2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Fractional Budget Allocation for Influence Maximization

Abhishek K. Umrawal, Vaneet Aggarwal and Christopher J. Quinn

***Abstract*— We consider a generalization of the widely studied discrete influence maximization problem. We consider that instead of marketers using a budget to send free products to a few influencers, they can provide discounts to partly incentivize a larger set of influencers with 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 constraint. We propose and analyze an efficient $(1-1/e)$-approximation algorithm. We run experiments on a real-world social network to show the performance of our method in contrast to methods proposed for other generalizations of influence maximization.**

I. INTRODUCTION

With increasing numbers 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, namely by providing free samples of a product to those influencers so that they will discuss and promote the product on their social media accounts.

The question of which set of influencers to sponsor is known as influence maximization (IM) [1]. Under standard diffusion models, this discrete optimization problem is NP-hard [2]. Since it exhibits a diminishing returns property (the objective function is submodular) a simple greedy algorithm [3] achieves a $(1-1/e)$ approximation [2].

Influence maximization has been widely studied. One important limitation of the associated optimization problem as a model for viral marketing is the binary nature of influencer sponsorships—either an influencer is 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 discuss 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 could have been done with free products alone.

We consider this continuous optimization problem modeling influence maximization, where the marketer can choose to provide discounts instead of only free samples (i.e. few, full discounts). We characterize the problem and show that not only is it no more challenging to approximate than the widely-studied discrete problem, but near-optimal approximations to the continuous problem can be constructed from near-optimal approximations for the discrete problem, for which there are many off-the-shelf methods available.

Furthermore, the method we propose, which uses a subroutine to solve the discrete problem, is more efficient than methods for general classes of continuous submodular optimization problems and, using a value oracle for the discrete problem, does not use random sampling to simulate a value or gradient oracle for the continuous objective.

A. Our Contributions

The main contributions of the paper are

1) We propose and analyze a procedurally simple and efficient $(1-1/e)$ approximation algorithm for the problem of 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.
2) With minimal additional overhead, our method can identify a continuous path of near-optimal solutions up to the specified budget, allowing the marketer to evaluate cost/benefit tradeoffs over a range of budgets.

We empirically demonstrate the performance of our proposed method, in terms of solution quality and efficiency.

B. Literature Review

There is a large literature on maximizing submodular set functions and continuous analogs such as DR-submodular functions. Due to space limitations, we only discuss a few works that are most closely related to ours. Here we simply note that there are works in submodular optimization that solve a discrete problem using a continuous relaxation (we instead solve a continuous problem) and that there are other works that maximize continuous submodular functions using

This material is based upon work supported in part by the National Science Foundation under Grants No. 1742847, 2149588, and 2149617.

Abhishek K. Umrawal is with the Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, IL, 61801, USA. He was with the School of Industrial Engineering, Purdue University, West Lafayette, IN, 47907, USA, when this work was performed. aumrawal@illinois.edu.

Vaneet Aggarwal is with the School of Industrial Engineering and the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907 USA. vaneet@purdue.edu.

Christopher J. Quinn is with the Department of Computer Science, Iowa State University, Ames, IA, 50011, USA. cjquinn@iastate.edu.

oracle access for a related discrete problem, but those works use many oracle queries (random sampling) to construct estimates of the continuous function or its gradient. Our method instead constructs near-optimal solutions to the continuous problem directly from solutions for the discrete problem.

We next discuss works in the influence maximization literature. Kempe et al. [2] proposed using a greedy algorithm from [3] for discrete IM (DIM), where the decision maker chooses a subset of users to seed the diffusion. They also considered a much more general setting 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. They proposed a continuous greedy algorithm for solving this problem.

Yang et al. [4] considered a special case of Kempe et al. [2]'s general marketing problem, known as continuous influence maximization (CIM), where each marketing strategy only directly affects a single user. Yang et al. [4] then proposed a heuristic procedure based on coordinate descent. We focus on an important special case of CIM where the incentives are proportional to the users becoming seeds. (The case we consider still generalizes the widely studied discrete IM problem.) We propose a procedurally simple and computationally efficient algorithm and then prove it yields a $(1-1/e)$ approximation guarantee. We further discuss how the problem we consider relates to CIM in Section III-B.

Chen et al. [5] considered another special case of Kempe et al. [2]'s generalization, known as lattice influence maximization (LIM), where they focused on discretized marketing strategies with some granularity parameter. They proposed a continuous greedy algorithm using reverse influence sampling [6], [7], [8]. It is an adaptation of gradient methods for DR-submodular maximization for influence maximization. Demaine et al. [9] also proposed an influence model considering strategies that affect the peer-to-peer influences (i.e. edge weights in the independent cascade model).

C. Organization

The rest of the paper is organized as follows. In Section II, we review background material. In Section III, we introduce the fractional influence maximization problem we study. In Section IV, we propose a greedy algorithm. In Section V, we analyze the algorithm. In Section VI, we provide experiments. In Section VII, we conclude the paper.

II. Background

In Section II-A, we review submodular optimization. In Section II-B, we review influence maximization.

A. Submodular Optimization

We first review submodular optimization. See [2], [10] for more details. Let Ω denote the ground set of n elements. Let 2^Ω denote the set of all subsets of Ω. A set function $f : 2^\Omega \to \mathbb{R}$ is said to be *submodular* if it satisfies a natural "diminishing returns" property: the marginal gain from adding an element v to a set $S \subseteq \Omega$ is at least as high as the marginal gain from adding the same element v to a superset $S' \subseteq \Omega$ of S. Formally, for any sets $S \subseteq S' \subseteq \Omega$, f satisfies $f(S \cup \{v\}) - f(S) \geq f(S' \cup \{v\}) - f(S')$. A set function f is said to be *monotone* (non-decreasing) if for any $S \subseteq S' \subseteq \Omega$, f satisfies $f(S) \leq f(S')$. Let $[0,1]^n$ denote the unit hypercube. The *multilinear extension* $F : [0,1]^n \to \mathbb{R}$ of the set function $f : 2^\Omega \to \mathbb{R}$ is defined as $F(x) = \sum_{S \subseteq \Omega} f(S) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i)$. At the corners $x \in \{0,1\}^n$ of the hypercube $[0,1]^n$, the multilinear extension $F(x)$ is equal to the set function $f(S)$, with $S = \{i \mid i \in \{1, \ldots, n\},\ x_i = 1\}$. F is the expected value $F(x) = \mathbb{E}[f(\hat{x})]$, where $\hat{x}$ is a random vector where each element i is distributed as Bernoulli(x_i) and is independent of other elements [10]. This relationship is the basis for the widely used strategy of estimating a multilinear extension F by averaging f over randomly sampled subsets.

B. Influence Maximization

Diffusion models describe how information and content spread over a social network. For the purpose of our research, we focus on the widely used *independent cascade* model [11], [12]. Given a graph $G = (V, E)$, the process starts at time 0 with an initial set of active nodes S, called the *seed set*. When a node $v \in S$ first becomes active at time t, it will be given a single chance to activate each currently inactive neighbor w, it succeeds with a probability $p_{v,w}$ (independent of the history thus far). If w has multiple newly activated neighbors, their attempts are sequenced in an arbitrary order. If v succeeds, then w will be active at time $t+1$. Let $y_t^{(v)} \in \{0,1\}$ denote whether v was activated or not by time t.

The *influence* $\sigma(S)$ of a seed set S is the expected number of active nodes at the end of the cascade (denoted by time T), given that the users in S seed the diffusion, $\sigma(S) = \mathbb{E}[\sum_{v \in V} y_T^{(v)} \mid \text{seed set } S]$, where $y_0^{(v)} = 1$ if $v \in S$ and 0 otherwise. Kempe et al. [2] showed that under the independent cascade model, $\sigma(S)$ is a monotone submodular set function. They proposed the discrete influence maximization (DIM) problem under a budget k as

Problem 1 (Discrete Influence Maximization (DIM)):

$$\max_{S \subseteq V:\ |S| \leq k} \sigma(S).$$

III. Problem Formulation

A. Social Influence Maximization Under Partial Incentives

We are interested in solving a generalization of Problem 1 we refer to as fractional influence maximization (FIM) where the decision maker can *partially incentivize* users to seed the diffusion through discounts, in contrast to Problem 1 where users are either fully incentivized to seed the diffusion (if they receive a free product) or are not incentivized at all.

Let $\mathbf{d} \in [0,1]^n$ denote the vector of normalized user discounts, such that user i becomes influenced by the incentive at time 0 with probability d_i. Analogous to the set function $\sigma(S)$, we consider the influence $\tilde{\sigma}(\mathbf{d})$ of a vector of incentives $\mathbf{d} \in [0,1]^n$ as the expected number of active nodes at the end of the cascade (denoted by time T), $\tilde{\sigma}(\mathbf{d}) = \mathbb{E}[\sum_{v \in V} y_T^{(v)} \mid \text{incentives } \mathbf{d}]$.

Lemma 0.1: The expected influence function $\tilde{\sigma}$ for the relaxation Problem 2 of Problem 1 is the multilinear extension of the monotone submodular set function σ in Problem 1, specifically, $\tilde{\sigma}(\mathbf{d}) = \sum_{S \subseteq V} \sigma(S) \prod_{i:v_i \in S} d_i \prod_{i:v_i \notin S} (1 - d_i)$.
The proof is straightforward using properties of expectation.

B. Problem Statement

We now formally state the problem of *fractional influence maximization* (FIM) we consider. FIM generalizes the widely studied DIM by allowing for fractional incentives and we show is much more tractable to solve than more general model classes. Let G be a weighted directed social network and $k \in (0, n)$ the budget.

Problem 2 (Fractional Influence Maximization (FIM)):

$$\max_{\mathbf{d} \in [0,1]^n:\ \langle \mathbf{1}, \mathbf{d} \rangle \leq k} \tilde{\sigma}(\mathbf{d}). \tag{1}$$

FIM is a subclass of the problem of maximizing the multilinear extension of a monotone submodular set function subject to an L_1 constraint. It is also a special case of CIM [4]. CIM permits the probability of node i being a seed to be a non-linear function of the discount d_i, though like in FIM users are fully incentivized with one unit of budget (i.e. discounts $d_i = 0$ and $d_i = 1$ result in $y_0^{(i)} = 0$ and $y_0^{(i)} = 1$ respectively). The objective function for CIM [4] is not a multilinear function and is more challenging to optimize.

IV. Main Results - Approximation Algorithm

In this section, we present a greedy $\left(1 - \frac{1}{e}\right)$-approximation algorithm, *Multilinear Extension Greedy* (MLE-Greedy), for Problem 2. It is procedurally simple. Unlike other methods, our method does not need a value oracle for the multilinear function, a gradient oracle, nor to estimate gradients with random sampling from a value oracle for the corresponding set 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 k, with negligible overhead.

A. Multilinear Extension Greedy—Approximation Algorithm

Algorithm 1 calls a subroutine Oracle-DIM to obtain a sequence of nested solutions with increasing cardinality. We will discuss Oracle-DIM in Section IV-B. For a budget $k \in \mathbb{R}_+$, denote the floor as $\lfloor k \rfloor$, so $k - \lfloor k \rfloor$ is the factional part. Let $x(\lfloor k \rfloor)$ and $x(\lfloor k \rfloor + 1)$ denote the approximate solutions to the discrete problem Problem 1 for integer budgets $\lfloor k \rfloor$ and $\lfloor k \rfloor + 1$ respectively, corresponding to nested subsets, with $x(\lfloor k \rfloor) \leq x(\lfloor k \rfloor + 1)$ (element-wise) so they differ in a single coordinate.

We construct the solution to Problem 2 as

$$x(k) := x(\lfloor k \rfloor) + (k - \lfloor k \rfloor)(x(\lfloor k \rfloor + 1) - x(\lfloor k \rfloor)). \tag{2}$$

Despite this method's simplicity, both in computation and in selecting a vector on the boundary of $[0, 1]^n$, we will show that (2) is nonetheless near-optimal.

Algorithm 1 MLE-Greedy

1: **Input** budget k, access to Oracle-DIM
2: $\{\ldots, x(\lfloor k \rfloor), x(\lfloor k \rfloor + 1)\} \leftarrow$ Oracle-DIM $(\lfloor k \rfloor + 1)$.
3: **Return** $x(\lfloor k \rfloor) + (k - \lfloor k \rfloor)(x(\lfloor k \rfloor + 1) - x(\lfloor k \rfloor))$

Algorithm 2 Oracle-DIM (version [3])

1: **Input** integer budget k', access to value oracle for σ
2: $S_0 \leftarrow \emptyset$
3: **for** $i \in \{1, \ldots, k'\}$ **do**
4: $\quad e_i \leftarrow \arg\max_{e \in V \setminus S_{i-1}} \sigma(S_{i-1} \cup \{e\}) - \sigma(S_{i-1})$
5: $\quad S_i \leftarrow S_{i-1} \cup \{e_i\}$
6: **end for**
7: **Return** $\{S_1, \ldots, S_{k'}\}$

B. DIM Subroutine

We now discuss the subroutine Oracle-DIM. In [3], the authors proposed a well-known greedy algorithm for maximizing a monotone submodular function (of which Problem 1 is a special case) that returns such a nested sequence with an approximation ratio $(1 - 1/e)$ in $O(nk)$ time. The pseudo-code is shown as Algorithm 2. Lazy evaluations can improve the empirical run-time of Algorithm 2, though its worst case complexity is still $O(nk)$. For the application of social influence maximization, more efficient greedy methods have been proposed, such as CELF [13], CELF++ [14], Community-IM [15], [16], and reverse influence sampling (RIS) based methods [6], [7], [8], [17]. Any of these or other $(1 - 1/e)$-approximation algorithms could be used as Oracle-DIM, provided they return a nested sequence of solutions (each subset an $(1 - 1/e)$ approximation for the respective cardinality).

Also, faster algorithms with weaker guarantees, such as the linear time randomized greedy method proposed in [18], could be used as subroutines and their weaker approximation guarantees would carry over to our proposed MLE-Greedy.

C. MLE-Greedy Performance

We next state properties of our proposed procedure Algorithm 1, namely that it efficiently provides a near-optimal solution to Problem 2. Since Algorithm 1 is applicable not just for the FIM problem, but more generally for optimizing a multilinear extension $F(x)$ (of a monotone submodular set function $f(S)$), in the following we will use the more generic notation "$F(x)$" instead of the FIM specific "$\tilde{\sigma}(\mathbf{d})$." Let $x^*(k)$ denote an optimal solution to Problem 2.

Theorem 1: Algorithm 1 is an $\left(1 - \frac{1}{e}\right)$-factor approximation algorithm for Problem 2. Equivalently, (2) satisfies $F(x(k)) \geq \left(1 - \frac{1}{e}\right) F(x^*(k))$.
Theorem 1 will follow immediately from Theorem 2 below.

The computational complexity of Algorithm 1 is dominated by calling Oracle-DIM which has computational complexity $O(nk)$. We next discuss how a slight extension allows us to recover a continuous path of near-optimal solutions for any budget, allowing the user to make a cost-benefit analysis for how much budget to use.

D. Continuous Near-Optimal Path Construction

We can extend (2) to a continuous path for any budget $t \in [0, n]$ such that each point along the path is near optimal for its respective L_1 norm. We begin by fixing $x(t)$ for integer values of t, $t \in \{0, 1, \ldots, n\}$. Let $\{S_t\}_{t=0}^n$ denote the sequence of nested subsets produced by Algorithm 2 for budget n, each subset being an $(1 - 1/e)$-approximate solution for its respective cardinality. For integer values of t, set $x(t)$ to match S_t ($x_i(t) = 1$ if $i \in S_t$). For non-integer values, set $x(t)$ as

$$x(t) = x(\lfloor t \rfloor) + (t - \lfloor t \rfloor)(x(\lfloor t \rfloor + 1) - x(\lfloor t \rfloor)). \quad (3)$$

By construction, the coordinates of $x(\lfloor t \rfloor)$ are binary valued and differ with those of $x(\lfloor t \rfloor + 1)$ in a single coordinate, namely the element j that gets added to $S_{\lfloor t \rfloor}$ to form $S_{\lfloor t \rfloor + 1}$. Let $x^*(t)$ denote an optimal solution for a budget t.

Theorem 2: For all $t \in [0, n]$, the continuous path $x(t)$ is feasible for Problem 2 for budget t and near optimal, with $F(x(t)) \geq \left(1 - \frac{1}{e}\right) F(x^*(t))$.

We defer the proof to Section V-B. With effectively *the same computational expense* of identifying a near-optimal solution for the (maximum) budget, and armed only with access to a value oracle for a set function, the decision maker can identify near-optimal solutions *for any budget* up to the maximum. This gives the decision-maker greater flexibility.

Remark 1: The continuous path we construct is on the hyper-cube boundary and efficiently constructed using solutions (subsets) to the discrete problem DIM. There are several other algorithms that optimize continuous analogs of submodular set functions, such as [10], [19], [20], [5], which construct a "continuous greedy" path *inside* the unit hypercube in iterative steps beginning at $x = 0$. Those procedures typically require discretization or a step size parameter. They also require a value oracle for the continuous function or its gradients, else they use random samples from a value oracle for the set function to estimate those.

E. Problem 2 Characterization

We next identify a property of Problem 2 that motivates the design of our proposed Algorithm 1 and path construction.

Theorem 3: For Problem 2 with any budget $0 < t < n$, there is an optimal solution $x^*(t)$ that is the convex combination of just two, nested, and not necessarily optimal, solutions to Problem 1 for integer budgets, one for budget $\lfloor t \rfloor$ and one for budget $\lfloor t \rfloor + 1$.

The proof of Theorem 3 will follow directly from Lemma 3.2, which we defer to Section V-A.

We next explore discontinuity of $x^*(t)$ in an experiment.

F. Synthetic experiment to show discontinuity of $x^*(t)$

We ran influence maximization experiments using the independent cascade model on the network shown in Figure 1.

Experiment Details. We exhaustively evaluated $F(x)$ for all $x \in \{0, 1\}^n$ with $\langle \mathbf{1}, x \rangle \leq 3$. Each evaluation entailed running 1000 Monte-Carlo simulations and averaging.

Results and Discussion. We found that the optimal solutions to Problem 1 for budgets $k \in \{1, 2, 3\}$ were not fully nested. Namely, with index relabeling for simplicity, the optimal set for budget $k = 1$ was $S_1 = \{1\}$, for $k = 2$ was $S_2 = \{2, 3\}$, and for $k = 3$ was $S_3 = \{1, 2, 3\}$. We then searched over pairs of nested solutions with integer-valued L_1 norms where one was optimal (for a budget of value equal to its L_1 norm). With this, we identified a discontinuous path that is near-optimal, though we did not exhaustively check if that path was indeed optimal. The values $F(x)$ for a sequence of solutions we identified are plotted in Figure 2. The blue segments (for $k \in [0, 1] \cup [2, 3]$) are optimal values. Over $k \in [2, 3]$, the set of optimal solutions is discontinuous in k and the optimal value function is piece-wise linear with at least two parts (max over orange and green segments).

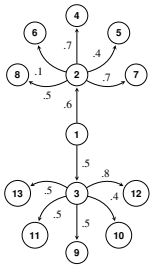


Fig. 1: Network.

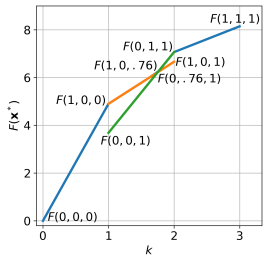


Fig. 2: $F(x^*(k))$, $k \in [0, 3]$.

V. Main Results - Analysis

We next characterize properties of (near)-optimal solutions to Problem 2 which will then be used to prove Theorem 1.

We first state important properties of multilinear functions. We then use those properties to show that there is an optimal solution that is an extreme point (i.e. a corner) of the polytope formed by the intersection of the hyperplane $\langle \mathbf{1}, x \rangle = k$ and the hypercube $[0, 1]^n$. From there we bound the value of the optimal solution for any non-integer budget k using the values of the optimal solutions for budgets $\lfloor k \rfloor$ and $\lfloor k \rfloor + 1$, which are also optimal solutions to the discrete Problem 1.

A. Properties of Multilinear Extensions

Let $\boldsymbol{e}_i$ denote the basis vector for the dimension i. For a vector $\mathbf{d}$, the inequality $\mathbf{d} \geq 0$ denotes element-wise non-negativity. Let $\mathcal{P}_k = \{x \mid \mathbf{0} \leq x \leq \mathbf{1},\ \langle \mathbf{1}, x \rangle = k\}$ denote the polytope formed by the intersection of the hypercube $[0, 1]^n$ and the hyperplane $\langle \mathbf{1}, x \rangle = k$.

Lemma 3.1 ([10], [21]): 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 $\mathbf{d} \geq 0$,
- F is **concave** along any line of direction $\mathbf{d} \geq 0$, and
- F is **convex** along any line of direction $\boldsymbol{e}_i - \boldsymbol{e}_j$.

Corollary 3.1: There is an optimal solution x^* to Problem 2 satisfying $\langle \mathbf{1}, x^* \rangle = k$.

Proof: This follows directly from Lemma 3.1. ■

By Corollary 3.1, we can restrict our attention to $\mathcal{P}_k$. By the third property of Lemma 3.1, we will be able to further restrict our attention to the extreme points of $\mathcal{P}_k$, namely those $x \in \mathcal{P}_k$ with at most one integer-valued coordinate.

Lemma 3.2: For Problem 2, there is an optimal solution x^* with at most one coordinate that is not integer-valued.

Proof: By Corollary 3.1, there is an optimal solution $x^* \in \mathcal{P}_k$. Without loss of generality, suppose x^* has non-integer coordinates for dimensions 1 and 2. Let $\tilde{x}(t) = x^* + t(\mathbf{e}_1 - \mathbf{e}_2)$, $t_{\min} = -\min\{x_1, 1 - x_2\}$, and $t_{\max} = \min\{1 - x_1, x_2\}$. By construction, $\tilde{x}(t) \in \mathcal{P}_k$ for $t \in [t_{\min}, t_{\max}]$ (for other t it is outside the unit hypercube) and both $\tilde{x}(t_{\min})$ and $\tilde{x}(t_{\max})$ have one more integer-valued coordinate than x^* does. $F(\tilde{x}(t))$ is convex in t by Lemma 3.1, so $F(x^*) \leq \max\{F(\tilde{x}(t_{\min})),\ F(\tilde{x}(t_{\max}))\}$. Replace x^* with the $\arg\max$ and repeat. ∎

For the special case of an integer budget k, this further means that there is an optimal solution to FIM with integer-valued coordinates, which necessarily is optimal for DIM.

Corollary 3.2: For an integer-valued k, there is an optimal solution to Problem 1 that is also optimal for Problem 2.

Proof: This follows from Lemma 3.2. ∎

As noted in Section IV-E, Theorem 3 follows from Lemma 3.2. That result characterizing solutions (motivating the algorithm design) leads to the following result on function values (which leads to the near-optimality guarantees).

Corollary 3.3: The optimal value $F(x^*(t))$ for Problem 2 over $[0, n]$ is a piece-wise linear, continuous function of the budget t. Between successive integer values, $t' \in [\lfloor t \rfloor, \lfloor t \rfloor + 1]$, $F(x^*(t'))$ is convex.

Proof: Let $\mathcal{E}_t$ denote the subset of points in $\mathcal{P}_t$ that have at most one non-integer valued coordinate (of value $t - \lfloor t \rfloor$). For non-integer t, these are extreme points of $\mathcal{P}_t$. By Theorem 3, for a budget t there is an optimal solution $x^* \in \mathcal{E}_t$ to Problem 2 that can be expressed as a convex combination $x^* = x' + \theta(x'' - x')$ of nested solutions ($x' \leq x''$) to Problem 1, where $\theta = t - \lfloor t \rfloor$.

Let $\widetilde{\mathcal{E}}_{\lfloor t \rfloor}$ denote the set of nested pairs of solutions to Problem 1 for budgets $\lfloor t \rfloor$ and $\lfloor t \rfloor + 1$, with the first element x' an extreme point $x' \in \mathcal{E}_{\lfloor t \rfloor}$ of the polytope $\mathcal{P}_{\lfloor t \rfloor}$ for integer budget $\lfloor t \rfloor$ and the second an extreme point for budget $\lfloor t \rfloor + 1$, with just one coordinate differing in value,

$$\widetilde{\mathcal{E}}_{\lfloor t \rfloor} = \{(x', x'') \mid x' \in \mathcal{E}_{\lfloor t \rfloor},\ x'' \in \mathcal{E}_{\lfloor t \rfloor + 1},\ x' \leq x''\}. \quad (4)$$

For each pair $(x', x'') \in \widetilde{\mathcal{E}}_{\lfloor t \rfloor}$, we can consider the function $\psi_{(x', x'')}(\theta) = F(x' + \theta(x'' - x'))$. F is linear along any unit direction (i.e. all but one coordinate fixed), so ψ is a linear function, $\psi_{(x', x'')}(\theta) = F(x') + (F(x'') - F(x'))\theta$.

Thus, we can express the optimal value as

$$F(x^*) = \max_{(x', x'') \in \widetilde{\mathcal{E}}_{\lfloor t \rfloor}} \psi_{(x', x'')}(\theta). \quad (5)$$

Since (5) is a maximization over the same finite set of linear functions for any $\theta \in [0, 1]$ (with $\lfloor t \rfloor$ fixed), the optimal objective value is a piece-wise linear, continuous, and convex function of θ. Each pair $(x', x'') \in \widetilde{\mathcal{E}}_{\lfloor t \rfloor}$ that achieves the maximum value $\psi_{(x', x'')}(\theta)$ for some θ does so for either a single point or a single interval of $[0, 1]$. We can form (possibly discontinuous) piece-wise linear paths $x(\theta)$ and we can restrict ourselves to (possibly discontinuous) piece-wise linear paths with the minimal number of segments. ∎

B. Proof of Theorem 2

Proof: Continuity and feasibility of the path (in particular $x(t) \in \mathcal{P}_t$) follow by construction. We now show near-optimality. Writing $t' = t - \lfloor t \rfloor$ and x^*_t for $x^*(t)$,

$$\begin{aligned} F(x(t)) &= f(S_{\lfloor t \rfloor}) + t'(f(S_{\lfloor t \rfloor + 1}) - f(S_{\lfloor t \rfloor})) && (6)\\ &\geq (1 - \frac{1}{e})\big(f(S^*_{\lfloor t \rfloor}) + t'(f(S^*_{\lfloor t \rfloor + 1}) - f(S^*_{\lfloor t \rfloor}))\big) && (7)\\ &= (1 - \frac{1}{e})\big(F(x^*_{\lfloor t \rfloor}) + t'(F(x^*_{\lfloor t \rfloor + 1}) - F(x^*_{\lfloor t \rfloor}))\big) && (8)\\ &\geq (1 - \frac{1}{e})F(x^*(t)), && (9) \end{aligned}$$

where (6) uses linearity of multilinear extensions along unit directions and construction of $x(t)$, (7) uses near-optimality of the output of Algorithm 2 for the discrete problem, Problem 1 [3]; (8) follows by construction; (9) follows from the piecewise-linearity of $F(x^*(t))$—the value in (8) is a convex combination of F for integer valued budgets that upper bounds $F(x(t))$ and equality is achieved only if there are a pair of nested optimal solutions to Problem 1 for budgets $\lfloor t \rfloor$ and $\lfloor t \rfloor + 1$ (i.e. $S^*_{\lfloor t \rfloor} \subset S^*_{\lfloor t \rfloor + 1}$ and consequently (element-wise) $x^*(\lfloor t \rfloor) \leq x^*(\lfloor t \rfloor + 1)$).

∎

VI. Experiments

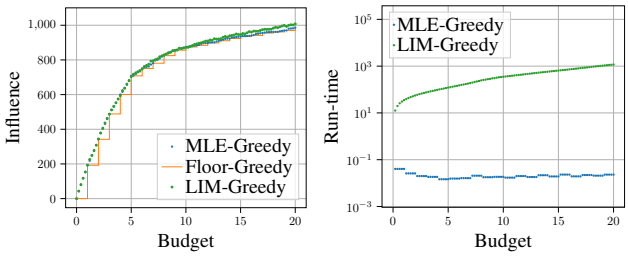
We ran experiments to evaluate our greedy approximation algorithm on a large real-world social network.

Network Data. We used an undirected Facebook [22] network with 4,039 nodes and 88,234 edges from [23]. Each edge in this undirected network was replaced by two directed edges. For edge weights, we used the *weighted cascade model* [2] with weights set as one over the in-degrees.

Algorithms. We instantiated Oracle-DIM with a state-of-the-art IM specific greedy algorithm [17] based on reverse influence sampling (RIS). We use the following baselines.

1) Floor-Greedy – for any budget k, calculate Oracle-DIM($F, \lfloor k \rfloor$), i.e. ignore the fractional budget.
2) LIM-Greedy [5] – a state-of-the-art, RIS-based adaptation of gradient methods for DR-submodular maximization for the problem of IM.

Experimental Details. For LIM-Greedy [5], we set the discretization parameter, $\delta = 0.1$ and approximation parameter, $\epsilon = 0.5$ as suggested by the authors. For state-of-the-art RIS-based greedy, we set the approximation parameter, $\epsilon = 0.01$ as suggested by the authors. For our experiments, we compared our proposed Algorithm 1 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 $k \in \{0.2, 0.4, \ldots, 20\}$. After the methods finished, to compare their solutions, we separately evaluated the outputted solutions using 1000 Monte Carlo simulations each. The experiments were carried out on a computer with a 2.3 GHz 2-core Intel Core i5 processor and 8 GB of memory. We used Python for our implementation. The source codes of RIS-based greedy algorithm [17] and LIM-Greedy [5] provided by their authors are written in C++.



(a) Influence. (b) Run-time.

Fig. 3: Results for the Facebook network.

Results. Figure 3a shows the influence (on the y-axis) achieved by different algorithms as a function of the budget (on the x-axis). For each algorithm, the approximate solutions are obtained using a single run for budget 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. The influence values in the plots are the expected numbers calculated using 1000 Monte Carlo simulations. Figure 3b depicts the run-time in seconds (on the y-axis) using different algorithms for different values of the budget (on the x-axis). We use a log scale for plotting as actual run-time values for MLE-Greedy are much smaller (fractions of a second) than those for LIM-Greedy. The run times for MLE-Greedy are averaged over 10 runs.

Discussion. From Figure 3a, we note that in terms of influence, the proposed MLE-Greedy algorithm outperforms the Floor-Greedy algorithm for non-integer budgets. The influence values are almost the same for MLE-Greedy and LIM-Greedy algorithms except for high values of the budget ($k > 15$), where LIM-Greedy performs slightly better. From Figure 3b, we note that our proposed MLE-Greedy takes orders of magnitude less time than LIM-Greedy. The run-time savings due to MLE-Greedy compared to LIM-Greedy increase as the budget increases.

VII. Conclusion

For an important fractional generalization of the widely studied discrete influence maximization problem, we proposed an efficient $(1 - 1/e)$-approximation algorithm. Our algorithm interpolates the solutions to the discrete problem and is significantly more tractable than other generalizations that adopt a continuous greedy strategy. Furthermore, we demonstrated the performance of our algorithm using experimental evaluations.

References

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

[2] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. Ninth ACM SIGKDD Int. Conf. on Know. Disc. and Data Mining*. ACM, 2003, pp. 137–146.

[3] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

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

[5] W. Chen, R. Wu, and Z. Yu, "Scalable lattice influence maximization," *IEEE Trans. Comp. Soc. Sys.*, vol. 7, no. 4, pp. 956–970, 2020.

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

[7] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proc. 2014 ACM SIGMOD Int. Conf. on Management of Data*, 2014, pp. 75–86.

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

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

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

[11] J. Goldenberg, B. Libai, and E. Muller, "Talk network: A complex systems look at the underlying process of word-of-mouth," *Marketing letters*, vol. 12, no. 3, pp. 211–223, 2001.

[12] ——, "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*, vol. 9, no. 3, pp. 1–18, 2001.

[13] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. Thirteenth ACM SIGKDD Int. Conf. on Know. Disc. and Data Mining*, 2007, pp. 420–429.

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

[15] A. K. Umrawal and V. Aggarwal, "Leveraging the community structure of a social network for maximizing the spread of influence," *ACM SIGMETRICS Performance Eval. Rev.*, vol. 50, no. 4, pp. 17–19, 2023.

[16] A. K. Umrawal, C. J. Quinn, and V. Aggarwal, "A community-aware framework for social influence maximization," *IEEE Trans. on Emerging Topics in Computational Intelligence*, 2023.

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

[18] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy," in *Proc. AAAI Conf. on Artificial Intelligence*, vol. 29, no. 1, 2015.

[19] G. Calinescu, C. Chekuri, M. Pal, and J. Vondrák, "Maximizing a monotone submodular function subject to a matroid constraint," *SIAM J. on Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.

[20] C. Chekuri, J. Vondrák, and R. Zenklusen, "Submodular function maximization via the multilinear relaxation and contention resolution schemes," *SIAM J. on Computing*, vol. 43, no. 6, pp. 1831–1879, 2014.

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

[22] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 539–547.

[23] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, June 2014.