Multi-Objective Influence Maximization

Shay Gershtein
Tel Aviv University
shayg1@mail.tau.ac.il

Tova Milo
Tel Aviv University
milo@post.tau.ac.il

Brit Youngmann
Tel Aviv University
brity@mail.tau.ac.il

ABSTRACT

Influence Maximization (IM) is the problem of finding a set of influential users in a social network, so that their aggregated influence is maximized. The classic IM problem focuses on the single objective of maximizing *the overall number of influenced users*. While this serves the goal of reaching a large audience, users often have multiple specific sub-populations they would like to reach within a single campaign, and consequently multiple influence maximization objectives. As we show, maximizing the influence over one group may come at the cost of significantly reducing the influence over the others. To address this, we propose IM-Balanced, a system that allows users to explicitly declare the desired balance between the objectives. IM-Balanced employs a refined notion of the classic IM problem, called Multi-Objective IM, where all objectives except one are turned into constraints, and the remaining objective is optimized subject to these constraints. We prove Multi-Objective IM to be harder to approximate than the original IM problem, and correspondingly provide two complementary approximation algorithms, each suiting a different prioritization pertaining to the inherent trade-off between the objectives. In our experiments we compare our solutions both to existing IM algorithms as well as to alternative approaches, demonstrating the advantages of our algorithms.

1 INTRODUCTION

Social networks attracting millions of people, such as Twitter and LinkedIn, have emerged recently as a prominent marketing medium. *Influence Maximization* (IM) is the problem of finding a set of influential network users (termed a *seed-set*), so that their aggregated influence is maximized [23]. IM has a natural application in viral marketing, where companies promote their brands through the word-of-mouth propagation. This has motivated extensive research [7, 26], emphasizing the development of scalable algorithms [20, 33].

The classic IM problem focuses on the single objective of maximizing *the overall number of influenced users*, given a bound on the seed-set size. While this serves the goal of reaching a large audience, IM algorithms may obliviously focus on certain well-connected populations, at the expense of other demographics of interest. Indeed, marketing campaigns often have multiple objectives, and consequently multiple subpopulations they would like to reach within a single campaign. In this paper we refer to the subpopulations of interest as *emphasized groups*, and assume the existence of boolean functions over user profile attributes, which identify these groups. We introduce the Multi-Objective IM problem, which refines the IM problem, handling multiple emphasized groups.

Ideally, one would like to find a seed-set which simultaneously maximizes the influence over all emphasized groups. However, as we demonstrate, maximizing influence over one group may come at the cost of significantly reducing the influence over another group. Hence, we devise a framework enabling users to explicitly specify the desired trade-off. Concretely, our system, called IM-Balanced, allows the user to prioritize the objectives and declare what portion of the influence over specific groups she is willing to compromise, in order to increase influence over the others.

For simplicity of presentation, we initially focus on the case where the user has two (possibly overlapping) emphasized groups, denoted as g_1 and g_2, and she is willing to compromise a certain percentage of the maximal possible influence over one group for an influence increase over the other. We then extend our discussion to multiple groups, and shortly discuss alternative problem definitions.

We illustrate the problem that we study in this paper via the following two examples.

Example 1.1. Consider a government office aiming to spread a message regarding a new vaccination policy, across a social network. The main goal is to reach the largest possible number of users, but at the same time, it is also desirable to maximize the number of reached anti-vaccination users. Here g_1 consists of all users, and g_2 is the group of anti-vaccination users. A standard IM algorithm will maximize the overall influence (g_1), possibly at the expense of not reaching sufficient g_2 members. A partial solution can be found in targeted IM algorithms (e.g., [9]), which maximize the influence over a particular group (here - g_2). But if this (possibly small) group is somewhat socially isolated, the message may not reach a sufficient number of users overall.

Example 1.2. Consider a tech company running a recruitment campaign over a social network, with the goal of hiring both engineers (g_1) and researchers (g_2). Assume that there are far more engineers than researchers, and that the two groups are not strongly connected socially (though some users may belong to both groups). A targeted IM algorithm focusing, e.g., on users belonging to the union of the groups, may fail to reach a sufficiently large fraction of the researchers. On the other hand, a targeted IM focusing on the researchers may result in too few engineers being reached.

In both examples, there is a trade-off between the influence over two groups of interest. One simple solution is to split the budget (i.e., seed-set size) and run two separate (single-objective) targeted IM algorithms. However, it is not clear how to split the seed-set to obtain the desired balance between the objectives. An alternative approach to tackle multi-objective optimization problems is the weighted-sum approach, where the objectives are combined into a single objective. In the IM setting this involves assigning each user a weight depending on the groups(s) to which she belongs (e.g. [26, 31]). A main difficulty in applying this approach is assigning the weights that achieve a desired influence balance [21]. Indeed, as we demonstrate in our experiments, the exploration for the optimal weights results in poor runtime performance.

Another more direct approach to multi-objective optimization problems is the constraints method [12], where all objectives except one are transformed into constraints, and the remaining objective is optimized subject to these constraints. Our work employs this approach for IM. Concretely, in IM-Balanced users can define the emphasized groups, and specify for each group the fraction of its optimal influence that they are willing to compromise in order to increase influence over other groups. An easily operated UI allows users to view the maximal possible influence for each group (and what influence it entails over other groups), specify the constraints, and view the corresponding derived influence.

Continuing with Example 1.1, if the UI indicates that the overall number of users that can be influenced is rather high, one may be willing to sacrifice a certain amount in order to increase the influence over anti-vaccination users. In Example 1.2, assuming that the company is interested in recruiting a small number of researchers and a larger number of engineers, one can set a constraint on the minimal number of researchers to be informed, and maximize the influence over engineers under this constraint.

Next, we provide a brief overview of our contributions.

Multi-Objective IM. To allow users to balance the objectives we formalize the Multi-Objective IM problem, which extends the IM problem as follows. Given two emphasized groups g_1 and g_2 and a threshold $0 \leq t \leq 1$, we add a requirement that the solution must exceed a t-fraction of the optimal influence over g_2. Then, subject to this constraint, we maximize the influence over g_1. For $t = 0$ one gets a single-objective targeted IM problem solely over g_1 users, whereas for $t = 1$ one gets a single-objective targeted IM solely over g_2 users (Section 3).

Approximation lower bound. We prove that, like IM, Multi-Objective IM is NP-hard. We show that when the constraint threshold t is $> (1 - \frac{1}{e})$, then no seed-set satisfying the constraint can be found in PTIME. Moreover, we prove that the $(1 - \frac{1}{e})$-approximation factor for g_1, which is optimal in the (unconstrained) IM problem, is unattainable in our setting. We show however that it can nevertheless be achieved if the constraint imposed on g_2 is also approximated by a $(1 - \frac{1}{e})$ factor. This bound exposes the trade-off between the approximation factor for the g_1 users and the relaxation of the constraint imposed on the g_2 users. We therefore provide two approximation algorithms, each suiting a different prioritization pertaining to this trade-off.

The MOIM algorithm. Our first algorithm is simple yet highly efficient. It follows the budget splitting approach mentioned above, but rather than requiring the user to specify the partition, it derives it by itself. MOIM runs two single-objective targeted IM algorithms, each focusing on a different group, and combines their outputs. It guarantees that the constraint is fully satisfied, while providing a $(1 - \frac{1}{e \cdot (1-t)})$-approximation for the g_1 users, which equals $1 - \frac{1}{e}$ for $t = 0$, but decreases as t increases. A key advantage of MOIM is its modularity: MOIM maintains the properties of its input IM algorithm, carrying over all of its optimizations, and therefore it achieves near linear time performance. Such good performance is critical for scaling successfully to massive networks (Section 4).

The RMOIM algorithm. To get a tighter approximation ratio one needs to compromise on (i) how strictly the constraint is maintained, and (ii) performance. The RMOIM algorithm relaxes the constraint, allowing its approximation by a $(1 - \frac{1}{e})$ factor, achieving in return near optimal approximation ratio for the influence over g_1. RMOIM extends a Linear program (LP) for Maximum Coverage [38], and thus its performance becomes polynomial (but still practical for real-life social networks including tens of thousands users, as our experiments indicate). One point to note is that building the LP assumes knowledge of the optimal influence over the constrained g_2 group. As this value is incomputable in PTIME, we approximate it, and provide worst case guarantees for this as well.

Implementation and Experimental study. We have implemented our algorithms as part of the IM-Balanced system and experimentally compare our algorithms to (targeted) IM algorithm and alternative approaches. We show that while the weighted-sum approach, *when assigned optimal weights*, is able to achieve results of quality close to ours, our algorithms are significantly more efficient. In terms of runtime performance, we show that the quality advantage comes with a reasonable performance cost for MOIM, which scales well for massive networks. For RMOIM the decrease in scalability turns out to be moderate, proving it practical for non-massive networks, while often exceeding worst-case guarantees to satisfy the constraint (Section 6).

A demonstration of IM-Balanced's usability and its suitability to end-to-end employment was presented in [16]. The short paper accompanying the demonstration provides only a brief, high-level description of the system, whereas the present paper provides the theoretical foundations and algorithms underlying the demonstrated system, as well as the experimental study.

For space constraints, all proofs are deferred to our technical report [3].

2 PRELIMINARIES

This section presents the standard IM problem, and introduces the auxiliary problem of *Group-Oriented IM*. Our multi-objective variant of the IM problem is then presented in the next section.

2.1 Influence Maximization

We model a social network as a weighted graph $G=(V, E, W)$, where V is the set of nodes and every edge $(u, v) \in E$ is associated with a weight $W(u, v) \in [0, 1]$, which models the probability that u will influence v. Given a function $I(\cdot)$ dictating how influence is propagated in the network, the IM problem [23] is defined as follows.

Definition 2.1 (IM [23]). Given a weighted directed graph G and a natural number $k \leq |V|$, find a set O that satisfies: $O = argmax_{\{T: T \subseteq V, |T|=k\}} I(T)$, where $I(T)$ is the expected number of nodes influenced by the seed set T.

Naturally, every node v in a seed set T is influenced by itself, and hence, by definition, T is influenced by T with probability 1. In what follows, we refer to influenced nodes as *covered*.

The function $I(\cdot)$ is defined by the influence propagation model. The majority of existing IM algorithms apply for the two most researched models [7, 20], the Independent Cascade (IC) and the Linear Threshold (LT) models. Both models define the function $I(\cdot)$ as non-negative, submodular and monotonically rising. Our results hold under both models. For simplicity of presentation, in our numeric examples throughout the paper we focus on the LT model.

In the LT model, each node v chooses a threshold $\theta_v \in [0, 1]$ uniformly at random, which represents the weighted fraction of v's neighbors that must become covered in order for v to become covered. Given a random choice of thresholds and an initial set of seed nodes, the diffusion process unfolds deterministically in discrete steps: in step t, all nodes that were covered in step

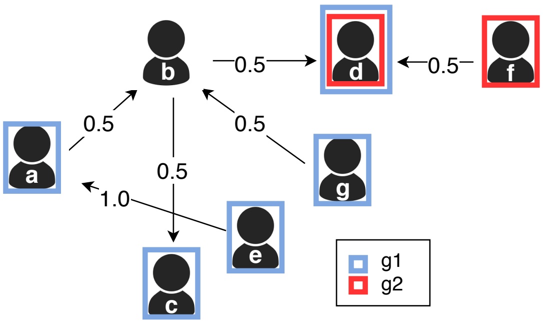


Figure 1: Example social network with two emphasized groups.

$t-1$ remain covered, and we cover any node v for which the total weight of its covered neighbors is at least θ_v. To illustrate, consider the example network presented in Figure 1, ignoring for now the users' border colors. For k=2, the optimal 2-size solution is O=$\{e, g\}$, where $I(O)$=5. Throughout the paper, the threshold for each node was sampled uniformly at random from $[0, 1]$.

Existing IM algorithms. Selecting the optimal seed set is NP-hard, and hard to approximate beyond a factor of $(1-\frac{1}{e})$ [23].The subsequent work on IM following [23], which had already achieved the optimal approximation, has focused on scalability [7, 13, 29]. In what follows, whenever we refer to an IM algorithm, we in fact refer to a probabilistic algorithm, which, given the parameters $0\leq\epsilon, \delta\leq 1$, achieves, with probability $\geq(1-\delta)$, the optimal approximation factor up to an additive error of ϵ. To ease the presentation, we omit the discussion of ϵ and δ whenever possible.

State-of-the-art IM algorithms are based on the Reverse Influence Sampling (RIS) framework, achieving near optimal time complexity [33] of $\tilde{\Theta}(k\cdot(|V|+|E|))$. The RIS framework utilizes sampling over the transpose graph, to reduce the problem to an instance of the *Maximum Coverage* (MC) problem [38]. For completeness of this paper, we formally define this problem.

Definition 2.2 (MC [38]). Given subsets $S_1, ..., S_m$ of elements from $U = \{u_1, ..., u_n\}$ and a natural number $k \leq m$, the goal is to find k subsets from $S_1, ..., S_m$ so as to maximize the number of covered elements in their union.

The well-known MC problem has a simple greedy approximation procedure [38], achieving an optimal approximation factor of $(1-\frac{1}{e})$. The RIS framework consists of two steps: First, θ nodes are sampled uniformly, then, for each sampled node u, a backward influence propagation is simulated from it, with all nodes covered in a simulation constituting a *Reverse Reachability* (*RR*) set. This *RR* set plays the role of possible influence sources for u. Next, each node is associated with the set of *RR* sets containing it, then, using a greedy algorithm, k nodes are selected with the goal of maximizing the number of covered *RR* sets. The observation underpinning this approach is that influential nodes will appear more frequently in *RR* sets, and that the share of *RR* sets covered by a seed set implies an unbiased estimator for its influence.

Example 2.3. Let k=2, θ=4 and four random *RR* sets G_{d_1}=$\{b, d, f\}$, G_e=$\{e\}$, G_{d_2}=$\{d, f\}$ and G_b=$\{a, b, e\}$ are generated from the graph depicted in Figure 1 (d was sampled twice). The corresponding MC instance is: S_b=$\{G_{d_1}, G_b\}$, S_d=$\{G_{d_1}, G_{d_2}\}$, S_f=$\{G_{d_1}, G_{d_2}\}$, S_e=$\{G_b, G_e\}$, S_a=$\{G_b\}$. W.h.p. the sets S_e, S_f will be selected by the greedy algorithm for MC, as they cover all *RR* sets, and hence the nodes e, f will be selected as the seed nodes.

Most recent works focused on optimizing this approach by minimizing the number of sampled *RR* sets [20, 28, 34].

An important observation is that the second step of RIS can also be achieved using Linear Programming (LP), yielding the same guarantees. However, in terms of time complexity, IM algorithms are nearly linear, compared to PTIME LP solvers [22].

2.2 Group-Oriented IM

In our setting users are associated with profile properties such as their profession or political opinion. Characterized by these properties, the end-user provides her *emphasized groups*, i.e., groups which she wishes to ensure are sufficiently covered. An emphasized group may be defined using a boolean query over (multiple) user profile attributes. Figure 1 depicts two emphasized groups: the group of users with red border (g_1), and the group of users with blue border (g_2). In this example, user d belongs to both groups and user b to none.

Recall that $I(S)$ denotes the expected number of nodes covered by a seed-set S. Let $g\subseteq V$ be a group of emphasized users, and $I_g(S)$ denote the expected number of g members covered by S, referred to as the g-cover. We present the auxiliary *Group-Oriented IM* problem, denoted as IM_g, which instead of maximizing $I(\cdot)$, maximizes $I_g(\cdot)$.

Definition 2.4 (The IM_g problem). Given a group $g \subseteq V$ and a number $k\leq|V|$, find a set O_g satisfying: $O_g = argmax_{\{T:T\subseteq V,|T|=k\}} I_g(T)$.

To illustrate, consider the following example.

Example 2.5. Consider again Figure 1 and assume that k=2. The optimal solution for g_2 is O_{g_2}=$\{d, f\}$, where $I(O_{g_2})$=$I_{g_2}(O_{g_2})$=2 and $I_{g_1}(O_{g_2})$=0. The solution that maximizes the g_1-cover is O_{g_1}=$\{e, g\}$, where $I_{g_1}(O_{g_1})$=4 and $I_{g_2}(O_{g_1})$=0.5. Observe that covering a greater number of users from one group may come at the cost of significantly reducing the cover size of users from another group.

The hardness result of IM also applies to this variant, following a straightforward reduction from IM, where g=V.

Proposition 2.6. *The IM_g problem is hard to approximate beyond a factor of* $(1 - \frac{1}{e})$ *in PTIME.*

In Section 4.1 we explain how a given IM algorithm can be adapted to its group-oriented version, retaining all its theoretical properties. Note that this variant can be seen as a special case of the *Targeted IM* problem [26], where the goal is to maximize influence over a targeted group of users, with relevance of users modeled by weights in $[0, 1]$. The IM_g problem is further imposing a dichotomy where the weights are in $\{0, 1\}$, modeling discrete properties.

3 PROBLEM FORMULATION

As mentioned, our results support multiple, possibly overlapping, emphasized groups. However, for simplicity, we initially focus on the two groups scenario and imposed a size constraint on one group. In Section 5.1 we extend our results to multiple emphasized groups, and discuss alternative problem definitions.

3.1 Multi-Objective IM

Let g_1, g_2 to be two emphasized groups. Our goal is to assure the obtained solution will ensure sufficient cover of the two groups. To this end, we add a constraint on the IM_{g_2} problem (pertaining to the g_2 group), which explicitly models how much the user is willing to settle on the g_2-cover, in order to increase the g_1-cover.

Definition 3.1 (Multi-Objective IM). Given a network G, two emphasized groups $g_1, g_2 \subseteq V$, a threshold parameter $0 \leq t \leq 1$ and a number k, find a k-size seed-set O^* that maximizes the g_1-cover size, subject to the constraint on the g_2-cover being above a t-fraction of its optimal size. Namely, find a set O^* s.t:

$$O^* = argmax_{\{T:|T|=k, I_{g_2}(T) \geq t \cdot I_{g_2}(O_{g_2})\}} I_{g_1}(T)$$

where $I_{g_1}(T)$ (resp., $I_{g_2}(T)$) denote the expected size of the g_1 (resp., g_2) cover by T, and O_{g_2} denotes the optimal k-size solution for g_2.

Throughout the paper, we refer to the expected g_1 and g_2 influences, resp., as the objective and the constraint. To illustrate, in Example 1.1, one may wish to maximize the influence over the anti-vaccination users, while ensuring that the influence over all users is at least 60% of its optimal value. Alternately, continuing with Example 1.2, a user may wish to maximize the influence over engineers, while ensuring that the influence over researchers is no less than 50% of its optimal value.

To illustrate how the constraint affects the selected seed-set, consider the following example.

Example 3.2. Consider again Figure 1 and let $k = 2$. For $t = 0.1$ the optimal solution is $S = \{e, g\}$ since $I_{g_2}(S) = 0.5 \geq 0.1 \cdot I_{g_2}(O_{g_2}) = 0.2$ (O_{g_2} is the optimal solution for g_2), and among all 2-size seed-sets satisfying the constraint, its g_1-cover size is maximal with $I_{g_1}(S) = 4$. However, for $t = 0.5$, S no longer satisfies the constraint, and $S' = \{e, d\}$ becomes the optimal solution, with $I_{g_1}(S') = 3.25$ and $I_{g_2}(S') = 1$. This demonstrates that higher values of t put more emphasis on the g_2-cover, possibly at the expense of eliminating seed-sets with high approximation factor for the g_1-cover.

Recall that the IM problem is closely related to the MC problem, as explained in Section 2.1. We define the *Multi-Objective MC* problem, analogous to Multi-Objective IM, which will serve us for deriving our lower bound and for devising the RMOIM algorithm.

Definition 3.3 (Multi-Objective MC). Given subsets $S_1, \ldots, S_m$ of elements from $U = \{u_1, \ldots, u_n\}$, two groups of elements $g_1, g_2 \subseteq U$, a threshold parameter $0 \leq t \leq 1$, and a number $k \leq m$, a constraint is imposed on the number of covered elements from g_2, requiring it to exceed a t-fraction of the optimal cover size. The goal is to find, among all k sets from $S_1, \ldots, S_m$ satisfying the constraint, the one covering a maximal number of elements belong to g_1.

The constraint threshold. Before presenting our algorithms, let us highlight important properties of the constraint threshold parameter t.

First, consider again Example 3.2, demonstrating that setting higher values for t restricts the solution space and diminishes the optimal value for the objective among remaining k-size seed-set. This exposes the inherent trade-off between the objective and the constraint threshold. A higher threshold is at odds with optimizing the main objective.

We note that the actual value of the optimal g_2-cover size, $I_{g_2}(O_{g_2})$, can only be approximated up to a $(1 - \frac{1}{e})$ factor in PTIME. Thus, the exact value can only be referred to implicitly. Hence, to allow the user to make an informed decision for the value of t, our system uses an IM_g algorithm (as we explain in Section 4), yielding the optimal PTIME approximation for $I_{g_2}(O_{g_2})$.

Observe that setting t to 0 nullifies the constraint, producing the IM_g problem for g_1. Therefore, we only examine cases where $t > 0$. Moreover, it is easy to show that for $t > 1 - \frac{1}{e}$, following the hardness results of IM [23], merely finding a single (not necessarily optimal) k-size seed set satisfying the constraint cannot be done in PTIME.

COROLLARY 3.4. *A k-size seed set satisfying the constraint can always be found in PTIME only if $0 \leq t \leq (1 - \frac{1}{e})$. For higher t values, this claim no longer holds.*

We therefore restrict our attention to cases where $0 \leq t \leq (1 - \frac{1}{e})$. In cases where the user is interested in higher values of t, as no PTIME algorithm which satisfies the constraint exists, one would need to employ an exhaustive search over the $|V|^k$ possible k-size seed-sets to find the optimal solution.

3.2 Approximation lower bound

In order to devise efficient algorithms for Multi-Objective IM, it is useful to understand which properties are attainable for a PTIME algorithm. We next formally define the solution space, then present a lower bound for Multi-Objective IM.

The solution space. We generalize the solution space to *bicriteria approximation*, where an algorithm approximates the objective and may also approximate the constraint, up to multiplicative factors of α and β, resp. For β=1 the solution strictly satisfies the constraint. To accommodate practical algorithms we consider, as in standard IM, randomized algorithms that may add an error margin ϵ to the approximation factors, while requiring the stated factors to hold with probability $\geq(1-\delta)$. Formally, given $0 \leq \epsilon, \delta \leq 1$, an algorithm computes a (α, β)-solution S, with $0 \leq \alpha, \beta \leq 1$, if for every instance (G, g_1, g_2, k, t) of Multi-Objective IM, the following holds with probability $\geq 1-\delta$: $I_{g_2}(S) \geq (\beta-\epsilon) \cdot t \cdot I_{g_2}(O_{g_2})$ and $I_{g_1}(S) \geq (\alpha-\epsilon) I_{g_1}(O^*)$, where O^* is the optimal constrained solution w.r.t. Def. 3.1. We assume ϵ and δ are implicitly provided. However, for simplicity, we omit discussions of these parameters whenever possible.

We emphasize that α is derived from comparing the returned solution not to the optimal unconstrained solution, but rather to an optimal solution which satisfies the constraint. This highlights the difference between approximating the constraint by a factor of β and replacing t with $\beta \cdot t$, as the solution space is affected only in the latter case. Namely, when examining a seed-set which relaxes the constraint, the optimal value for the objective is still taken only over the subset of solutions satisfying the constraint. We refer to an algorithm *as dominant over another algorithm* if it computes an approximated solution for higher values of at least one parameter (α, β), with the other parameter being at least equal. We refer to a tuple (α, β) as an *optimum*, if no (PTIME) algorithm that generates an approximated solution dominant over it exists. One immediate such optimum is $(1 - \frac{1}{e}, 1)$, which follows directly from the hardness result of IM [23]. However, as we prove, there exists no PTIME algorithm which can achieve this bound. Moreover, we show that to achieve $\alpha = (1 - \frac{1}{e})$, β must be reduced to $(1 - \frac{1}{e})$ as well.

Hardness of approximation. As mentioned, the optimal objective approximation of Multi-Objective IM is α=$1-\frac{1}{e}$. We next prove that in order to achieve this optimal α value, a relaxation of the constraint is necessary. Concretely, we prove that Multi-Objective IM has no PTIME algorithm with approximation guarantees (even in expectation) dominant over $(1 - \frac{1}{e}, 1 - \frac{1}{e})$, via a reduction from MC. This result is independent of t, yet, surprisingly, holds for all its values in $(0, 1 - \frac{1}{e}]$.

Theorem 3.5. *Multi-Objective IM has no approximation factor dominant over* $(1-\frac{1}{e}, 1-\frac{1}{e})$ *(unless NP = BPP).*

Next, we provide a proof sketch for Theorem 3.5 using a novel reduction from MC.

Proof. (sketch). Given an MC instance along with k and t, let k_t denote the smallest natural number s.t. $I(O_{k_t}) \geq t \cdot I(O_k)$. We first fix any arbitrary k and $t \in (0, 1-\frac{1}{e}]$, then sample two disjoint MC instances, $\mathcal{I}_1$ and $\mathcal{I}_2$, s.t. the seed set size requirements are $k - k_t$ and k_t, resp. We construct a Multi-Objective MC instance by taking the union of both collection of sets, and defining the g_1 and g_2 groups as follows: g_1 comprises of all elements of $\mathcal{I}_1$, and g_2 comprises of all elements of $\mathcal{I}_2$. The cardinality constraint is k along with threshold t. This construction implies a dichotomy where choosing sets from the g_1 collection only affects the objective, while choosing sets from the g_2 collection only affects the constraint. We show that, in the worst case, one needs to choose as many g_2 sets as in the optimal solution (i.e. k_t sets), up to a $o(1)$ factor, to achieve a $(1-\frac{1}{e})$ approximation of the constraint, and therefore with the remaining slots one cannot guarantee any factor beyond $(1-\frac{1}{e})$ for the objective.

Last, we extend this result to Multi-Objective IM via a reduction from Multi-Objective MC. In essence, we reduce a given Multi-Objective MC instance to a graph s.t. each element is mapped to a new node, carrying over any membership in g_1 and g_2 groups. Additionally, for each subset S_i, we create a new node, and add an edge from it into every nodes corresponding to an element in this set, with the constant edge weight of 1. □

Note that this lower bound holds even for the easier version of the problem, where explicit values are known for both the constraint threshold and the constrained optimum for the objective.

4 ALGORITHMS

As mentioned, the approximation factor of the objective depends on how strictly the constraint is preserved. We, therefore, provide two complementary algorithms for Multi-Objective IM. Our first algorithm, named *the Multi-Objective IM (MOIM) algorithm*, finds a seed-set that strictly satisfies the constraint, at the cost of influence decrease for the objective. Its key advantage is that it achieves near-linear time complexity, which, as we show, is critical for scaling successfully to massive networks. To get a tighter approximation ratio for the objective, our second algorithm, named *the Relaxed Multi-Objective IM (RMOIM) algorithm*, relaxes the constraint, allowing its approximation by a $(1-\frac{1}{e})$-factor, achieving in return near optimal approximation for the objective. This however comes at the cost of performance - its time complexity is polynomial.

4.1 The MOIM algorithm

MOIM is a simple yet efficient algorithm achieving state-of-the-art performance by leveraging existing IM algorithms. Intuitively, using a modular approach where given an IM algorithm, it generically modifies it to create two group-oriented versions of it, then combines them together to produce a single seed set. We next detail our modification of a given IM algorithm, followed by the full algorithm scheme.

Given an IM algorithm $\mathcal{A}$ and an emphasized group g, we define $\mathcal{A}_g$ as its IM_g counterpart - an analogous algorithm that maximizes $I_g(\cdot)$ instead of $I(\cdot)$. Any RIS-based algorithm, $\mathcal{A}$, can be adapted to $\mathcal{A}_g$ via a single modification: the *RR* sets are generated from nodes from g only, independently and uniformly as before. We can prove that $\mathcal{A}_g$ outputs a seed-set covering at least $(1-\frac{1}{e}) \cdot I_g(O_g)$ nodes from g, which is optimal [23].

A method of weighted RIS sampling for solving Targeted IM was presented in [26]. Concretely, instead of using the uniform distribution, nodes are sampled according to their weights, which model their relevance to a given context. Our adaptation for IM_g can be seen as a special case of this method with binary weights. Nonetheless, the authors of [26] have focused in cases where there is only one emphasized group. As we show in our experiments, choosing the weights achieving sufficient covers for more than one group requires further effort.

Algorithm 1 The MOIM algorithm.

1: **Input:** A network G; emphasized groups $g_1, g_2 \subseteq V$; $k \in [n]$; $t \leq 1-\frac{1}{e}$; an IM algorithm $\mathcal{A}$.
2: **Output:** A k-size seed set S.
3: We run independently the following two procedures:
  i $S_1 \leftarrow$ Run algorithm $\mathcal{A}_{g_2}$, where the seed set size is fixed to $\lceil -\ln(1-t) \cdot k \rceil$.
  ii $S_2 \leftarrow$ Run algorithm $\mathcal{A}_{g_1}$, where the seed set size is fixed to $\lfloor (1+\ln(1-t)) \cdot k \rfloor$.
4: $S \leftarrow S_1 \cup S_2$
5: **if** $|S| < k$ **then**
6: Run $\mathcal{A}_{g_1}$ on the residual network until enough seeds are gathered.
7: **end if**
8: **return** S

The MOIM algorithm is depicted in Algorithm 1. MOIM runs independently two procedures: The first ensures satisfaction of the constraint (line 3.i), while the second maximizes the objective (line 3.ii). We return the union S of the selected seeds (line 4). If S contains less than k seeds, we run $\mathcal{A}_{g_1}$ on the residual problem (by eliminating the respective sets of the seeds selected so far), s.t. additional nodes are added to S (lines 5-7). In practice, this could be achieved by initially running $\mathcal{A}$. Note that this can only improve the accuracy guarantees. In our analysis we assume that the returned set is of size exactly k.

We now state the approximation factor of MOIM.

Theorem 4.1. *For* $0 \leq t \leq 1-\frac{1}{e}$, *MOIM provides a* $(1-\frac{1}{e\cdot(1-t)}, 1)$*-approximation to the Multi-Objective IM problem.*

Example 4.2. Consider again Figure 1, and let k=2. Recall that the optimal solution for g_2 is O_{g_2}=$\{d, f\}$, with $I_{g_2}(O_{g_2})$=2. For t=$1-\frac{1}{e}$, MOIM would be equivalent to running $\mathcal{A}_{g_2}$ with k=2. It would w.h.p. output, if not O_{g_2}, then a set S, s.t. $I_{g_2}(S) \geq 2\cdot(1-\frac{1}{e}) \approx$ 1.26, with no particular regard for g_1 cover, which may be as small as 1.5 (for S=$\{c, f\}$), or as high as 3 (for S=$\{e, f\}$). For t=$1-\frac{1}{\sqrt{e}}$, MOIM runs $\mathcal{A}_{g_1}$ and $\mathcal{A}_{g_2}$ while setting k=1 for both, which would presumably output $\{e\}$ and $\{f\}$ resp., combining for a seed set S s.t $I_{g_1}(S)$=3 and $I_{g_2}(S)$=1.75. This approximated solution comes close to both O_{g_1} and O_{g_2}, in terms of g_1/g_2 cover size, resp.

The time complexity of MOIM depends only on that of its input IM algorithm $\mathcal{A}$, which is assumed to be near optimal [33].

4.2 The RMOIM algorithm

We first describe a theoretical algorithm which, given the optimal cover size of g_2, $I_{g_2}(O_{g_2})$, exactly matches our hardness bound. We then discuss the practical case where $I_{g_2}(O_{g_2})$ is unknown (and can only be approximated in PTIME), proving that the scale of the reduction in the approximation factors is not too high.

Theorem 4.3. *There exists a PTIME randomized algorithm that, given $I_{g_2}(O_{g_2})$, in expectation, outputs a $(1-\frac{1}{e}, 1-\frac{1}{e})$ approximation for the Multi-Objective IM problem.*

We described the reduction from IM to MC suggested in [7], utilized by the RIS framework. We extend this reduction to the multi-objective variants, implying that any algorithm for Multi-objective MC can be extended to Multi-Objective IM, retaining the same guarantees. Therefore, all that is left to prove is that one can get a $(1-\frac{1}{e}, 1-\frac{1}{e})$-approximation for Multi-Objective MC.

Given an instance $\mathcal{I}$ of Multi-Objective MC with m subsets $S_1, ..., S_m$ and two groups $g_1, g_2 \subseteq U$, we construct LP($\mathcal{I}$), the corresponding LP instance, where $Y = |g_2 \backslash g_1|$, $Z = |g_1 \backslash g_2|$, $W = |g_1 \cap g_2|$:
variables: $x_1, \ldots, x_m, y_1, \ldots, y_Y, z_1, \ldots, z_Z, w_1, \ldots, w_W$ (x_i is an indicator for selecting S_i, y_i for covering element in $g_2 \backslash g_1$, z_i for covering element in $g_1 \backslash g_2$, and w_i for covering elements in $g_1 \bigcup g_2$)
constraints: $\sum_{i=1}^{m} x_i = k$ (cardinality constraint)

$$\sum_{i:u_j \in S_i} x_i \geq y_j, \sum_{i:u_j \in S_i} x_i \geq z_j, \sum_{i:u_j \in S_i} x_i \geq w_j \text{ (coverage constraint)}$$

$$(\sum_{i=1}^{Y'} y_i \cdot \frac{Y}{Y'} + \sum_{i=1}^{W'} w_i \cdot \frac{W'}{W}) \geq t \cdot I_{g_2}(O_{g_2}) \text{ (size constraint)}$$

$$\forall i \in \{1, ..., m\}, 0 \leq x_i \leq 1; \forall i \in \{1, \ldots, W'\}, 0 \leq w_i \leq 1$$

$$\forall i \in \{1, \ldots, Y'\}, 0 \leq y_i \leq 1; \forall i \in \{1, \ldots, Z'\}, 0 \leq z_i \leq 1$$

objective: maximize $\sum_{i=1}^{Z'} z_i + \sum_{i=1}^{W'} w_i$.
where $I_{g_2}(O_{g_2})$ is the optimal g_2-cover size and Y', Z', W' are the number of sampled nodes from $g_2 \backslash g_1$, $g_1 \backslash g_2$ and $g_1 \cap g_2$, resp.

The solution is determined by the values of the variables x_i, indicating the selected sets. This LP relaxes the Integer LP which precisely models the Multi-Objective MC problem. We can compute an optimal solution by using any LP solver, then apply the following randomized rounding procedure [30]: (1) Interpret the numbers $\frac{x_1}{k}, .., \frac{x_m}{k}$ as probabilities corresponding to $S_1, .., S_m$, resp. (2) Choose k sets independently w.r.t. the probabilities. By adapting the proof in [32], we show that this procedure yields a seed set whose cover, in expectation, for each group separately, is at least a $1-\frac{1}{e}$ fraction of the corresponding optimal cover size, thus proving Theorem 4.3.

Omitting the optimal-value knowledge assumption. As mentioned, the optimal value of the g_2-cover is uncomputable in PTIME. We, therefore, first run a IM_{g_2} algorithm which outputs a seed set S, s.t. $I_{g_2}(O_{g_2}) \cdot (1-\frac{1}{e}) \leq I_{g_2}(S) \leq I_{g_2}(O_{g_2})$. We then set the constraint threshold in $LP(\mathcal{I})$ to $t \cdot (1-\frac{1}{e})^{-1} \cdot I_{g_2}(S)$ instead of $t \cdot I_{g_2}(O_{g_2})$, with the rest of the algorithm remaining the same. This substitution can only increase the constraint threshold, which in turn, reduces the set of valid solutions, possibly diminishing the objective value of the optimal solution subject to the stricter constraint. However, as we prove, the scale of the reduction in α is not arbitrarily large.

The RMOIM algorithm is depicted in Algorithm 2. Given an IM algorithm $\mathcal{A}$, we first run $\mathcal{A}_{g_2}$ to estimate $I_{g_2}(O_{g_2})$ (line 3). Next, using $\mathcal{A}$, we sample the *RR* sets needed for constructing the Multi-Objective MC instance, and build the corresponding LP (lines 4 − 5). Then, we employ an LP solver, obtaining the fractional solution (line 6). Last, we employ the rounding procedure to select k sets for the Multi-Objective MC instance, and return their corresponding nodes as the selected seed-set S (lines 7 − 8).

Given an IM_g algorithm, let S denote its output. We define $\lambda \in [0, \frac{1}{e-1}]$ s.t. $I_g(S) = (1+\lambda) \cdot (1-\frac{1}{e}) \cdot I_g(O_g)$.

Algorithm 2 The RMOIM algorithm.

1: **Input:** A network G; emphasized groups $g_1, g_2 \subseteq V$; $k \in [n]$; $t \leq 1-\frac{1}{e}$; an RIS-based IM algorithm $\mathcal{A}$ and an LP solver.
2: **Output:** A k-size seed set S.
3: $I_{g_2}(\tilde{O_{g_2}}) \leftarrow$ Run $\mathcal{A}_{g_2}$ on the input.
4: RR $\leftarrow$ Construct the *RR* sets using $\mathcal{A}$.
5: LP($\mathcal{I}$) $\leftarrow$ Construct the LP from RR, replacing $t \cdot I_{g_2}(O_{g_2})$ with $t \cdot (1-\frac{1}{e})^{-1} \cdot I_{g_2}(\tilde{O}_{g_2})$.
6: $\vec{X} \leftarrow$ Solve LP($\mathcal{I}$), and output the values for the x_i variables.
7: $S \leftarrow$ Run the randomized rounding procedure on $\vec{X}$.
8: **return** S

Theorem 4.4. *The RMOIM algorithm provides, in expectation, a $((1-\frac{1}{e}) \cdot (1-t \cdot (1+\lambda)), (1+\lambda) \cdot (1-\frac{1}{e}))$ approximation to Multi-Objective IM, where $\lambda \in [0, \frac{1}{e-1}]$.*

The time complexity of RMOIM is dominated by its input LP solver, whose complexity is polynomial in the input size [22].

5 EXTENSIONS

We present an extension of our results to multiple groups, then briefly discuss on alternative problem definitions. We conclude with a discussion regarding a well-studied related problem.

5.1 Multiple Emphasized Groups

The Multi-Objective IM problem naturally extends to multiple groups. Given m emphasized groups, the user can impose size constraints on all but one groups, and subject to these constraints, maximize the cover size of the remaining group. W.l.o.g. let us assume that the user imposed size constraints on the last $m-1$ groups. Given the $m-1$ constraint threshold parameters $t_2, ..., t_m$, analogously to the binary scenario, we can show that a k-size seed set satisfying all constraints can always be found in PTIME if $0 \leq \sum_i t_i \leq (1-\frac{1}{e})$. We prove that in PTIME, one cannot attain an approximation factor dominant over $(1-\frac{1}{e}, \ldots, 1-\frac{1}{e})$. Moreover, our generalized random algorithm matches our lower bound for multiple groups.

Both our algorithms can be generalized to solve the multiple groups scenario. In MOIM we run (independently) $m-1$ $IM_{g_i}, i \in [2, m]$ algorithms, where the seed set size in each algorithm is fixed to $\lceil -\ln(1-t_i) \cdot k \rceil$, and run an $\mathcal{IM}_{g_1}$ algorithm, where the seed set size is fixed to $\lfloor (1+\ln(1-\sum_i t_i)) \cdot k \rfloor$. As in Algorithm 1, we then return the union of the selected seeds. We can show that this algorithm provides a $(1-\frac{1}{e \cdot (1-\sum_i t_i)}, 1, \ldots, 1)$-approximation to Multi-Objective IM with m emphasized groups.

In RMOIM, we first estimate the $I_{g_i}(O_{g_i})$ values for the constrained $m-1$ groups, to include these values in the LP described in Section 4.2. Given an IM_{g_i} algorithm, let S_i denote its output. Recall that $\lambda_i \in [0, \frac{1}{e-1}]$ was defined s.t. $I_{g_i}(S_i) = (1+\lambda_i) \cdot (1-\frac{1}{e}) \cdot I_{g_i}(O_{g_i})$. We prove that RMOIM provides, in expectation, a $((1-\frac{1}{e}) \cdot (1-\sum_i t_i \cdot (1+\sum_i \lambda_i)), (1+\lambda_1) \cdot (1-\frac{1}{e}), \ldots, (1+\lambda_{m-1}) \cdot (1-\frac{1}{e}))$-approximation to Multi-Objective IM with m emphasized groups.

5.2 Alternative problem definitions

We next briefly discuss alternative problem definitions. An alternative variant of Multi-Objective IM is where the user specifies an explicit value constraint (rather than specifying a fraction of the optimal possible value). For instance, continuing with Example 1.2, one may request to maximize the cover over engineers, subject to a constraint requiring that at least 1K researchers are influenced. Both our algorithms support this variant as well. Specifically, in MOIM, we can run an IM_{g_2} algorithm until it exceeds the constraint value, and with the remaining seeds we run an IM_{g_1} algorithm, which can only improve the guarantees as we no longer overestimate the constraint. In RMOIM, the problem becomes much simpler, since now the exact value for the size constraint is known. Therefore, here RMOIM is optimal as it matches our lower bound (which holds here as well). We focus on the implicit size constraint variant, as the analysis of the explicit value constraint variant is contained in it as a simpler case.

Our definition provides cardinality guarantees over the emphasized groups. An alternative definition may be to constrain the *ratio* of different cover cardinalities. We note that this definition is essentially different form our definition, as maximizing the ratio between the cover cardinalities can dramatically reduce the number of covered users from each group. Therefore, such definition is ill-suited to our motivation where the underlying goal is to reach as many as possible users from the emphasized groups. We further note that the analysis of such ratio-based definitions differs from the one we have provided, and therefore we leave the study of ratio-based constraints for future research.

In our analysis so far the user imposes constraints on all but one group. Our results also support the case where the user imposes constraints on all emphasized groups (see details in [3]).

5.3 Connection to the RSOS problem

The closely related problem of multi-objective maximization of monotone submodular functions subject to a cardinality constraint (known as the RSOS problem) was introduced in [24].

Given m monotone submodular functions $f_i(\cdot), i \in \{1, .., m\}$ and a target value V_i for each function f_i, the goal in the RSOS problem is to find a k-size set A s.t. $\forall i : f_i(A) \geq V_i$, or provide a certificate that there is no feasible solution. A solution S is an *alpha*-approximation if $\forall i : f_i(S) \geq \alpha \cdot V_i$.

In contrast to Multi-Objective IM, where users can specify for each group the *fraction* of the optimal influence that they wish to retain, in RSOS only explicit values can be used. Nonetheless, we establish the connection between the two problems. Specifically, we prove that the two problems are equally hard, and that any algorithm solving RSOS, could in principle also solve Multi-Objective IM. However, as we show in our experiments, top performing RSOS algorithms can only process small networks.

We next briefly present our main results.We restrict our analysis of the RSOS problem to its applicability in an IM setting, s.t. all functions are IM-functions. To simply the presentation, we focus here on the two groups scenario, and defer the analogous results regarding multiple groups to [3].

We reduce RSOS to Multi-Objective IM, showing that any (α, α)-approximation to Multi-Objective IM implies an α-approximation to RSOS. It follows that leveraging existing techniques in RSOS works yields at best an $(1-\frac{1}{e}, 1-\frac{1}{e})$-approximation for Multi-Objective IM, which is an optimum we have already achieved with RMOIM.

THEOREM 5.1. *RSOS* $\leq_p$ *Multi-Objective IM.*

We further provide a reduction in the other direction, showing that any α-approximation algorithm for RSOS, implies an (α, α)-approximation algorithm for Multi-Objective IM.

THEOREM 5.2. *Multi-Objective IM* $\leq_p$ *RSOS.*

Table 1: Datasets.

Datasets	Dimensions	Profile properties
Facebook	|V|=4K, |E|=168K	Gender, Education type.
DBLP	|V|=80K, |E|=514K	Gender, country, age, h-index.
Pokec	|V|=1M, |E|=14M	Gender, age, region
Weibo-Net	|V|=1.5M, |E|=369M	Gender, city.
YouTube	|V|=1M, |E|=3M	-
LiveJournal	|V|=4.8M, |E|=69M	-

However, to do so, we need to know both the optimal cover size of the constrained group $I_{g_2}(O_{g_2})$ (as in RMOIM), and (additionally) the constrained optimal objective value $I_{g_1}(O^*)$. $I_{g_2}(O_{g_2})$ may be estimated, as done in RMOIM, by running an IM_{g_2} algorithm. Here again, we may overestimate this value by a $(1-\frac{1}{e})$ factor, yielding the same guarantees as RMOIM. To efficiently estimate $I_{g_1}(O^*)$, we can examine only $O(log(n))$ guesses for $I_{g_1}(O^*)$, which increases the time complexity of an RSOS algorithm by an $O(log(n))$ factor.

A state-of-the-art algorithm for RSOS, which achieves the optimal $(1-\frac{1}{e})$-approximation, has been introduced in [36]. As we show in our experiments, this algorithm can only process small networks (even without the $log(n)$ multiplicative overhead).

6 EXPERIMENTAL STUDY

We have implemented our prototype in Python 2.7. We use as the input IM algorithm, for both of our algorithms, *IMM*[1] [33], a top performing IM algorithm. We solve the LP in RMOIM using Gurobi LP solver [2][2]. We have conducted an experimental study to evaluate (1) The quality of results achieved by our algorithms. We demonstrate the advantages of our algorithms in multiple scenarios over real-life datasets, compared to existing and alternatives approaches; (2) The performance of our algorithms in terms of execution times and scalability.

6.1 Experimental setup

We conducted all experiments on a Linux server with a 2.1GHz CPU and 96GB memory. Next, we describe the examined datasets, the considered emphasized groups, the competing algorithms, and the parameters setup.

Datasets. We have focused on social networks which include user profile properties, to characterize the emphasized groups. We have examined 6 commonly used datasets: Facebook, DBLP, Pokec, Weibo-Net, Twitter and Google+ (extracted from [4, 25]). For space constraints, we omit the results over Twitter and Google+, as they were similar to those obtained over the other 4 datasets (depicted in Table 1). To further examine our algorithms scalability, we considered two additional large-scale datasets: YouTube and LiveJournal [25]. These datasets do not include user properties. To nevertheless examine them in our context, we randomly assigned users to emphasized groups (see details below). Following the conventional method as in [28, 34], we set the weight of each edge (u, v) as $w(u, v) = \frac{1}{d_{in}(v)}$, where $d_{in}(v)$ denotes the in-degree of v. To ensure uniformity, undirected networks were made directed by considering, for each edge, the arcs in both directions (as was done in [5]).

[1]We used the corrected version described in [10].
[2]Our code will be publicly available upon acceptance.

Emphasized groups. The benefit that our approach brings is in particular critical for subpopulations that are typically not covered by standard IM algorithms. To identify such groups, we have run, for each network, a grid search over the extracted profile properties. We have considered all groups that are characterized by a single or a combination of two profile properties. For each such group g, we have examined the expected g-cover size of standard IM algorithms, as well as the expected g-cover size of their IM_g counterparts. We are focusing here only on groups in which the results showed that standard IM algorithms tend to overlook their users, while targeted IM algorithms showed that a different choice of seed-set significantly increase their expected cover size. Interestingly, our experiments indicate that all analyzed datasets include several such groups. For example, female Indian researchers in DBLP and females over the age of 50 in Pokec, are typically neglected by standard IM algorithms. Additional examples are provided in [3]. For YouTube and LiveJornal, we have considered random emphasized groups, defined as follows. Given a number $c \in (0, 1]$ (sampled uniformly at random), every node $v \in V$ is a member of the emphasized group with probability of c. Note that this simple definition allows for overlapping emphasized groups of different cardinalities.

Examined scenarios. We examine the following two scenarios: ***Scenario I.*** In this scenario the user wishes to maximize the overall influence (g_1), subject to a constraint requiring that at least a given portion of a group's members (g_2) are influenced (a scenario analogous to that of Example 1.1). We focus on this particular scenario as it allows to compare, in a single setting, algorithms for standard IM (that maximize the overall influence), targeted IM (that maximize the influence solely over the g_2 members), and ours. We present the results while setting g_2 to be a group which is not covered by standard IM algorithms (see full details in [3]). We have also run all experiments while choosing all possible pairs of g_1 and g_2 to be groups that are typically not covered by standard IM algorithms. We report that all experiments show similar trends and therefore we omit from presentation these results. ***Scenario II.*** Next we consider multiple-groups, to demonstrate the effect of multiple objectives on performance. We present a scenario where the user provides 5 emphasized groups, specifies constraints on 4 of them, and asks to maximize the influence over the remaining group, subject to these constraints. We have also experimented with other numbers of emphasized groups and report that all results have shown similar trends. In real-life scenarios, the number of emphasized groups is typically small [26, 36] and thus we focus on realistic number ranges $(2 - 10)$. Here again we have considered groups that are typically not covered by standard IM algorithm.

Competing algorithms. We consider the following baselines. **Standard IM algorithms.** We have examined the results of *IMM* [33] and SSA [28], top preforming RIS-based algorithms, as well as SKIM [13] and Celf++ [17], greedy-based IM algorithms. As all algorithms demonstrated similar trends, we detail here only *IMM*.
(Single objective) Targeted IM algorithms. We examine IMM_g, a variant of *IMM* (based on [26]) which maximizes exclusively the cover of a given emphasized group g. In scenario *II* we have defined the target group to be the union of all emphasized groups. **Weighted IM.** An alternative is to assign different weights to users, reflecting their relevance to the objectives. The authors of [26] introduced a weighted RIS sampling method, that maximizes the influence over a targeted group. We examined the results for Weighted *IMM* (*WIMM*), a variant of *IMM* which is based on a weighted RIS sampling method presented in [26]. We apply a (multi-dimensional) binary search to find the optimal weights[3]. We examined the results while substituting the weights of users in the constrained group(s) and the objective group with c_i and $1 - \sum_i c_i$, resp[4]., for varying values of $c_i \in [0, 1]$.
We have also examined a variant of *WIMM* that skips the search and instead uses some default weights given as input. **RSOS algorithms.** We examine the RSOS algorithm of [36] (used to solve Multi-Objective IM). Additionally, the authors of [36] have studied the problem of fair resource allocation in IM, and proposed two fairness concepts: *MaxMin*, which maximizes the minimum fraction of users within each group that are influenced, and Diversity Constraints (*DC*), which guarantees that every group receives influence proportional to what it could have generated on its own, based on a number of seeds proportional to its size. They have shown that both fairness concepts can be reduced to RSOS, for which they provided the state-of-the-art algorithm. For completeness, we have included the *MAXMIN* and *DC* baselines. As we show, all RSOS-based algorithms can only process small networks. A more recent fairness-aware IM framework was presented in [15]. However, in this work as well, only small-size networks were examined[5].

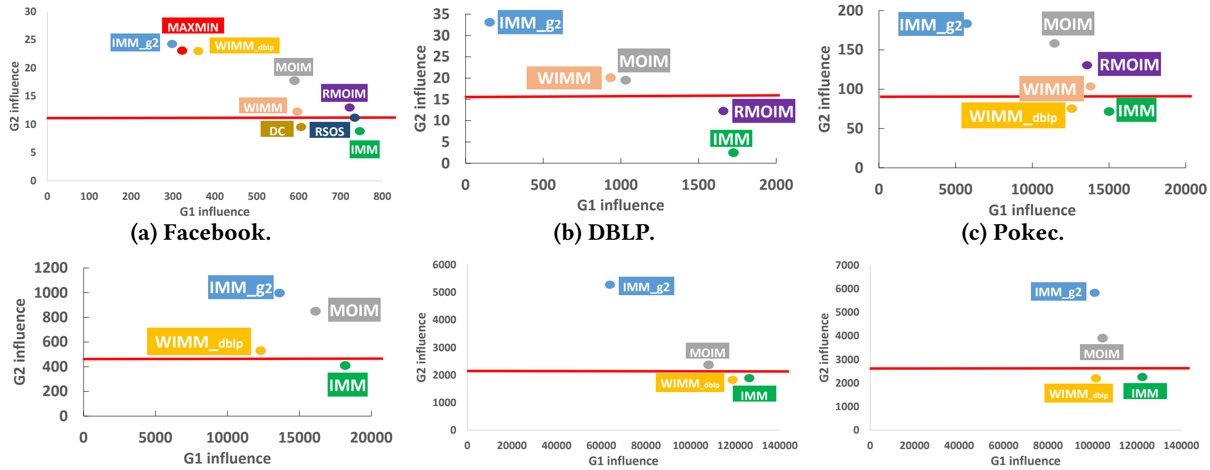
Parameter Settings. Recall that RMOIM requires to estimate $I_{g_i}(O_{g_i})$, the optimal cover cardinality for all constrained groups g_i. For that we use the following estimation strategy (as described in Section 4.2): for each emphasized group g we ran IMM_g for 10 times, selecting the minimal obtained value to derive an estimate for $I_g(O_g)$. Unless mentioned otherwise, we set $k = 20$, and $\epsilon = 0.1$. In scenario *I* we have set the threshold parameter $t = 0.5 \cdot (1 - \frac{1}{e})$, and in scenario *II* we have set the threshold parameters $t_i = 0.25 \cdot (1 - \frac{1}{e}), \forall i \in 1, ..., 4$. We also use, as a default setting, the LT model (when setting uniformly random threshold for every node). In all experiments, the time-out limit is 24 hours (or out of memory exception). For the RSOS baselines, we use the default parameters as provided in [1]. We report for each baseline the averaged measurements of 10 runs.

6.2 Quality Evaluation

Scenario I results. The results are depicted in Figure 2, where the x and y axes represent, resp., the g_1 and g_2 influences, and red lines are the estimated constraint thresholds. A desirable solution should be above (or near) the red lines (i.e., satisfying the constraint), and, at the same time, the right as much as possible (i.e., covering as many g_1 users as possible). For *WIMM*, we present the results obtained by selecting the optimal weights for each dataset (pink points). We have also examined multiple settings of default weights for *WIMM*, however, none of these options yielded satisfying results across all datasets. In particular, the optimal weights per network were different, and to illustrate that, we show how the optimal weights for DBLP operate on the other datasets (yellow points).

In all cases, MOIM managed to match (and sometimes even exceed) the results of *WIMM*, which uses the optimal weights for each dataset. For example, over Facebook, while *WIMM* and MOIM influenced almost the same number of g_1 users (601 and

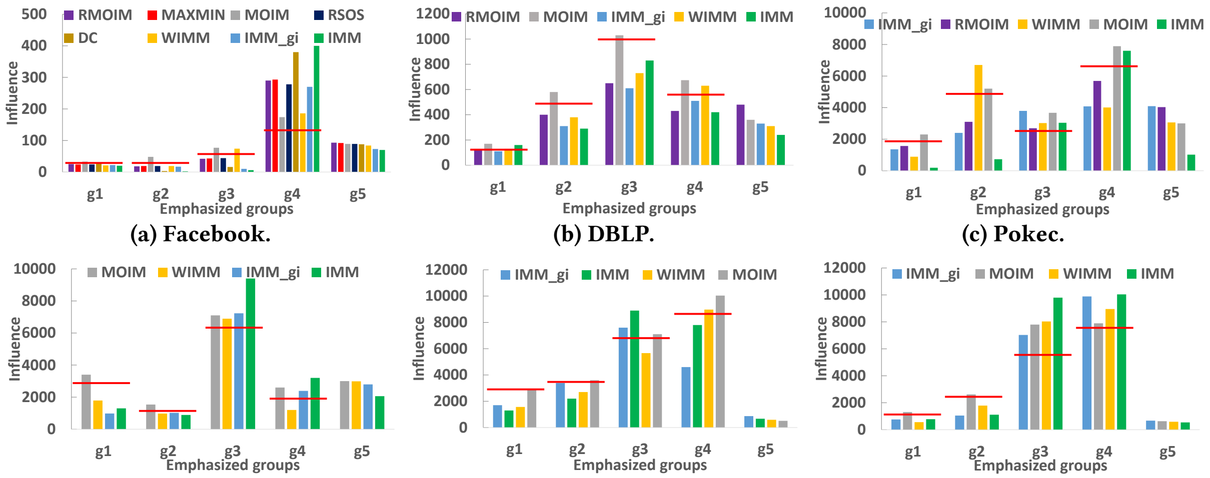
[3]The optimal choice is the one that satisfies all constraints, while maximizing the value for the objective.
[4]Users belong to multiple groups are assigned with the sum of weights of their groups.
[5]In both [36] and [15], the largest examined network included 500 nodes.



(a) Facebook. (b) DBLP. (c) Pokec.

(d) Weibo-Net. (e) YouTube. (F) LiveJournal.

Figure 2: Expected influence with 2 emphasized groups. The red horizontal lines represent the estimated constraints.



(a) Facebook. (b) DBLP. (c) Pokec.

(d) Weibo-Net. (e) YouTube. (F) LiveJournal.

Figure 3: Expected influence with 5 emphasized groups. The red horizontal lines represent the estimated constraints.

599, resp.), MOIM succeeded in covering more g_2 users (19 vs. 12 for MOIM and *WIMM*, resp.). Observe that using the optimal weights for DBLP over Pokec for *WIMM*, result in not satisfying the constraint. The exploration of *WIMM* for optimal weights significantly increases its runtime, making it impractical for massive networks like Weibo-Net, YouTube and LiveJournal (exceeded our time cutoff). In all cases, not only did MOIM satisfy the constraint, it also came very close to the results of IMM_{g_2} in terms of covering g_2 users, which returns the optimal solution. For example, over Pokec, where IMM_{g_2} covered 189 g_2 users, MOIM covers 159, as opposed to *IMM* covering only 73 such users.

Although RMOIM allows for some relaxation of the constraint, it in-fact fully satisfied it in most cases. Moreover, its overall influence was consistently higher than those of *WIMM* and MOIM. In particular, in all but one of the cases, the g_1 influence of RMOIM was very close to that of *IMM*. For example, over DBLP, RMOIM and *IMM* covered $1,661$ and $1,712$ users, resp., with RMOIM covering over 6 times more g_2 members. RMOIM is incapable of processing massive networks like Weibo-Net (out of memory).

Not surprisingly, the results RMOIM and *RSOS* were similar. Nonetheless, as opposed to RMOIM, all RSOS-based baselines were incapable of even processing medium-size networks (exceeded our time cutoff). Recall that *MAXMIN* aims to maximize the minimum influence over the emphasized groups, and therefore here it behaves similarly to IMM_{g_2} (as $g_2 \subseteq g_1$). As for *DC*, since it guarantees that every group receives influence proportional to what it could have generated on its own, it ignores the constraint. This demonstrates that *MAXMIN* and *DC* are ill-suited for Multi-Objective IM.

Observe that the single objective algorithms were either far from satisfying the constraint (*IMM*) or covered significantly less g_1 users (IMM_{g_2}). Contrarily, both our algorithms succeeded in covering almost as many g_1 users as *IMM*, and almost as many g_2 users as IMM_{g_2}. For example, over DBLP, *IMM* covered only 2 g_2 users and $1,712$ users in total (g_1 users), whereas IMM_{g_2} covered 33 g_2 users, and less than 155 in total. MOIM and RMOIM covered 20 and 13 g_2 users, resp., and covered each more than $1,050$ users in total. This demonstrates the advantage of our approach over solutions which are focused only on a single objective.

Last, consider Figures 2 (e) and (f). Among all competitors that satisfy the constraints, MOIM has influenced the largest number of users. Interestingly, even though the emphasized groups were randomly generated, *IMM* did not satisfy the constraints. As for IMM_{g_i}, it influences significantly less users than MOIM. This demonstrates that existing single-objective IM algorithms do not ensure the desired balance between the objectives. Note that here the differences in the cover cardinalities among all competitors were smaller than in other networks. This stems from the fact that the benefit our approach provides is particularly critical for groups that are typically not covered by standard IM algorithms (which is mostly not the case in random emphasized groups).

Scenario II results. The results are depicted in Figure 3, where the y-axis is the influence over the emphasized groups, and red lines represent the estimated constraint thresholds. A desirable solution should be above (or near) the red lines for the constrained groups $g_1, \ldots, g_4$ groups, and, at the same time, should be as high as possible for g_5 (i.e., maximizing the objective). For the $WIMM$ baseline we only present the results obtained by using default weights set to 0.2 for all 5 groups (we report that similar results were obtained when using other weighting schemes), as the search for the optimal weights was infeasible in all cases (it exceeded our time cutoff).

MOIM is the only algorithm satisfying all constraints over each dataset. On top of that, its g_5 influence (i.e., objective value) competes nicely with all competitors. For example, over Weibo-Net, MOIM succeeded to cover the greatest number of g_5 members, while over YouTube it covered 510 g_5 members, compared with the best competitor (here - IMM_{g_i}) that covered 810 g_5 users (yet did not satisfy the constraints). In the datasets which RMOIM has managed to process, its g_5 influence was the best or slightly below the best value achieved. E.g., over Pokec, RMOIM and IMM_{g_i} covered 4036 and 4090 g_5 users, resp., while over Facebook and DBLP RMOIM covered the greatest number of g_5 users.

Here again, all RSOS baselines could only process the small Facebook network (exceeded our time cutoff in other datasets), and, as expected, the results of *RSOS* and RMOIM were similar. Here, *MAXMIN* also behaves similarly to RMOIM, however, as noted above, in other scenarios it may behave differently. This stems from the fact that *MAXMIN* optimizes for equality of outcomes, which may be undesirable when some groups are much better connected than others. For instance, if one group is poorly connected, *MAXMIN* would require that a large number of seeds is "spent" on reaching it, even though these seeds may have a relatively small impact on other groups. As the *DC* baseline ignores the constraints, it did not satisfy them.

As opposed to the binary scenario where the objective was to maximize the overall influence, here *IMM* has no advantage over the competitors. Indeed, in all except one of the examined cases, *IMM*'s objective value was the lowest among all algorithms. Furthermore, regarding IMM_{g_i}, as can be seen, covering a greater number of users from one group may come at the cost of significantly reducing the cover sizes of users from other groups. For example, in LiveJournal (Figure 3 (F)), while the g_4 and g_5 cover sizes of IMM_{g_i} were the largest, its g_1 and g_2 cover sizes were significantly lower than the competitors (and below the required constraints). This demonstrates that existing (single-objective) IM algorithms do not ensure the desired balance between the objectives.

6.3 Parameter Tuning

Next, we examine how varying the input parameters affects the results. To illustrate, we present here the results using a range of values for k and t over the DBLP dataset (the other datasets show similar trends). We note that a desirable behavior of a Multi-Objective IM algorithm is as follows. As k increases, we expect both the g_1 (i.e., overall) and the g_2 (i.e., emphasized group) influences to increase as well. As t increases, i.e., the constraint threshold is elevated, the g_2 influence should increase, possibly at the cost of reducing the g_1 (i.e., overall) influence. Naturally, as only our algorithms and $WIMM$ take into account the parameter t, other competitors are indifferent to it.

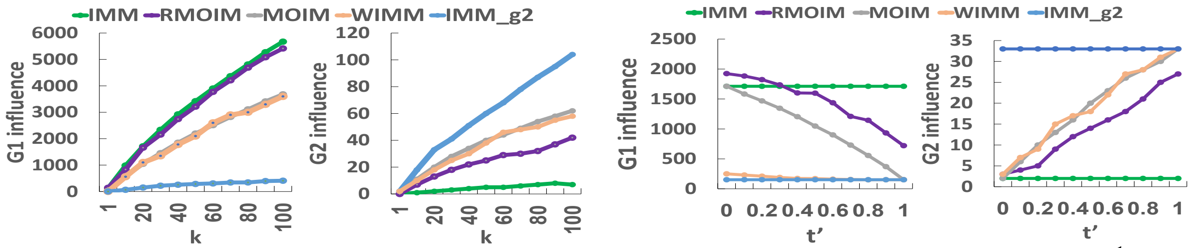
The results are depicted in Figure 4. We first examine Figure 4(a). Interestingly, for all examined k values, the targeted IM algorithm, IMM_g, has shown almost no growth in the overall number of influenced users (less than 400), compared to *IMM* and RMOIM, which, already for $k = 10$, are influencing twice as many users (more than 800). Analogously, for all k values, there is almost no increase in the number of emphasized users influenced by *IMM* (8 such users at most), while IMM_g, already for $k = 10$, influenced twice as many emphasized users (more than 18 such users). Contrarily, MOIM, RMOIM and $WIMM$ have demonstrated the desired behavior when k increases. As expected, MOIM, RMOIM and $WIMM$, as t increases, cover a greater number of g_2 users, and fewer users in total, as illustrated in Figure 4(b). Note that in these experiments $WIMM$ exhibit the desired behavior, almost identical to that of MOIM. However, as we will see next, its execution times are significantly longer.

6.4 Performance Evaluation

We next measure the cost of enriching the IM problem by incorporating multiple objectives, studying how different parameters affect running times of our algorithms. For brevity, we present the results only for scenario *II*, as the results for scenario *I* show similar trends (see [3]).

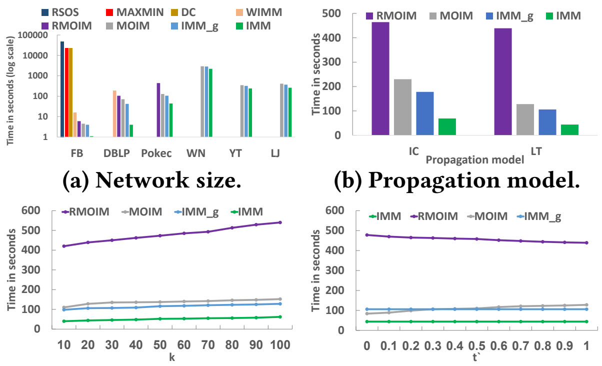
Recall that MOIM runs targeted IM algorithms (i.e., IMM_g) as subroutines. As we show, the overhead for MOIM turns out to be negligible compared to IMM_g, and it can process massive networks efficiently. Naturally, MOIM behaves similarly to its current input algorithm *IMM*, whose optimizations and shortcomings both carry over to MOIM. In particular, as mentioned in [33], when k decreases, so does the optimal expected influence, $I(O)$ (resp. $I_g(O_g)$), in which case it is more challenging for *IMM* (resp. IMM_g) to estimate $I(O)$ (resp. $I_g(O_g)$). Contrarily, for larger k values, *IMM* (resp. IMM_g) is optimized to reuse *RR* sets produced in earlier stages. Thus, the two main factors affecting *IMM* (resp. IMM_g) are k and $I(O)$ (resp. $I_g(O_g)$). Consequently, these factors have a similar effect on MOIM. Regarding RMOIM, we show that solving an LP is indeed costlier than employing an IM algorithm. We will see that when it comes to medium or large scale networks, RMOIM's overhead turns out to be moderate, but when it comes to massive networks it is incapable of processing them. We further show that RMOIM's scalability is not affected by the same factors as MOIM, and its running times are barely affected by those of its input IM algorithm.

Network size. . We first report the running times for the cases presented above in Figure 5(a). Naturally, all competitors' running times increase for larger networks. Although we see that MOIM and RMOIM are naturally slower than *IMM* and IMM_g, they run in approximately 2 and 7 minutes, resp., even on Pokec, which includes 1M nodes and 14M edges. That is, both our algorithms can process large-scale networks in feasible running times. Importantly, note that the running times of MOIM are very close to those of IMM_{g_i} (i.e., MOIM and IMM_{g_i} have processed YouTube in 5.7 and 5.3 minutes, resp.). When it comes to massive networks such as Weibo-Net, while MOIM processed it in less than 49 minutes (in comparison, IMM_{g_i} processed it in 47 minutes), RMOIM can not process it, since the LP program was too big for the LP solver to handle (out of memory). According to our experiments, RMOIM is feasible for graphs including up



(a) Varying values of k. (b) Varying values of t' (t is set to $t' \cdot (1 - \frac{1}{e})$).

Figure 4: The expected influence of different baselines on the DBLP network, using varying values of k and t.



(a) Network size. (b) Propagation model.

(c) Seed-set size k. (d) Constraint threshold.

Figure 5: Averaged execution times (for scenario *II*).

to 20M edges and nodes. Regarding *WIMM*, as it searches for optimal weights, its running times were significantly longer than both our algorithms. For example, on Facebook, it took *WIMM* 16 seconds - almost 4 times slower than MOIM (which ended after 4.5 seconds). Observe that all RSOS-based algorithms ran in more than 6 hours, even on the small Facebook instance network.

In what follows we focus on the Pokec dataset, as this is the largest dataset RMOIM can process. We omit the results of the RSOS-based and *WIMM* baselines, as they cannot process it.

Propagation model. We present the effect of the propagation model on running times in Figure 5(b). As reported in [5], while *IMM* scales well under the LT model, it shows inferior performance under the IC model, as it samples more *RR* sets. Consequently, all *IMM* variants, MOIM included, run slower under the IC model. Indeed, it took all *IMM* variants almost twice the time to process Pokec when using the IC model. Contrarily, as RMOIM is less sensitive to the increase in the number of *RR* sets, and it behaves similarly under both propagation models (the difference was less than a minute). As explained in [5], besides *IMM*, multiple top performing IM algorithms are not robust across different propagation models (e.g., [18], [34]). This property of *IMM* is naturally carried over to MOIM. In cases where the user is interested in a different propagation model, she can take a different IM algorithm optimized for this model (e.g., [13] for IC) as an input for MOIM.

Seed-set size. . In Figure 5(c) we examine the effect of the parameter k on running times. As mentioned, when k increases *IMM* employs an optimized computation and hence we observe almost no change in running times for all *IMM* variants, MOIM included. This behavior of MOIM is a consequence of employing *IMM*, and therefore using an alternative IM algorithm (e.g., [17]) could lead to a linear growth in running times. As expected, RMOIM demonstrates nearly linear growth as a function of k, as more k-size seed sets are considered.

Constraint threshold parameter. . In Figure 5(d) we examine how the parameters $t_i, i \in [1,4]$ affect performance. Here we tested all t_i values of the form $t_i = 0.25 \cdot t' \cdot (1 - \frac{1}{e})$, where $t' \in [0.1, 0.2, \ldots, 1]$. Note that this parameter only affects the behavior of our algorithms. In MOIM it dictates the required seed-set size for the procedures it employs. Observe that when all $t_i = 0$ MOIM only runs IMM_{g_5}, while for other t_i values it employs 5 versions of IMM_{g_i} with smaller k values, therefore it cannot use *IMM* optimizations for large k values. On the other hand, as the solution space becomes smaller for higher t_i values (i.e., less k-size seed-sets satisfy the constraint), the running time of RMOIM decreases.

7 RELATED WORK

The seminal work of [23], the first to formulate the IM problem, has motivated extensive research [5, 13], which can be classified into three main approaches: (i) The greedy framework [18, 23, 29], which iteratively adds nodes to the seed-set, maximizing the expected marginal influence gain; (ii) The RIS framework [7], where, while retaining optimal accuracy, running times were gradually improved, resulting in highly scalable algorithms [20, 28, 33]; (iii) In cases where scalability is preferred over accuracy, there are heuristic algorithms that have been shown to perform well in practice (e.g., [11]), despite not having theoretical guarantees. Any greedy or RIS-based IM algorithm can be embedded in MOIM, retaining the same features and drawbacks. In our experiments we have examined the results of top performing IM algorithms (e.g., [17, 33]), showing them all to be ill-suited for the Multi-Objective IM problem.

An extension of IM, which we also examined in our experiments, is *targeted IM*, where the goal is to maximize the influence over a target group of users [6, 9, 26]. As demonstrated, this extension as well is ill-suited for the Multi-Objective IM problem, as maximizing the influence over one group of users may come at the cost of influence decrease for other groups. Therefore, unlike our solutions, it does not provide theoretical guarantees for the influence over each emphasized group separately.

Multi-Objective optimization problems (also known as Pareto optimization) involve several (possibly conflicting) objectives, which are required to be optimized simultaneously. Such problems have been studied in numerous fields, including economics [27], finance [35], social-network analysis [19] and engineering [14]. A classic approach to tackle such problems, which was adopted by targeted IM algorithms [26, 31]), is the weighted-sum method (e.g., [21]), which scalarizes the objectives into a single objective, by assigning to each objective a user-defined weight (which is chosen in proportion to its relative importance). In the IM setting, the relative weights of users in the overall influence sum are altered in accordance with a context-based function

[6, 9, 26]. The main disadvantage of this method is the difficultly in setting the weights obtaining the desired trade-off between the objectives. Indeed, as we show in our experiments, adopting the weighted-sum approach for our context requires an exploration for the optimal weights which strike the desired balance. Hence, this solution results in poor performance.

An alternative, more direct approach to multi-objective optimization problems is the *constraints method* (e.g., [12]), that transforms all except one objectives into constraints, optimizing the remaining objective subject to these constraints. A typical challenge when applying this method is that the constraints have to be chosen within the minimum/maximum values of the individual objectives (which are generally unknown). Our solution follows this approach, which enables the user to prioritize her objectives and provides lower bound guarantees for all of them. As mentioned, to assist the user in choosing the minimum values of the objectives, IM-Balanced indicates to the user the range of possible constraints per objective.

We have discussed on the connection between Multi-Objective IM and the RSOS problem [24]. The authors of [8] provided an optimal $(1-\frac{1}{e})$-approximation algorithm for RSOS (assuming that number of objectives is $m = \Omega(k)$), which runs in $O(n^8)$. Udwani [37] has recently introduced two more efficient algorithms. The first is an optimal $(1 - \frac{1}{e})$-approximation algorithm, which runs in $\tilde{O}(mn^8)$. The second is a more efficient algorithm which runs in $O(n \log m \log n)$, yet achieves only a $(1 - \frac{1}{e})^2$ approximation. More recently, the authors of [36] remedy this gap by providing an optimal $(1 - \frac{1}{e})$-approximation algorithm, whose runtime is comparable to the second algorithm of Udwani. As mentioned, we have included this algorithm in the experimental study, showing that, unlike our algorithms, it fails to process large networks.

8 CONCLUSION AND FUTURE WORK

We have presented the IM-Balanced system, which employs Multi-Objective IM, a refined notion of the IM problem, handling multiple objectives. We motivate the practical relevance of this problem, and propose two algorithms: MOIM and RMOIM. IM-Balanced employs RMOIM for social networks including up to 20M users and links, and MOIM for larger networks. Our experimental study demonstrates the advantages of our algorithms in multiple real-life scenarios, compared to alternative approaches.

We are currently pursuing complementary Multi-Objective IM definitions, e.g., definitions aiming to maximize the *ratio* of different cover cardinalities, inspired by recent work on fairness-aware IM [15, 36]. We identify several interesting directions for future research, which include confirming the tightness of MOIM, and identifying other optimum values for Multi-Objective IM.

Acknowledgment. This work has been partially funded by the Israel Science Foundation, the Binational US-Israel Science Foundation, Tel Aviv University Data Science center, and eBay Israel.

REFERENCES

[1] 2019. Code for the paper: Group-Fairness in Influence Maximization. https://github.com/bwilder0/fair_influmax_code_release.
[2] 2020. Gurobi LP solver. http://www.gurobi.com/.
[3] 2020. Technical Report. https://bit.ly/2V9nXIS.
[4] AMiner 2018. AMiner datasets. https://aminer.org/data-sna.
[5] Akhil Arora, Sainyam Galhotra, and Sayan Ranu. 2017. Debunking the Myths of Influence Maximization: An In-Depth Benchmarking Study. In *SIGMOD*.
[6] Cigdem Aslay, Nicola Barbieri, Francesco Bonchi, and Ricardo A Baeza-Yates. 2014. Online Topic-aware Influence Maximization Queries.. In *EDBT*.
[7] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing Social Influence in Nearly Optimal Time. In *SODA*. Society for Industrial and Applied Mathematics.
[8] Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. 2010. Dependent randomized rounding via exchange properties of combinatorial structures. In *FOCS*. IEEE.
[9] Shuo Chen, Ju Fan, Guoliang Li, Jianhua Feng, Kian-lee Tan, and Jinhui Tang. 2015. Online Topic-aware Influence Maximization. *PVLDB* (2015).
[10] Wei Chen. 2018. An issue in the martingale analysis of the influence maximization algorithm IMM. In *International Conference on Computational Social Networks*. Springer.
[11] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient Influence Maximization in Social Networks. In *SIGKDD*.
[12] Kenneth Chircop and David Zammit-Mangion. 2013. On-constraint based methods for the generation of Pareto frontiers. *Journal of Mechanics Engineering and Automation* (2013).
[13] Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F. Werneck. 2014. Sketch-based Influence Maximization and Computation: Scaling Up with Guarantees. In *CIKM*. ACM.
[14] Kalyanmoy Deb and Rituparna Datta. 2012. Hybrid evolutionary multi-objective optimization and analysis of machining operations. *Engineering Optimization* (2012).
[15] Golnoosh Farnad, Behrouz Babaki, and Michel Gendreau. 2020. A Unifying Framework for Fairness-Aware Influence Maximization. In *WWW*.
[16] Sahy Gershtein, Tova Milo, Brit Youngmann, and Gal Zeevi. 2018. IM Balanced: Influence Maximization Under Balance Constraints. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM.
[17] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. 2011. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*. ACM.
[18] Amit Goyal, Wei Lu, and Laks V.S. Lakshmanan. 2011. CELF++: Optimizing the Greedy Algorithm for Influence Maximization in Social Networks. In *WWW*. ACM.
[19] R Chulaka Gunasekara, Kishan Mehrotra, and Chilukuri K Mohan. 2014. Multi-objective optimization to identify key players in social networks. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*. IEEE.
[20] Keke Huang, Sibo Wang, Glenn Bevilacqua, Xiaokui Xiao, and Laks V. S. Lakshmanan. 2017. Revisiting the Stop-and-stare Algorithms for Influence Maximization. *PVLDB* (2017).
[21] C-L Hwang and Abu Syed Md Masud. 2012. *Multiple objective decision making—methods and applications: a state-of-the-art survey*. Vol. 164. Springer Science & Business Media.
[22] Narendra Karmarkar. 1984. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM.
[23] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the Spread of Influence Through a Social Network. In *KDD*.
[24] Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. 2008. Robust submodular observation selection. *Journal of Machine Learning Research* (2008).
[25] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.
[26] Yuchen Li, Dongxiang Zhang, and Kian-Lee Tan. 2015. Real-time Targeted Influence Maximization for Online Advertisements. *PVLDB* (2015).
[27] SJ Mardle, Sean Pascoe, and Mehrdad Tamiz. 2000. An investigation of genetic algorithms for the optimization of multi-objective fisheries bioeconomic models. *International Transactions in Operational Research* (2000).
[28] Hung T Nguyen, My T Thai, and Thang N Dinh. 2016. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *SIGMOD*. ACM.
[29] Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-Ichi Kawarabayashi. 2014. Fast and Accurate Influence Maximization on Large Networks with Pruned Monte-Carlo Simulations. In *AAAI*. AAAI Press.
[30] Prabhakar Raghavan and Clark D Tompson. 1987. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica* (1987).
[31] Chonggang Song, Wynne Hsu, and Mong Li Lee. 2016. Targeted Influence Maximization in Social Networks. In *CIKM*. ACM.
[32] David Steurer. 2014. Max Coverage—Randomized LP Rounding. http://www.cs.cornell.edu/courses/cs4820/2014sp/notes/maxcoverage.pdf.
[33] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence Maximization in Near-Linear Time: A Martingale Approach. In *SIGMOD*.
[34] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *SIGMOD*. ACM.
[35] Ma Guadalupe Castillo Tapia and Carlos A Coello Coello. 2007. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *2007 IEEE Congress on Evolutionary Computation*. IEEE.
[36] Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. 2019. Group-fairness in influence maximization. *arXiv preprint arXiv:1903.00967* (2019).
[37] Rajan Udwani. 2018. Multi-objective maximization of monotone submodular functions with cardinality constraint. In *NeurIPS*.
[38] Vijay V Vazirani. 2013. *Approximation algorithms*. Springer Science & Business Media.