A Game-Theoretic Approach to Influence in Networks

Mohammad T. Irfan and **Luis E. Ortiz**
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794
{*mtirfan,leortiz*}@*cs.stonybrook.edu*

Abstract

We propose *influence games*, a new class of graphical games, as a model of the behavior of large but finite networked populations. Grounded in non-cooperative game theory, we introduce a new approach to the study of influence in networks that captures the strategic aspects of complex interactions in the network. We study computational problems on influence games, including the identification of the most influential nodes. We characterize the computational complexity of various problems in influence games, propose several heuristics for the hard cases, and design approximation algorithms, with provable guarantees, for the most influential nodes problem.

Introduction

The influence of an entity on its peers is a commonly noted phenomenon in both online and physical social networks. This is also a common perception that such influences have the potential to cause behavioral changes among the nodes of the network. Indeed, the general perception that influences can instill new behavioral patterns in the network is backed up by scientific evidence. For example, recent works in medical social sciences posit the intriguing hypothesis that smoking, obesity, and even happiness is contagious within a social network (Fowler and Christakis 2008). The underlying system under study in that research exhibits several core features. First, it is often very *large and complex*, with many individual entities exhibiting different behaviors and interactions. The *network structure of complex interactions* is central. The *directions and strengths of local influences* are highlighted as very relevant to the global behavior of the system as a whole.

The prevalence of systems and problems like the ones just described in the context of social medical science, combined with the obvious issue of often limited control over individuals, raises immediate, broad, difficult, and longstanding policy questions: e.g., *Can we achieve a desired objective, such as reducing the level of smoking, or controlling obesity via targeted, minimal interventions in a system? How do we optimally allocate our often limited resources to achieve the largest impact in such systems?* Clearly, these issues are not exclusive to obesity, smoking, or happiness; similar issues arise in a large variety of settings: drug use, vaccination, crime networks, security, marketing, markets, the economy, and public policy-making and regulations, to name a few.

The work reported in this paper is in large part motivated by such questions and their broader implication. Our major contributions here are (1) a new approach to influence in networks grounded in non-cooperative game theory; (2) *influence games*, a new class of graphical games, to model the behavior of individuals in networks; and (3) a study of computational aspects of influence games, including an algorithm for the identification of the most influential nodes.

Influence in Networks

A very important problem in social network analysis is the identification of "influential" individuals (Wasserman and Faust 1994; Kleinberg 2007). Roughly speaking, in our approach we consider a set of individuals S in a network to be *most influential*, *with respect to some objective of interest*, if S satisfies the following condition: were the individuals in S to choose the behavior $\mathbf{x}_S$ prescribed to them by some *stable outcome* of the system $\mathbf{x} \equiv (\mathbf{x}_S, \mathbf{x}_{-S})$ that achieves the given objective of interest, the *only* stable outcome of the system that remains consistent with their choices $\mathbf{x}_S$ is $\mathbf{x}$ itself. Said differently, once the most influential nodes follow the behavior $\mathbf{x}_S$ prescribed to them by a stable outcome $\mathbf{x}$ achieving the objective of interest, they become collectively so much influential that their behavior forces every other individual to a unique choice of behavior! Figure 1 provides an example illustrating our concept of "most influential." In contrast to a mechanism-design approach, we are not interested in changing the system—*the system is what it is*—but are rather interested in altering the behavior to "tip" the same system to some desirable stable outcome.

On Diffusion Models. To date, the study of influence in networks has concentrated mostly on analyzing the diffusion (or "contagion") processes induced by influences with the goal of maximizing the spread of a new behavior (see (Kleinberg 2007) for a comprehensive survey). A subtle aspect of the diffusion models is that each node in the network behaves as an independent agent. Any observed influence that a node's neighbors impose on the node is the result of the same node's "rational" or "natural" response to the

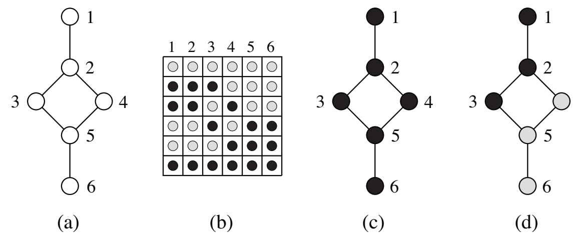

Figure 1: ***Illustration of our approach to influence in networks.*** *Each node has a binary choice of behavior, $\{-1,+1\}$, and wants to behave like the majority of its neighbors (and is indifferent if there is a tie). The network is shown in (a) and the enumeration of PSNE (a row for each PSNE, where black denotes node's behavior 1, gray -1) in (b). We want to achieve the objective of every node choosing 1 (desirable outcome). Selecting the set of nodes $\{1,2,3\}$ and assigning these nodes behavior prescribed by the desirable outcome (i.e., 1 for each) lead to two consistent stable outcomes of the system, shown in (c) and (d). Thus, $\{1,2,3\}$ cannot be a most influential set of nodes. On the other hand, selecting $\{1,6\}$ and assigning these nodes behavior 1 lead to the desirable outcome as the unique stable outcome remaining. Therefore, $\{1,6\}$ is a most influential set. Furthermore, note that $\{1,6\}$ is not most influential in the diffusion setting, since it does not maximize the spread of behavior 1. (It should be mentioned that we study a much richer class of games in this paper than the one shown in this example.)*

neighbors' behavior. If we select a set of nodes with the goal of maximizing the spread of the new behavior then it might very well happen that some of the selected nodes are "unhappy" being the initial adopters of the new behavior relative to their neighbors' final behavior after diffusion. Thus, isn't it more natural to require that the desired final state of the system be stable (in which everyone is "happy" with their behavioral response)?

For the most part, the diffusion process has been modeled as a "monotonic" process (in the sense that once an agent adopts a behavior, it cannot go back) to address the question of finding the most influential nodes. However, if we think of an application such as reducing the incidence of smoking or obesity, then a model that allows a "change of mind" based on the response of the immediate neighborhood may make more sense. While generalized versions of threshold models that allow "reversals" have been derived in the social science literature (Granovetter 1978), to the best of our knowledge, there is no substantive work on the most influential nodes problem in the context of such generalized diffusion models.

Although our model is inspired by the same threshold models that are used in modeling diffusion processes, in contrast to diffusion models, by concentrating on stable outcomes in a strictly game-theoretic setting we capture significant *strategic* aspects of complex interaction in networks that naturally appear in many real-world problems (e.g., identifying the most influential senators in the US Congress).

Influence Games

Inspired by the threshold models (Granovetter 1978), we design *influence games* as a model of influence in large networked populations. Even though the model falls within the general class of graphical games (Kearns, Littman, and Singh 2001), a distinctive feature of influence games is its compact, parametric representation.

General Game-Theoretic Model. Let us first formalize influence games. Let n be the number of individuals in the population and $x_i \in \{-1,1\}$ denote the *behavior* of individual i, where $x_i = 1$ indicates that i "adopts" a particular behavior and $x_i = -1$ indicates i "rejects" it.

Definition 1. *Denote by $f_i : \{-1,1\}^{n-1} \to \mathbb{R}$ the function that quantifies the* "influence" *of other individuals on i. In influence games, we define the* payoff *function $u_i : \{-1,1\}^n \to \mathbb{R}$ quantifying the preferences of each player i as $u_i(x_i, \mathbf{x}_{-i}) \equiv x_i f_i(\mathbf{x}_{-i})$, where $\mathbf{x}_{-i}$ denotes the vector of all joint-actions excluding that of i.*

For each player i in an influence game $\mathcal{G}$, the *best-response correspondence* $\mathcal{BR}_i^{\mathcal{G}} : \{-1,1\}^{n-1} \to 2^{\{-1,1\}}$ is, by definition, $\mathcal{BR}_i^{\mathcal{G}}(\mathbf{x}_{-i}) \equiv \arg\max_{x_i \in \{-1,1\}} u_i(x_i, \mathbf{x}_{-i})$, for any $\mathbf{x}_{-i} \in \{-1,1\}^{n-1}$. Hence, for all individuals i, and any possible behavior $\mathbf{x}_{-i} \in \{-1,1\}^{n-1}$ of the other individuals in the population, the *best-response behavior* x_i^* of individual i to the behavior $\mathbf{x}_{-i}$ of others satisfies

$$
\begin{aligned}
f_i(x_{-i}) > 0 &\implies x_i^* = 1, \\
f_i(x_{-i}) < 0 &\implies x_i^* = -1, \text{ and} \\
f_i(x_{-i}) = 0 &\implies x_i^* \in \{-1,1\}.
\end{aligned}
$$

Informally, "positive influences" lead an individual to adopt the behavior, while "negative influences" lead the individual to "reject" the behavior; the individual is indifferent if there is "no influence." A *stable outcome* of the system, by which we formally mean a *pure-strategy Nash equilibrium (PSNE)* of the corresponding influence game $\mathcal{G}$, is a behavior assignment $\mathbf{x}^* \in \{-1,1\}^n$ that satisfies all those conditions: Each player i's behavior x_i^* is a (simultaneous) best-response to the behavior $\mathbf{x}_{-i}^*$ of the rest. Denote by $\mathcal{NE}(\mathcal{G}) \equiv \{\mathbf{x}^* \in \{-1,1\}^n \mid x_i^* \in \mathcal{BR}_i^{\mathcal{G}}(\mathbf{x}_{-i}^*) \text{ for all } i\}$ the set of PSNE of game $\mathcal{G}$.

Most Influential Nodes: Problem Formulation. In formulating the most influential nodes problem in a network, we depart from the traditional model of diffusion and adopt influence games as the model of strategic behavior among the nodes in the network.

Definition 2. *Let $\mathcal{G}$ be an influence game, $g : \{-1,1\}^n \times 2^{[n]} \to \mathbb{R}$ be the* goal or objective function *mapping a joint-action and a subset of the players in $\mathcal{G}$ to a real number quantifying the general preferences over the space of joint-actions and players' subsets, and $h : 2^{[n]} \to \mathbb{R}$ be the* set-preference function *mapping a subset of the players to a real number quantifying the a priori preference over the space of players' subsets. Denote by $\mathcal{X}_g^*(S) \equiv \arg\max_{\mathbf{x} \in \mathcal{NE}(\mathcal{G})} g(\mathbf{x}, S)$ the* optimal set of PSNE *of $\mathcal{G}$, with respect to g and a fixed subset of players $S \subset$*

$[n]$. *We say that a set of nodes/players* $S^* \subset [n]$ *in* $\mathcal{G}$ *is* most influential *with respect to* g *and* h, *if* $S^* \in \arg\max_{S \subset [n]} \{h(S), s.t., |\{\mathbf{x} \in \mathcal{NE}(\mathcal{G}) \mid \mathbf{x}_S = \mathbf{x}^*_S, \mathbf{x}^* \in \mathcal{X}^*_g(S)\}| = 1\}$.

As mentioned earlier, we can interpret the players in S^* to be collectively so influential that they are able to restrict every other player's choice of action to a unique one: the action prescribed by some desirable stable outcome $\mathbf{x}^*$.

An example of the goal function g that captures our objective to achieve a specific stable outcome $\mathbf{x}^* \in \mathcal{NE}(\mathcal{G})$ is $g(\mathbf{x}, S) \equiv \mathbf{1}[\mathbf{x} = \mathbf{x}^*]$. Another example that captures our objective to achieve a stable outcome with the largest number of individuals adopting the behavior is $g(\mathbf{x}, S) \equiv \sum_{i=1}^n \frac{x_i+1}{2}$. A common example of the set-preference function h that captures our preference for sets of small cardinality is to define h such that $h(S) > h(S')$ iff $|S| < |S'|$.

Linear Influence Games. A simple instantiation of the general influence game model above is when the influences are linear.

Definition 3. *In a* linear influence game (LIG)*, the influence function of each individual* i*, which leads to a quadratic payoff function, is defined as* $f_i(\mathbf{x}_{-i}) \equiv \sum_{j \neq i} w_{ji} x_j - b_i$ *where for any other individual* j*,* $w_{ji} \in \mathbb{R}$ *is a* weight *parameter quantifying the "influence factor" that* j *has on* i*, and* $b_i \in \mathbb{R}$ *is a* threshold *parameter for* i*'s level of "tolerance" for negative effects.*

In general, the influence weights w_{ji} induce a directed graph, where nodes represent individuals and there is a directed edge from individual j to i iff $w_{ji} \neq 0$, and therefore, we obtain a graphical game having a *linear* representation size, as opposed to the *exponential* representation size of graphical games in normal form (Kearns, Littman, and Singh 2001). Furthermore, LIGs can be shown to be equivalent to 2-action poly-matrix games (Janovskaja 1968), modulo their PSNE (details omitted).

Equilibria Computation in Influence Games

We first study the problem of computing and counting PSNE in LIGs. We show that several special cases of LIGs present us with attractive computational advantages, while the general problem is intractable unless P = NP. We also give a heuristic to compute PSNE in general LIGs that has produced promising empirical results.

Nonnegative Influences. When all the influence weights are non-negative, an LIG is supermodular (Milgrom and Roberts 1990). In particular, the game exhibits what is called *strategic complementarity*. Hence, the best-response dynamics converges in at most n rounds. From this, we obtain the following result.

Theorem 4. *The problem of computing a PSNE is in P for LIGs on general graphs with only non-negative influences.*

Special Influence Structures and Potential Games. Several special subclasses of LIGs are potential games (Monderer and Shapley 1996). This connection guarantees the existence of PSNE in such games.

Proposition 5. *If the influence weights of an LIG* $\mathcal{G}$ *are symmetric (i.e.,* $w_{ji} = w_{ij}$*, for all* i, j*), then* $\mathcal{G}$ *is a potential game.*

Proof Sketch. We show that the game has a cardinal potential function, $\Phi(\mathbf{x}) = \sum_{t=1}^n x_t (\sum_{i \neq t} \frac{x_i w_{it}}{2} - b_t)$. □

If, in addition, the threshold $b_i = 0$ for all i, the game is a *party-affiliation game*, and computing a PSNE in such games is PLS-complete (Fabrikant, Papadimitriou, and Talwar 2004).

The following result is on a large class of games that we call *indiscriminate* LIGs, where for every player i, the influence factor, $w_{ij} \equiv \delta_i \neq 0$, that i imposes on every other player j is the same.

Proposition 6. *Indiscriminate LIGs having either all positive or all negative influence factors are potential games.*

Proof Sketch. Let $\mathcal{G}$ be an indiscriminate LIG in which all δ_i for all i, have the same sign, denoted by $\rho \in \{-1, +1\}$. Then $\mathcal{G}$ is an ordinal potential game with the potential function $\Phi(\mathbf{x}) = \rho[(\sum_{i=1}^n \delta_i x_i)^2 - 2\sum_{i=1}^n b_i \delta_i x_i]$. □

The interesting aspect of this result is that such indiscriminate LIGs are potential games despite being possibly *asymmetric* and exhibiting strategic substitutability if $\rho = -1$. If, for some $\delta \in \mathbb{R}$, we have $\delta_i = \delta$ and $b_i = 0$ for all i, then best-response dynamics converges in a number of rounds polynomial in the number of players.

Tree-Structured Influence Graphs. The next result follows from a careful modification of the **TreeNash** algorithm (Kearns, Littman, and Singh 2001). Note that a naive application of **TreeNash** over the space of pure strategies has a running time of $O(n2^d)$, exponential in the maximum degree d of a node, and thus also in the representation size of the LIG!

Theorem 7. *There exists an* $O(nd)$ *time algorithm to find a PSNE, or to decide that there exists none, in LIGs with tree structures, where* d *is the maximum degree of a node.*

Proof Sketch. We use similar notations as in (Kearns, Littman, and Singh 2001). The modification of the **TreeNash** involves efficiently (in $O(d)$ time) determining the existence of a witness vector and constructing one, if exists, at each node during the downstream pass, in the following way.

Suppose that an internal node i receives tables $T_{ki}(x_k, x_i)$ from its parents k, and that i wants to send a table $T_{ij}(x_i, x_j)$ to its unique child j. We first partition i's set of parents into two sets in $O(d)$ time: $Pa_1(i, x_i)$ consisting of the parents k of i that have a unique best response $\hat{x}_k$ to i's playing x_i and $Pa_2(i, x_i)$ consisting of the remaining parents of i. We show that $T_{ij}(x_i, x_j) = 1$ iff

$$x_i(x_j w_{ji} + \sum_{k \in Pa_1(i,x_i)} \hat{x}_k w_{ki} + \sum_{t \in Pa_2(i,x_i)} \underbrace{(2 \times \mathbf{1}[x_i w_{ti} > 0] - 1)}_{t\text{'s action in witness vector}} w_{ti}) \geq 0,$$

from which we get a witness vector, if exists. □

Hardness Results

Computing PSNE in a general graphical game is known to be an intractable problem (Gottlob, Greco, and Scarcello 2003). However, that result does not imply intractability in our problem, primarily because LIGs are a special type of succinctly representable graphical game with quadratic payoffs.

Theorem 8. *(1) It is NP-complete to decide the following questions in LIGs: (a) Does there exist a PSNE? (b) Given a designated set of $k \geq 1$ players, does there exist a PSNE consistent with those k players playing 1? (c) Given a number $k \geq 1$, does there exist a PSNE with at least k players playing 1?*
(2) Given an LIG and a designated set of $k \geq 1$ players, it is co-NP-complete to decide if there exists a unique PSNE with those players playing 1.
(3) It is #P-complete to count the number of PSNE, even if the graph of the LIG is a star.

Proof Sketch. To enhance the clarity of the hardness proofs we have reduced existing NP-complete problems to LIGs with binary actions $\{0, 1\}$, instead of $\{-1, 1\}$. It can be shown, via a simple linear transformation, that any LIG with actions $\{0, 1\}$ can be reduced to an LIG with the same underlying graph, but with actions $\{-1, 1\}$.

1(a). Let I be a 3-SAT instance with m clauses and n variables. Let C_i be the set of clauses in which variable i appears and V_k be the set of variables appearing in clause k. We construct an LIG J with the variables and the clauses of I as players; the influence factor from each "clause player" k to each "variable player" $i \in V_k$ being $1 - 2l_{k,i}$ where $l_{k,i} = \mathbf{1}[i \text{ appears non-negated in } k]$ and that in the reverse direction being $2l_{k,i} - 1$; the threshold of each clause k and variable i being $1 - \epsilon - \sum_{i \in V_k}(1 - l_{k,i})$ for $0 < \epsilon < 1$ and $\sum_{k \in C_i}(1 - 2l_{k,i})$, respectively. We show that there exists a satisfying truth assignment in I iff there exists a PSNE in J. The forward direction is easy to verify (by translating the truth assignments *true* and *false* to actions 1 and 0, respectively). For the reverse direction, we first show, by contradiction, that at any PSNE of J, every clause player must play 1. We then show, also by contradiction, that for every clause player k, there exists a variable player $i \in V_k$ such that $x_i = l_{k,i}$. We map the actions to truth values in I to obtain a satisfying truth assignment. Due to the construction, we obtain, as a corollary, that the NP-completeness of the PSNE-existence question holds even for LIGs with bipartite graphs.

1(b). We reduce a *monotone one-in-three SAT* instance I with $m > 1$ clauses and n variables to an LIG instance J in a similar way. In this case, the thresholds of the variable and clause players are defined to be 0 and ϵ, for $0 < \epsilon < 1$, respectively. There are arcs with influence factors -1 between two variable players iff they appear in the same clause. There is an arc with an influence factor 1 from each variable player to each clause player in which it appears, and there is no arc in the reverse direction. Assign $k = m$, and let the designated set of players be the set of clause players. A solution to I easily leads to a PSNE of J with the clause players playing 1. To show the reverse direction, we first show that *at most* one variable player connected to each clause player can play 1 at any PSNE. Furthermore, for each clause player to play 1, *at least* one variable player connected to that clause must play 1. Thus, for each clause player, *exactly one* variable player connected to it plays 1, which leads to a solution to I.

1(c). The proof is similar to 1(b). We augment the construction of 1(b) by adding $m(m-1)$ *extra players*, each with threshold ϵ, and add arcs, each with an influence factor 1, from each clause player to distinct $m - 1$ extra players (so that no two clause players have arcs to the same extra player), and set $k = m^2$. We show that there exists a solution to I iff there exists a PSNE in J with at least k players playing 1.

2. Again, we augment the construction of 1(b) by adding the following players: an *all-satisfied-verification player* v_{all} (with threshold $m - \epsilon$), a *none-satisfied-verification player* v_{none} (with threshold $-\epsilon$), and m^2 *extra players* (each with threshold ϵ). We add arcs from each clause player to v_{all} and v_{none} with influence factors 1 and -1, respectively. We also add arcs from v_{all} and v_{none} to each extra player with influence factor 1. Let the set of $k \equiv m^2$ extra players be the designated set of players. First, we show that at any PSNE, v_{all} plays 1 iff every clause player plays 1, v_{none} plays 1 iff no clause player plays 1, and every extra player plays 1 iff either v_{all} or v_{none} plays 1. Let $\mathbf{x}_0$ be the following PSNE: v_{none} playing 1, every extra player playing 1, and every other player playing 0. We show that there exists another PSNE in the LIG instance with every extra player playing 1 iff there exists a solution to the monotone one-in-three SAT instance.

3. Given a #KNAPSACK instance I with n items, along with weight $a_i \in \mathcal{Z}^+$ of every item i and the maximum capacity of the sack $W \in \mathcal{Z}^+$, we construct an LIG instance J having a star-structure where the influence factor from the central player v_0 to every other player is 1 and that in the reverse direction is $-a_i$. In addition, the threshold of v_0 is defined to be $-W$ and that of every other player 1. We first show that at every PSNE, v_0 must play 1. We then show that the number of PSNE in J is the same as the number of feasible solutions to I. □

Note that the #P-completeness result for LIGs is in contrast to that for general graphical games with tree graphs, for which not only deciding the existence of a PSNE is in P but also counting PSNE on general graphical games with tree graphs is in P (due to the representation size).

Heuristics for Computing and Counting Equilibria

The fundamental computational problem at hand is that of computing PSNE in LIGs. We have just seen that various computational questions pertaining to LIGs on general graphs, sometimes even on bipartite graphs, are NP-hard.

A natural approach to finding all the PSNE in an LIG would be to perform a backtracking search. However, a naive backtracking method that does not consider the structure of the graph would be destined to failure in practice. Thus, we need to order the node selections in a way that would facilitate pruning the search space.

The following is an outline of a backtracking search procedure that we have used in practice. The first node selected by the procedure is a node with the maximum outdegree. Intuitively, this node is the "most constraining" in terms of the number of nodes that a node directly influences. Subsequently, we select a node i that will most likely show that the current partial joint action cannot lead to a PSNE and explore the two actions of i, $x_i \in \{-1, 1\}$ in a suitable order. A good node selection heuristic that has worked well in our experiments is to select the one that has the maximum influence on any of the already selected nodes.

Suppose that the nodes are selected in the order $1, 2, ..., n$ (wlog). After selecting node $i+1$ and assigning it an action x_{i+1}, we determine if the partial joint action $\mathbf{x}_{1:(i+1)} \equiv (x_1, \ldots, x_{i+1})$ can possibly lead to a PSNE and prune the corresponding search space if not. Note that a desirable "no" answer to this requires a proof that one of the players j, $1 \leq j \leq i+1$, can never play x_j according to the partial joint action $\mathbf{x}_{1:(i+1)}$. A straightforward way of doing this is to consider each player j, $1 \leq j \leq i+1$, and compute the quantities $\gamma_j^+ \equiv \sum_{k=1,k\neq j}^{i+1} x_k w_{kj} + \sum_{k=i+2}^{n} |x_k w_{kj}|$ and $\gamma_j^- \equiv \sum_{k=1,k\neq j}^{i+1} x_k w_{kj} - \sum_{k=i+2}^{n} |x_k w_{kj}|$, and then test if the logical expression $((\gamma_j^- > b_j) \wedge x_j = -1) \vee ((\gamma_j^+ < b_j) \wedge x_j = 1)$ holds, in which case we can discard the partial joint action $\mathbf{x}_{1:(i+1)}$ and prune the corresponding search space. Furthermore, it may happen that due to $\mathbf{x}_{1:(i+1)}$, the choices of actions of some not-yet-selected players have become restricted. To this end, we apply **NashProp** (Ortiz and Kearns 2002) with $\mathbf{x}_{1:(i+1)}$ as the starting configuration, and see if the choices of the other players have become restricted because of $\mathbf{x}_{1:(i+1)}$. Although each round of updating the table messages in **NashProp** takes exponential time in the maximum degree in general graphical games, we can show in a way similar to Theorem 7 that we can adapt the table updates to the case of LIGs so that it takes polynomial time.

A Divide-and-Conquer Approach. To further exploit the structure of the graph in computing the PSNE, we propose a divide-and-conquer approach that relies on the following separation property of influence games.

Property 9. *Let $G = (V, E)$ be the underlying graph of an influence game and S be a vertex separator of G such that removing S from G results in $k \geq 2$ disconnected components: $G_1 = (V_1, E_1)$, ..., $G_k = (V_k, E_k)$. Let G_i' be the subgraph of G induced by $V_i \cup S$, for $1 \leq i \leq k$. Consider the influence games on these (smaller) graphs G_i''s, where we retain all the weights of the original graph, except that we treat the nodes in S to be indifferent (that is, we remove all the incoming arcs to these nodes and set their thresholds to 0). Computing the set of PSNE on G_i''s and then merging the PSNE (by performing outer-joins of joint actions and testing for PSNE in the original influence game), we obtain the set of all PSNE of the original game.*

To obtain a vertex separator, we first find an edge separator (using well-known tools such as METIS (Karypis and Kumar 1995)), and then convert the edge separator to a vertex separator (by computing a maximum matching on the bipartite graph spanned by the edge separator). We then use this vertex separator to compute all PSNE, as outlined in Property 9. The benefits of this approach are two-fold: (1) for graphs that have good separation properties (such as preferential-attachment graphs), we have found this approach to be computationally effective in practice; and (2) this approach leads to an *anytime algorithm* for enumerating or counting PSNE: Observe that ignoring some edges from the edge separator may result in a smaller vertex separator, which greatly reduces the computation time of the divide-and-conquer algorithm at the expense of producing only a subset of all PSNE (Note that the edges that are ignored from the edge separator are not permanently removed from the original graph, and that after merging, every joint action is tested for PSNE in the original game). We can obtain progressively better result as we ignore less number of edges from the edge separator.

Computing the Most Influential Nodes

We now focus on the problem of computing the most influential set of nodes with respect to a specified desirable PSNE and a preference for sets of minimal size. In the discussion below, we also assume, *only* for the purpose of establishing the equivalence to the *minimum hitting set problem* (Karp 1972), that we are given the set of all PSNE. (As we will see, a counting routine is all that our algorithm requires, not an enumeration.) We give a hypergraph representation of this problem that would lead us to a logarithmic-factor approximation by a natural greedy algorithm.

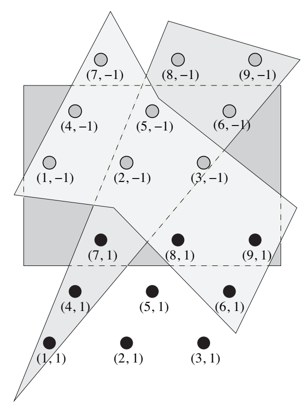

Figure 2: *A hypergraph representation of three PSNE in a 9-player game with binary actions. The PSNE shown here are the followings:* $(1, -1, -1, 1, -1, -1, 1, -1, -1)$ *(triangle),* $(-1, -1, -1, -1, -1, -1, 1, 1, 1)$ *(rectangle), and* $(-1, -1, -1, -1, -1, 1, -1, 1, 1)$ *(6-gon).*

Let us start by building a hypergraph that can represent the PSNE of a binary-action game. The nodes of this hypergraph are the player-action tuples of the game. That is, for each player i of the game, there are two nodes in the hypergraph: one in which i plays -1 (tuple $(i, -1)$, colored gray in Figure 2) and the other in which i plays 1 (tuple $(i, 1)$, colored black). For every PSNE $\mathbf{x}$ we construct a hyperedge $\{(i, x_i) \mid 1 \leq i \leq n\}$. Let us call this hypergraph the *game hypergraph*. By construction, a set of players S play the same joint-action $\mathbf{a}_S \in \{-1, 1\}^{|S|}$ in two distinct PSNE $\mathbf{x}$ and $\mathbf{y}$ of the influence game iff both of the corresponding hyperedges $e_{\mathbf{x}}$ and $e_{\mathbf{y}}$ (resp.) of the game hypergraph contains $T = \{(i, a_i) \mid i \in S\}$.

We can use the above property to translate the most influential nodes selection problem, given all PSNE, to an equivalent combinatorial problem on the corresponding game hypergraph H. Let $e_{\mathbf{x}^*}$ be the hyperedge in H corresponding to the desirable PSNE $\mathbf{x}^*$. Let us call $e_{\mathbf{x}^*}$ the *goal hyperedge*. Then the most influential nodes selection problem is the problem of selecting a minimum-cardinality set of nodes $T \subseteq e_{\mathbf{x}^*}$ such that T is contained in no other hyperedge of H. Let us call the latter problem the *unique hyperedge problem*.

Theorem 10. *The unique hyperedge problem having $2n$ nodes and h hyperedges is equivalent to the minimum hitting set problem having n nodes and h hyperedges.*

Proof Sketch. Let J be an instance of the minimum hitting set problem, specified by a hypergraph $G' = (V, E)$. Wlog, we assume that E contains the hyperedge $e^* \equiv V$. We now construct an instance I of the unique hyperedge problem, specified by the hypergraph $G = (V \times \{1, -1\}, \{e^* \times \{1\}\} \cup \{\bar{e} \times \{1\} \cup e \times \{-1\} \mid e \in E \text{ and } e \neq e^*\})$ and the goal hyperedge $e^* \times \{1\}$. We show that $S \subseteq V$ is a feasible solution to J iff $S \times \{1\}$ is a feasible solution to I.

For the reduction in the reverse direction, let us consider an instance I of the unique hyperedge problem, given by a hypergraph $G = (V, E)$, along with the goal hyperedge $e_{\mathbf{x}^*}$. We construct an instance J of the minimum hitting set problem, specified by the hypergraph $G' = (e_{\mathbf{x}^*}, \{e_{\mathbf{x}^*}\} \cup \{\bar{e} \cap e_{\mathbf{x}^*} \mid e \in E \text{ and } e \neq e_{\mathbf{x}^*}\})$ and show that a set S of nodes is a feasible solution to I iff it is a feasible solution to J. □

Immediate consequences of Theorem 10 are that the unique hyperedge problem is not approximable within a factor of $c \log h$ for some constant $c > 0$, and that it admits a $(1 + \log h)$-factor approximation (Raz and Safra 1997; Johnson 1974), where h is the total number of PSNE. The approximation algorithm in our case can be outlined as follows: At each step, select the least-degree node v of the goal hyperedge, remove the hyperedges that do not contain v, remove v from the game hypergraph, and include v in the solution set, until the goal hyperedge becomes the last remaining hyperedge in the hypergraph. In the context of the original influence game, at every round, this algorithm is essentially picking the node whose assignment would reduce the set of PSNE consistent with the current partial assignment the most. Hence, the algorithm only requires a subroutine to *count* the PSNE extensions for some given partial assignment to the players' actions, not an *a priori* full list or enumeration of all the PSNE.

Experimental Results

We have performed empirical studies on three different types of LIGs—random LIGs, preferential-attachment LIGs, and the LIGs among the US senators that has been learned from the real-world voting data of the US Congress using machine learning techniques (Honorio and Ortiz 2010).

Random Influence Games. We have studied LIGs on uniform random directed graphs. While constructing the random graphs, we have independently chosen each arc with a probability of 0.50, and assigned it a weight of -1 with a probability p (named *flip probability*) and 1 with probability $1 - p$. Several interesting findings have emerged from our study of this parameterized family of LIGs on uniform random graphs. For various flip probabilities, we have independently generated 100 uniform random graphs of 25 nodes each, and for each of these random graphs, we have first computed all PSNE using our heuristic. We have then applied the greedy approximation algorithm to obtain a set of the most influential nodes in each graph and compared the approximation results to the optimal ones.

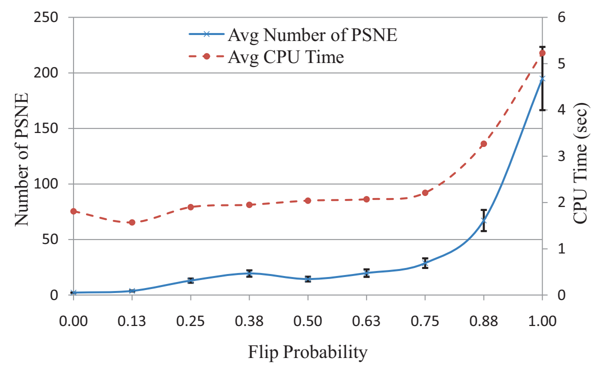

Figure 3: *PSNE computation on random LIGs. The vertical bars denote 95% confidence intervals.*

As shown in Figure 3, the number of PSNE usually increases if we have more negative-weighted arcs than positive ones, although the number of PSNE is very small relative to the maximum possible number. We have further found that although the approximation algorithm for influential nodes selection problem has a logarithmic factor worst case bound, most often the result of the approximation algorithm is very close to the optimal solution. For example, for the random games having all negative weights, in 87% of the trials the approximate solution size $\leq$ optimal size $+1$, and in 99% of the trials the approximate solution size $\leq$ optimal size $+2$.

Preferential-Attachment LIGs. We have also experimented with LIGs based on preferential-attachment graphs (Albert and Barabási 2002). In constructing these graphs, we have started with three nodes in a triangle and then progressively added each node to the graph, connecting it with three existing nodes with probabilities proportionate to the degrees. We have made each connection bidirectional and imposed the same weighting scheme as above: with the flip probability p, the weight of an arc is -1 and with probability $1 - p$ it is 1. The threshold of each node has been set to 0. We have observed that for $0 < p < 1$, these games have very few PSNE, while for $p = 0$ and $p = 1$ the number of PSNE is considerably large. Furthermore, these games show very good separation properties, making the computation amenable to the divide-and-conquer approach. We show the average number of PSNE and the average computation time for graphs of sizes 20 to 50 nodes in Figure 4 for $p = 1$ (each average is over 20 trials). Note that in contrast to the random LIGs, preferential-attachment graphs show an exponential increase in the number of PSNE as the number of nodes increase, although the number of PSNE is still a small fraction of the maximum possible number.

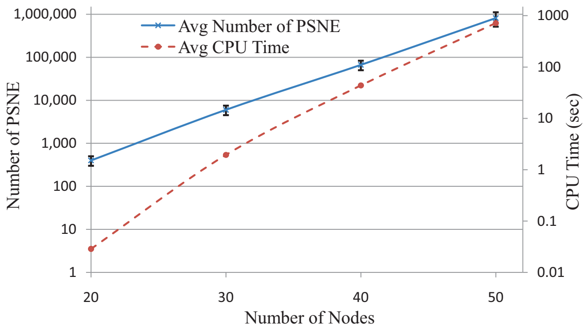

Figure 4: *PSNE computation on preferential-attachment LIGs (log-log scale). The vertical bars denote 95% confidence intervals.*

Congressional Voting. We will further illustrate our computational scheme in a real-world scenario where the strategic aspects of the agents' behavior are of prime importance. We have learned the LIGs among the senators of the 101st and the 110th US Congress (Honorio and Ortiz 2010). The 101st Congress LIG consists of 100 nodes, each representing a senator, and 936 weighted arcs among these nodes. On the other hand, the 110th Congress LIG has the same number of nodes, but it is a little sparser than the 101st one, having 762 arcs. In these LIGs, each node can play one of the two actions: 1 (yes vote) and -1 (no vote). First, we have applied the divide-and-conquer algorithm that exploits the nice separation properties of these LIGs, to find the set of all PSNE (this has been done for convenience; as discussed earlier, counting alone would have been sufficient). We have obtained a total of 143,601 PSNE for the 101st Congress graph and 310,608 PSNE for the 110th one. Note that the number of PSNE in these games is extremely small relative to the maximum possible 2^{100}. Regarding the computation time, solving the 110th Congress using the divide-and-conquer approach takes about seven hours, whereas solving the same without this approach, simply relying on the backtracking search, takes about 15 hours.

Next, we have computed the most influential senators using the approximation algorithm outlined earlier. We have obtained a solution of size five for the 101st Congress graph, which we have verified to be an optimal solution. The solution consists of the following senators: Rockefeller (Democrat, WV), Sarbanes (Democrat, MD), Thurmond (Republican, SC), Symms (Republican, ID), and Dole (Republican, KS). Interestingly, none of the maximum-degree nodes has been selected. Similarly, the six most influential senators of the 110th Congress are: Kerry (Democrat, MA), Bennett (Republican, UT), Sessions (Republican, AL), Enzi (Republican, WY), Rockefeller (Democrat, WV), and Lautenberg (Democrat, NJ).

Conclusion

We have studied the problem of identifying the most influential nodes in a network from a new game-theoretic perspective. To that end, we have introduced a rich class of games, named influence games, to capture the core strategic component of complex interactions in a network. Our computational complexity and algorithmic results carry over to other areas of game theory such as poly-matrix games.

References

Albert, R., and Barabási, A.-L. 2002. Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74(1):47–97.

Fabrikant, A.; Papadimitriou, C.; and Talwar, K. 2004. The complexity of pure nash equilibria. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, 604–612. New York, NY, USA: ACM.

Fowler, J. H., and Christakis, N. A. 2008. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study. *BMJ* 337(dec04_2):a2338–.

Gottlob, G.; Greco, G.; and Scarcello, F. 2003. Pure Nash equilibria: hard and easy games. In *Proceedings of the 9th conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 215–230.

Granovetter, M. 1978. Threshold models of collective behavior. *The American Journal of Sociology* 83(6):1420–1443.

Honorio, J., and Ortiz, L. E. 2010. Learning graphical games from behavioral data. Preprint.

Janovskaja, E. B. 1968. Equilibrium situations in multimatrix games. *Litovsk. Mat. Sb.* 8:381–384.

Johnson, D. S. 1974. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences* 9(3):256–278.

Karp, R. 1972. Reducibility among combinatorial problems. In Miller, R., and Thatcher, J., eds., *Complexity of Computer Computations*. Plenum Press. 85–103.

Karypis, G., and Kumar, V. 1995. Metis - unstructured graph partitioning and sparse matrix ordering system, version 2.0. Technical report.

Kearns, M.; Littman, M.; and Singh, S. 2001. Graphical models for game theory. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 253–260.

Kleinberg, J. 2007. Cascading behavior in networks: Algorithmic and economic issues. In Nisan, N.; Roughgarden, T.; Éva Tardos; and Vazirani, V. V., eds., *Algorithmic Game Theory*. Cambridge University Press. chapter 24, 613–632.

Milgrom, P., and Roberts, J. 1990. Rationalizability, learning, and equilibrium in games with strategic complementarities. *Econometrica* 58(6):1255–1277.

Monderer, D., and Shapley, L. 1996. Potential games. *Games and Economic Behavior* 14:124–143.

Ortiz, L. E., and Kearns, M. 2002. Nash propagation for loopy graphical games. In *Neural Information Processing Systems (NIPS)*.

Raz, R., and Safra, S. 1997. A sub-constant error-probability low-degree-test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the 29th ACM Symposium on Theory of Computing (STOC)*, 475–484.

Wasserman, S., and Faust, K. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press.