector was reported separately from the reported running times for Algorithms 2 and 3 in Tables 2 and 3, respectively, as Algorithms 2 and 3 are agnostic to how the initial DIM solution vector is computed. They are reported in Table 3 for each network (and are less than 1 second for each network). We used Python to implement Algorithms 2 and 3. The source code of the RIS-based greedy algorithm [Guo et al. 2020] provided by their authors is written in C++.

Results and Discussion. The results of these experiments are shown in Tables 2 and 3. Table 2 provides the desired influence, actual influence, budget used, and runtime using Budget-Influence for different networks. Table 3 provides the maximum profit, budget used, and runtime using Budget-Profit for different networks. In addition, Table 3 includes the initial computation time of the RIS-based greedy algorithm for each network for the stated maximum budget. The difference in running time is due to the sample efficiency of this algorithm, along with the difference in programming language used.

From Table 2, we observe that the runtime of our algorithm scales well with both the size of the input graph and the desired influence, as the running time is dominated by the evaluations of the fractional influence function. The actual influence differs from the desired influence because of the inclusion of the error parameter ε_2 and the fact that the fractional influence function is evaluated with Monte Carlo simulations. From Table 3, we observe Budget-Profit efficiently scales for some of the largest networks we considered in this work.

Furthermore, from Tables 2 and 3, we note that the desired budgets are real-valued, pointing out the benefit of partial incentives. Additionally, from Table 3, for the chosen per unit price and cost values, the maximum profit is achieved quite before the maximum budget $\tilde{k}_3 = 20$. This illustrates that the submodularity (diminishing returns) of the influence suggests the marketer stops spending the budget beyond when the total cost of sponsorships outweighs the marketing gains through product adoptions.

6 Conclusion and Future Work

We considered an important generalization of the widely studied discrete influence maximization problem, which accounted for users being partially incentivized. Despite this generalization being a continuous optimization problem, we proposed an efficient $(1 - 1/e)$-approximation algorithm that interpolates solutions to the discrete problem, making it significantly more tractable than other generalizations. Furthermore, we discussed how this algorithm can be used by a marketer to evaluate cost–benefit trade-offs over a range of budgets, including

Table 2. Experiments demonstrating the practicality of the near-optimal path for budget selection for desired influence. All experiments in this table used error parameter $\varepsilon_2 = 0.1$ and maximum budget $\tilde{k}_2 = 20$. The listed runtime (in seconds) does not include the time taken in computing the initial DIM solution—our algorithm is agnostic to the DIM solution method used. See Table 3 for those times.

Network	Desired influence	Actual influence	Budget used	Runtime
Amazon	100	106.15	0.23	10.47
	200	208.94	0.47	10.94
	500	501.37	1.25	13.31
	1,000	999.76	3.98	21.07
DBLP	100	105.42	0.70	11.19
	200	205.68	1.41	12.73
	500	502.74	3.52	17.10
	1,000	994.99	6.88	27.16
Deezer	100	100.54	0.70	6.28
	200	201.35	1.48	7.58
	500	508.52	4.84	12.23
	1,000	1,005.11	12.42	22.91
Facebook	100	105.17	0.55	20.41
	200	204.97	1.09	24.12
	500	503.06	3.12	34.95
	1,000	989.84	20.0	65.31
Wikipedia	100	115.66	0.39	29.57
	200	208.73	0.70	32.69
	500	522.52	1.72	44.84
	1,000	1,011.63	5.00	64.92

Table 3. Experiments demonstrating the practicality of the near-optimal path for budget selection for profit maximization. All experiments in this table used error parameter $\varepsilon_3 = 0.1$, cost per unit $c = 100$, price per unit $p = 1$, and maximum budget $\tilde{k}_3 = 20$. The listed runtime (in seconds) does not include the time taken in computing the initial DIM solution—our algorithm is agnostic to the DIM solution method used.

Network	Maximum profit	Budget used	Runtime	Initial computation time
Amazon	870.09	13.73	123.45	0.68
DBLP	417.10	17.08	133.57	0.65
Deezer	62.23	2.12	17.68	0.12
Facebook	200.88	4.82	60.59	0.02
Wikipedia	510.71	4.97	88.98	0.03

budget selection for a desired level of influence and profit maximization. We also evaluated the performance and scalability of the methods proposed in this paper through extensive theoretical and empirical analyses.

One important future direction is to explore whether the theoretical and algorithmic results can be extended to broader classes of problems, such as those involving more complex constraints or alternative diffusion models. Other future directions include studying partial incentives in competitive settings with multiple marketers

[Bharathi et al. 2007] and in online settings [Agarwal et al. 2022; Nie et al. 2022; Xu and Umrawal 2026]. Investigating fairness in influence maximization [Becker et al. 2022; Lin et al. 2023; Nguyen et al. 2022] under partial incentives is also a compelling direction for future work.

7 Acknowledgments

This work was supported in part by the National Science Foundation under Grants No. 1742847, 2149588, and 2149617. A preliminary version of this paper appeared in the 2023 IEEE Conference on Decision and Control (CDC) [Umrawal, Aggarwal, and Quinn 2023].

References

M. Agarwal, V. Aggarwal, A. K. Umrawal, and C. J. Quinn. 2022. "Stochastic Top K-Subset Bandits with Linear Space and Non-Linear Feedback with Applications to Social Influence Maximization." *ACM/IMS Trans. on Data Science (TDS)*, 2, 4, 1–39.

A. A. Ageev and M. I. Sviridenko. 2004. "Pipage rounding: A new method of constructing algorithms with proven performance guarantee." *Journal of Combinatorial Optimization*, 8, 3, 307–328.

CP-Algorithms. 2025. *Ternary Search*. https://cp-algorithms.com/num_methods/ternary_search.html. Accessed: 2025-03-23. (2025).

R. Becker, G. D'Angelo, S. Ghobadi, and H. Gilbert. 2022. "Fairness in Influence Maximization through Randomization." *J. Artif. Intell. Res.*, 73, 1251–1283. doi:10.1613/JAIR.1.13367.

S. Bharathi, D. Kempe, and M. Salek. 2007. "Competitive influence maximization in social networks." In: *International Workshop on Web and Internet Economics*. Springer, 306–311.

A. Bhimaraju, E. W. Robson, L. R. Varshney, and A. K. Umrawal. 2024. "Fractional Budget Allocation for Influence Maximization under General Marketing Strategies." In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 3627–3631.

A. A. Bian, B. Mirzasoleiman, J. Buhmann, and A. Krause. 2017. "Guaranteed non-convex optimization: Submodular maximization over continuous domains." In: *Artificial Intelligence and Stat.* PMLR, 111–120.

C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. 2014. "Maximizing social influence in nearly optimal time." In: *Proc. Twenty-fifth Annual ACM-SIAM Symp. on Discrete algorithms*. SIAM, 946–957.

R. L. Breiger and P. E. Pattison. 1986. "Cumulated social roles: The duality of persons and their algebras." *Social Networks*, 8, 3, 215–256.

G. Calinescu, C. Chekuri, M. Pal, and J. Vondrák. 2011. "Maximizing a monotone submodular function subject to a matroid constraint." *SIAM J. on Computing*, 40, 6, 1740–1766.

G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. 2007. "Maximizing a submodular set function subject to a matroid constraint." In: *Int. Conf. on Int. Prog. and Comb. Opt.* Springer, 182–196.

C. Chekuri and K. Quanrud. 2019. "Submodular function maximization in parallel via the multilinear relaxation." In: *Proc. 30th Annual ACM-SIAM Symp. on Discrete Algorithms*. SIAM, 303–322.

C. Chekuri, J. Vondrák, and R. Zenklusen. 2014. "Submodular function maximization via the multilinear relaxation and contention resolution schemes." *SIAM J. on Computing*, 43, 6, 1831–1879.

L. Chen, M. Zhang, H. Hassani, and A. Karbasi. 2020. "Black box submodular maximization: Discrete and continuous settings." In: *Int. Conf. on Artificial Intelligence and Stat.* PMLR, 1058–1070.

W. Chen, R. Wu, and Z. Yu. 2020. "Scalable lattice influence maximization." *IEEE Trans. Comp. Soc. Sys.*, 7, 4, 956–970.

E. Cohen, D. Delling, T. Pajor, and R. F. Werneck. 2014. "Sketch-based influence maximization and computation: Scaling up with guarantees." In: *Proc. 23rd ACM Int. Conf. on Info. and Know. Mgmt.* 629–638.

E. D. Demaine, M. Hajiaghayi, H. Mahini, D. L. Malec, S. Raghavan, A. Sawant, and M. Zadimoghadam. 2014. "How to influence people with partial incentives." In: *Proc. 23rd Int. Conf. WWW*, 937–948.

P. Domingos and M. Richardson. 2001. "Mining the network value of customers." In: *Proc. Seventh ACM SIGKDD Int. Conf. on Know. Disc. and Data Mining*. ACM, 57–66.

M. Frank, P. Wolfe, et al.. 1956. "An algorithm for quadratic programming." *Naval research logistics quarterly*, 3, 1-2, 95–110.

J. Goldenberg, B. Libai, and E. Muller. 2001a. "Talk network: A complex systems look at the underlying process of word-of-mouth." *Marketing letters*, 12, 3, 211–223.

J. Goldenberg, B. Libai, and E. Muller. 2001b. "Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata." *Academy of Marketing Science Review*, 9, 3, 1–18.

A. Goyal, F. Bonchi, and L. V. Lakshmanan. 2011. "A data-based approach to social influence maximization." *Proc. VLDB Endowment*, 5, 1, 73–84.

A. Goyal, F. Bonchi, L. V. Lakshmanan, and S. Venkatasubramanian. 2013. "On minimizing budget and time in influence propagation over social networks." *Social network analysis and mining*, 3, 2, 179–192.

A. Goyal, W. Lu, and L. V. Lakshmanan. 2011. “CELF++: Optimizing the greedy algorithm for influence maximization in social networks.” In: *Proc. 20th Int. Conf. Comp. WWW*, 47–48.

M. Granovetter. 1978. “Threshold models of collective behavior.” *American journal of sociology*, 83, 6, 1420–1443.

Q. Guo, S. Wang, Z. Wei, and M. Chen. 2020. “Influence maximization revisited: Efficient reverse reachable set generation with bound tightened.” In: *Proc. 2020 ACM SIGMOD Int. Conf. on Management of Data*, 2167–2181.

A. A. Hagberg, D. A. Schult, and P. J. Swart. 2008. “Exploring Network Structure, Dynamics, and Function using NetworkX.” In: *Proceedings of the 7th Python in Science Conference.* Ed. by G. Varoquaux, T. Vaught, and J. Millman. Pasadena, CA USA, 11–15.

H. Hassani, M. Soltanolkotabi, and A. Karbasi. 2017. “Gradient methods for submodular maximization.” *Advances in Neural Information Processing Systems*, 30.

H. Kamarthi, P. Vijayan, B. Wilder, B. Ravindran, and M. Tambe. 2020. “Influence Maximization in Unknown Social Networks: Learning Policies for Effective Graph Sampling.” In: *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS ’20, Auckland, New Zealand, May 9-13, 2020.* Ed. by A. E. F. Seghrouchni, G. Sukthankar, B. An, and N. Yorke-Smith. International Foundation for Autonomous Agents and Multiagent Systems, 575–583. doi:10.5555/3398761.3398831.

D. Kempe, J. Kleinberg, and É. Tardos. 2003. “Maximizing the spread of influence through a social network.” In: *Proc. 9th ACM SIGKDD Int. Conf. on Know. Disc. and Data Mining*. ACM, 137–146.

S. Kumar, A. Mallik, A. Khetarpal, and B. Panda. 2022. “Influence maximization in social networks using graph embedding and graph neural network.” *Information Sciences*, 607, 1617–1636. doi:https://doi.org/10.1016/j.ins.2022.06.075.

J. Leskovec, D. Huttenlocher, and J. Kleinberg. 2010a. “Predicting positive and negative links in online social networks.” In: *Proc. 19th Int. Conf. WWW*, 641–650.

J. Leskovec, D. Huttenlocher, and J. Kleinberg. 2010b. “Signed networks in social media.” In: *Proc. SIGCHI Conf. on Human Factors in Computing Systems*, 1361–1370.

J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. 2007. “Cost-effective outbreak detection in networks.” In: *Proc. 13th ACM SIGKDD Int. Conf. on Know. Disc. and Data Mining*, 420–429.

J. Leskovec and A. Krevl. June 2014. *SNAP Datasets: Stanford Large Network Dataset Collection.* http://snap.stanford.edu/data. (June 2014).

J. Leskovec and J. J. Mcauley. 2012. “Learning to discover social circles in ego networks.” In: *Advances in Neural Information Processing Systems*, 539–547.

Y. Li, H. Gao, Y. Gao, J. Guo, and W. Wu. 2023. “A Survey on Influence Maximization: From an ML-Based Combinatorial Optimization.” *ACM Trans. Knowl. Discov. Data*, 17, 9, 133:1–133:50. doi:10.1145/3604559.

M. Lin, L. Sun, R. Yang, X. Liu, Y. Wang, D. Li, W. Li, and S. Lu. 2023. “Fair Influence Maximization in Large-scale Social Networks Based on Attribute-aware Reverse Influence Sampling.” *J. Artif. Intell. Res.*, 76, 925–957. doi:10.1613/JAIR.1.14450.

C. Ling, J. Jiang, J. Wang, M. T. Thai, R. Xue, J. Song, M. Qiu, and L. Zhao. 2023. “Deep Graph Representation Learning and Optimization for Influence Maximization.” In: *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA* (Proceedings of Machine Learning Research). Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. PMLR, 21350–21361. https://proceedings.mlr.press/v202/ling23b.html.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. 1978. “An analysis of approximations for maximizing submodular set functions—I.” *Mathematical programming*, 14, 1, 265–294.

B.-N. T. Nguyen, P. N. H. Pham, V.-V. Le, and V. Snášel. 2022. “Influence Maximization under Fairness Budget Distribution in Online Social Networks.” *Mathematics*, 10, 22. doi:10.3390/math10224185.

G. Nie, M. Agarwal, A. K. Umrawal, V. Aggarwal, and C. J. Quinn. 2022. “An Explore-then-Commit Algorithm for Submodular Maximization Under Full-bandit Feedback.” In: *The 38th Conf. on Uncertainty in Artificial Intelligence.*

E. W. Robson, D. Reddy, and A. K. Umrawal. 2024. “CyNetDiff: A Python Library for Accelerated Implementation of Network Diffusion Models.” *Proc. VLDB Endow.*, 17, 12, 4409–4412. https://www.vldb.org/pvldb/vol17/p4409-umrawal.pdf.

E. W. Robson and A. K. Umrawal. 2025. “A Community-Aware Framework for Influence Maximization with Explicit Accounting for Inter-Community Influence.” *arXiv preprint arXiv:2512.23973.*

B. Rozemberczki and R. Sarkar. 2020. “Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models.” In: *Proc. 29th ACM Int. Conf. on Info. and Know. Mgmt.* ACM, 1325–1334.

T. C. Schelling. 2006. *Micromotives and macrobehavior.* WW Norton & Company.

T. Soma and Y. Yoshida. 2015. “A generalization of submodular cover via the diminishing return property on the integer lattice.” *Advances in Neural Information Processing Systems*, 28, 847–855.

C. Stutsman, E. W. Robson, and A. K. Umrawal. 2026. “A pressure-based diffusion model for influence maximization on social networks.” In: *Proceedings of the International AAAI Conference on Web and Social Media* 1. Vol. 20, 2238–2251.

Y. Tang, Y. Shi, and X. Xiao. 2015. “Influence maximization in near-linear time: A martingale approach.” In: *Proc. 2015 ACM SIGMOD Int. Conf. on Management of Data*, 1539–1554.

Y. Tang, X. Xiao, and Y. Shi. 2014. “Influence maximization: Near-optimal time complexity meets practical efficiency.” In: *Proc. 2014 ACM SIGMOD Int. Conf. on Management of Data*, 75–86.

L. Uieda, S. R. Soler, R. Rampin, H. van Kemenade, M. Turk, D. Shapero, A. Banihirwe, and J. Leeman. 2020. "Pooch: A friend to fetch your data files." *Journal of Open Source Software*, 5, 45, 1943. doi:10.21105/joss.01943.

A. K. Umrawal and V. Aggarwal. 2023. "Leveraging the Community Structure of a Social Network for Maximizing the Spread of Influence." *ACM SIGMETRICS Performance Eval. Rev.*, 50, 4, 17–19.

A. K. Umrawal, V. Aggarwal, and C. J. Quinn. 2023. "Fractional budget allocation for influence maximization." In: *2023 62nd IEEE Conference on Decision and Control (CDC).* IEEE, 4327–4332.

A. K. Umrawal, C. J. Quinn, and V. Aggarwal. 2023. "A community-aware framework for social influence maximization." *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7, 4, 1253–1262.

J. Vondrák. 2008. "Optimal approximation for the submodular welfare problem in the value oracle model." In: *Proc. 40th Annual ACM Symp. on Theory of Computing*, 67–74.

L. A. Wolsey. 1982. "An analysis of the greedy algorithm for the submodular set covering problem." *Combinatorica*, 2, 4, 385–393.

J. Xu and A. K. Umrawal. 2026. "LOFA: Online Influence Maximization under Full-Bandit Feedback using Lazy Forward Selection." *arXiv preprint arXiv:2601.00933.*

J. Yang and J. Leskovec. 2015. "Defining and evaluating network communities based on ground-truth." *Knowledge and Information Systems*, 42, 1, 181–213.

Y. Yang, X. Mao, J. Pei, and X. He. 2016. "Continuous influence maximization: What discounts should we offer to social network users?" In: *Proc. 2016 Int. Conf. on Management of Data*, 727–741.

A Table of Notations

Table 4. Table of notations.

Symbol	Explanation
Ω	Ground set.
2^{Ω}	Set of all subsets of Ω.
$\mathbb{R}$	Set of all real numbers.
$[0,1]^n$	Unit hypercube.
e	Euler's constant ($e \approx 2.718$).
u	Current time during independent cascade.
U	Ending time of independent cascade process.
$\mathbb{E}(X)$	Expectation of a random variable X.
$G = (V, E)$	Directed graph.
$V = (v_1, \dots, v_n)$	Set of vertices or nodes.
n	Size of V.
$E = (e_1, \dots, e_n)$	Set of directed edges where e_i are ordered pairs of nodes.
$p_{v,w}$	Weight of the edge $v \to w$.
$Y_t^{(v)}$	Activation/state of node v at time t.
k	Budget (real-valued).
t	Intermediate budget (real-valued).
$\sigma(S)$	Influence of a set $S \subseteq V$.
$\mathbf{d} = (d_1, \dots, d_n)$	Discount vector.
$\tilde{\sigma}(\mathbf{d})$	Influence of a discount vector $\mathbf{d}$.
$f(S)$	A monotone, non-decreasing, submodular set function with argument $S \subseteq V$.
$F(\mathbf{x})$	Multilinear extension of f with argument $\mathbf{x} \in [0,1]^n$.
$\langle \mathbf{1}, \mathbf{x} \rangle = k$	The budget hyperplane.
$\mathscr{P}$	Polytope formed by the intersection of $[0,1]^n$ and $\langle \mathbf{1}, \mathbf{x} \rangle = k$.

Table 5. Table of notations (Contd.).

$\mathbf{x}^*$	Optimal solution to FIM.
$\lfloor k \rfloor$	Largest integer smaller than k.
(k)	Fractional portion of k, equal to $k - \lfloor k \rfloor$.
$\mathbf{x}_{\lfloor k \rfloor}$	Greedy approximate solution to DIM with integer budget $\lfloor k \rfloor$.
$\mathbf{x}(k)$	Convex combination: $(1-(k))\mathbf{x}(\lfloor k \rfloor) + (k)\,\mathbf{x}(\lfloor k \rfloor + 1)$.
δ	Discretization parameter (0.1) for the LIM-Greedy algorithm.
ϵ	Error parameter (0.5) for the LIM-Greedy algorithm.
ϵ_1	Error parameter (0.01) for the RIS-based DIM-Oracle.
$\tilde{\sigma}_{\text{desired}}$	Desired influence for Budget-Influence.
ϵ_2	Error parameter for Budget-Influence.
$\tilde{k}_2$	Maximum budget for Budget-Influence.
p	Price per unit of the product.
c	Cost of sponsoring one unit of product.
$\pi(t)$	Profit function: $\pi(t) = p\,\tilde{\sigma}(\mathbf{x}(t)) - ct$.
ϵ_3	Error parameter for Budget-Profit.
$\tilde{k}_3$	Maximum budget for Budget-Profit.

Received 12 September 2025; accepted 23 March 2026