

Bayesian Security Games for Controlling Contagion

Jason Tsai, Yundi Qian,
and Milind Tambe
University of Southern California
Los Angeles, CA
{jasontts,yundi.qian,tambe}@usc.edu

Yeveniy Vorobeychik
Sandia National Laboratories
Livermore, CA
yvorobe@saniam.gov

Christopher Kiekintveld
University of Texas at El Paso
El Paso, TX
cdkiekintveld@utep.edu

ABSTRACT

Influence blocking games have been used to model adversarial domains with a social component, such as counterinsurgency. In these games, a mitigator attempts to minimize the efforts of an influencer to spread his agenda across a social network. Previous work has assumed that the influence graph structure is known with certainty by both players. However, in reality, there is often significant information asymmetry between the mitigator and the influencer. We introduce a model of this information asymmetry as a two-player zero-sum Bayesian game. Nearly all past work in influence maximization and social network analysis suggests that graph structure is fundamental in strategy generation, leading to an expectation that solving the Bayesian game exactly is crucial. Surprisingly, we show through extensive experimentation on synthetic and real-world social networks that many common forms of uncertainty can be addressed near-optimally by ignoring the vast majority of it and simply solving an abstracted game with a few randomly chosen types. This suggests that optimal strategies of games that do not model the full range of uncertainty in influence blocking games are typically robust to uncertainty about the influence graph structure.

I INTRODUCTION

Social contagion has long been of great interest in the literature on marketing, the spread of rumors, and, recently, in the context of Arab Spring [1–3]. Our specific focus is on counterinsurgency, which we view as a competition for the support of local leaders. Counterinsurgency can be modeled as a game with two strategic players, the insurgents and the peacekeepers, in which the insurgents aim to spread their views, unrest, etc. among the local population, while the peacekeepers wish to minimize the resulting contagion by engaging in their own influence campaign [4–6]. The key *computational* question we address is: given limited resources, how to select which of the local leaders to influence to minimize the global impact of the insurgency.

These ‘influence blocking’ games have received recent attention in the security games literature [6], where they have been modeled using graphs with nodes representing the tribal leaders and edges representing possible transmission of influence. However, this line of work has assumed that full information about network structure is available to both players. In practice, informational challenges abound in counterinsurgency, where the insurgents are typically an indigenous group that has an informational advantage, and the mitigators are often uncertain about the the social network [4].

We model counterinsurgency as an influence blocking game with asymmetric information. Specifically, we assume that the influencer (an insurgent group) has perfect knowledge of the influence graph structure, while the mitigator is uncertain about it. In the resulting Bayesian game, an influencer type is a particular instantiation of the influence graph, and the mitigator must reason over the distribution over these graphs (i.e., influencer types) in order to compute an optimal strategy.

Past work in influence maximization and social network analysis highlights the importance of graph structure in strategy generation [1, 7, 8]. In addition, previous work on Bayesian security games has shown that not accounting for even small degrees of payoff uncertainty can lead to large drops in solution quality [9]. Thus, we expect strategies generated without modeling most of the uncertainty about graph structure to do far worse than the optimal solution to the Bayesian game. Supporting this intuition, we show that there are cases where a mitigator who has incorrect information about a single edge can suffer unbounded loss and that quantifying the impact of changing a single edge in a given graph is #P-Hard. We also show empirically that, indeed, under our models of uncertainty, optimal mitigator strategies for different influencer types are vastly different.

However, while past work has focused on sophisticated algorithms for Bayesian security games [9–11], we showcase the opposite approach that runs directly counter to what intuition and our initial experiments

suggest: ignoring the vast majority of uncertainty has minimal impact. Specifically, we show through extensive experiments that computing a mitigation strategy based on a game with only a few *randomly* sampled influencer types yields near-optimal rewards for widely varied models of uncertainty. We experiment on 3 different synthetic graph models with and without resource imbalances on both sides, 5 models of uncertainty, weighted/unweighted counting of nodes, varied edge weight distributions, varied graph sizes, varied degrees of uncertainty, and varied degrees of sampling. We also conduct experiments on two real-world social networks using two different models graph construction. In all, we studied over 200 experimental settings and consistently observe the same result: simple sampling techniques perform near-optimally. This suggests that even in domains as challenging as ours, models which ignore uncertainty may nevertheless be robust to it.

II MODEL

1 ASYMMETRIC INFORMATION GAME

We model counterinsurgency as a two-player Bayesian zero-sum game situated on a graph in which two players, the influencer (denoted by I) and the mitigator (denoted by M) compete to maximize influence over the nodes. Formally, let $G = (V, E)$ be a graph with weighted nodes V and edges E , and for each edge $(i, j) \in E$, let p_{ij} be the probability that node i 's opinion will influence node j . Suppose that the influencer initially attempts to influence a subset of nodes $S_I \subseteq V$ to his cause, and the mitigator's initial influence is aimed at a subset of nodes $S_M \subseteq V$. We model propagation of influence in the graph as a synchronized independent cascade process [1, 6] as follows. For nodes $v \in S_I \cap S_M$ which both players initially try to influence, initial "activation" (e.g., actual opinion adoption) happens in either player's favor with equal probability, while all the remaining nodes adopt the view of (are activated by) the player who directly targets them. Next, we activate *all* edges (i, j) in the graph with the corresponding probability of influence, p_{ij} , yielding a subgraph upon which influence can spread. At that point, the influence process proceeds through a sequence of iterations. In each iteration, if a node j has not yet adopted an opinion but has active edges to neighbors who have, j either adopts the opinion of these neighbors when it is unanimous, or adopts each opinion with equal probability if j 's active neighbors disagree. Viewing now the initial target nodes S_I and S_M as the strate-

gies of the players I and M respectively, let $\sigma(S_I, S_M)$ be the expected value of nodes that adopt the influencer's opinion following the independent cascade process described above. We define the utility of the influencer to be $U_I(S_I, S_M) = \sigma(S_I, S_M)$.

We now depart from the model of Tsai et al. [6] by relaxing the complete/symmetric information assumption. Specifically, we assume that the influencer knows the actual influence graph G exactly, while the mitigator is uncertain about its true structure, and only knows the probability distribution over possible graphs. Let λ be an index identifying a particular graph G_λ , and let us make explicit the dependence of the expected influence on the graph, denoting it by $\sigma(S_I, S_M, \lambda)$. Finally, we denote by P the probability distribution over λ , with P_λ being the probability that the true graph is the one identified by λ . From the mitigator's perspective, the influencer's decision will depend on his type; that is, on the true graph which the influencer observes. Thus, we view the influencer's strategy S_I as a function of λ , with S_I^λ denoting the influencer's strategy when his type is λ . The mitigator's utility is then $U_M(S_I, S_M) = -E_{\lambda \sim P}[\sigma(S_I^\lambda, S_M, \lambda)]$.

2 MODELS OF NETWORKS AND UNCERTAINTY

Numerous stochastic generative models for graphs have been proposed to generate synthetic instances that resemble real social networks [12]; some of the best known examples are the preferential attachment process, which generates scale-free graphs [13], and the process of generating small-world networks pioneered by Watts and Strogatz [14]. Recently, a new generative model, BTER, has been developed, and the authors convincingly demonstrated that this model matches the important properties of real-world networks, such as the distribution of degrees and clustering coefficients, far better than alternatives [15]. BTER graphs feature a scale-free collection of densely clustered community structures (dense Erdős-Rényi subgraphs), which are sparsely interconnected by 'inter-community' edges. We conducted experiments on BTER graphs (including variations in community density and interconnectedness), small-world graphs (Watts-Strogatz), preferential attachment graphs, and real-world networks from two villages in India.

We consider several ways to model the mitigator's uncertainty about the graph. *Influential Node* un-

certainty models uncertainty about which nodes are most connected, motivated by the fact the identity of the most socially connected and influential individuals is a function of the local culture which is more familiar to the influencer than the mitigator. Specifically, we start with a baseline graph, then, for each type, choose a set of j nodes and add k new randomly chosen edges from each of these nodes to others. It is important to note that in BTER graph, these j nodes are the *only* nodes that can potentially have inter-community edges under this uncertainty. These inter-community edges are particularly important in contagion games because they enable the spread of influence across groups. The second model, *Random Edge* uncertainty, is the simplest: the mitigator has perfect information about the nodes in the graph, and is uncertain about which edges out of a given set exist. The third model of uncertainty, *Inter-community Edge* uncertainty, models the mitigator's uncertainty about a subset of the inter-community edges. The fourth model of uncertainty, *Inter/Intra-Community Edge* uncertainty, models uncertainty about a combination of inter-community *and* intra-community edges and addresses the concern that Inter-community Edge Uncertainty may provide additional information by being restricted to the critical inter-community edges. Note that in Inter-community Edge, Inter/Intra-Community Edge, and Random Edge uncertainty, we have a type λ for each possible subset of uncertain edges in the graph so the number of types could be as large as $2^{|E|}$. The fifth model of uncertainty, *Inter-community Edge Set* uncertainty, models uncertainty over which *set* of inter-community edges exists (i.e., which set of 8 edges exists). The final three uncertainties, which highlight inter-community edges, apply only to BTER graphs.

The counterinsurgency literature [4] makes clear that military intelligence explicitly performs ‘intelligence preparation of the battlefield (IPB)’ to ascertain the structure and dynamics of a local population with high fidelity. Therefore, we are not interested in cases in which the entire social network is largely unknown or misunderstood. Instead, we focus on situations with a generally correct social network in which uncertainty is about the details of the network structure.

III THE CHALLENGES OF UNCERTAINTY

The first question is whether we can bound the impact of a small amount of uncertainty, as that may help us bound the resulting loss in solution quality.

We show that, in general, ignoring uncertainty can yield an arbitrarily poor solution for the mitigator. Consider the graph shown in Figure 1 in which the edge from A to B is uncertain, $N > M$, and both players have a single resource. Suppose that the influencer chooses to influence node A with probability 1. If the mitigator mistakenly assumes the edge does not exist, then his best response is to influence node C with probability 1, but his actual loss amounts to $\frac{N}{2}$ as compared to the true best-response, for a sufficiently large N , of playing B ($\frac{M}{2}$). A similar situation arises when the mitigator assumes the opposite. Thus, since N is arbitrary, by ignoring the uncertainty of just a single edge the mitigator can suffer unbounded loss.

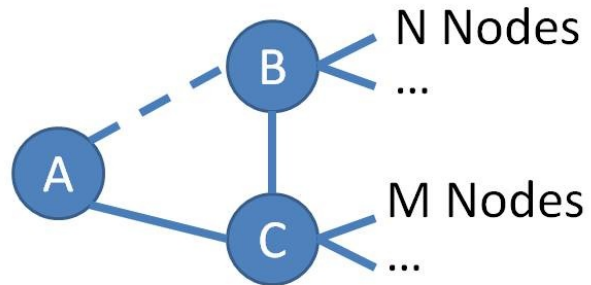


Figure 1: *Schematic representation of the graph that yields unbounded loss.*

The network in the above example is rather artificial, so it is natural to wonder what happens under a more realistic model of a network and uncertainty. To this end, we investigate the following empirical question: under our models of uncertainty, if we were to compute an optimal strategy assuming a single influencer type, how much would that strategy vary for different types? To answer this, we take a Bayesian game with 40 types and compute an optimal mitigation strategy for each possible influencer type λ under the assumption of complete information. This yields a mixed strategy, S_M^λ , for each possible influencer type. Next, we select a type b uniformly at random and measure the fraction of pure strategies in the support of S_M^b that is different from the pure strategies in the support of each S_M^λ for $\lambda \neq b$. In Figure 2 we report the *average* fractional difference over 20 independent instances of 40-type Bayesian games on 40-node BTER graphs (edges vary from 130 to 200) with Influential Node uncertainty (more details on our standard setup will be presented shortly). Note that 1 in this case indicates that the mixed strategy for a randomly chosen type does not share a *single* pure strategy with

the mixed strategy computed for *any* other type. As can be seen here for this instance, and is generally true under this uncertainty, nearly all instances show minimal overlap in the pure strategies used by each of the type-specific optimal strategies.

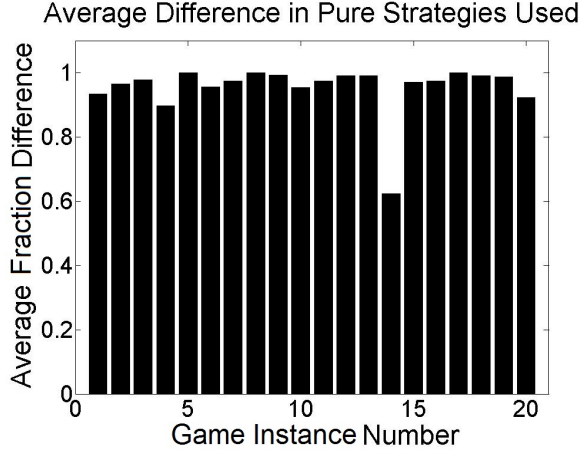


Figure 2: *Comparison of mixed strategies*

Finally, we turn to the question of complexity, where the result is very clear and very negative. At a high level, the challenge of efficiently reducing the runtime of computing equilibria in our setting lies in quantifying the impact of even small changes in the graph structure. If this could quickly and accurately be determined, then types could be efficiently clustered and bounds could be placed on the quality loss. The fact that computing the expected influence is #P-Hard [16] should already give us pause. Indeed, a simple corollary of this result reveals that such quantification is intractable in general.

Proposition 1. *Computing the difference in expected influence for a given seed set even when a single edge is added to a graph is #P-Hard.*

Proof. We prove this by contradiction. Call the difference function, $d(S, G, e)$, where S is the given seed set, $G = (N, E)$ is the base graph, and e is the edge to be added. Assume $d(\cdot)$ can be calculated in polynomial time. Define a graph $G' = (N, \emptyset)$. $\sigma_{G'}(S)$ can be calculated in polynomial time. Repeatedly add edges from E to G' until G is fully reconstructed, computing $d(S, G, e)$ in each iteration. Since the total influence of G is $\sum_{e \in E} d(S, G, e)$, this implies that we have computed influence in polynomial time, since only $|E|$ iterations were executed, which contradicts that fact that computing the expected influence of a graph is #P-Hard. \square

IV DOUBLE ORACLE ALGORITHM

Even though we formulated influence blocking as a zero-sum game which can be solved using linear programming computing, an equilibrium of this game in our case is challenging for three reasons. First, payoff estimation requires determining the value of $\sigma(S_I^\lambda, S_M, \lambda)$, which has been shown to be #P-Hard [16]. Therefore, even constructing the payoff matrix for this linear program is non-trivial. Second, the strategy sets for both players are exponentially large, making it impractical to store the entire payoff matrix even if we could compute payoffs efficiently for a pair of player strategies. Third, because we model uncertainty over graph instances, the number of influencer types can be exponentially large.

The first problem was previously addressed by introducing the LSMI heuristic for faster estimation of $\sigma(\cdot)$, which we also use here [6]. The Bayesian double-oracle algorithm introduced by Halvorson et al. [17] offers a solution to the second problem. This algorithm begins with a small subset of pure strategies for each player and iteratively adds best-response strategies to the existing subgame. The algorithm ends when no new best-responses need to be added, at which point it has provably converged to the equilibrium of the full game. In the context of Bayesian games, Halvorson et al. propose computing the best response for every player type, which in our case means that we compute the influencer's best response for each type (graph), and add all of these pure strategies in each iteration. This approach runs into our third and final problem: the exponential number of types. Since computing a best response for a given type requires a non-negligible amount of computation, doing it for every type will not scale. To address this, we now show empirically that simple heuristics actually produce near-optimal solutions.

V THE POWER OF SIMPLE

The results presented thus far, as well as the intuition from the vast literature on influence maximization [2, 7, 16], suggest that carefully accounting for our uncertainty about graph structure is crucial to obtaining high quality solutions. Our hardness and initial empirical results about the Bayesian counter-contagion problem also suggest that it will be difficult to scale up to large graphs while carefully accounting for uncertainty. However, we will now present a representative subset of an extensive collection of experiments, all showing precisely the opposite: we

need only to randomly sample a few types from the type distribution and solve the resulting game as if no other types exist, to obtain solutions that are nearly optimal. This is quite surprising, particularly since we have already shown, via the example in Figure 2, that ignoring even a single influencer type can yield arbitrarily poor solutions even with only two types.

All the results below are an average of 20 game instances and were run on machines with CPLEX 12.2, 2.8 GHz CPU, and 4GB of RAM. Unless otherwise stated, experiments were run on 40-node graphs (130 to 200 edges), contagion probabilities on edges drawn from a $\mathcal{N}(0.4, 0.2)$ distribution, node values varying uniformly from 1-10, each player having two seed nodes ($|S_I| = |S_M| = 2$), and payoffs estimated using the LSMI heuristic introduced by Tsai et al. [6]. Monte Carlo payoff estimations produced similar results but could not be meaningfully scaled. Since an optimal benchmark is necessary, the best-response oracles iteratively evaluate each available action to determine the best response, rather than using greedy hill-climbing common in the influence maximization literature. Unless otherwise stated, Influential Node uncertainty selects 3 nodes and gives each 4 additional edges. Moreover, *only* these 12 edges could potentially connect communities, making the chosen nodes not only more connected (average degree, excluding uncertain edges, varies from 3-5 with maximums of 9), but also incident to the more consequential edges. For Inter-Community Edge uncertainty we varied the number of uncertain edges between 1 and 6 (the optimal technique could not scale to more edges). We focus throughout on the mitigator strategy obtained by drawing a random subset of the influencer’s types and solving the game assuming no other types exist (referred to as *Random Sampling*).

1 EXPERIMENTS

In our first set of results, shown in Figures 3, 4, and 5 we consider the impact of the number of randomly sampled types on solution quality. The three different sets of figures show results for different methods of graph generation, including scale free graphs, small world graphs, and BTER graphs. The x -axis shows the number of sampled types, while the mitigator utility is plotted on the y -axis. These experiments use the *same* 40-node games that were featured in Figure 1 (right) that showed pure strategies used by individual types have minimal overlap. The key point is that with only about 2-5 randomly sampled types

we obtain a solution that is very nearly optimal, despite the fact that only using a single influencer type yields a relatively poor mitigator reward (Figure 5b). This is true regardless of the method used for generating the underlying graph. While results in the optimization literature such as sample average approximation theory [18] show that random sampling can converge exponentially fast to optimal solutions, our “convergence” is uncannily quick.

Next, we fix the number of randomly sampled types used to generate a solution at 2, and increase the number of *actual* types (increasing the degree of uncertainty). The graph sizes were fixed to 40 nodes. Intuitively, we would expect that the performance of *Random Sampling* should degrade significantly as we increase uncertainty by adding types. In addition, we compare the random sampling strategy to an even simpler heuristic which uses only a single type with the highest probability; we call this *Max Prob*. Figures 6 and 7 are representative of a broad array of experiments we ran in this space. In addition to considering several types of uncertainty, we also varied the density of connections among communities (low density uses $\rho = 0.5$ as the probability of inter-community edges, while high density uses $\rho = 0.9$). Perhaps the most surprising finding in these experiments is that the quality of *Random Sampling* relative to optimal degrades very little as we increase the number of types. While we could not compute optimal solutions for games involving more types, this finding suggests that we may need to sample a decreasing (rather than a constant) fraction of all possible types as the number of total types increases.

In our final set of results using synthetic graphs, we study the impact of the size of the underlying network. The number of edges varied from 28 (20 nodes) to 188 (40 nodes) with up to 6 edges differing between types for Inter-community Edge uncertainty and up to 24 edges for Influential Node uncertainty (12 new edges per type). Here, we keep the number of nodes/edges about which we are uncertain fixed, and increase the network size. Consequently, we expect that smaller networks would exhibit significantly greater difference between random sampling and optimal, since uncertainty involves a greater fraction of the graph.

Figure 8 shows little evidence of this: the quality of simple heuristics relative to optimal is little affected by the fraction of the graph that is uncertain.

¹Abhijit Banerjee; Arun Chandrasekar; Esther Duflo; Matthew Jackson, 2011-08, “Social Networks and Microfinance”,

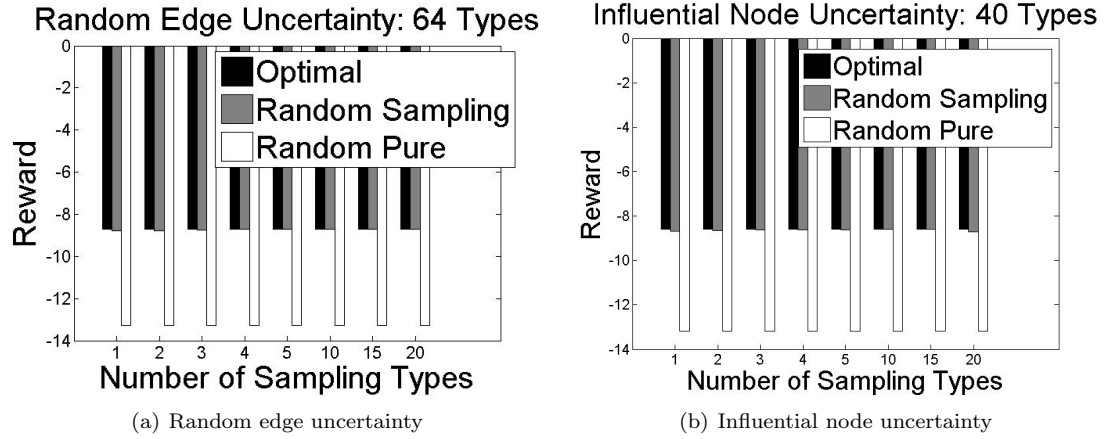


Figure 3: Reward comparison, Scale-Free graphs

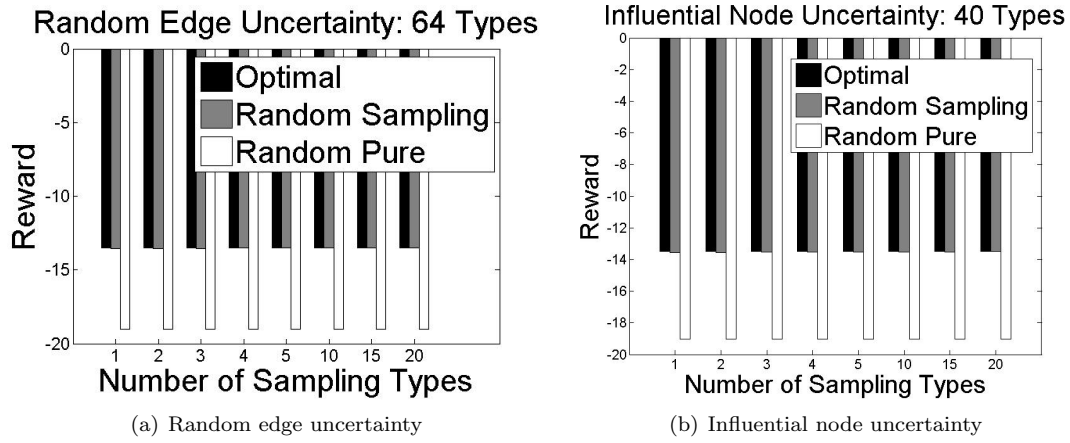


Figure 4: Reward comparison, Small-world graphs

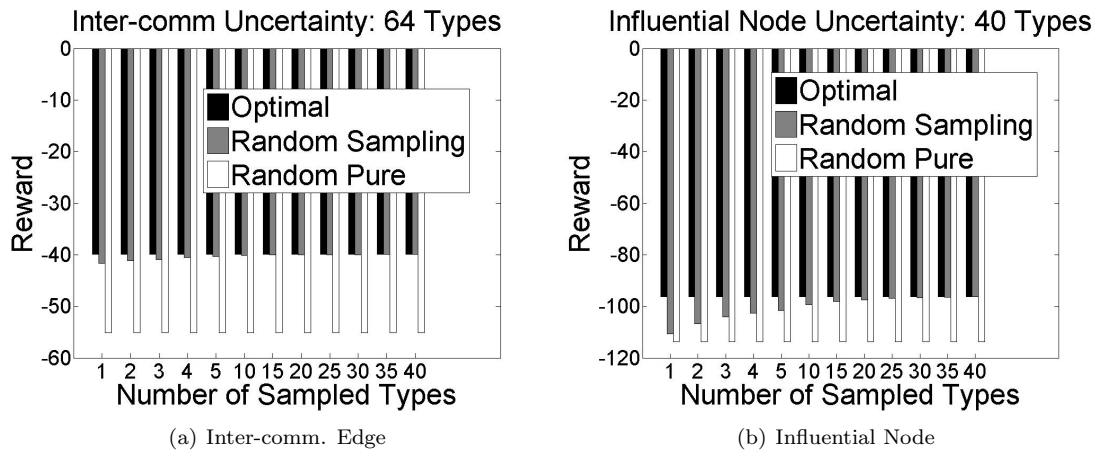


Figure 5: Reward comparison, BTER graphs

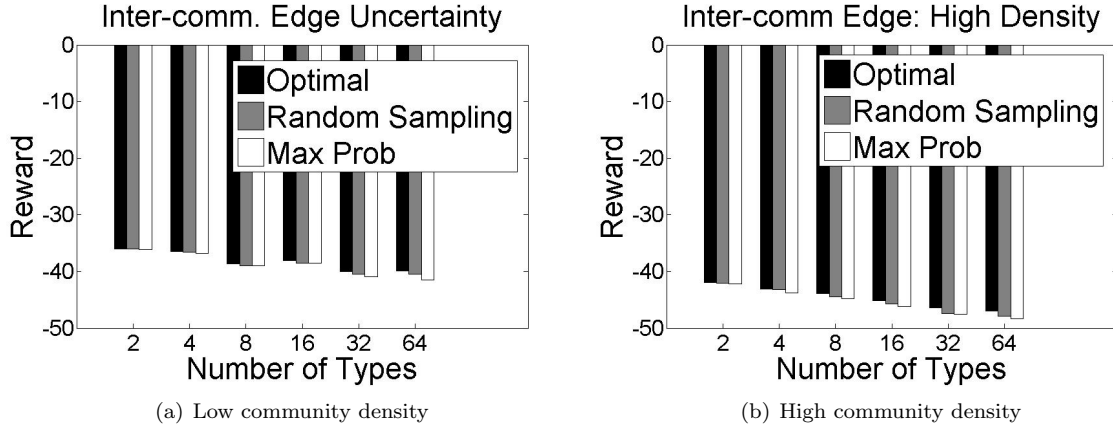


Figure 6: Inter-community Edge uncertainty

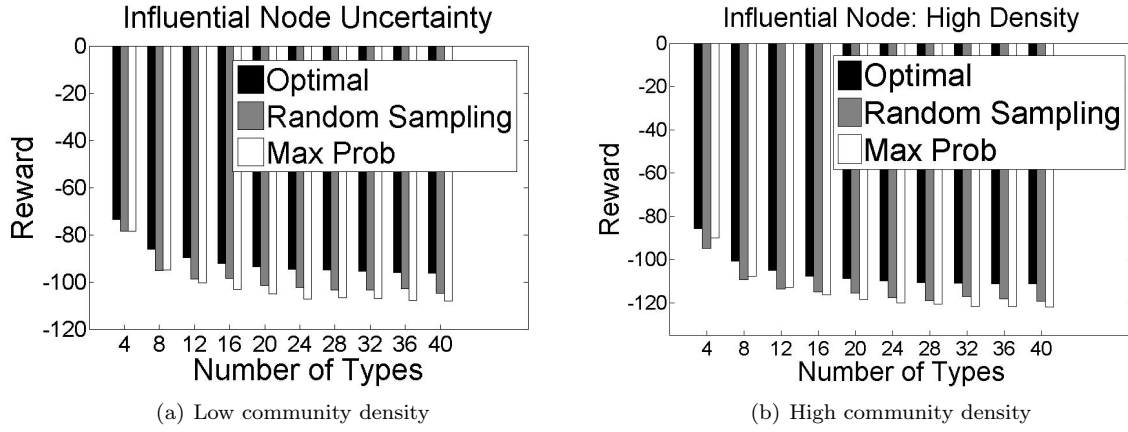


Figure 7: Influential Node uncertainty

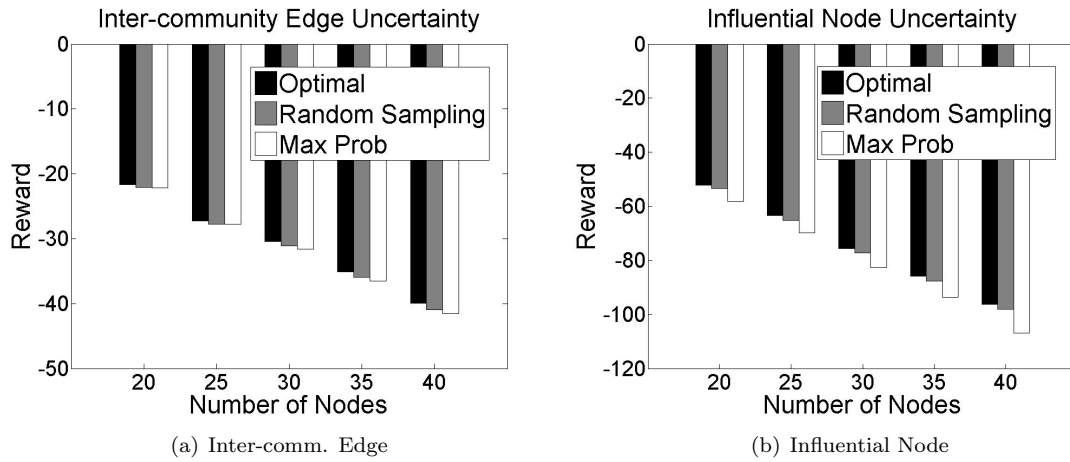


Figure 8: Scale-up of graph size

Parameter	Variations
Graph type	Scale-free, small-world, BTER, Indian villages
Graph size	20, 25, 30, 35, 40 node synthetic graphs, 80-100 node real graphs
Uncertainty model	Random edge, Influential node, Intercommunity edge, Intercommunity edge set, Inter/Intracommunity edge
Number of sampled types	1-40
Total number of types	4-40
Influential node uncertainty: edges per node	4, 10-50
Intercommunity edge set uncertainty: number of edges per type	5-40
Average contagion probability	0.1, 0.4, 0.7
Community density	0.6, 0.9
Indian village graph weighting scheme	uniform (0.1 increments), weighted

Table 1: Summary of parametric variations we had explored, all yielding broad consensus on the overall conclusion that only very few types need to be sampled for near-optimal results.

Finally, we conducted a set of experiments on a real-world social network dataset released in 2012 that was obtained via survey data in 75 Indian villages.¹ The survey asked the inhabitants of the villages a series of questions to ascertain their relationship with other people in the village (e.g., would you invite him in for tea, do you go to temple with him, would you loan him money, etc.). From this data, a social network can be constructed by beginning with a complete graph with edge weights of 0.0, increasing the weight of an edge corresponding to a positive answer to a survey question by 0.1, and then normalizing all weights. We also tried a second weighting scheme to address concerns about edge-weight translation choices. This scheme is referred to as the Varied-Weight-Scheme, and places greater weight on intuitively stronger relationships to provide greater variation in the weights.

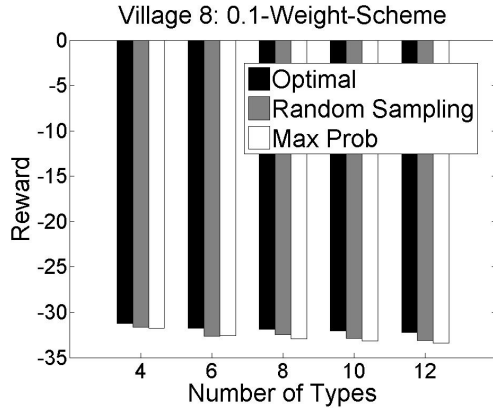
For our experiments, we use the household-level data for two of the smaller villages (8 and 10), because even the double-oracle optimization does not scale to larger networks. The results in Figures 9 and 10 (reflecting the two different ways to generate weights for edges) use Influential Node uncertainty, and each type now chooses 8 random nodes and gives each 10 new edges to maintain the same fraction of uncertainty, since the India data sets have 77 or 94 nodes and an average degree of 7.7 or 7.4. As the figure testifies, our results are not an artifact of synthetic graph models that we generate, but can be observed on graphs based on actual social network data as well.

As mentioned previously, in addition to using scale-

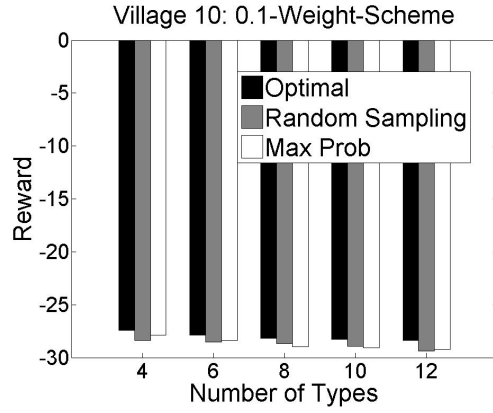
free and small-world graphs and additional models of uncertainty, we also varied a number of other parameters. We created resource imbalances by testing games in which the mitigator was allowed 3 or 4 nodes against the influencer’s 2. We also tried the opposite, varying the number of nodes allowed for the influencer. We also varied the distribution of contagion probabilities on edges from the 0.4 we set it to for the experiments shown to 0.7 to examine the impact of changes in contagion probability. For BTER graphs we also examined multiple degree distributions in addition to the variation of community density that we show here. Overall, we studied over 200 parametric variations, which we summarize in Table 1, with the results virtually identical to what we show here: a simple heuristic which samples only a few types from the type distribution performs near optimally.

2 ANALYSIS OF THE POWER OF SIMPLE

The results shown are surprising in their extremity, especially in light of the result presented previously demonstrating minimal overlap of pure strategies in optimal strategies for individual types. We now explore why this might be occurring in these games. A plausible hypothesis is that the problems are simple to begin with and that most actions provide very high reward to the mitigator. To examine this, we plot the distribution of performance of *pure* strategies available to the mitigator by evaluating the expected reward obtained by each of the mitigator’s pure strate-

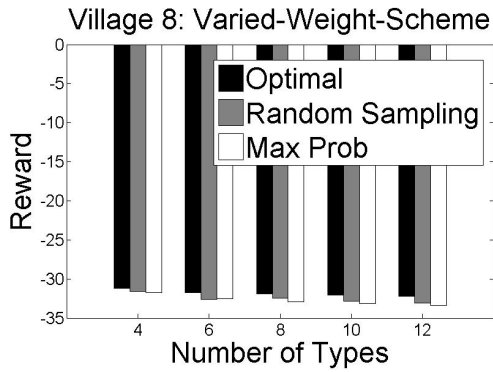


(a) Village 8

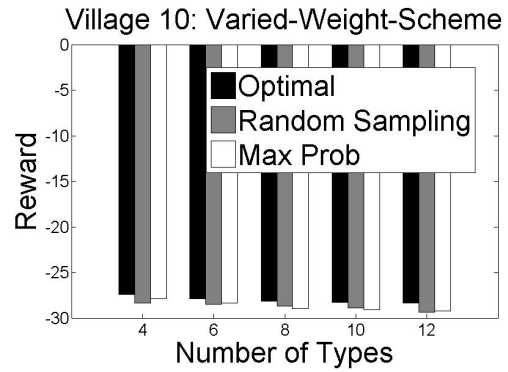


(b) Village 10

Figure 9: Influential Node uncertainty, 0.1-Weight-Scheme, that is, set the weights by adding 0.1 for every positive response and normalizing.



(a) Village 8



(b) Village 10

Figure 10: Influential Node uncertainty, Varied-Weight-Scheme, that is, greater weight was given for intuitively stronger relationships.

gies against the best response of the entire range of influencer types. To ease analysis, we bucket the rewards obtained into integer values. We show the results for two prototypical game instances to illustrate our findings. As can be seen in Figure 11a, the majority of actions lie in two clusters near -67 and -57. Figure 11b shows a distribution resembling a normal distribution. Neither of these shows a high percentage of high-reward actions, indicating that the initial hypothesis is incorrect and that substantial value is gained by optimally solving a subgame composed of a subset of types.

While we have previously already shown that type-by-type action overlap in mixed strategies is quite small, this in itself does not rule out substantial overlap in *nodes* used by the type-by-type strategies. This is because each pure strategy is a subset of nodes, and two different pure strategies may be different, but may actually share many of the nodes that the strategy uses. To study this, we consider the overlap of the *nodes* used in mixed strategies between each type's optimal strategy and, first, a randomly chosen type, and, additionally, the optimal strategy in the full Bayesian game. Figure 12 shows the results for 20 trials where each bar represents the average percentage of node overlap for a single trial (averaged over all types). We see in Figure 12a that there is a 60-80% difference (average of 74%) in the nodes used by individual types when compared against a randomly chosen type's optimal strategy. Thus, while each type's strategy may differ, they may all contain a set of core nodes that overlap more with the optimal strategy, which may cause the type-specific strategies to perform well overall. In Figure 12b we show the results when comparing the overlap in nodes used between the optimal strategy and each type's individual optimal strategy. The difference drops to the 40-70% range (average of 62%), suggesting that each individual type's optimal strategy uses nearly half of the nodes used by the optimal Bayesian strategy. The existence of such a core of nodes that are part of the optimal strategy for many types can partially explain the success of simple sampling techniques.

Next, we examine the type-by-type reward obtained by the sampling strategy versus the reward obtained by the optimal Bayesian strategy under Influential Node uncertainty. These experiments were examined instance by instance, and we show a typical example to illustrate the trends. In Figure 13a, we show results for the optimal strategy's performance on each type, where each bar shows the mitigator's reward on the y -axis for a particular type. Note that there is

one extremely high reward type and the others are all in the -80 to -100 range. In Figure 13b, we show the results for a randomly chosen single type's optimal strategy. Here the majority of rewards range from -100 to -120, suggesting that the overlap in nodes showed previously may mitigate some of the losses (maximum loss is -200 in expectation), but does not explain the whole story. An additional clue, however, is that the single type's strategy actually performed extremely well for two types here, leading to a comparable average reward despite the poorer performance in most cases. While not universal, this was a typical phenomenon, suggesting that within the forms of uncertainty explored here, an optimal strategy for one type tends to be near-optimal for a handful of others (despite minimal overlap in optimal support sets). Under what formal circumstances this reliably occurs, however, remains an open question.

Our final experiment draws a direct comparison between the impact of uncertainty on single-player influence maximization, as well as our two-player zero-sum contagion blocking setting. To make this comparison most revealing, we construct the following class of graphs: first, we generate 5 cliques of 5 nodes each, and then add 10 isolated nodes. Then, we generate each influencer type by connecting a single isolate to a subset of the cliques (possibly with multiple edges to the same clique); see Figure 14 for several example types produced in this construction. In this graph, the optimal strategy given one type is the pessimal for all other types.

Figure 15 shows a comparison of single-player (a) and two-player (b) results with and without reasoning about all possible graph types. In the single-player setting, ignoring uncertainty is markedly detrimental; in contrast, the two-player game is extremely robust to it. This evidence suggests that the minimax nature of the contagion blocking problem itself confers upon our setting high resilience to uncertainty about the influence graph.

To sum up, it appears that the phenomena we observed are due to a combination of at least the following two factors: there is non-trivial overlap in nodes used by type-optimal strategies with the Bayesian optimal, and the minimax nature of the game itself makes the solutions robust to uncertainty about the game structure.

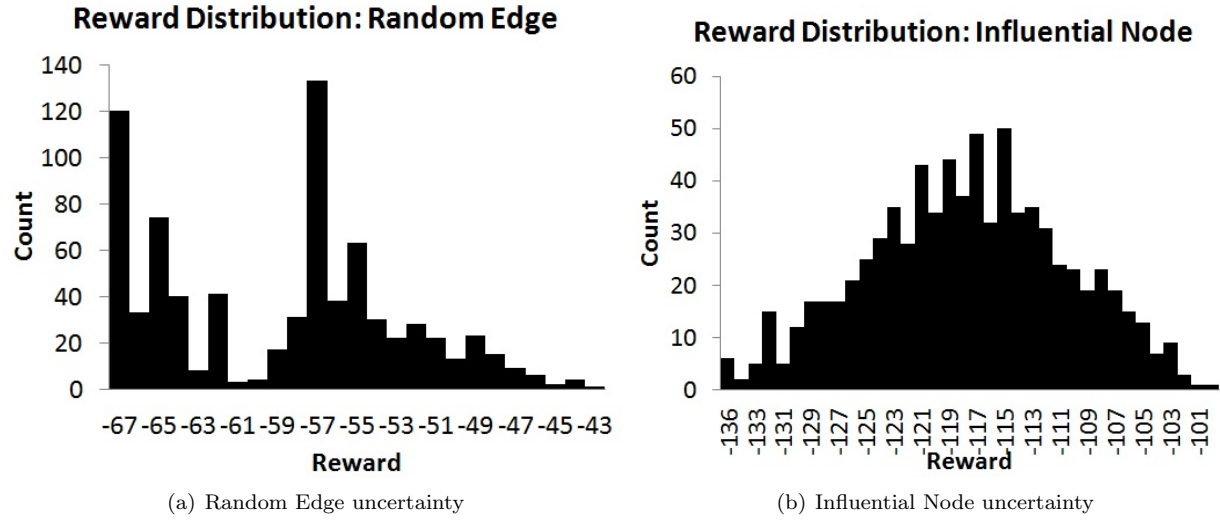


Figure 11: Distribution of pure strategy performance

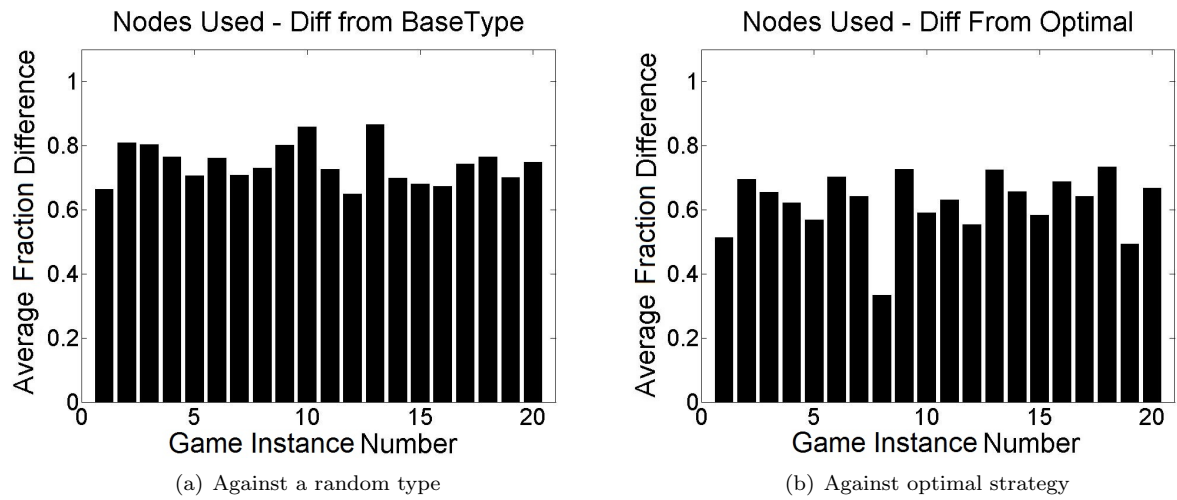


Figure 12: Overlap in nodes used

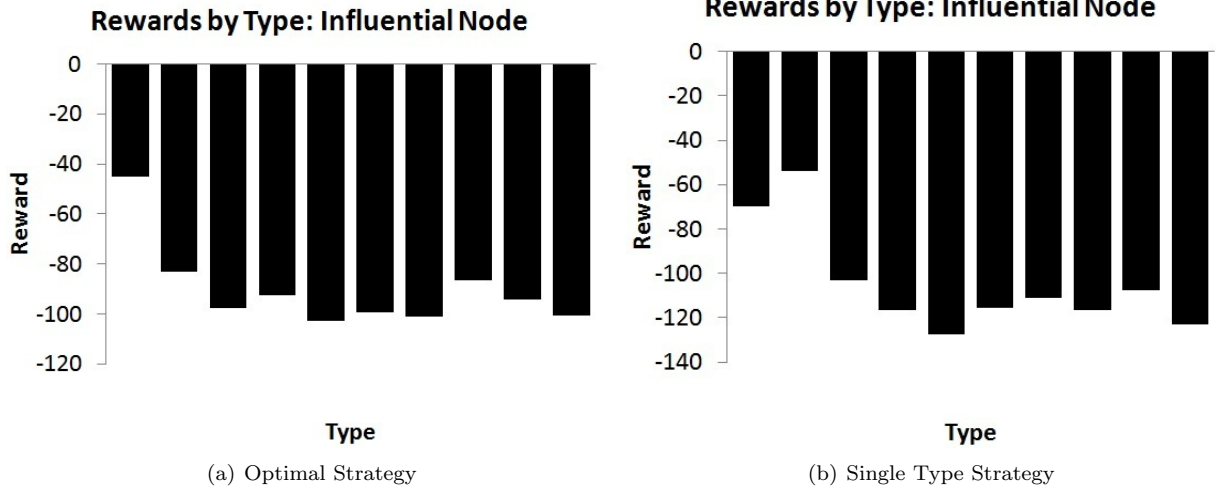


Figure 13: Performance distribution, Influential Node uncertainty

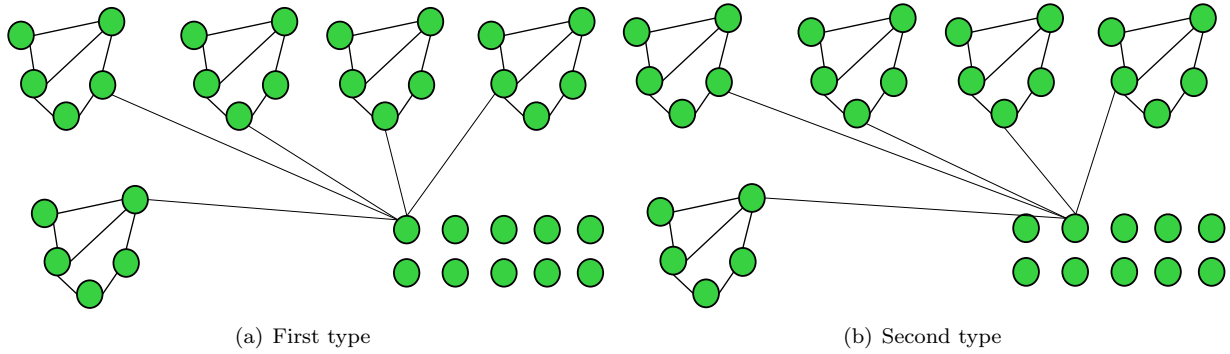


Figure 14: Two example types (graph realizations) for the construction in which optimal strategy for one type is pessimal for the others.

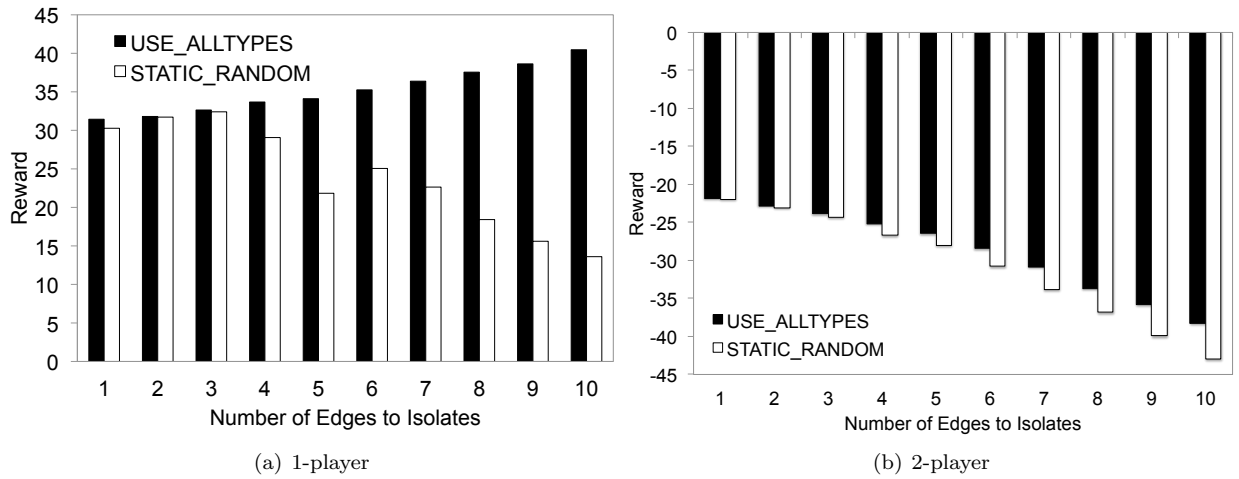


Figure 15: (a) 1-player (maximum influence) problem; (b) 2-player (influence blocking) problem. Black lines consider all types, white considers just a single type.

VI DISCUSSION

The phenomenon of simple techniques providing highly effective solutions has also been observed elsewhere [18–21]. In addition to novelty of our influence-driven, network-based model, our work differs from these in other important ways. As noted earlier, our results differ from previous work using sampling techniques in how dramatic they are, and unlike research in heuristic techniques for equilibrium computation, our work focuses on the power of extremely few samples instead of general heuristics.

A closer examination of previous literature in security games that addresses uncertainty reveals that similar phenomena may have been true elsewhere but went unexplored. In Yin et al. [11], the authors provide a novel algorithm (HUNTER) for optimally handling Bayesian Stackelberg games with many types. While the algorithm is orders of magnitude faster than previously proposed optimal algorithms, the authors report that BRASS, a far less complex solution method [22], achieves an average loss of 0.7 in a game where the range of rewards for optimal solutions ranged from -26 to 17 compared against their algorithm. One again wonders whether a sampling approach would have worked extremely well here too.

Our work does not dispute the fact that extremely large Bayesian zero-sum games remain very challenging to solve well in general and there are certainly problem classes that are not amenable to simple heuristics. In Kiekintveld et al. [9], for example, the authors introduce several techniques for handling large numbers of Bayesian types to address payoff uncertainty and they show that simple techniques do not perform near-optimally. Our work stresses the need to verify whether or not simple techniques work before embarking on extensive algorithmic gymnastics to achieve minimal gains in solution quality. Although we have provided some analysis of why this occurs in our domain, this is only the beginning, and more work is necessary. Still, our findings give hope that many very challenging problems in computational game theory may be effectively addressed by simple techniques.

References

- [1] D. Kempe, J. M. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in *KDD*, 2003, pp. 137–146.
- [2] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance, “Cost-effective outbreak detection in networks,” in *KDD*, 2007, pp. 420–429.
- [3] M. Trusov, R. E. Bucklin, and K. Pauwels, “Effects of word-of-mouth versus traditional marketing: Findings from an internet social networking site,” *Journal of Marketing*, vol. 73, September 2009.
- [4] B. W. K. Hung, “Optimization-based selection of influential agents in a rural Afghan social network,” Masters Thesis, MIT, Sloan School of Management, June 2010.
- [5] N. J. Howard, “Finding optimal strategies for influencing social networks in two player games,” Masters Thesis, MIT, Sloan School of Management, June 2011.
- [6] J. Tsai, T. H. Nguyen, and M. Tambe, “Security games for controlling contagion,” in *AAAI*, 2012.
- [7] C. Budak, D. Agrawal, and A. E. Abbadi, “Limiting the spread of misinformation in social networks,” in *WWW*, 2011, pp. 665–674.
- [8] X. He, G. Song, W. Chen, and Q. Jiang, “Influence blocking maximization in social networks under the competitive linear threshold model,” in *SDM*, 2012, pp. 463–474.
- [9] C. Kiekintveld, J. Marecki, and M. Tambe, “Approximation methods for infinite Bayesian Stackelberg games: Modeling distributional payoff uncertainty,” in *International Conference on Autonomous Agents and Multiagent Systems*, 2011.
- [10] M. Jain, M. Tambe, and C. Kiekintveld, “Quality-bounded solutions for finite Bayesian Stackelberg games: Scaling up,” in *In AAMAS*, 2011, pp. 997–1004.
- [11] Z. Yin and M. Tambe, “A unified method for handling discrete and continuous uncertainty in Bayesian Stackelberg games,” in *In AAMAS*, 2012, pp. 855–862.
- [12] M. Newman, *Networks: An Introduction*. Oxford University Press, 2010.
- [13] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999. [Online]. Available: <http://www.sciencemag.org/content/286/5439/509.abstract>
- [14] D. J. Watts and S. H. Strogatz, “Collective dynamics of small-world networks,” *Nature*, vol.

- 393, no. 6684, pp. 440–442, June 1998. [Online]. Available: http://tam.cornell.edu/tam/cms/manage/upload/SS_nature_smallworld.pdf
- [15] C. Seshadhri, T. G. Kolda, and A. Pinar, “Community structure and scale-free collections of Erdős-Rényi graphs,” *Phys. Rev. E*, vol. 85, p. 056109, May 2012.
 - [16] W. Chen, C. Wang, and Y. Wang, “Scalable influence maximization for prevalent viral marketing in large-scale social networks,” in *KDD*, 2010, pp. 1029–1038.
 - [17] E. Halvorson, V. Conitzer, and R. Parr, “Multi-step multi-sensor hide-seeker games,” in *IJCAI*, 2009, pp. 159–166.
 - [18] A. Shapiro and T. Homem-de Mello, “On the rate of convergence of optimal solutions of monte carlo approximations of stochastic programs,” *SIAM Journal on Optimization*, vol. 11, no. 1, pp. 70–86, 2000. [Online]. Available: <http://epubs.siam.org/doi/abs/10.1137/S1052623498349541>
 - [19] M. Kearns, Y. Mansour, and A. Y. Ng, “A sparse sampling algorithm for near-optimal planning in large markov decision processes,” *Machine Learning*, vol. 49, pp. 193–208, 2002.
 - [20] R. Porter, E. Nudelman, and Y. Shoham, “Simple search methods for finding a nash equilibrium,” in *Games and Economic Behavior*, 2004, pp. 664–669.
 - [21] O. Armantier, J.-P. Florens, and J.-F. Richard, “Approximation of Nash equilibria in Bayesian games,” *Journal of Applied Econometrics*, vol. 23, no. 7, pp. 965–981, 2008.
 - [22] J. Pita, M. Jain, F. Ordonez, M. Tambe, S. Kraus, and R. Magori-Cohen, “Effective solutions for real-world Stackelberg games: When agents must deal with human uncertainties,” in *The Eighth International Conference on Autonomous Agents and Multiagent Systems*, 2009.