

# Uncharted but not Uninfluenced: Influence Maximization with an Uncertain Network

Bryan Wilder<sup>1</sup>, Amulya Yadav<sup>1</sup>, Nicole Immorlica<sup>2</sup>, Eric Rice<sup>1</sup>, Milind Tambe<sup>1</sup>  
<sup>1</sup> Center for AI in Society, University of Southern California <sup>2</sup> Microsoft Research, New England  
{bwilder, amulyaya, eric, tambe}@usc.edu nicimm@gmail.com

## ABSTRACT

This paper focuses on new challenges in influence maximization inspired by non-profits' use of social networks to effect behavioral change in their target populations. Influence maximization is a multi-agent problem where the challenge is to select the most influential agents from a population connected by a social network. Specifically, our work is motivated by the problem of spreading messages about HIV prevention among homeless youth using their social network. We show how to compute solutions which are provably close to optimal when the parameters of the influence process are unknown. We then extend our algorithm to a dynamic setting where information about the network is revealed at each stage. Simulation experiments using real world networks collected by the homeless shelter show the advantages of our approach.

## Keywords

Dynamic influence maximization, game theory, HIV prevention

## 1. INTRODUCTION

Many behaviors are mediated by social influences, so a population's social network is essential to spreading a desired behavior. The influence maximization problem models a multi-agent setting over a social network where a population of agents can influence each other via the links of the network. The challenge is to select the most influential nodes, a task with many applications. For instance, nonprofit organizations harness social networks to improve health in disadvantaged groups by conducting in-person social interventions. Such interventions have proven successful in many domains [24], including reducing HIV spread [18], improving nutrition [12], and reducing smoking [22]. In this paper, we use as an example the problem of spreading messages about HIV prevention among homeless youth. HIV/AIDS kills 10,000 people each year just in the U.S. [3], and the proportion of homeless youth who are HIV positive is over ten times that for populations with stable housing [17]. However, as is typical of nonprofits, homeless shelters only have sufficient resources to accommodate a few youth in each intervention [19]. They depend on word of mouth to expand the reach of their message.

While influence maximization has been well studied in online social networks [11, 5, 8], many domains (e.g., HIV prevention in homeless youth) raise three new challenges. First, organizations

face *parameter uncertainty*. Models of influence spread [11] require numerous parameters which give the probability that influence will propagate along each edge. However, organizations acquire only a partial picture of the network from field observations, Facebook contacts, etc., and there will inevitably be many links of uncertain strength [18]. Thus, we require algorithms which perform well with large uncertainty about the model parameters.

Second, the problem is *dynamic*. The organization (e.g., the homeless shelter) can ask participants about their immediate social circle during each intervention. Hence, updates about the true parameters are given at each stage, informing node selections for future interventions. We discuss the shortcomings of some related work which uses POMDPs for dynamic influence maximization in Section 3 [27, 26], but this related work has certainly failed to address the challenges of parameter and execution uncertainty.

Third, organizations face *execution uncertainty* because they cannot be sure that the youth (i.e., nodes in the social network) they select for an intervention will actually show up. This problem is particularly acute in our example domain because a number of events could prevent a youth's participation. For instance, a youth might have been arrested or gone to stay with relatives.

While we illustrate these issues through health-related interventions, the underlying principles apply to many influence maximization domains. Take, for instance, a viral marketer who generates word of mouth by giving away a limited number of discounts. While the marketer likely will not know how influence propagates ahead of time, they can learn more during the campaign by surveying their users (e.g. "name five friends to receive this coupon").

**Overview of our approach:** We start by addressing parameter uncertainty in the nondynamic setting by showing how to compute solutions whose value under *any* parameter setting is close to the optimum if the true parameters were known. We view the problem as a zero sum game between the influencer and an adversary who chooses the worst case parameters, and present techniques which provably approximate the minimax strategies. The key challenge is that the algorithm's strategy space is exponentially large, while the adversary's is infinite. We give a polynomial time primal-dual algorithm, as well as DOSIM, a more practical algorithm based on a double oracle approach. DOSIM uses a greedy algorithm as the inner oracle for the influencer. DOSIM also handles execution uncertainty. We then extend these techniques to the dynamic setting by showing that DOSIM's greedy oracle still gives excellent empirical performance. Our approach is evaluated on real world social networks of homeless youth in a large U.S. city. We show that it produces policies which perform near optimally regardless of the unknown parameters, while failing to consider this uncertainty results in substantially worse performance.

Our algorithm has been piloted by a homeless shelter in a ma-

**Appears in:** *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.

Copyright © 2017, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

for US city. A companion paper [28] shows results and detailed analysis from a pilot test of this and other algorithms at homeless shelters in a major US city, and shows the benefits of DOSIM in practical settings. A large-scale deployment of our algorithm with approximately 300 youth is scheduled to begin in Spring 2017.

## 2. DYNAMIC INFLUENCE MAXIMIZATION

**Problem overview:** A social network is a directed graph  $G = (V, E)$ , with  $|V| = n$  and  $|E| = m$ . Each node  $u \in V$  represents an agent and a directed edge from  $u$  to  $v$  indicates that  $u$  is a friend of  $v$ . While friendships are typically reciprocal,  $G$  is directed because influence is often asymmetric [20]. An influencer does the following for  $T$  stages (where  $T$  is the total number of interventions). First, she selects a set of  $K$  nodes (participants) to attend the intervention and be influenced. Each node actually attends with an independent probability. Then, the influencer receives *observations* about the propagation probability on each edge outgoing from nodes that did attend, clarifying those nodes' immediate social circle. Finally, the influencer selects the next  $K$  nodes and the process repeats. The objective is to maximize the expected number of influenced nodes at the end of stage  $T$ .

**Influence model:** Influence propagates according to a variant of the standard independent cascade model [11]. In the independent cascade model, each edge  $e = (u, v)$  is labeled with a probability  $p_e$ . The vector  $\mathbf{p}$  contains  $p_e$  for each  $e \in E$ . If  $u$  is influenced, it makes one attempt to influence  $v$  which succeeds with probability  $p_e$ . Each influenced node remains influenced for all stages. We use a variant where a node tries to influence its neighbor at each stage (until it succeeds), instead of just the first. This has been shown to better match empirical diffusion patterns [6].

It has been observed in many domains [1, 13] that the likelihood of influence spread depends largely on the kind of relationship. To model this phenomenon, each edge  $e \in E$  has a type  $\theta_e$  drawn from a set  $\Theta$ . In our example, homeless shelter officials can annotate the edges as being either strong or weak ties. Each type  $\theta \in \Theta$  is associated with a prior distribution over propagation probabilities and each edge  $e$  independently samples  $p_e$  from the prior for  $\theta_e$ . This is a natural framework for our domain because organizations cannot accurately estimate each  $p_e$ , but they may classify the edges into types. We focus on uniform priors but our work can be extended to other distributions. We will assume a fixed width  $w$  for each uniform prior and overload notation to let  $\theta_e$  give the center of the distribution. Hence, each edge samples  $p_e \sim U[\theta_e - \frac{w}{2}, \theta_e + \frac{w}{2}]$ , where  $U$  denotes a uniform distribution. Figure 1 provides an example. The vector  $\theta$  contains  $\theta_e$  for each  $e \in E$ .

**Observation model:** The influencer receives observations at each stage as she asks participants (nodes) about their immediate social circle. Importantly, it is not possible to ask participants for the exact  $p_e$ . Instead, the influencer only learns proxies for this probability, such as how frequently the participant interacts with the other person. We abstract this feedback as updates *relative* to the prior. For example, if  $e = (u, v)$  and  $u$  and  $v$  interact more often than typical acquaintances, then we would infer that  $p_e$  is on the upper end of the prior  $\theta_e$ . Accordingly, for each outgoing edge  $e$ , we assume that the algorithm observes the quantile of  $p_e$  relative to  $\theta_e$ .

**Example 1:** If  $\theta_e = 0.3$  and  $w = 0.4$ , then our prior distribution would be  $p_e \sim U[0.1, 0.5]$ . If frequent interaction was observed, then the influencer would infer that  $p_e$  lies in the upper quantile of the prior, and the updated posterior distribution would be  $U[0.3, 0.5]$ .

**Objective formalization:** Our algorithm starts with an input consisting of a graph  $G$  (with all nodes uninfluenced) and prior parameters  $\theta$ . Unknown to the algorithm, the vector of edge prop-

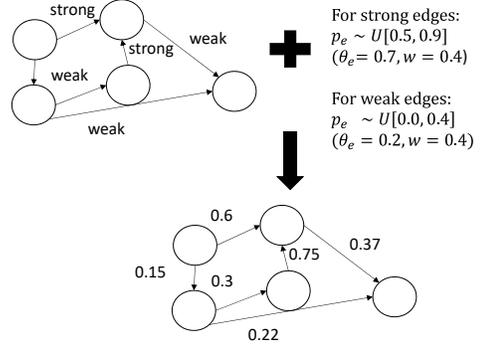


Figure 1: Sampling propagation probabilities from the prior. Each edge  $e$  with the “strong” type has  $\theta_e = 0.7$ . Since  $w = 0.4$ ,  $p_e$  is drawn from the uniform distribution  $U[0.5, 0.9]$ . Analogously, weak edges have  $\theta_e = 0.2$  and hence  $p_e \sim U[0.0, 0.4]$ .

agation probabilities  $\mathbf{p}$  is drawn from the factorized prior described by  $\theta$ . At each stage  $t \in [1, T]$  the algorithm picks an action  $S_t$  from the set  $\mathcal{A} = \{S \subset V \text{ s.t. } |S| = K\}$ . Each node  $v \in S_t$  attends with an independent probability  $q_v$ , which is known in advance.  $q_v$  models execution uncertainty. The algorithm then receives an observation  $O_t$  containing each outgoing edge from the subset of  $S_t$  that attended, which allows it to update its posterior distribution over the edge probabilities. The algorithm’s state is given by the actions taken and observations received so far,  $\{(S_1, O_1) \dots (S_t, O_t)\}$ . A policy  $\pi$  is a mapping from states to actions, and hence specifies the action to take in each stage of the problem. Let  $f(\pi, \mathbf{p})$  be the expected number of nodes influenced by  $\pi$  by the end of stage  $T$  when the propagation probabilities are  $\mathbf{p}$ <sup>1</sup>.  $f$  can be evaluated by averaging over random samples of the influence process [11, 23]. Our goal is to find a policy which maximizes  $\mathbb{E}_{\mathbf{p} \sim \theta}[f(\pi, \mathbf{p})]$ .

## 3. UNCERTAINTY

Thus far, our model contains uncertainty about the realized value of the probability  $p_e$  given its expectation  $\theta_e$ . This reflects variance in the distribution, and is captured via  $w$  (the width of the support of the prior over  $p_e$ ). This uncertainty is important in the dynamic setting, because  $w$  controls the amount of information which can be gained via observations. For instance, if  $w = 0$ , then the prior for any edge is a point distribution and subsequent observations cannot reveal anything new. However, if  $w = 0.4$ , then the posterior mean on a given edge will shift by 0.1 if the algorithm observes whether  $p_e$  for that edge lies in the top or bottom quantile.

On its own, though, placing a distribution over  $p_e$  is inadequate in the following sense. As noted by He and Kempe [9], distributional uncertainty does not impact the average behavior of the model. This is because the following processes are analytically equivalent: (1) draw  $p_e \sim U[\theta_e - \frac{w}{2}, \theta_e + \frac{w}{2}]$  and then propagate influence with probability  $p_e$  (2) propagate influence with probability  $\theta_e$ . That is, in the classical influence maximization problem, placing a distribution over  $p_e$  is no different from setting  $p_e$  deterministically to its mean  $\theta_e$ . Simply increasing the width  $w$  does not change this: the priors  $p_e \sim U[0.4, 0.6]$  and  $p_e \sim U[0.2, 0.8]$  both imply that edge  $e$  will propagate influence with probability  $\mathbb{E}[p_e] = 0.5$ . While distributional uncertainty does have a role

<sup>1</sup>All of our techniques also apply to an infinite time horizon with discounted rewards.

in our dynamic problem because we allow the influencer to learn more about the realized values, it is not sufficient to capture uncertainty about the average likelihood that influence will spread. For instance, the influencer may know that  $p_e$  is drawn from the “strong tie” prior, but this will mean entirely different things if a strong tie has  $\theta_e = 0.1$  than if  $\theta_e = 0.9$ . In domains such as HIV prevention, we will not know what  $\theta_e$  should be set to because, e.g., we do not know the average probability that one friend will be able to convince another to get tested for HIV.

We thus consider a second, higher-order uncertainty over the value of  $\theta$ . We assume interval uncertainty, that is,  $\theta \in [a_\theta, b_\theta] \forall \theta \in \Theta$ . Since the influencer never learns the true value of  $\theta$ , our goal is to produce an algorithm which simultaneously performs well for all possible values of  $\theta$ . Robustness to this second kind of uncertainty is what was considered by Chen et al. [4] and He and Kempe [9]. The key difference from the first level of uncertainty is that we do not have a distribution over  $[a_\theta, b_\theta]$ . This leads to an adversarial model, requiring robustness to a worst-case choice of  $\theta$  [9]. A worst-case approach is further motivated by the fact that organizations may not be able to quantify higher-order uncertainty, i.e., provide a prior over the prior parameters. *In short, we have a more expressive model to represent two levels of uncertainty.* The following example shows how the entire model works together, incorporating both levels of uncertainty and our observation model.

**Example 2:** Suppose that for some edge  $e$ , our interval uncertainty on  $\theta$  gives  $\theta_e \in [0.2, 0.8]$  and  $w = 0.4$ . If the actual value is  $\theta_e = 0.2$ , then this would imply that  $p_e \sim U[0.0, 0.4]$ . If instead the value is  $\theta_e = 0.6$ , then we would have  $p_e \sim U[0.4, 0.8]$ . Similar distributions are implied for every  $\theta_e$  contained in our uncertainty interval  $[0.2, 0.8]$ . The influencer does not know which distribution is the true one. However, suppose she observes that  $p_e$  is in the upper quantile of the prior. Then in the case that  $\theta_e = 0.2$ , the updated posterior would be  $p_e \sim U[0.2, 0.4]$ , and in the case  $\theta_e = 0.6$ , the posterior would be  $p_e \sim U[0.6, 0.8]$  (and so on). The influencer still does not know the true value of  $\theta_e$ . But she now knows that (whatever it is)  $p_e$  lies in  $U[\theta_e, \theta_e + \frac{w}{2}]$ .

When our problem contains only the first level of uncertainty (about the realized value of  $p_e$ ), we call this the *known parameter* case (as the key parameters  $\theta$  are known). When the second level of uncertainty is also present, and we are hence uncertain about  $\theta$ , we call this the *robust* case. Our objective for the robust case is formalized in Section 5.

## 4. RELATED WORK

We discuss work on three related topics. First is work which deals with parameter uncertainty in classical influence maximization. Two recent papers [9, 4] analyze a similar model of parameter uncertainty, where the parameter for each edge is chosen adversarially within an interval. In their model, the edge parameters are totally independent. However, due to the domains we consider, we assume that edges belong to a small number of types. Therefore, our models are not directly comparable, and in particular the hardness result proved by He and Kempe [9] does not apply (we elaborate in Section 5). Further, neither of [9, 4] have a dynamic component, which is a key feature of our domain. Independently, several recent papers deal with learning either the structure [7, 16] or parameters [15] of a network. All of these papers assume that the algorithm observes many independent influence cascades, so our contributions are largely orthogonal to learning based approaches. Similarly, Vaswani et al. [25] introduce a bandit-inspired setting where the algorithm learns the reward it obtained after each influence campaign. In our case, an organization will never observe the hundreds of campaigns required by such approaches since each one

takes several months.

Second is work by Yadav et al. [27, 26] on applying dynamic influence maximization to HIV prevention. This is the most closely related work to ours, and hence Section 7 empirically compares our algorithm and theirs. Yadav et al. formulate the problem as a POMDP. Our contributions extend their work in two ways. First, unlike their work, we address the robust case by giving an algorithm which is provably robust to parameter and execution uncertainty. Second, for known parameters, we show that dynamic influence maximization can be solved by a greedy algorithm instead of POMDP and that only greedy scales beyond small networks.

Lastly, Seeman and Singer [21] also consider an adaptive influence maximization problem but our settings are very different. In their work, the goal is to select an initial seed set so that we may then recruit a second set from the friends of the first. In contrast, our problem is adaptive since the algorithm learns more about the network in each stage.

## 5. ROBUSTNESS TO PARAMETER UNCERTAINTY

We start by showing how to obtain robust results under parameter uncertainty. For ease of explanation, we focus first on the nondynamic setting where the algorithm picks a single seed set in the first stage instead of a multistage policy. We also defer consideration of execution uncertainty and assume  $q_v = 1 \forall v \in V$ . When there is no parameter uncertainty, Kempe, Kleinberg and Tardos [11] showed that a greedy algorithm obtains a  $(1 - 1/e)$  approximation to the optimal value because the objective function is submodular. We now show how to extend this approximation to a robust optimization over uncertain parameters.

Recall that we assumed interval uncertainty over the key parameters  $\theta$ , that is,  $\theta \in [a_\theta, b_\theta] \forall \theta \in \Theta$ . The influencer’s goal is to find a seed set  $S \in \mathcal{A}$  which leads to near-optimal influence spread no matter where each entry in  $\theta$  lies within its interval. Our full model also includes lower-order uncertainty concerning the realized value of  $\mathbf{p}$  given  $\theta$ . Since we focus for now on a single stage, the influencer does not receive observations about the realized  $\mathbf{p}$ . Thus (as noted in Section 3), the lower-order uncertainty is irrelevant because distributional uncertainty over  $\mathbf{p}$  can be removed via setting  $\mathbf{p}$  to its mean. This lower order uncertainty becomes relevant when we extend our algorithm to the dynamic setting in Section 6.

First, we formally define our objective. For each  $S \in \mathcal{A}$ , let  $g_S(\theta)$  give the expected influence spread of seed set  $S$  with priors  $\theta$ . Let the set of possible  $\theta$  be  $\mathcal{P} = \{\theta \mid \theta_e \in [a_{\theta_e}, b_{\theta_e}] \forall e \in E\}$ .  $\mathcal{P}$  is the  $m$ -dimensional box where each  $\theta_e$  may lie anywhere within the interval for that type. Let the optimal influence spread for each  $\theta$  be  $g_{OPT}(\theta) = \max_S g_S(\theta)$ . The influencer’s utility for choosing seed set  $S$  when the parameters are  $\theta$  is

$$R(S, \theta) = \frac{g_S(\theta)}{g_{OPT}(\theta)},$$

which quantifies the value that the influencer received compared to the optimum. This objective was also discussed by He and Kempe [9] and Chen et al. [4]. Since  $g_{OPT}$  is NP-hard to compute exactly,  $R$  cannot be efficiently evaluated. However, we can compute a greedy approximation  $g_G$  to the optimal value. Hence, we follow [4, 9] and instead optimize

$$R_G(S, \theta) = \frac{g_S(\theta)}{g_G(\theta)}.$$

This compares the influence spread of  $S$  to the value which would

have been obtained by greedy. Note that  $R(S, \theta) \leq R_G(S, \theta) \leq \frac{e}{e-1} R(S, \theta)$ , so  $R_G$  is always within a constant factor of  $R$ . A randomized algorithm for the influencer is a vector  $x \in \Delta^{|\mathcal{A}|}$  (where  $\Delta^n$  denotes the  $n$ -dimensional probability simplex) giving the probability of selecting each seed set. As explained in Section 3, we take a worst-case approach to maximizing  $R_G$ . Hence, we seek the strategy  $x^*$  which maximizes the minimum value:

$$x^* = \arg \max_{x \in \Delta^{|\mathcal{A}|}} \min_{\theta \in \mathcal{P}} \sum_S x_S R_G(S, \theta). \quad (1)$$

This problem can be viewed as a zero-sum game between an influencer, who chooses the seed set  $S \in \mathcal{A}$ , and an adversary (nature), who chooses the parameters  $\theta \in \mathcal{P}$ . The influencer's payoff is given by  $R_G(S, \theta)$ . Since we solve for a mixed strategy for the influencer, our algorithm is randomized: the influencer samples a pure strategy each time she wishes to choose a seed set. Randomization is necessary to guard against weakness to a particular set of parameters (a player in a game may be exploited if they commit to a deterministic strategy). However, it is possible that an influencer might prefer a pure strategy with a lower but "guaranteed" payoff to a probabilistic strategy which has a small chance of yielding very bad utility. In this case, it suffices to express the payoffs of our game through a risk aversion function and then solve the game normally using the algorithms described below; this will yield a desired guaranteed payoff. In practice, our proposed algorithm produces strategies with very sparse support; the few non-zero components all have comparable worst-case value so this problem does not arise.

We now introduce our techniques for solving the game. Normal methods do not apply because  $\mathcal{A}$  (the set of all sets of  $K$  nodes) is exponentially large, while  $\mathcal{P}$  (the set of all allowable parameter values) is infinite. However, we demonstrate that both sources of intractability can be overcome and present a constant-factor approximation to the minimax robust solution in polynomial time. Recall that a mixed strategy  $x$  is a vector giving the probability of choosing each seed set. Our approximation notion is as follows:

**DEFINITION 1.** *An algorithm produces an  $(\alpha, \epsilon)$ -minimax robust solution if it always returns a mixed strategy  $x$  satisfying*

$$\min_{\theta} \sum_S x_S R_G(S, \theta) \geq \alpha \max_{x'} \min_{\theta} \sum_S x'_S R_G(S, \theta) - \epsilon.$$

We first tackle the adversary's infinite strategy space by showing that this space can be reduced to a discrete set  $\mathcal{P}^*$  such that the adversary loses only an arbitrarily small  $\epsilon$  by choosing from  $\mathcal{P}^*$  instead of  $\mathcal{P}$ . Recall that  $\mathcal{P}$  has infinite size because it is the continuous set of all allowable parameter values. For example, if we have types corresponding to strong and weak edges,  $\mathcal{P}$  could be the set of parameters which places the  $\theta_e$  for strong edges anywhere in the continuous interval  $[0.4, 0.8]$  and the  $\theta_e$  for the weak edges anywhere in the interval  $[0.2, 0.4]$ . There are no general approaches for solving games with infinite strategy spaces, so we show how to reduce our original game to one with a finite set of adversary strategies. The challenge is to ensure that only a small amount of approximation error is incurred in the reduction. Towards this goal, the main technical step is to prove that each  $g_S$  is sufficiently smooth for such a discretization to be possible. It is not immediately clear that  $g_S$  should always be smooth. For instance, in a complete graph,  $g_S$  rapidly increases from 0 to  $n$  even at a very small propagation probability because of the combinatorial number of potential paths (this is related to the emergence of a giant connected component in Erdős-Rényi graphs). However, Lemma 1, stated below, shows that  $g_S$  is smooth for any fixed size graph.

Lemma 2 will then use Lemma 1 to show that there is a discretization  $\mathcal{P}^*$  where  $|\mathcal{P}^*|$  scales reasonably with the problem size.

**LEMMA 1.** *For any  $g_S, S \in \mathcal{A}$ , (or  $g_G$ ) and any  $\theta_1, \theta_2 \in \mathcal{P}$ ,  $|g_S(\theta_1) - g_S(\theta_2)| \leq nT \|\theta_1 - \theta_2\|_1$ .*

The proof of Lemma 1 is fairly long and technical, and so is deferred to the supplemental material (along with the proof of all other lemmas). In Lemma 2, we use Lemma 1 to construct a  $\mathcal{P}^*$  which ensures that the loss incurred by discretizing is at most  $\epsilon$ . The idea is as follows. Suppose we cover the hyperrectangle of allowable type values  $\times_{\theta} [a_{\theta}, b_{\theta}]$  with a regular grid. Let our discretization  $\mathcal{P}^* \subset \mathcal{P}$  be the set of  $\theta$  which correspond to the grid points (by setting each  $\theta_e$  to its value at the point). Then we have:

**LEMMA 2.** *Fix  $\epsilon > 0$ , and construct  $\mathcal{P}^*$  using a grid with  $\left(\frac{2nmT}{\epsilon}\right)^{|\Theta|}$  points. Then for any seed set  $S \in \mathcal{A}$  and any point  $\theta \in \mathcal{P}$  there is a  $\theta^* \in \mathcal{P}^*$  satisfying  $|R_G(S, \theta) - R_G(S, \theta^*)| \leq \epsilon$ .*

Although  $|\mathcal{P}^*|$  is exponential in  $|\Theta|$ , we assume that  $|\Theta|$  is a small constant because organizations can only provide a small number of edge types. Moreover, this exponential dependence is likely unavoidable since when  $|\Theta| = m$ , we recover a model for which He and Kempe [9] showed even approximating the minimax robust strategy is NP-hard.

By reducing the adversary's pure strategy space to a finite set  $\mathcal{P}^*$ , Lemma 2 allows us to formulate the minimax robustness problem as the following LP:

<b>Primal:</b>	<b>Dual:</b>
$\max_{x \in \mathbb{R}^{ \mathcal{A} }} U$ s.t.	$\min_{y \in \mathbb{R}^{ \mathcal{P}^* }} W$ s.t.
$\sum_{S \in \mathcal{A}} x_S = 1$	$\sum_{\theta \in \mathcal{P}^*} y_{\theta} = 1$
$x_S \geq 0 \quad \forall S \in \mathcal{A}$	$y_{\theta} \geq 0 \quad \forall \theta \in \mathcal{P}^*$
$U \leq \sum_{S \in \mathcal{A}} x_S R_G(S, \theta) \quad \forall \theta$	$W \geq \sum_{\theta \in \mathcal{P}^*} y_{\theta} R_G(S, \theta) \quad \forall S$ (2)

The primal is intractable due to the exponential number of variables. However, since  $\mathcal{P}^*$  has size  $\text{poly}(n, m, \frac{1}{\epsilon})$ , we can instead work with the dual. Although the dual has exponentially many constraints, all that we need to ensure polynomial time solvability is a separation oracle which checks if Constraint 2 is violated for any  $S$ . This reduces to finding the "tightest" constraint; i.e., the seed set  $S$  which maximizes the expected value of  $R_G$  under the adversary mixed strategy  $y$ . We want to compute

$$\max_{S \in \mathcal{A}} \sum_{\theta \in \mathcal{P}^*} y_{\theta} R_G(S, \theta) = \max_{S \in \mathcal{A}} \sum_{\theta \in \mathcal{P}^*} \frac{y_{\theta}}{g_G(\theta)} g_S(\theta)$$

Note that the term  $\frac{y_{\theta}}{g_G(\theta)}$  is independent of the seed set  $S$  which we choose; it is just a nonnegative coefficient. Further, computing  $\max_{S \in \mathcal{A}} g_S(\theta)$  is the problem of finding the most influential seed set for the fixed parameters  $\theta$ , which is a submodular maximization problem. Since a nonnegative linear combination of submodular functions is itself submodular, greedy can be used to find an approximately tightest constraint:

**LEMMA 3.** *For any adversary mixed strategy  $y \in \Delta^{|\mathcal{P}^*|}$ , running greedy with the objective  $\max_{S \in \mathcal{A}} \sum_{\theta \in \mathcal{P}^*} \frac{y_{\theta}}{g_G(\theta)} \mathbb{E}_{\mathbf{p} \sim \theta} [f(\cdot, \mathbf{p})]$  produces a  $(1 - 1/e)$ -approximate best response to  $y$ .*

This is exactly what we need to check Constraint 2 since a particular setting of  $y$  corresponds to an adversary mixed strategy, so Constraint 2 is violated if and only if the best response to  $y$  has expected  $R_G$  higher than  $W$ . Although greedy may not always detect violated constraints, we show that the ellipsoid algorithm can still be used to approximately solve the dual, and that we can then recover an approximate solution to the primal.

**THEOREM 1.** *There is an algorithm which obtains an  $(1-1/e, \epsilon)$ -minimax robust solution to the robust influence maximization problem in time  $\text{poly}(m, n, \frac{1}{\epsilon})$ .*

**PROOF.** The dual exchanges an exponential number of variables for an exponential number of constraints. Thus, we can solve it using the ellipsoid algorithm as long as we have an efficient separation oracle for Constraint 2. By Lemma 3, using greedy as a separation oracle will detect if  $W \leq (1-1/e) \max_S \sum_{\theta} y_{\theta} R_G(S, \theta)$  (and explicitly return an  $S$  which gives a violated constraint). The challenge is that greedy may or may not report a violation when  $(1-1/e) \max_S \sum_{\theta} y_{\theta} R_G(S, \theta) < W < \max_S \sum_{\theta} y_{\theta} R_G(S, \theta)$ . We handle this approximation using an argument similar to that employed by Jain, Mahdian, and Salavatipour in the context of a problem related to Steiner trees [10]. Suppose that we solve the dual by adding an additional constraint  $W \leq W^*$  and binary searching for the smallest  $W^*$  which makes the dual feasible. It is known that the ellipsoid algorithm will still terminate in polynomial time even using an approximate separation oracle [2]. If the ellipsoid algorithm claims that  $(W^*, y)$  is dual feasible when using greedy as a separation oracle, we know via the approximation guarantee for the separation oracle that  $(\frac{e}{e-1}W^*, y)$  must be feasible in truth. That is, the optimal solution of the dual lies between  $W^*$  and  $\frac{e}{e-1}W^*$ . This portion of the algorithm runs in time  $\text{poly}(n, m, \frac{1}{\epsilon})$  since at most a polynomial number of calls are made to the separation oracle, which itself runs in polynomial time.

Now we show how to use the dual solution to recover a solution to the primal with polynomial size. When the ellipsoid algorithm is run on the dual, it makes a polynomial number of calls to the separation oracle. Checking only the constraints which correspond to the actions returned by the oracle is sufficient to certify that the optimal solution to the dual is at least  $W^*$ . Hence by strong duality, we know that there is a solution to the primal which uses only the variables corresponding to those actions and achieves value  $W^*$ . If we solve a polynomially-sized version of the primal which uses only those variables, then we obtain such a solution,  $x^*$ . Also by strong duality, we know that the optimal solution to the primal cannot have value better than  $\frac{e}{e-1}W^*$ , so

$$\begin{aligned} \min_{\theta \in \mathcal{P}^*} \sum_S x_S^* R_G(S, \theta) &\geq W^* \\ &= \left(1 - \frac{1}{e}\right) \left(\frac{e}{e-1}W^*\right) \\ &\geq \left(1 - \frac{1}{e}\right) \max_{x'} \min_{\theta \in \mathcal{P}^*} \sum_S x'_S R_G(S, \theta). \end{aligned}$$

Note that by Lemma 2 the value of this solution can decrease by at most  $\epsilon$  if the adversary chooses from  $\mathcal{P}$  instead of  $\mathcal{P}^*$ , which establishes the claim.  $\square$

**A practical algorithm:** Theorem 1 accomplishes our goal of providing a polynomial time approximation algorithm. However, this algorithm is likely impractical since the ellipsoid algorithm is known to perform poorly. Moreover, the strategy returned could be quite large (although polynomial). While we view the construction

of a polynomial time approximation algorithm as an important theoretical contribution, we also present a practical algorithm using a double oracle approach [14]. DOSIM (Double Oracle for Social Influence Maximization) solves the game using best response oracles for each player. Initially, both players choose only from a small set of arbitrarily chosen pure strategies (lines 1-2 of Algorithm 1). At

---

#### Algorithm 1 DOSIM

---

```

1:  $\mathcal{A}_0 =$  arbitrary seed set
2:  $\mathcal{P}_0 = \{(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})\}$ 
3:  $i = 0$ 
4: while  $\mathcal{A}_i \neq \mathcal{A}_{i-1}$  or  $\mathcal{P}_i \neq \mathcal{P}_{i-1}$  do
5:    $x_i, y_i = \text{SolveLP}(\mathcal{A}_i, \mathcal{P}_i)$  //Solve restricted game
6:    $\mathcal{A}_{i+1} = \mathcal{A}_i \cup \{\text{INFLUENCERBESTRESPONSE}(y_i)\}$ 
7:    $\mathcal{P}_{i+1} = \mathcal{P}_i \cup \{\text{ADVERSARYBESTRESPONSE}(x_i, \epsilon)\}$ 
8:    $i = i + 1$ 
9: end while
10: return  $\text{SolveLP}(\mathcal{A}_i, \mathcal{P}_i)$ 

```

---

each iteration, we compute the minimax mixed strategies assuming the game is restricted to just the current strategy set (line 5). The payoff matrix of this restricted game is given by  $R_G(S, \theta)$  for each pair of strategies  $S \in \mathcal{A}_i, \theta \in \mathcal{P}_i$  contained in the current strategy sets. Then, we find the best response of each player to their opponent's mixed strategy (lines 6-7). If either of these responses is not in the corresponding strategy set, the new entry is added and the algorithm proceeds. If both strategies are present, the algorithm terminates with a provably optimal solution. While double oracle is well known [14], it is not an out-of-the-box approach. Our contribution is constructing appropriate oracles for robust influence maximization. Both oracles follow naturally from our earlier results:

**Influencer oracle:** Run the greedy algorithm on the objective given by the current adversary strategy  $y$ . Lemma 3 shows that greedy is a  $(1-1/e)$ -approximate oracle.

**Adversary oracle:** Search  $\mathcal{P}^*$  and choose the parameters  $\theta \in \mathcal{P}^*$  which minimize  $\sum_{S \in \mathcal{A}_{i-1}} x_S R_G(S, \theta)$ . By Lemma 2, this approximates the adversary's best response to within an additive  $\epsilon$ .

Hence, we can provide an approximation guarantee for the best response oracle for both players. McMahan et al. [14] showed that when double oracle is given an *optimal* best response oracle for each player, the strategies it returns form an equilibrium of the game. We now generalize this reasoning to show that *approximate* best response oracles yield an approximate equilibrium. We say that a mixed strategy  $x$  is a  $c$ -approximate minimax strategy if it obtains worst case utility within a factor  $c$  of the optimum:

$$\min_{\theta \in \mathcal{P}^*} \sum_{S \in \mathcal{A}} x_S R_G(S, \theta) \geq c \max_{x' \in \Delta^{|\mathcal{A}|}} \min_{\theta \in \mathcal{P}^*} \sum_{S \in \mathcal{A}} x'_S R_G(S, \theta)$$

Likewise, a mixed strategy  $y$  for the adversary (the minimizing player) is a  $c$ -approximate minimax strategy if the influencer can obtain utility at most a factor  $\frac{1}{c}$  greater than the value of the game in response to it. We prove the following guarantee on the output of the double oracle algorithm:

**THEOREM 2.** *If double oracle is given an  $\alpha$ -approximation to the best response of Player 1 and a  $\beta$ -approximation to the best response of Player 2, on termination it returns  $\alpha\beta$ -approximate minimax strategies for both players.*

**PROOF.** Consider a zero sum game. We will retain our earlier notation, letting  $\mathcal{A}$  be the pure strategy space for Player 1 and  $\mathcal{P}^*$  be the pure strategy space for Player 2, with payoff function  $R_G$ . However, all of our reasoning applies to any zero sum game. Sup-

pose that the double oracle algorithm terminates with mixed strategies  $x$  and  $y$ . We will prove the theorem for the maximizing player (assumed to be Player 1). The proof for the minimizing player is analogous. Since  $x$  is an  $\alpha$ -best response to  $y$ , for any alternate strategy  $S' \in \mathcal{A}$ ,

$$\sum_{\theta \in \mathcal{P}^*} y_{\theta} R_G(S', \theta) \leq \frac{1}{\alpha} \sum_{S \in \mathcal{A}} \sum_{\theta \in \mathcal{P}^*} x_S y_{\theta} R_G(S, \theta).$$

Similarly, for any alternate adversary strategy  $\theta'$ ,

$$\sum_{S \in \mathcal{A}} x_S R_G(S, \theta') \geq \beta \sum_{S \in \mathcal{A}} \sum_{\theta \in \mathcal{P}^*} x_S y_{\theta} R_G(S, \theta).$$

Combining these inequalities yields that:

$$\begin{aligned} \max_{x' \in \Delta^{|\mathcal{A}|}} \min_{\theta' \in \mathcal{P}^*} \sum_{S \in \mathcal{A}} x'_S R_G(S, \theta') &\leq \max_{x'} \sum_{S \in \mathcal{A}} \sum_{\theta \in \mathcal{P}^*} x_S y_{\theta} R_G(S, \theta) \\ &\leq \frac{1}{\alpha} \sum_{S \in \mathcal{A}} \sum_{\theta \in \mathcal{P}^*} x_S y_{\theta} R_G(S, \theta) \\ &\leq \frac{1}{\alpha\beta} \min_{\theta' \in \mathcal{P}^*} \sum_{S \in \mathcal{A}} x_S R_G(S, \theta') \end{aligned}$$

which completes the proof.  $\square$

By taking  $\beta = 1$  and accounting for the additive loss of  $\epsilon$  introduced by discretizing the adversary's strategy space, we get:

**COROLLARY 1.** *DOSIM obtains a  $(1 - 1/e, \epsilon)$ -minimax robust solution to the robust influence maximization problem.*

DOSIM may require an exponential number of steps to converge (since it may need to add all pure strategies in the worst case). However, we see in Section 7 that it converges quickly in practice and so Corollary 1 provides guaranteed solution quality.

**Robustness to execution uncertainty:** Recall that each node  $v$  is present with probability  $q_v$ . To illustrate the implication for the optimal strategy, suppose that there are two very influential nodes  $u$  and  $v$  with  $q_u = q_v = q$  whose neighborhoods entirely overlap. Clearly, we should only select one of the two if  $q$  is high. However, if  $q$  is low, we might select both to ensure that we reach their large set of neighbors. We create a new graph  $G'$  to represent this tradeoff as follows. For each node  $v \in V$ , add a new "designer" node  $v'$ , which has a single edge  $(v', v)$ . This edge has a new type  $\theta_d$  whose prior outputs  $p_{v',v} = 1$  with probability  $\theta_d = q_v$  and  $p_{v',v} = 0$  otherwise. If actions are restricted to the designer nodes, i.e.,  $\mathcal{A} = \{\{v'_1, v'_2, \dots, v'_K\} | v_i \in V\}$ , then maximizing influence in  $G'$  exactly corresponds to maximizing influence under execution errors in  $G$ . We can also incorporate our higher-level uncertainty by allowing  $\theta_d$  to be adversarially chosen within an interval. Therefore, we can compute a seed set which is robust to an adversarially chosen  $\theta_d$  by running DOSIM on  $G'$ .

## 6. DOSIM IN THE DYNAMIC SETTING

We now extend DOSIM to the dynamic setting, where we seek a multistage policy instead of a seed set. Recall that a policy selects a set of nodes at each stage conditioned on observations received from intervention participants in earlier stages. We will show how to find a policy which is robust to uncertainty over the prior  $\theta$ . When we seek a robust policy, observations give a quantile of the prior but do not fully specify the posterior since the exact prior is unknown. E.g., if  $\theta_e \in [0.2, 0.7]$  and  $w = 0.2$ , the prior could be either  $U[0.2, 0.4]$  or  $U[0.5, 0.7]$  and an observation of "top half"

implies a posterior of either  $U[0.3, 0.4]$  or  $U[0.6, 0.7]$ . Our algorithm uses this information while remaining robust to the higher-level uncertainty over  $\theta$ . Recall that the lower-level uncertainty cannot be replaced by setting  $p_e = \theta_e$  anymore because the algorithm receives observations about the realized value. Thus, our algorithm must simultaneously handle both levels of uncertainty.

DOSIM translates naturally into the dynamic setting. Specifically, our scheme for discretizing the adversary's strategy space (Lemmas 1 and 2) generalizes to the case where the influencer chooses a multistage policy instead of a single seed set (see supplemental material for proof). Thus, our adversary oracle, and its theoretical guarantees, are unchanged. Theorem 2 applies to any zero-sum game, and hence implies that if we can supply an influencer oracle for the dynamic setting, then running DOSIM with this oracle will produce robust policies.

Unfortunately, supplying an influencer oracle with guaranteed approximation ratio is difficult; there are no known theoretical guarantees for dynamic influence maximization because the objective function is not adaptive submodular. Yadav et al. [26] proved this for their DIME model, which our model generalizes. Our approach is to use a heuristic oracle and show experimentally that it produces good influence spread. This is sufficient to ensure that we obtain robust results on real world problems since DOSIM performs well on a particular network so long as its influencer oracle does:

**LEMMA 4.** *If the influencer oracle achieves an  $\alpha$ -approximation for any  $\theta \in \mathcal{P}^*$  on a specific graph  $G$ , then DOSIM provides an  $(\alpha, \epsilon)$ -minimax robust solution on  $G$ .*

Essentially, we can be sure that DOSIM's solution is robust if the oracle performs well on our particular network since DOSIM preserves the quality of its input oracle.

We now turn to finding an appropriate heuristic for the influencer oracle. One natural idea would be to use the current state of the art for dynamic influence maximization, the POMDP based heuristic algorithm HEAL [26]. However, HEAL is unsuitable for two reasons. First, our model is more general than HEAL's. Specifically, HEAL cannot represent the  $p_e$  as being drawn from an interval. Second, we show experimentally that HEAL is not scalable and is hence unsuitable as a subroutine which will be called many times by DOSIM. Hence, we use an alternative heuristic oracle, presented below.

When examining Algorithm 1, our extension of DOSIM to the dynamic setting only needs to supply an influencer oracle (line 6). The rest of DOSIM is completely unchanged because we can use the same adversary oracle. For the influencer oracle, which best responds to an adversary mixed strategy  $y$ , we will use a dynamic version of the greedy influence maximization algorithm, described in Algorithm 2. The dynamic greedy algorithm operates greedily on two layers. In each stage (line 3), it chooses the set of  $K$  nodes which maximize the immediate gain in influence spread, conditioned on the observations gathered thus far. However, since finding this set is essentially a classical influence maximization problem (and hence NP-hard), we use a further greedy selection process to approximate the optimal set of  $K$  nodes for each stage. Line 5 selects the node  $v^*$  which maximizes the expected marginal gain. The expectation is over parameters  $\theta$  sampled from the adversary mixed strategy  $y$ , and propagation probabilities  $p$  sampled according to  $\theta$ . After identifying these  $K$  nodes, the algorithm marks them as influenced and simulates new observations (lines 10-11). We can repeatedly simulate different sets of observations (repeating line 10) to build up the best response policy. Thus, we can construct an influencer oracle.

Explicitly representing the entire policy is infeasible because a

---

**Algorithm 2** Dynamic greedy
 

---

```

1:  $S_{\text{prev}} = \emptyset$  //nodes selected in previous stages
2:  $O = \emptyset$  //set of all observations
3: for  $t \leftarrow 1$  to  $T$  do
4:    $S_t = \emptyset$  //nodes selected in this stage
5:   for  $i \leftarrow 1$  to  $K$  do
6:      $v^* = \arg \max \mathbb{E}_{\mathbf{p}} [f(S_{\text{prev}} \cup S_t \cup \{v\}, \mathbf{p})$ 
7:        $\quad \quad \quad - f(S_{\text{prev}} \cup S_t, \mathbf{p}) | O]$ 
8:      $S_t = S_t \cup \{v^*\}$ 
9:   end for
10:  Influence  $S_t$ ; Update  $S_{\text{prev}} = S_{\text{prev}} \cup S_t$ 
11:  Receive observation  $O_t$ ; Update  $O = O \cup O_t$ 
12: end for
  
```

---

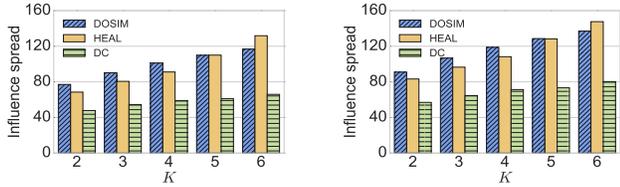


Figure 2: Influence with varying  $K$  on Network A (left) and Network B (right).

policy specifies the action to take in response to *any* possible set of observations; even storing it would require exponential space. However, the oracle simulates a *sample* of random observations, which is sufficient to implement DOSIM. Suppose we store only the adversary mixed strategy  $y_t$  that the oracle best responds to on each iteration  $t$ . In future iterations, we can run the oracle with input  $y_t$  and sample from the policy generated in iteration  $t$  as needed.

## 7. EXPERIMENTS

We present experimental results for our example domain, preventing HIV spread in homeless youth via dynamic influence maximization under parameter uncertainty. We use two datasets (Network A and Network B) collected by homeless shelters using surveys and interviews [26]. Both have 140-170 nodes and 300-400 edges. We also show results on artificial Watts-Strogatz networks (model parameters  $p = 0.1$ ,  $k = 7$ , all results averaged over 30 networks), since these mirror our datasets' small diameter.

### Dynamic influence maximization with known parameters:

The most closely related work to ours is HEAL algorithm [26], so we begin by setting up a comparison between DOSIM and HEAL. HEAL only handles the known parameter case, where only the lower order uncertainty over the realized value of  $\mathbf{p}$  is present. Thus, we run DOSIM assuming that the adversary strategy is fixed

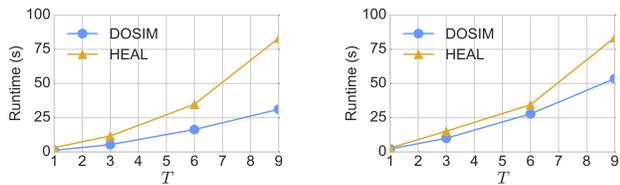


Figure 3: Runtime with varying  $T$  on Network A (left) and Network B (right).

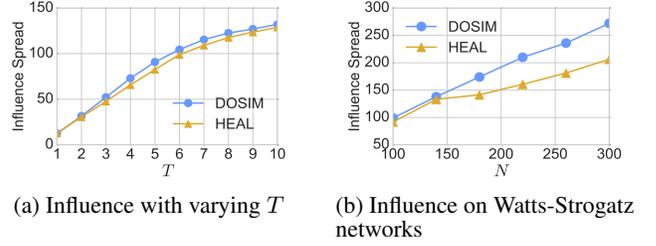


Figure 4: Additional influence spread results

$p \setminus u$	0.1	0.2	0.3	$p \setminus u$	0.1	0.2	0.3
0.5	-0.1	15.0	6.0	0.5	23.1	-1.8	10.5
0.6	6.4	13.3	24.9	0.6	0.6	18.7	18.3
0.7	8.9	21.0	-3.4	0.7	-0.4	2.5	6.7

(a) Network A (b) Network B

Table 1: % gain over HEAL with varying parameters

to a single, known  $\theta$ . Our results validate that even when only lower-level uncertainty is present, the approach used by DOSIM still outperforms the closest competitor.

Our model is more general than HEAL's, so we align the models as follows. There is one type of "certain" edges with fixed  $p_e = p$ , and one type of "uncertain" edges which have  $p_e = p$  with probability  $u$  and  $p_e = 0$  with probability  $1 - u$ . We give DOSIM the same observations as HEAL by revealing whether an edge is in the top  $100 \cdot u\%$  of the prior or bottom  $100 \cdot (1 - u)\%$ . Figure 2 shows the influence spread of both algorithms on the two real networks as we vary the intervention size  $K$  (fixing  $T = 5$ ,  $p = 0.6$ ,  $u = 0.1$  as in [26]). We also compare to picking nodes with the highest degree centrality (DC), since this is standard in health policy [24]. The maximum intervention size is  $K = 6$  since the shelter cannot accommodate more participants.

We see that DOSIM and HEAL perform very similarly, while DC performs poorly (and is omitted hereafter). Next, Figure 4a shows influence spread on Network A as the time horizon  $T$  varies ( $K = 2$ ,  $p = 0.6$ ,  $u = 0.1$ ); again the algorithms perform very similarly. Results for Network B are almost identical (see supplemental material). Figure 4b shows influence spread on Watts-Strogatz networks of varying size. DOSIM performs better as the size increases. Lastly, Table 1 shows the percentage gain of DOSIM over HEAL as the parameters  $u$  and  $p$  are varied (for  $K = 2$ ,  $T = 10$ ). DOSIM performs no more than 4% worse, and up to 25% better. Overall, the influence obtained by DOSIM is comparable to HEAL, and sometimes better.

However, only DOSIM is scalable. Figures 3 and 5 show runtime on our two real-world networks and Watts-Strogatz networks

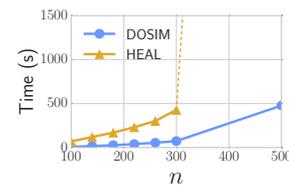


Figure 5: Runtime on Watts-Strogatz networks.

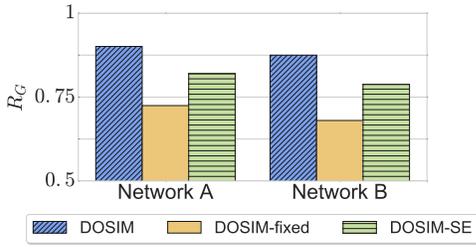


Figure 6: Robustness comparison. Each bar gives the worst-case value of  $R_G$  for the corresponding algorithm.

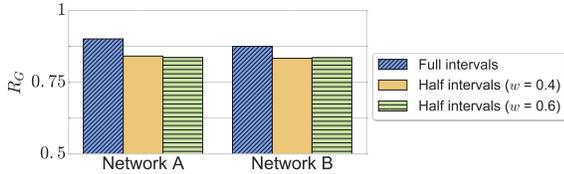


Figure 7: DOSIM's performance with different intervals for  $\theta$  and values of  $w$ .

respectively. DOSIM performs better as the problem size grows. For  $T = 10$  on Network A, it is roughly 2.5 times faster. On Watts-Strogatz networks with 500 nodes, DOSIM finishes in about 8 minutes while we cut off HEAL's execution after 5 hours. For networks larger than 500 nodes, HEAL runs out of memory. We do not attempt to run either algorithm on larger networks because in the domains we are concerned with network sizes will typically be in the small hundreds of nodes. Both algorithms were run on a 2.33 GHz Intel processor with 48 GB of RAM.

**Robustness:** We now investigate the performance of the full DOSIM algorithm when we have interval uncertainty over  $\theta$ . Shelter officials annotated the edges as being either strong or weak ties, so we have two corresponding types. We add a third type for the designer edges reflecting execution uncertainty. A uniform prior is adversarially chosen for the first two types, using width  $w = 0.4$  (truncated if the support would exceed  $[0,1]$ ). We allow any mean  $\theta_{\text{strong}} \in [0.2, 0.8]$  and  $\theta_{\text{weak}} \in [0, 0.4]$ . These ranges contain the parameters used in previous work [26], but are large enough to represent our considerable uncertainty. We allow any  $\theta_d \in [0.2, 0.8]$ . Observations specified whether outgoing edges were in the bottom or top quantile of the prior.  $K = 4$  nodes were selected in each of  $T = 5$  stages with  $\epsilon = 0.1$ . We compare DOSIM to two benchmarks. Since no previous work (including HEAL) handles both levels of uncertainty simultaneously, our benchmarks run DOSIM with a fixed adversary strategy. "DOSIM-fixed" uses fixed parameter settings reported previously ( $\theta_{\text{strong}} = 0.6$ ,  $\theta_{\text{weak}} = 0.05$ ) without considering execution uncertainty ( $\theta_d = 1$ ). "DOSIM-SE" uses the same settings for  $\theta_{\text{strong}}$  and  $\theta_{\text{weak}}$ , but also accounts for stochastic execution uncertainty by setting  $\theta_d = 0.5$  (based on shelter officials' experience). These two algorithms assume a single allowable point value for  $\theta$ . Figure 6 shows the worst case ratio  $R_G$  (over all allowable parameter values) on the  $y$  axis for DOSIM and the benchmarks. DOSIM obtains approximately 90% of the optimal value across all parameter settings while using fixed parameters leads to substantially worse performance (about 72% in the worst case). Incorporating execution uncertainty results in a value roughly halfway in between.

Next, we examine the consequences when DOSIM's uncertainty intervals for  $\theta$  might not contain the true value. Specifically, we run

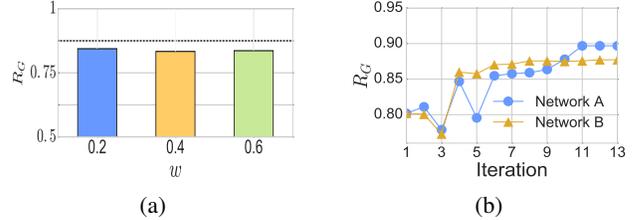


Figure 8: (a) Worst-case  $R_G$  over larger intervals for  $\theta$  when DOSIM plans on smaller intervals. The dashed line gives  $R_G$  when DOSIM plans on larger intervals. (b) Convergence of DOSIM.

DOSIM with half-sized uncertainty intervals:  $\theta_{\text{strong}} \in [0.35, 0.65]$ ,  $\theta_{\text{weak}} \in [0.1, 0.3]$ ,  $\theta_d \in [0.35, 0.65]$ , and show the worst case value of  $R_G$  when  $\theta$  could lie anywhere in the larger intervals used above. The first two bars of Figure 7 compare the performance of DOSIM with half-sized intervals to its performance when given the true, larger intervals with  $w = 0.4$ . DOSIM's performance with half-sized intervals is approximately 84% of the optimal value, compared to 90% given the true intervals. Hence, DOSIM performs fairly well even when its uncertainty intervals are misspecified. However, we cannot make up this gap (which is caused by underestimating the higher-order uncertainty) by increasing our amount of lower-level uncertainty. The last bar in Figure 7 shows that DOSIM's performance is essentially equivalent if  $w$  is increased from 0.4 to 0.6.

Expanding on this point, Figure 8a shows the value of  $R_G$  which DOSIM obtains on Network A when it plans on half-sized intervals for  $\theta$ , with each bar representing a different  $w$ . The dashed horizontal line gives DOSIM's performance when it correctly planned on the larger uncertainty intervals ( $\theta_{\text{strong}}, \theta_d \in [0.2, 0.8]$  and  $\theta_{\text{weak}} \in [0, 0.4]$ ). We infer that increasing the amount of lower-order uncertainty is not a substitute for incorporating higher-order uncertainty. Results for Network B are similar and are shown in the supplemental material. This confirms that our more expressive model, ranging over both levels of uncertainty, captures features of the problem which cannot be represented using only lower-order uncertainty.

Lastly, Figure 8b shows DOSIM's performance at each iteration (for large uncertainty intervals). On both datasets, it converges rapidly (within 13 iterations). Convergence is similar for half-sized intervals, and given in the supplemental material. This confirms our earlier claims that DOSIM only needs a small number of iterations in practice and that the strategies it generates have sparse support.

## 8. CONCLUSION

We address dynamic influence maximization under uncertainty about both the network parameters and the efficacy of interventions. First, we give algorithms with provable guarantees for minimax robustness under unknown parameters. Second, our algorithm handles execution uncertainties. Third, we extend these results to the dynamic setting using an experimentally validated greedy algorithm. Lastly, experiments on real world networks collected from homeless youth demonstrate our approach's advantages.

## Acknowledgments

This research was supported by MURI Grant W911NF-11-1-0332. Wilder was supported by a NSF Graduate Research Fellowship. Rice was supported by NIMH Grant number R01-MH093336.

## REFERENCES

- [1] J. Brown and P. Reingen. Social ties and word-of-mouth referral behavior. *Journal of Consumer Research*, 14(3):350–362, 1987.
- [2] Y. Cai, C. Daskalakis, and S. M. Weinberg. Reducing revenue to welfare maximization: Approximation algorithms and other generalizations. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 578–595. Society for Industrial and Applied Mathematics, 2013.
- [3] CDC. HIV surveillance report. 26, 2014.
- [4] W. Chen, T. Lin, Z. Tan, M. Zhao, and X. Zhou. Robust influence maximization. *CoRR*, abs/1601.06551, 2016.
- [5] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.
- [6] J.-P. Cointet and C. Roth. How realistic should knowledge diffusion models be? *Journal of Artificial Societies and Social Simulation*, 10(3):5, 2007.
- [7] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *KDD*, 2010.
- [8] A. Goyal, F. Bonchi, and L. V. Lakshmanan. A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5(1):73–84, 2011.
- [9] X. He and D. Kempe. Robust influence maximization. *CoRR*, abs/1602.05240, 2016.
- [10] K. Jain, M. Mahdian, and M. R. Salavatipour. Packing steiner trees. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 266–274. Society for Industrial and Applied Mathematics, 2003.
- [11] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.
- [12] D. Kim et al. Social network targeting to maximise population behaviour change: a cluster randomised controlled trial. *The Lancet*, 386(9989):145–153, 2015.
- [13] D. McAdam and R. Paulsen. Specifying the relationship between social ties and activism. *American Journal of Sociology*, 99(3):640–667, 1993.
- [14] H. B. McMahan, G. J. Gordon, and A. Blum. Planning in the presence of cost functions controlled by an adversary. In *ICML*, pages 536–543, 2003.
- [15] H. Narasimhan, D. C. Parkes, and Y. Singer. Learnability of influence in networks. In *NIPS*, 2015.
- [16] P. Netrapalli and S. Sanghavi. Learning the graph of epidemic cascades. *SIGMETRICS Perform. Eval. Rev.*, 40(1):211–222, June 2012.
- [17] N.H. Council. HIV/AIDS among persons experiencing homelessness: Risk factors, predictors of testing, and promising testing strategies. 2012.
- [18] E. Rice. The positive role of social networks and social networking technology in the condom-using behaviors of homeless young people. *Public Health Reports*, pages 588–595, 2010.
- [19] E. Rice, A. Fulginiti, H. Winetrobe, J. Montoya, A. Plant, and T. Kordic. Sexuality and homelessness in Los Angeles public schools. *American Journal of Public Health*, 102(2):200a–3201, 2012.
- [20] E. Rice, E. Tulbert, J. Cederbaum, A. B. Adhikari, and N. G. Milburn. Mobilizing homeless youth for HIV prevention: a social network analysis of the acceptability of a face-to-face and online social networking intervention. *Health Education Research*, 2012.
- [21] L. Seeman and Y. Singer. Adaptive seeding in social networks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 459–468. IEEE, 2013.
- [22] F. Starkey, S. Audrey, J. Holliday, L. Moore, and R. Campbell. Identifying influential young people to undertake effective peer-led health promotion. *Health Education Research*, 2009.
- [23] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 75–86. ACM, 2014.
- [24] T. W. Valente. Network interventions. *Science*, 337(6090):49–53, 2012.
- [25] S. Vaswani, L. Lakshmanan, and M. Schmidt. Influence maximization with bandits. In *NIPS*, 2015.
- [26] A. Yadav, H. Chan, A. Jiang, H. Xu, E. Rice, and M. Tambe. Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty. In *AAMAS*, 2016.
- [27] A. Yadav, L. Marcolino, E. Rice, R. Petering, H. Winetrobe, H. Rhoades, M. Tambe, and H. Carmichael. Preventing HIV spread in homeless populations using PSINET. In *IAAI*, 2015.
- [28] A. Yadav, B. Wilder, R. Petering, E. Rice, and M. Tambe. Influence maximization in the field: The arduous journey from emerging to deployed application. In *In Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), 2017*, 2017.