Multi-Objective Influence Maximization

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 subpopulations 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 [24]. IM has a natural application in viral marketing, where companies promote their brands through the word-of-mouth propagation. This has motivated extensive research [6, 27], emphasizing the development of scalable algorithms [21, 36].

The classical 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 the *emphasized groups*, and assume the existence of boolean functions over (multiple) user profile attributes, which identify these groups. We introduce the Multi-Objective IM problem, which refines the classical 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 in Section 2, 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.

To illustrate the problem that we study in this paper, consider 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 anti-vaccination users that are being reached. Here g_1 consists of all users, while g_2 is the group of anti-vaccination users. A standard IM algorithm will maximize the overall influence (i.e., 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 aim to maximize the influence over a particular group (here - g_2). But if this (possibly small) group is somewhat socially disconnected from the general crowd, the message may not reach a sufficient number of users overall (i.e., g_1).

Example 1.2. Consider a tech company interested in running a recruitment campaign over a social network, with the goal of hiring both engineers (g_1) and researchers (g_2). For the sake of this example 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 these 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 classical 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. [27, 32]). A main difficulty in applying this approach is assigning the weights that achieve a desired influence balance [22]. 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. Alternately, 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.

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 [25]. In contrast to our settings where the user can specify for each group the *fraction* of the optimal influence that they wish to retain, in RSOS, only explicit values can be used. As in the weighted-sum approach mentioned above, it is not clear what values should be used to obtain desired results, and the exploration for the optimal values is expensive. Furthermore, existing RSOS algorithms are not scalable for large graphs [40] (see Sections 5 6 for a detailed theoretical and experimental comparison resp.).

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 exceeds 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).

Multi-objective MC. State-of-the-art IM algorithms [21, 36] are based on a reduction from IM to the classical *Maximum Coverage* (MC) problem [6]. Analogously, we define the *Multi-objective MC* problem, and establish the corresponding relation between Multi-Objective IM and Multi-objective MC. Thus, we state that, as an independent contribution, all our results hold for Multi-objective MC as well.

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 (not even a non-optimal one) can be found in PTIME. Moreover, we prove that the $(1 - \frac{1}{e})$ approximation factor for g_1, which is optimal in the standard (unconstrained) IM problem, is unattainable in our setting. We show however that such an optimal approximation factor can nevertheless be achieved if the constraint imposed on g_2 is also somewhat relaxed and 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. Specifically, MOIM runs two single-objective targeted IM algorithms, each focusing on a different group, and combines their outputs. MOIM 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 ones 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 MC [41], and thus its performance becomes polynomial, rather than near-linear as in MOIM

(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 approximates it, and provide worst case guarantees for this as well.

Connecting to the RSOS problem. As mentioned, we examine the connection between Multi-Objective IM and the RSOS problems. We prove that the two problems are equally hard, and that any algorithm solving RSOS, could in principle also solve Multi-Objective IM, yielding the same guarantees as in RMOIM. However, to do so, one would need to examine $O(log(n))$ instances of RSOS (where n is the number of nodes), which yields significant overhead performance-wise (Section 5).

Experimental study. We experimentally compare our algorithms to (targeted) IM algorithm as well as to 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. We further show that, as opposed to our algorithms, top performing RSOS algorithms are impractical for large-scale networks (even without the $log(n)$ multiplicative overhead). We then examine the runtime performance of our algorithms, showing 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 recently presented in [17]. 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.

Last, related work is presented in Section 7 and we conclude in Section 8.

2 PRELIMINARIES

This section presents the formal definition of standard IM, reviews existing IM algorithms, and introduces the auxiliary problem of *Group-Oriented IM*.

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 $0 \leq W(u, v) \leq 1$, which models the probability that a node u will influence its neighbor v. Given

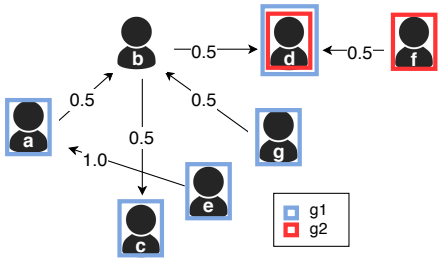


Figure 1: Example social network with two (user-defined) emphasized groups of users.

a function $I(\cdot)$ dictating how influence is propagated in the network, the IM problem [24] is defined as follows.

Definition 2.1 (IM [24]). 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 seed node v in a seed set T is influenced by itself, and hence, by definition, we have: $T \subseteq I(T)$. 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 [6, 13, 21], 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, but 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 as well. 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 $t-1$ remain covered, and we cover any node v for which the total weight of its covered neighbors is at least θ_v. To illustrate the definition, 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 (as well as in this example), the threshold for each node was sampled uniformly at random from $[0, 1]$.

2.2 Existing IM algorithms

Selecting the optimal seed set is NP-hard [24], with Feige [16] proving inapproximability beyond a factor of $(1 - \frac{1}{e})$. The subsequent work on IM following [24], which had already achieved the optimal approximation, has focused on scalability [6, 13, 30]. In what follows, whenever we refer to an IM algorithm, we in fact refer to a probabilistic algorithm, which, given the parameters $0 \leq \varepsilon, \delta \leq 1$, 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, with [36] achieving nearly optimal time complexity 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 [41]. For completeness of this paper, we next provide a formal definition of this problem.

Definition 2.2 (MC [41]). 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.

This well-known problem was proven to be NP-hard, but has a simple greedy approximation procedure [41], achieving an optimal approximation factor of $(1 - \frac{1}{e})$.

The RIS framework consists of two phases: First, θ nodes are sampled independently and 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 sources of influence for u. Next, each node is associated with a set whose members are the *RR* sets containing it, then, using the greedy algorithm for MC, 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 $k = 2, \theta = 4$ 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.

An important observation is that the second phase of RIS, i.e., solving MC, can also be achieved using Linear Programming (LP), yielding the same approximation factor. However, in terms of time complexity, IM algorithms are nearly linear [36], compared to PTIME LP solvers [23].

2.3 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 g_1 and g_2. Continuing with Example 1.2, the smaller researchers group may be the group of users with red border, and g_2 is the engineers group (users with blue border). In this example the 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. We denote by $I_g(S)$ the expected number of members of g which are covered by S, referred to as the g-cover. To define our problem we first present the auxiliary *Group-Oriented IM* problem, denote as IM_g, which instead of maximizing $I(\cdot)$, maximizes $I_g(\cdot)$.

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

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.

We can easily show that the hardness result of IM also applies to this variant, following a straightforward reduction from IM, where $g = V$. In Section 4.1, we explain how a given (RIS-based) IM algorithm can be adapted to its group-oriented version, retaining all its theoretical properties.

We note that this variant can be seen as a special case of the *Targeted IM* problem (e.g., [27]), 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. In Section 4.3 we extend our results to multiple emphasized groups.

3.1 Multi-Objective IM

As explained in the Introduction, it is often desirable to assure the obtained solution will ensure sufficient cover of two emphasized groups. To this end, we add a constraint on the IM_g 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.

Formally, given two emphasized groups, denoted as $g_1, g_2 \subseteq V$, a threshold parameter $0 \leq t \leq 1$ and a number k, find a k-size seed-set 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:

$$O^* = argmax_{\{T:|T|=k, I_{g_2}(T) \geq t \cdot I_{g_2}(O_{g_2})\}} I_{g_1}(T) \quad (1)$$

where $I_{g_1}(T)$ and $I_{g_2}(T)$ denote, resp., the expected size of the g_1 and g_2 covers by T, and O_{g_2} denotes the optimal k-size solution for g_2 (Def. 2.4). For simplicity of presentation, we first present our results adapted to Equation (1), termed in the sequel, for brevity, as the *Multi-Objective IM* problem.

Throughout the paper, pertaining to Equation (1), 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.

Example 3.1. Consider again Figure 1 and assume that $k = 2$. For Equation (1) with $t = 0.1$ 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 and equals $I_{g_1}(S) = 4$. 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 (by definition) 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 2.2. We define the *Multi-Objective MC* problem, analogous to Multi-Objective IM, which will serve us for deriving our lower bound as well as for devising the RMOIM algorithm.

Definition 3.2 (Multi-Objective MC). Given subsets $S_1, ..., S_m$ of elements from $U = \{u_1, ..., u_n\}$, 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 collections of k sets from $S_1, ...S_m$ satisfying the constraint, the one covering a maximal number of elements belong to g_1.

3.2 The constraint threshold

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 (targeted) IM [24], merely finding a single (not necessarily optimal) k-size seed set satisfying the constraint cannot be done in PTIME.

Corollary 3.3. *A k-size seed set satisfying the constraint can always be found in PTIME only if $0 \leq t \leq (1 - \frac{1}{e})$.*

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.

3.3 Approximation lower bound

We first formally define the solution space, then present an approximation 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 $\beta = 1$ the solution strictly satisfies the constraint. To accommodate practical algorithms we consider, as in standard IM, randomized algorithms that may add an error margin $0 \leq \varepsilon \leq 1$ to the approximation factors, while requiring the stated factors to hold with probability $\geq (1 - \delta)$ for $0 \leq \delta \leq 1$.

Definition 3.4 ((α, β)-approximation). Given $0 \leq \varepsilon, \delta \leq 1$, 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 $\geq 1 - \delta$: $I_{g_2}(S) \geq (\beta - \varepsilon) \cdot t \cdot I_{g_2}(O_{g_2})$ and $I_{g_1}(S) \geq (\alpha - \varepsilon) \cdot I_{g_1}(O^*)$, where O^* is the optimal solution w.r.t. Equation (1).

We assume ε and δ are implicitly provided as part of the input. Our algorithms take these requirements into account, however, for simplicity of presentation, we omit discussions of these parameters in our analysis 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 in (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. We note that one immediate such optimum is $(1 - \frac{1}{e}, 1)$, which follows directly from the hardness result of IM [24]. However, as we prove next, there exists no PTIME algorithm which can achieve this bound.

Hardness of approximation. As mentioned, the optimal objective approximation of Multi-Objective IM is $\alpha = 1 - \frac{1}{e}$. We next prove that in order to achieve this optimal α value, a relaxation of the constraint is necessary. More concretely, we

prove that Multi-Objective IM has no PTIME algorithm with approximation guarantees (even in expectation) dominant over $(1-\frac{1}{e}, 1-\frac{1}{e})$, using a novel reduction from MC (see Appendix C). 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).*

4 ALGORITHMS

As we proved in the previous section, the approximation factor we can achieve for the objective depends on how strictly the constraint is preserved. We, therefore, provide two complementary algorithms for Multi-Objective IM. Our first algorithm, referred to as *the Multi-Objective IM (MOIM) algorithm*, finds a seed-set that strictly satisfies the constraint, at the cost of influence decrease for the objective. The key advantage of MOIM is that it achieves near linear-time complexity, which, as we show, is critical for scaling successfully to massive social networks. To get a tighter approximation ratio for the objective, our second algorithm, referred to as *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, rather than near-linear as in MOIM. We conclude with a generalization of our results to multiple emphasized groups and a discussion on alternative problem definitions.

4.1 The MOIM algorithm

We next detail our modification of a given IM algorithm, followed by the full algorithm scheme.

The IM_g algorithm. Given an IM algorithm $\mathcal{A}$ and an emphasized group $g \subseteq V$, 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. We 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 [16]. The key observation, following the same reasoning as in [6], is that for a given seed-set S, its intersection with an *RR* set generated by a uniformly chosen node from g, is an unbiased estimator of $\frac{I_g(S)}{|g|}$, where $|g|$ is the size of g.

Proposition 4.1. *For any RIS-based algorithm $\mathcal{A}$, let S denote the seed set obtained by its IM_g counterpart, $\mathcal{A}_g$, then:* $(1-\frac{1}{e}) \cdot I_g(O_g) \leq I_g(S) \leq I_g(O_g)$.

We note that a method of weighted RIS sampling for solving the Targeted IM problem was presented in [27]. 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 author of [27] has focused in cases where there is only one emphasized group. As we demonstrated 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 algorithm $\mathcal{A}_{g_1}$ on the residual network until enough seeds are gathered.
7: **end if**
8: **return** S

The full MOIM algorithm is depicted in Algorithm 1. Specifically, 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). Note that this can only improve the accuracy guarantees.

Example 4.2. Consider again Figure 1. Let $k = 2$ and $\mathcal{A}$ be an IM algorithm. 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 the 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., both optimal for each algorithm, combining for a seed set S s.t $I_{g_1}(S) = 3$ and $I_{g_2}(S) = 1.75$. Note that this approximated solution comes close to both O_{g_1} and O_{g_2}, in terms of g_1/g_2 cover size, resp.

Theorem 4.3. *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.*

The time complexity of MOIM depends only on that of its input IM algorithm $\mathcal{A}$ (we run $\mathcal{A}$ twice), which is assumed to be near optimal [36].

4.2 The RMOIM algorithm

We begin by describing a theoretical algorithm which, given the optimal size of the g_2-cover, $I_{g_2}(O_{g_2})$, exactly matches our hardness bound. We then extend this algorithm to handle 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 4.4. *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 have previously described the reduction from IM to MC suggested in [6], which is 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 $Y = |g_2 \setminus g_1|, Z = |g_1 \setminus g_2|, W = |g_1 \cap g_2|$:

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, and y_i - for covering element from $g_2 \setminus g_1$, z_i - for covering element from $g_1 \setminus g_2$ and w_i - for covering elements in the intersection of the groups)

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 $g_2 \setminus g_1, g_1 \setminus g_2$ and $g_1 \cap g_2$, resp., when sampling the *RR* sets.

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 to this LP by using any LP solver [23], then apply the following randomized rounding procedure [31]: (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 [33], 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 4.4.

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 (as described in Section 4.1) 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 RIS-based 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 the given 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. $I_g(S) = (1+\lambda) \cdot (1-\frac{1}{e}) \cdot I_g(O_g)$.

Theorem 4.5. *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 network size and the number of seeds [23].

Algorithm 2 The RMOIM algorithm.

1: **Input:** A network G; $k \in [n]$; $t \leq 1-\frac{1}{e}$; 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 algorithm $\mathcal{A}_{g_2}$ 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($\mathcal{I}$) using the LP-solver, and output the values for the x_i variables.
7: $S \leftarrow$ Run the randomized rounding procedure on $\vec{X}$.
8: **return** S

Setting the constraint threshold parameter. We conclude with a discussion on how IM Balanced assists users to set the constraint threshold. To allow users to make an informed decision for the value the parameter t imposed on g_2, our system runs an IM_{g_2} algorithm (as explained in Section 4.1), providing an estimation of the optimal achievable cover over this group, displays this value to the user. As mentioned in Section 3, a k-size seed-set satisfying the constraint can only be found if $0 \leq t \leq (1-\frac{1}{e})$. Therefore, to avoid the case

where IM Balanced outputs no solution, in case the user provided a threshold that do not satisfy this restriction, the system may alert the user and suggest to lower the threshold.

4.3 Extensions

We present an extension of our results to multiple groups, then briefly discuss on alternative problem definitions.

Multiple Emphasized Groups. The Multi-Objective IM problem naturally extends to multiple groups. Given m emphasized groups, the user can, in a similar manner, 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 $0 \leq \sum_i t_i \leq (1 - \frac{1}{e})$ (Corollary 3.3). Regrading our hardness bound, we prove that in PTIME, one cannot attain an approximation factor dominant over $(\underbrace{1 - \frac{1}{e}, \ldots, 1 - \frac{1}{e}}_{m \text{ times}})$ (see full details in Appendix C). Moreover, the generalized random algorithm described in Section 4.2 matches our lower bound for multiple groups.

Both our algorithms can be generalized to solve the multiple groups scenario as well. Specifically, 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 $(1 - \frac{1}{e \cdot (1 - \sum_i t_i)}, \underbrace{1, \ldots, 1}_{m-1 \text{ times}})$-approximation to the Multi-Objective IM problem with m emphasized groups. The time complexity of MOIM depends only on that of its input IM algorithm (as we can run all m IM_{g_i} algorithms simultaneously), which is assumed to be near optimal [36].

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. The time complexity of RMOIM is dominated by its input LP solver, whose complexity is polynomial [23].

Alternative problem definitions. One can in principle consider an alternative variant of Multi-Objective IM where the user specifies an explicit value constraint (rather than one that relates to the optimal possible value). For instance, continuing with Example 1.2 from the Introduction, the user may request to maximize the cover over engineers, subject to a constraint requiring that at least 1K researchers are influenced. It is interesting to note that both our algorithms naturally support such a variant as well. Specifically, in MOIM, we can run an IM_{g_2} algorithm until it exceeds the constraint value, and with the remaining number of 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 in the LP is known (and there is no need to estimate it). Therefore, here, RMOIM is optimal as it matches our lower bound. We focus in this paper 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 (we illustrate that via an example in Appendix D.) Therefore, such definition is ill-suited to our motivation (as presented in the Introduction) 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.

Last, note that in our analysis so far the user imposes constraints on all but one group. Our results also support the scenario where the user imposes constraints all emphasized groups. We deffer the discussion on this definition to Appendix D.

5 CONNECTION TO THE RSOS PROBLEM

As mentioned, Multi-Objective IM is closely related to the RSOS problem [25]. We next prove that the two problems are equally hard, and that any algorithm solving RSOS, could in principle also solve Multi-Objective IM. However, to do so one would need to examine $O(log(n))$ instances of RSOS (which yields significant overhead performance-wise).

Given m monotone submodular functions $f_i(\cdot), i \in \{1, .., m\}$ the RSOS problem is defined as follows[1]: We are given a target value V_i (positive real) with each function f_i, and the goal 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. An *alpha*-approximation S means that $\forall i : f_i(S) \geq \alpha \cdot V_i$.

[1]We discuss here the formulation presented in [8], which is equivalent to the classic definition of [25].

To simply the presentation, we focus here on the two groups scenario, and deffer the results regarding multiple groups to Appendix C. We first 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.*

Intuitively, in our proof we initially focus on the value constraint variant of Multi-Objective IM (discussed in Section 4.3), where the constraint threshold is an explicit value. We then extend the result to our standard implicit size constraint definition, by reducing an instance of RSOS to $O(n^2)$ instances of Multi-Objective IM, each corresponding to a different guess of the threshold parameter t.

We next 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.*

To prove Theorem 5.1 holds, here 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 optimal objective value $I_{g_1}(O^*)$. $I_{g_2}(O_{g_2})$ may be estimated, as we do for 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 in RMOIM. But to estimate $I_{g_1}(O^*)$, we can try all $O(n)$ possible guesses for the constrained optimal value $I_{g_1}(O^*)$. We can do this more efficiently by examining only $O(log(n))$ guesses, exponentially growing by a constant factor $delta > 1$, at the price of reducing the approximation guarantee by a multiplicative $1 - 1/delta$ factor, which we can make as small we want. Note that this increases the time complexity of an RSOS algorithm (adapted to solve Multi-Objective IM) by an $O(log(n))$ factor.

The state-of-the-art algorithm for RSOS, which achieves the optimal $(1 - \frac{1}{e})$-approximation, have been recently introduced in [39]. As we show in our experiments, as opposed to our algorithms, this algorithm can only process small networks (even without the $log(n)$ multiplicative overhead).

6 EXPERIMENTAL STUDY

We have conducted an extensive set of experiments to evaluate (1) The quality of the 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.

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	-

Table 1: Datasets.

6.1 Experimental setup

IM Balanced is implemented in Python 2.7 and its UI is implemented in HTML5/CSS3 (see details regarding the UI in [17]. We use as the input IM algorithm, for both of our algorithms, IMM[2] [36], a top performing IM algorithm. We solve the LP in RMOIM using Gurobi LP solver [2]. We conducted all experiments on a Linux server with a 2.1GHz CPU and 96GB memory.

Datasets. We have focused on social networks which include user profile properties, to allow for characterizing the emphasized groups. We have examined 6 commonly used graph datasets: Facebook, DBLP, Pokec, Weibo-Net, Twitter and Google+ (extracted from [3, 26]). For space constraints, we omit the results over Twitter and Google+, as they were similar to the those obtained over the other 4 datasets, which are depicted in Table 1. To further examine our algorithms scalability, we considered two additional large-scale datasets: YouTube and LiveJournal [26]. 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 [29, 37], 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 [4]).

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 users in g, while targeted IM algorithms showed that a different choice of seed-set significantly increase the expected g-cover size. Interestingly, our experiments indicate that all analyzed

[2]We used the corrected version described in [10].

datasets include several such groups. For example, female Indian researchers in the DBLP network and females over the age of 50 in Pokec, are typically neglected by standard IM algorithms. Additional examples are provided in Appendix E. For the YouTube and LiveJornal datasets 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 given 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 Appendix E). ***Scenario II***. Next we consider multiple-groups, to demonstrate the effect of multiple objectives on the performance. Specifically, 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 tested the performance for other numbers of emphasized groups and report that all results have shown similar trends. Note that in real-life scenarios, the number of emphasized groups is typically small [27, 39] 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 compare our algorithms, MOIM and RMOIM, with the following baselines.

Standard IM algorithms. We have examined the results of *IMM* [36] and SSA [29], top preforming RIS-based algorithms, as well as SKIM [13] and Celf++ [18], 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 [27]) which maximizes exclusively the cover of a given emphasized group g (as explained in Section 4.1). In scenario *II* we have defined the target group to be the union of all emphasized groups.

Weighted IM. An alternative approach is to assign different weights to individual users, reflecting their relevance to the objectives. The authors of [27] 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 the this method, where we apply a (multi-dimensional) binary search to find the optimal weights[3]. We examined the results while substituting in *WIMM* 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 have included the RSOS algorithm of [39] (used to solve Multi-Objective IM) as a baseline. Additionally, the authors of [39] have studied the problem of fair resource allocation in IM, and proposed two fairness concepts: the first MaxMin, which maximizes the minimum fraction of users within each group that are influenced. The second is 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[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 $\varepsilon = 0.1$. 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 $t_i = 0.25 \cdot (1 - \frac{1}{e}), \forall i \in 1, \ldots, 4$. We also use, as a default setting, the LT model (when setting uniformly random threshold for every node) and discuss the minor changes in the results, when using the IC model instead in Section 6.3. 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 axises represent, resp., the g_1 and g_2 influences, and red lines represent the estimated constraint thresholds. Note that 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

[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 the corresponding groups.

[5]In [39], the largest examined network included 500 nodes.

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 examined cases (which extend beyond the ones we present here) 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 have influenced almost the same number of g_1 users (601 and 599, resp.), MOIM succeeded in covering more g_2 users than *WIMM* (19 vs. 12 for MOIM and *WIMM*, resp.). Observe that when using the optimal weights for DBLP over Pokec for *WIMM*, result in not satisfying the constraint. We note that the exploration of *WIMM* for optimal weights significantly increases its runtime, and therefore it is 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 approximated 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 overall influence of RMOIM (i.e., g_1 influence) 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. As RMOIM runtime is polynomial, it is incapable of processing massive networks like Weibo-Net (out of memory).

Not surprisingly, as *RSOS* and RMOIM both ensure a $(1 - \frac{1}{e})$-approximation of the constraint and the objective, they results were similar. Nonetheless, as opposed to RMOIM, all RSOS-based baselines (i.e., *RSOS*, *MAXMIN* and *DC*) where incapable of even processing medium-size networks such as DBLP (they have all exceeded our time cutoff). Recall that *MAXMIN* aims to maximize the minimum influence over the emphasized groups, and therefore in this scenario 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.

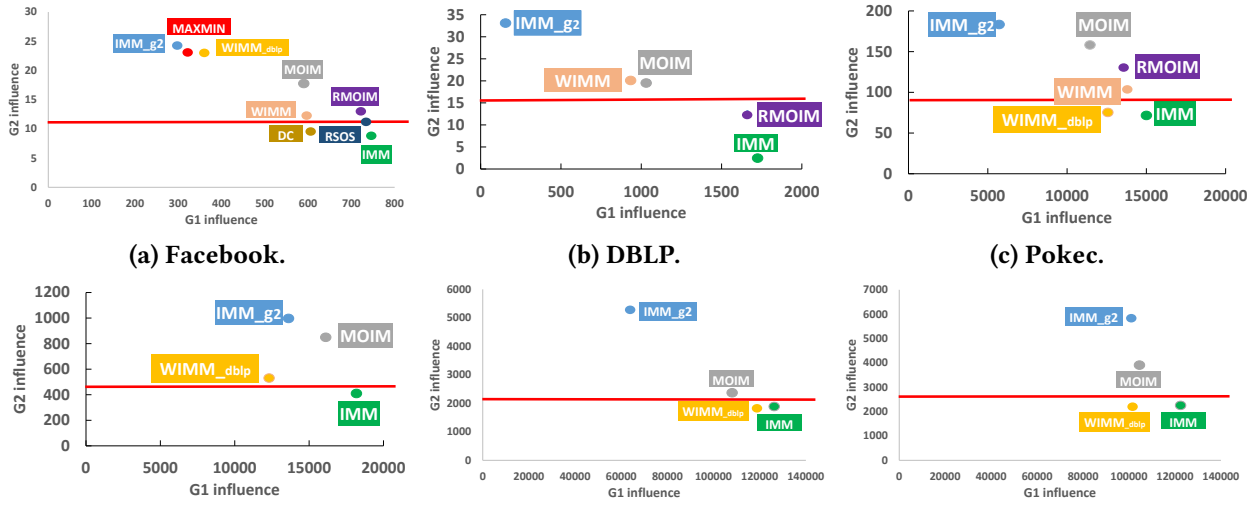
Observe that the single objective algorithms were either far from satisfying the constraint (*IMM*) or covered significantly less g_1 users (IMM_{g_2}). For example, over DBLP, *IMM* covered only 2 g_2 users and 1,712 users in total (i.e., g_1 users), whereas IMM_{g_2} covered 33 g_2 users, and less than 155 in total. Contrarily, MOIM and RMOIM covered 20 and 13 g_2 users, resp., and covered each more than 1,050 users in total. This demonstrate 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 managed to influence 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 (i.e., lower objective value). This demonstrates that existing single-objective IM algorithms do not ensure the desired balance between the objectives. We note that over YouTube and LiveJournal 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 subpopulations that are typically not covered by standard IM algorithms (which is mostly not the case in randomly generated 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. Note that a desirable solution should be above (or near) the red lines for the constrained $g_1, \ldots, g_4$ groups (i.e., satisfying the constraints), 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).

Observe that 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 has 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 - the targeted IM algorithm 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 objective value (i.e., 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 both Facebook and DBLP RMOIM covered the greatest number of g_5 users.

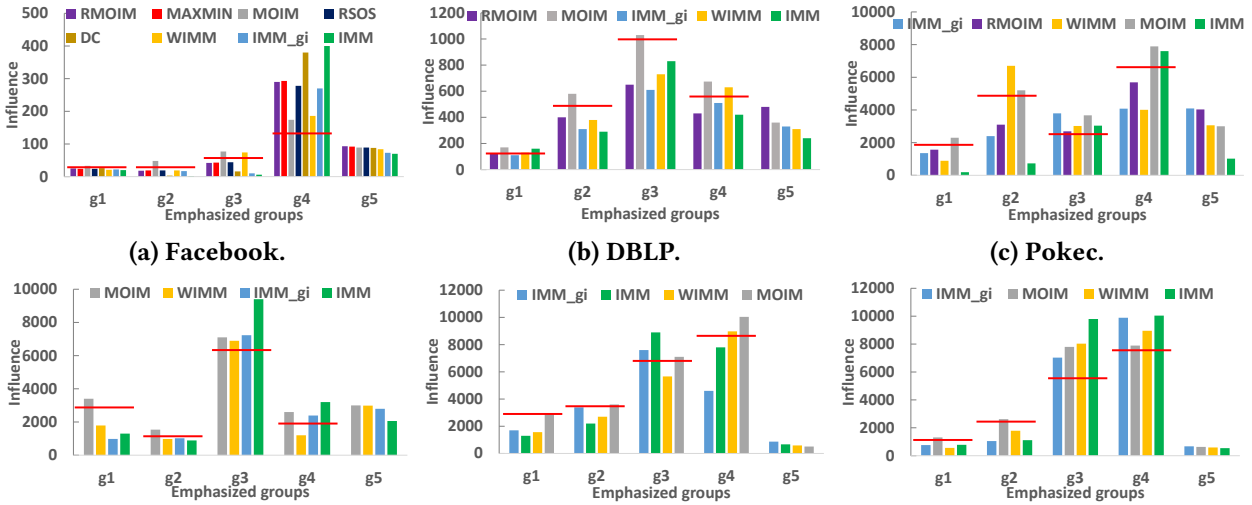
Here again, all RSOS baselines could only process the small Facebook network (they exceeded our time cutoff in other datasets), and, as expected, the results of *RSOS* and RMOIM over Facebook (Figure 3 (a)) were similar. Observe that here *MAXMIN* also behaves similarly to RMOIM, however, as was demonstrated above, in other scenarios it may behave



(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.



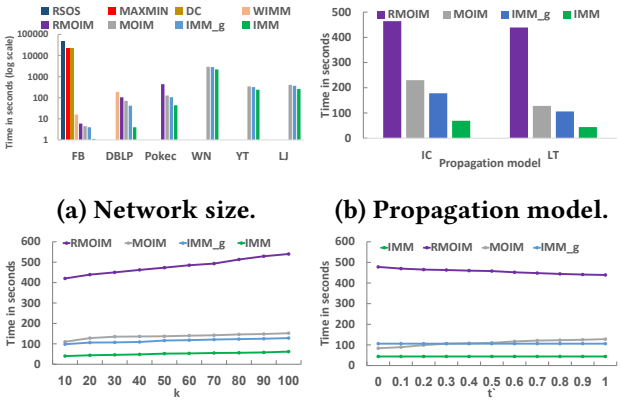
(a) Facebook. (b) DBLP. (c) Pokec.

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

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

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, in all cases it did not satisfy them.

As opposed to the binary case scenario where the objective was to maximize the overall influence, in this scenario, the standard IM algorithm, *IMM*, has no advantage over the competitors. Indeed, in all except one of the examined cases (i.e., YouTube), *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 (as was demonstrated in Example 2.5). 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.



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

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

Figure 4: Averaged execution times (for scenario *II*).

Note that a desirable behavior of a Multi-Objective IM algorithm is as follows. As k increases, all emphasized groups influences should increase as well. As the constraint threshold is elevated for one group, the influence over it should increase, possibly at the cost of reducing the objective value. Naturally, our algorithms, *WIMM*, and *RSOS* take into account the parameter t and therefore exhibit the desired behavior, while other competitors are indifferent to it. Additional experiments demonstrating this are provided in Appendix E.

6.3 Performance Evaluation

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

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 [36], 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, as we shall see, 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 4(a). Naturally, the larger the network, the longer it takes to compute the solution. Indeed, 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 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 4(b). As reported in [4], 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 [4], besides *IMM*, multiple top performing IM algorithms are not robust across different propagation models (e.g., [19], [37]). 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 4(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. We note that this behavior of MOIM is a consequence of employing *IMM*, and therefore using an alternative IM algorithm (e.g., [18]) 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 4(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 $t' \in [0.1, 0.2, \ldots, 1]$ (where all t_i values are equal). Note that this parameter only affects the running times of our algorithms, and in particular, in MOIM it dictates the required seed-set size for the procedures it employs. Observe that when all $t_i = 0$ it 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 [24], the first to formulate the IM problem, has motivated extensive research [4, 13], which can be classified into three main approaches: (i) The greedy framework [19, 30], proposed in [24], which iteratively adds nodes to the seed-set s.t. each addition maximizes the expected marginal influence gain; (ii) The RIS framework [6], where, while retaining optimal accuracy, running times were gradually improved, resulting in highly scalable algorithms of near-optimal time complexity [21, 29, 36]; (iii) In cases where scalability is preferred over accuracy, there are heuristic algorithms that have been shown to perform well in practice, most notably [11], despite not having theoretical guarantees for the quality of the returned output. As explained in Section 4.1, any given 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 (greedy and RIS based) IM algorithms (e.g., [18, 36]), showing them all to be ill-suited for the Multi-Objective IM problem.

An extension of IM, which we also have examined in our experiments, is *targeted IM* algorithms, which aim to maximize the influence over a (single) target group of users [5, 9, 27]. As demonstrated, this extension as well is ill-suited for the Multi-Objective IM problem, as maximizing the influence over one target group of users may come at the cost of influence decrease for another target group. 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 [28], finance [38], social-network analysis [20] and engineering [14]. A classical approach to tackle multi-objective optimization problems, which was adopted by targeted IM algorithms [27, 32]), is the weighted-sum method (e.g., [22]), 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 [5, 9, 27]. As mentioned, the main disadvantage of this method is the difficultly in setting the weights obtaining the desired trade-off between the objectives. Indeed, as we demonstrated in our experimental study, 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 end-user in choosing the minimum values of the objectives, IM Balanced's UI indicates to the user the range of possible constraints per objective.

We have discussed in Section 5 on the connection between Multi-Objective IM and the RSOS problem [25]. The authors of [8] have 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 [40] 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 [39] 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-scale networks.

8 CONCLUSION AND FUTURE WORK

We have presented the IM Balanced system, which employs Multi-Objective IM, a refined notion of the classic 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 extensive 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 algorithmic discrimination [7, 15, 34, 35, 39]. We identify several interesting directions for future research, which include confirming the tightness of MOIM, and identifying other optimum values for Multi-Objective IM.

REFERENCES

[1] Code for the paper: Group-fairness in influence maximization. https://github.com/bwilder0/fair_influmax_code_release, 2019.

[2] Gurobi lp solver. http://www.gurobi.com/, 2019.

[3] Aminer datasets, 2018. https://aminer.org/data-sna.

[4] A. Arora, S. Galhotra, and S. Ranu. Debunking the myths of influence maximization: An in-depth benchmarking study. In *SIGMOD*, 2017.

[5] C. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates. Online topic-aware influence maximization queries. In *EDBT*, 2014.

[6] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *SODA*. Society for Industrial and Applied Mathematics, 2014.

[7] L. E. Celis, D. Straszak, and N. K. Vishnoi. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840*, 2017.

[8] C. Chekuri, J. Vondrak, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *FOCS*. IEEE, 2010.

[9] S. Chen, J. Fan, G. Li, J. Feng, K.-l. Tan, and J. Tang. Online topic-aware influence maximization. *PVLDB*, 2015.

[10] W. Chen. An issue in the martingale analysis of the influence maximization algorithm imm. In *International Conference on Computational Social Networks*. Springer, 2018.

[11] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *SIGKDD*, 2009.

[12] K. Chircop and D. Zammit-Mangion. On-constraint based methods for the generation of pareto frontiers. *Journal of Mechanics Engineering and Automation*, 2013.

[13] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *CIKM*. ACM, 2014.

[14] K. Deb and R. Datta. Hybrid evolutionary multi-objective optimization and analysis of machining operations. *Engineering Optimization*, 2012.

[15] M. Drosou, H. Jagadish, E. Pitoura, and J. Stoyanovich. Diversity in big data: A review. *Big data*, 2017.

[16] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 1998.

[17] S. Gershtein, T. Milo, B. Youngmann, and G. Zeevi. Im balanced: Influence maximization under balance constraints. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018.

[18] A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*. ACM, 2011.

[19] A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In *WWW*. ACM, 2011.

[20] R. C. Gunasekara, K. Mehrotra, and C. K. Mohan. 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, 2014.

[21] K. Huang, S. Wang, G. Bevilacqua, X. Xiao, and L. V. S. Lakshmanan. Revisiting the stop-and-stare algorithms for influence maximization. *PVLDB*, 2017.

[22] C.-L. Hwang and A. S. M. Masud. *Multiple objective decision makingâĂŤmethods and applications: a state-of-the-art survey*, volume 164. Springer Science & Business Media, 2012.

[23] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM, 1984.

[24] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[25] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 2008.

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

[27] Y. Li, D. Zhang, and K.-L. Tan. Real-time targeted influence maximization for online advertisements. *PVLDB*, 2015.

[28] S. Mardle, S. Pascoe, and M. Tamiz. An investigation of genetic algorithms for the optimization of multi-objective fisheries bioeconomic models. *International Transactions in Operational Research*, 2000.

[29] H. T. Nguyen, M. T. Thai, and T. N. Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *SIGMOD*. ACM, 2016.

[30] N. Ohsaka, T. Akiba, Y. Yoshida, and K.-I. Kawarabayashi. Fast and accurate influence maximization on large networks with pruned monte-carlo simulations. In *AAAI*. AAAI Press, 2014.

[31] P. Raghavan and C. D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 1987.

[32] C. Song, W. Hsu, and M. L. Lee. Targeted influence maximization in social networks. In *CIKM*. ACM, 2016.

[33] D. Steurer. Max coverageâĂŤrandomized lp rounding. http://www.cs.cornell.edu/courses/cs4820/2014sp/notes/maxcoverage.pdf, 2014.

[34] J. Stoyanovich, S. Abiteboul, and G. Miklau. Data, responsibly: Fairness, neutrality and transparency in data analysis. In *EDBT*, 2016.

[35] J. Stoyanovich, K. Yang, and H. V. Jagadish. Online set selection with fairness and diversity constraints. In *EDBT*. OpenProceedings.org, 2018.

[36] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *SIGMOD*, 2015.

[37] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *SIGMOD*. ACM, 2014.

[38] M. G. C. Tapia and C. A. C. Coello. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2007.

[39] A. Tsang, B. Wilder, E. Rice, M. Tambe, and Y. Zick. Group-fairness in influence maximization. *arXiv preprint arXiv:1903.00967*, 2019.

[40] R. Udwani. Multi-objective maximization of monotone submodular functions with cardinality constraint. In *NeurIPS*, 2018.

[41] V. V. Vazirani. *Approximation algorithms.* Springer Science & Business Media, 2013.

A APPENDIX

B DETAILED RUNNING EXAMPLE

Consider again the example network in Figure 1, with $k = 2$, and recall that the assumed underlying propagation model is the LT model. Table 2 depicts the expected g_1, g_2 and overall influences for all possible 2-size seed sets.

S	$I(S)$	$I_{g_1}(S)$	$I_{g_2}(S)$
{e, g}	5	4	0.5
{e, f}	4.5	3	1.75
{b, e}	4	3	0.5
{d, e}	3.75	3.25	1
{c, e}	3.75	3.25	0.25
{a, g}	4	3	0.5
{a, f} , {f, g}	3.5	2	1.75
{b, f}	3.5	1.5	2
{a, b},{b, g}	3	2	0.5
{a, e}	3	2.5	2.25
{a, d}, {d, g}	2.75	2.25	1
{a, c}, {c, g}	2.75	2.25	0.25
{c, f}	2.5	1.5	1.5
{b, d}	2.5	1.5	1
{b, c}	2.5	1.5	0.5
{d, f}	2	1	2
{c, d}	2	2	1

Table 2: Expected influences of 2-size seed sets.

C MISSING PROOFS

C.1 Proofs of Section 3

Let S_k denote a k-size seed set. We omit the subscript k when it is clear from the context. We begin by presenting several useful properties of the propagation function $I(\cdot)$.

Observation C.1. *Given a k-size seed set S_k to an instance of the MC problem, for every $0 < c \leq 1$ there exists a $\lceil c \cdot k \rceil$-size seed-set $S_{\lceil c \cdot k \rceil} \subseteq S_k$ s.t. $I(S_{\lceil c \cdot k \rceil}) \geq \frac{\lceil c \cdot k \rceil}{k} \cdot I(S_k)$.*

Lemma C.2. *For any instance of the MC problem, we have: $\forall i \in [n-1]$: $I(O_{i+1}) - I(O_i) \leq \frac{I(O_i)}{i}$, where O_j is the j-size optimal solution.*

Proof. Suppose, for the sake of contradiction, that there exists $i \in [n-1]$ s.t. $\frac{I(O_{i+1})}{i+1} > \frac{I(O_i)}{i}$. According to Observation C.1, there exists a $S'_i \subset O_i$ of size (i) s.t. $I(S'_i) \geq \frac{i}{i+1} \cdot I(O_{i+1})$. Overall, we get: $I(S'_i) > I(O_i)$ contradicting the optimality of O_i. Therefore, $\frac{I(O_{i+1})}{i+1} \leq \frac{I(O_i)}{i}$, which implies $I(O_{i+1}) - I(O_i) \leq \frac{(i+1)}{i} \cdot I(O_i) - I(O_i) = \frac{1}{i} \cdot I(O_i)$. □

We next prove the following Lemma holds.

Lemma C.3. *Given an instance $\mathcal{I}$ of MC subject to constraints (1)-(2) and the parameter t, we have: $\frac{I(O_{k_t})}{I(O_k)} \leq t + o(1)$.*

Proof of Lemma C.3. (sketch). It follows from Lemma C.2 that: $I(O_{k_t}) - I(O_{k_t-1}) \leq \frac{I(O_{k_t-1})}{k_t-1}$. Also, by definition of k_t, we have: $I(O_k) \geq \frac{I(O_{k_t-1})}{t}$. Combined, we get: the desired bound: $\frac{I(O_{k_t})}{I(O_k)} \leq \frac{t}{I(O_{k_t-1})} \cdot (I(O_{k_t-1}) + \frac{I(O_{k_t-1})}{k_t-1}) = t \cdot (1 + O(1))$, where the last equality follows from the constraint $k = \omega(1)$, which is true for all hard instances, as for constant k the problem can be solved polynomial by considering all $O(n^k)$ possible solutions. □

Before proceeding to proving our main result, we note that we assume henceforth, in our approximation hardness results, that in any MC instance we are also given as part of the input the optimal value $I(O_k)$ for that instance. Observe that the MC problem retains its $(1 - \frac{1}{e})$ approximation hardness bound, even when given this knowledge. Otherwise, one could guess all $O(n)$ possible optimal values, and for each guess solve the problem, leading to an improved bound for the right guess, and therefore for the original instance as well, which is a contradiction.

Finally, we provide a formal proof of Theorem 3.5.

Proof of Theorem 3.5. Let $t \leq 1 - \frac{1}{e}$, and $\mathcal{I}_1, \mathcal{I}_2$ denote two instances of MC, $\mathcal{I}_1 = \langle S_1, k - k_t \rangle$ and $\mathcal{I}_2 = \langle S_2, k_t \rangle$, with the same number of elements up to a constant multiplicative factor. Recall the definition of k_t:

$$k_t = argmin_{l<k, l \in \mathbb{N}}(I(O_l) \geq t \cdot I(O_k))$$

where O_i is the optimal i-size solution.

Next, we build the following instance $\hat{\mathcal{I}}$ for multi-objective MC. The sets of $\hat{\mathcal{I}}$ are $S_1 \cup S_2$. The required number of sets in the output for $\hat{\mathcal{I}}$ is set to k. We mark all elements in S_1 and S_2 as g_1 and g_2 elements, resp. To satisfy the constraint, one must cover $t \cdot I_{g_2}(k)$ elements, which, by definition, requires choosing at least k_t sets. Therefore, the k-size optimal solution is $B_{k-k_t} \cup R_{k_t}$, where B_{k-k_t} is the optimal $(k - k_t)$-size solution to the original instance $\mathcal{I}_1$, and R_{k_t} is the optimal solution to the original instance $\mathcal{I}_2$, which uses all available k_t slots. If one could always exceed a $(1 - \frac{1}{e}) \cdot I_{g_1}(B_{k_t-k})$ overall cover size, it would require, in hard cases (for I_1), to use more than $(k - k_t)$ sets, implying the size constraint can always be achieved up to an additional $(1 - \frac{1}{e} + o(1))$ loss factor, using less than k_t sets, which carries over to the instance $\mathcal{I}_2$ and contradicts the hardness of the MC problem (note that, for any given algorithm, with positive probability both original instances may be "maximally hard").

We have implicitly leveraged Lemma C.3, which implies that using by "repurposing" a constant number of seeds to use in favor of covering the objective instead of the constraint,

this results in negligible $o(1)$-fractional addition to the value of the objective (The proof for this property of the objective is completely analogous to Lemma C.3), which when ignoring sub-constant factors, yields exactly the same approximation bounds.

Additionally, observe that our hardness bound holds even when in the multi-objective MC instance we are given prior knowledge of the optimal value for the (constrained) objective, and the explicit value for the constraint threshold, as these are known when reducing from MC instances whose optimal values are known, which as we have proved retain their hardness bound.

Finally, we provide more details regrading the reduction from the multi-objective MC problem to the Multi-Objective IM problem. Given an instance of multi-objective MC, $\langle S, k, t, g_1, g_2 \rangle$ we create a weighted directed graph, G, as follows: For each element u_i add node u_i with the same same g_1 or g_2 property to G; for each set $S_i \in S$ add a node v_i and add edges of weight 1 from it to all nodes $u_j \in S_i$. Any solution to an instance of the multi-objective MC, can be augmented by substituting any node u_i in the seed set with a node v_j such that $u_i \in S_j$, and then translated back to the corresponding k sets in S constituting a solution with the same approximation factors. Moreover, any solution to the Multi-Objective IM instance that consists of only v_i nodes (such a solution cannot be improved with u_j nodes), is equivalent in all parameters to the analogous solution for the multi-objective MC instance. □

The proof for the case with $m > 2$ groups $g_1, g_2, ..., g_m$ (where the lower bound once again implies a $1 - \frac{1}{e}$ bound for every group simultaneously) with a threshold parameter constraint t_i for each g_i where $i \geq 2$ is completely analogous. We once again, given apriori an arbitrary k and and $\{t_i\}_{i>2}$ s.t. $\sum_{i>1} t_i \leq 1 - \frac{1}{e}$, assign a different and disjoint MC instance for each group, such that the first instance has cardinality constraint $k - \sum_{i>1} k_{t_i}$, and every other instance, $i > 2$, has a cardinality constraint of k_{t_i}. The rest of the proof is completely analogous to the proof given above for 2 groups. Note that m is a constant, hence any algorithm polynomial in the size of an instance assigned to a specific group, remains polynomial in the size of the entire input, which is the union of all instances.

C.2 Proofs of Section 4.1

Here we provide the full proof of Theorem 4.3.

Proof of Theorem 4.3. Recall that any RIS-based IM algorithm, $\mathcal{A}$, uses a greedy selection during the second phase of solving the MC instance. Let S_{g_2} denote the g_2 elements covered by the optimal k-size selection O_{g_2}. Since at most k seed nodes (all seeds in O_{g_2} not chosen so far) can cover, after any iteration of the greedy procedure, the remaining uncovered elements in S_{g_2}, at least one set covers a $\frac{1}{k}$ fraction, and, thus, the greedy selection at each iteration, covers at least a $\frac{1}{k}$ fraction as well. Hence, the fraction of elements in S_{g_2} covered after $x \cdot k$ iterations is $1-(1-\frac{1}{k})^{x \cdot k} \geq 1-(\frac{1}{e})^{x}$. The adaptation to elements of all groups is immediate.

It follows that $I_{g_2}(S) \geq I_{g_2}(S_1) \geq 1-(\frac{1}{e})^{-\ln(1-t)} \cdot I_{g_2}(O_{g_2}) = t \cdot I_{g_2}(O_{g_2})$. Hence the constraint is satisfied. Similarly, we have $I_{g_1}(S) \geq I_{g_1}(S_2) \geq 1-(\frac{1}{e})^{1+\ln(1-t)} \cdot I_{g_1}(O_{g_1}) = (1-\frac{1}{e \cdot (1-t)}) \cdot I_{g_1}(O_{g_1})$. This proves the guarantee on the objective function. □

C.3 Proofs of Section 4.2

As mentioned, our proofs and algorithm follows similar lines to the proof provided for the original MC problem in [33].

To prove Theorem 4.4, we first show that the optimal objective value of the LP depicted in Section 4.2 exceeds the optimal objective value for the original multi-objective MC instance. Concretely we prove that $Opt(\mathcal{I})$, the optimal solution to the multi-objective MC instance $\mathcal{I}$, is a valid solution for LP(*mathcalI*), thus proving that the optimal fractional solution reaches an objective value exceeding that of $I_{g_1}(Opt(\mathcal{I}))$.

We construct a solution of $LP(\mathcal{I})$ with an objective value of at least $I_{g_1}(Opt(\mathcal{I}))$.

$$x_i = \begin{cases} 1, & \text{if } S_i \in Opt(\mathcal{I}) \\ 0, & \text{otherwise} \end{cases}, \; w_j = \begin{cases} 1, & \text{if } b_j \in \cup_{S_i \in Opt(\mathcal{I})} S_i \\ 0, & \text{otherwise} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{if } b_j \in \cup_{S_i \in Opt(\mathcal{I})} S_i \\ 0, & \text{otherwise} \end{cases}, \; z_j = \begin{cases} 1, & \text{if } r_j \in \cup_{S_i \in Opt(\mathcal{I})} S_i \\ 0, & \text{otherwise} \end{cases}$$

This solution satisfies the cardinality constraint because exactly k of the variables $x_1, ..., x_m$ are set to 1. The solution also satisfies the coverage constraints: If $y_j = 0$, then the corresponding coverage constraint is satisfied because all x_i values are non-negative. Otherwise, if $y_j = 1$, then b_j is covered by $Opt(\mathcal{I})$, which means that one of the sets contains b_j, therefore, at least one of terms of the sum $\sum_{i:b_j \in S_i} x_i$ is equal to 1 (which is enough to satisfy the inequality). Similarly, the coverage constraint for the variables $z_1, ..., z_Z, w_1, ..., w_W$ is satisfied as well. As for the size constraint, since $Opt(\mathcal{I})$ is the optimal solution, by definition it is also satisfies the size constraint, therefore, it follows that this solution satisfies the size constraint as well.

Next, we prove that for each element u_j the probability that the set S produced by the randomized rounding algorithm covers u_j is at least $(1-\frac{1}{e}) \cdot y_j$ (resp., $(1-\frac{1}{e}) \cdot z_j$, $(1-\frac{1}{e}) \cdot w_j$). We only prove here that this holds for g_1 elements, as the proof for g_2 elements is identical.

If we choose a random set according to the probabilities $\frac{x_1}{k}, ..., \frac{x_m}{k}$, it covers u_j with probability $\sum_{i:u_j \in S_i} \frac{x_i}{k} \geq \frac{y_j}{k}$, following the coverage constraints. Therefore, the probability

that none of the k selected sets covers u_j is at most $(1-\frac{y_j}{k})^k$. Thus, the node u_j is covered by the set S with probability at least $1-(1-\frac{y_j}{k})^k$. It remains to verify that $1-(1-\frac{y_j}{k})^k \geq (1-\frac{1}{e})\cdot y_j$. This follows immediately from the proof provided to the standard MC case in [33].

Finally, we prove the main pertaining to the approximation factors stated in Theorem 4.4. We focus on the g_2 elements, as the proof for the cover size is identical. Let R_j be the 0/1-valued random variable such that $R_j = 1$ indicates the event that S_k covers u_j. Then, the number of g_1 elements that S_k covers is equal to $\sum_{j=1}^{n} R_j$. Therefore, by linearity of expectation, the expected number of g_1 elements covered by S_k is: $\mathbb{E}[\sum_{j=1}^{n} R_j] = \sum_{j=1}^{n} \mathbb{E}[R_j]$. Since R_j is a 0/1-valued random variable, the expectation of R_j is equal to the probability that $R_j = 1$. Hence, the expected number of g_1 elements covered by S_k is equal to:

$$\sum_j \Pr[R_j = 1] = \sum_j \Pr[S_k \text{ covers } u_j] \geq (1-\frac{1}{e})\sum_j y_j$$

$$= (1-\frac{1}{e})\cdot I_{g_1}(OPT(LP(\mathcal{I}))) \geq (1-\frac{1}{e})\cdot I_{g_1}(OPT(\mathcal{I}))$$

Next, we prove that the scale of the reduction in α after omitting the assumption on $I_{g_2}(O_{g_2})$ is not arbitrarily large.

Proof of Proposition 4.5. Let S denote the output of an IM_{g_2} algorithm on the given network. Define $\lambda \in [0, \frac{1}{e-1}]$ s.t. $I_{g_2}(S) = (1+\lambda)\cdot(1-\frac{1}{e})\cdot I_{g_2}(O_{g_2})$. Using $(1-\frac{1}{e})^{-1}\cdot I_{g_2}(S)$ instead of $I_{g_2}(O_{g_2})$ in the formulation of the LP, results (following the same reasoning as in the proof for the original LP) in $\beta = (1+\lambda)\cdot(1-\frac{1}{e})$, as we lose a factor of $(1-\frac{1}{e})$ in the rounding procedure.

It follows from Observation C.1 that, for $k' = t\cdot(1+\lambda)\cdot k$, we have $I_{g_2}(O^{k'}_{g_2}) \geq t\cdot(1+\lambda)\cdot I_{g_2}(O^{k}_{g_2})$, where O^k_g is the optimal k-size solution for the group g. Also, from Observation C.1, we get that: $I_{g_1}(O^{k-k'}_{g_1}) \geq \frac{k-k'}{k}\cdot I_{g_1}(O^k_{g_1})$. By definition, we have $I_{g_1}(O^k_{g_1}) \geq I_{g_1}(O^{*k})$, where O^{*k} is the optimal solution to the Multi-Objective MC instance (the unconstrained optimal solution can only improve on a constrained optimal solution), hence $I_{g_1}(O^{k-k'}_{g_1}) \geq \frac{k-k'}{k}\cdot I_{g_1}(O^{*k}) = (1-t\cdot(1+\lambda))\cdot I_{g_1}(O^{*k})$. Finally, we can conclude that the set $S' = O^{k-k'}_{g_1} \cup O^{k'}_{g_2}$ (if it is not of size k, we can add arbitrary sets to it) is a solution which satisfies the new constraint ($I_{g_2}(S') \geq t\cdot I_{g_2}(S)$) s.t. $I(O^{k-k'}_{g_1}) \geq (1-t\cdot(1+\lambda))\cdot I_{g_1}(O^{*k})$. This implies that the optimal solution (to the modified problem with the elevated constraint) achieves at least the same value for $I_{g_1}(\cdot)$. Hence since we can approximate that optimal solution to a factor of at least $(1-\frac{1}{e})$, compared to the optimal solution of the (implicit) unmodified LP, we get: $\alpha = (1-\frac{1}{e})\cdot(1-t\cdot(1+\lambda))$. □

C.4 Proofs for Section 5

We present reductions in both directions between RSOS and a variant of our Multi-Objective IM problem where the constraint threshold is an explicit value C, and the goal is to find a k-size set S that maximizes $I_{g_1}(S)$ subject to $I_{g_2}(S) \geq C$. These reductions show equivalence between approximating RSOS by an α factor and approximating Multi-Objective IMby a $(\alpha, \alpha, ..., \alpha)$ factor. Later, we show how to extend the equivalence to our standard definition with a threshold parameter t.

Proof of Theorem 5.1. Given an instance of RSOS, assume w.l.o.g. that $v_2 \leq v_1$ and that v_1 and v_2 are integers (as for large enough values, rounding the values is asymptotically insignificant). We reduce it to an instance of Multi-Objective IM with $I_{g_i} = f_i$ and $C = v_2$. Let A denote the optimal solution for this Multi-Objective IM instance. We have $\forall i : I_{g_i}(A) \geq v_i$. Moreover, an (α, α)-approximation algorithm for Multi-Objective IM which produces a solution S implies that $\forall i : I_{g_i}(S) \geq \alpha v_i$, therefore, as the functions and input sets are exactly the same, this is also a solution S for the RSOS instance s.t. $f_i(S) = I_i(S) \geq \alpha \cdot v_i$.

As for the standard formulation of Multi-Objective IM with a t-threshold, we can reduce an RSOS instance to $O(n^2)$ instances of the (implicit) Multi-Objective IM, each corresponding to a different guess for the value of t, iterating over all possible ratios between two integers bounded by n. We then select of all solutions which exceed the explicit threshold of the RSOS instance the one which has the maximal value for f_1. At the very least, the correct guess will produce the desired approximation.

Finally, to handle multiple groups, we need to guess the constraint threshold value t_i for every group $g_i, i \in [2, m]$. As we do in the binary scenario, we can reduce an RSOS instance to $O(n^{(2m)})$ instances of the (implicit) Multi-Objective IM (recall that m is a constant, hence this is a polynomial reduction). We then once again choose the solution which exceeds all the explicit values in the RSOS instance (recall that the explicit values are the same for both instances, as the functions and input sets are exactly the same). □

Proof of Theorem. 5.2 Given an instance of Multi-Objective IM with an explicit value C for the constraint, we can try all $O(n)$ possible guesses for the optimal value for the objective (subject to the constraint) $I_{g_1}(A)$. We can also approximate this value more efficiently, by trying $O(lg(n))$ guesses exponentially growing by a some constant factor $delta > 1$, at the price of reducing the approximation guarantee by a multiplicative $1/delta$ factor, an error we can make arbitrarily small. We reduce per each guess of $I_1(A)$ to an instance of RSOS by setting V_1 to be the value of $I_1(A)$, $v_2 = C$ and $f_i = I_{g_i}$. Any α approximation algorithm for RSOS then implies that over the instance using the correct guess, we

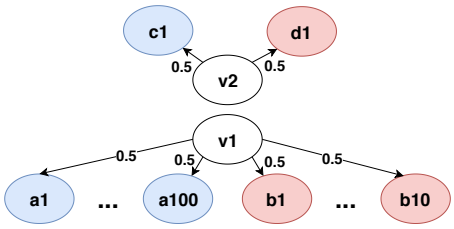


Figure 5: Example social network with two emphasized groups of users.

can produce an (α, α)-approximation of the original Multi-Objective IM instance.

As for the standard formulation of Multi-Objective IM with a t-threshold, we can estimate the possible range of values for the optimal value for I_{g_2} by running the IM_{g_2} algorithm, and solving an instance of RSOS for each guess in this range. One of the solutions provides the same approximation guarantees as before, however since we cannot identify it (in general we cannot determine which solutions exceed the threshold, as we do not know the explicit value up to a $(1-1/e)$ factor [16]), we must overestimate the optimal value produced by IM_{g_2} by a $(1-1/e)$ factor, as we do in RMOIM, yielding the same guarantees when using an $(1-1/e)$-approximation algorithm for RSOS.

Finally, to handle multiple groups, we use a completely analogous procedure to estimate the possible range of values for the optimal value for I_{g_i} by running the IM_{g_i} algorithm, $\forall i \in [2, m]$. As we may overestimate all these values by a $1-\frac{1}{e}$ factor, this reduction yields the same approximation guarantees as RMOIM (for multiple groups).

□

D ALTERNATIVE PROBLEM DEFINITIONS

As mentioned in Section 4.3, maximizing the ratio between the cover cardinalities can dramatically reduce the number of covered users from each emphasized group. We next illustrate this phenomenon via an example.

Example D.1. Consider the small network depicted in Figure 5. Assume that the user has defined two emphasized groups g_1 and g_2 to be the blue and red users, resp, and let $k = 1$. To maximize the ratio between the group covers, v_2 is the only optimal solution, as it covers the same number of blue and red users. However, in our Multi-Objective IM definition, with any value of the constraint threshold parameter t in $(0, 1-\frac{1}{e}]$ imposed, e.g., on the red users, v_1 is the optimal solution. One can see that even though v_1 does not cover the same number of blue and red users, it covers significantly more such users than v_2.

Note that in our analysis so far the user imposes constraints on all but one group. Next, we briefly discuss the scenario where given m emphasized groups, the user imposes constraints all m groups, and wishes to find a solution that satisfies all constraints. This can be easily reduced to our framework by examining the case where constraints are imposed on all but one emphasized groups. Given a solution in which all $m-1$ constraints are satisfied, we check whether it also satisfies the remaining constraint. In case it does, we return this solution. Otherwise, we can safely state that no solution satisfying all constraints simultaneously exists.

Datasets	Example neglected groups
Facebook	Users with a certain education type ; Users of a certain gender (the education type and gender properties are encrypted)
DBLP	Female Indian researchers ; (Female) Researchers from developing countries.
Pokec	Female users over the age of 50 ; (Female) Users over the age of 40/50/60; Users from different cities.
Weibo-Net	(Female) Users from different cities .

Table 3: Example groups of users which are typically neglected by IM algorithms.

E EXPERIMENTAL STUDY - ADDITIONAL DETAILS

Here provide more details regarding our experimental study and additional experimental results.

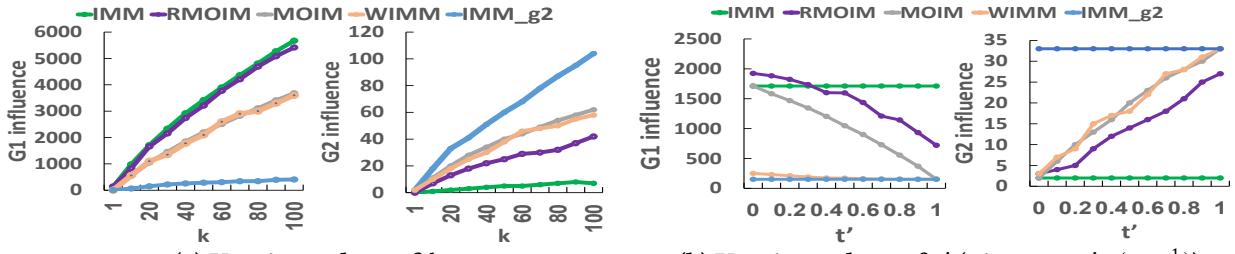
E.1 Example Emphasized Groups

Table 3 depicts example of groups that are typically not covered by standard IM algorithms. The examples highlighted in red are the ones we have examined in our experiments (for quality evaluation for scenario *I*).

E.2 Parameters Tuning

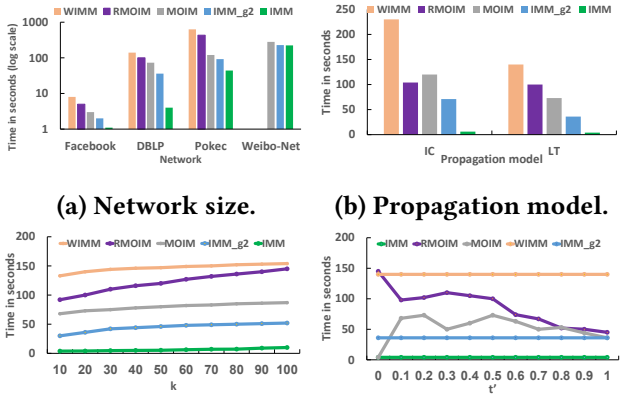
Next, we examine how varying the input parameters affects the results. To illustrate, we present here the results using a range of values for k and t over the DBLP dataset (the other datasets show similar trends). We note that a desirable behavior of a Multi-Objective IM algorithm is as follows. As k 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 6. 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



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

Figure 6: 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 7: Averaged execution times for scenario I.

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 6(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.

E.3 Performance Evaluation Experiments for Scenario I

We first report the running times for the cases presented in Section 6.2. Naturally, the larger the network, the longer it takes to compute the solution. Indeed, as depicted in Figure 7(a), all competitors' running times increase for larger networks. Although we see that MOIM and RMOIM are naturally slower than *IMM* and IMM_{g_2}, they run in less than 2 and 7 minutes, resp., even on the Pokec dataset, which includes almost 1M nodes and 15M edges. That is, both our algorithms can process large-scale networks in feasible running times. However, when it comes to massive networks such as Weibo-Net, while MOIM processed it in less than 47 minutes (in comparison, IMM_{g_2} processed it in 40 minutes), RMOIM can not process it, since there are too many parameters and inequalities for the LP solver to handle. According to our experiments, RMOIM is feasible for graphs including up to 20M edges and nodes.

Regarding *WIMM*, as it searches for the optimal weights, its running times were significantly longer than both our algorithms. For example, even on the small Facebook network, it took *WIMM* 8 seconds - almost 8 times more than *IMM* (which ended after 1.1 seconds) and almost 1.5 times slower than RMOIM (which ended after 5 seconds).

In the following experiments, when modifying the value of one parameter we fix other parameters to their default values. For space limitations, we present the results only for the DBLP network, and report that other networks have demonstrated similar trends.

Propagation model. We examine the effect of the underlying propagation model on running times. The results are depicted in Figure 7(b). Here again, all *IMM*-based baselines performed better with the LT as the underlying propagation model. Concretely, it took all *IMM* variants almost twice the time to process the DBLP network when using the IC model. As RMOIM is less sensitive to the increase in the number of *RR* sets, it behaves similarly under both propagation models. E.g., while MOIM and RMOIM running times were similar under the IC model (the difference was less than 3 seconds), under the LT model, MOIM runs almost two times faster than RMOIM.

Seed-set size. Here we examine the effect of the seed-set size k on running times. The results are depicted in Figure 7(c). 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. We note that this behavior of MOIM is a consequence of employing *IMM*, and therefore using an alternative IM algorithm (e.g., [18]) 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. Last, we examine how the parameter t affects performance. The results are depicted in Figure 7(d). Here we tested all t values of the form $t = t' \cdot (1 - \frac{1}{e})$, where $t' \in [0.1, 0.2, \ldots, 1]$. Note that, this parameter only affects the running times of our algorithms[6], and in particular, in the MOIM algorithm it dictates the required seed set size for the two procedures it employs. As expected, for very small or very large t values, MOIM is (almost) identical to IMM or IMM_{g_2} (e.g., when $t = 0$ it only runs IMM), while for other t values it employs both versions of IMM 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 values (i.e., less k-size seed-sets satisfy the constraint), the running time of RMOIM decreases.

[6]In the $WIMM$ baseline, t only dictates the optimal solution and hence has no effect on running times.