Using Social Sensors for Influence Propagation in Networks With Positive and Negative Relationships

Basak Guler, *Student Member, IEEE*, Burak Varan, *Student Member, IEEE*, Kaya Tutuncuoglu, *Student Member, IEEE*, Mohamed Nafea, *Student Member, IEEE*, Ahmed A. Zewail, *Student Member, IEEE*, Aylin Yener, *Fellow, IEEE*, and Damien Octeau

***Abstract*—Online social communities often exhibit complex relationship structures, ranging from close friends to political rivals. As a result, persons are influenced by their friends and foes differently. Future network applications can benefit from integrating these structural differences in propagation schemes through socially aware sensors. In this paper, we introduce a propagation model for such social sensor networks with positive and negative relationship types. We tackle two main scenarios based on this model. The first one is to minimize the end-to-end propagation cost of influencing a target person *in favor of* an idea by utilizing sensor observations about the relationship types in the underlying social graph. The propagation cost is incurred by social and physical network dynamics such as propagation delay, frequency of interaction, the strength of friendship/foe ties or the impact factor of the propagating idea. We next extend this problem by incorporating the impact of message deterioration and ignorance, and by limiting the number of persons influenced *against* the idea before reaching the target. Second, we study the propagation problem while minimizing the number of negatively influenced persons on the path, and provide extensions to elaborate on the impact of network parameters. We demonstrate our results in both an artificially created network and the Epinions signed network topology. Our results show that judicious propagation schemes lead to a significant reduction in the average cost and complexity of network propagation compared to naïve myopic algorithms.**

***Index Terms*—Network propagation for social media, recommender systems, signed networks, social networks, socially aware physical systems.**

Manuscript received March 15, 2014; revised August 14, 2014; accepted December 09, 2014. Date of publication January 06, 2015; date of current version February 11, 2015. This work was supported by the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance under Agreement W911NF-09-2-0053. The material in this paper was presented in part at the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining—Workshop on Curbing Collusive Cyber-gossips in Social Networks (ASONAM-C3), and 2014 IEEE Global Conference on Signal and Information Processing—Symposium on Network Theory (GlobalSIP'14). The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Christian Micheloni:.

B. Guler, B. Varan, K. Tutuncuoglu, M. Nafea, A. A. Zewail, and A. Yener are with the Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: basak@psu.edu; varan@psu.edu; kaya@psu.edu; msn139@psu.edu; aiz103@psu.edu; yener@ee.psu.edu).

D. Octeau is with the Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI 53706 USA, and also with the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: octeau@cs.wisc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2014.2386792

I. Introduction

SOCIAL media has become the primary platform for the spread of information, due to the recent proliferation of smart mobile phones, tablets and computers [1]. Social relationships and their impact on information flow [2] have been studied in various works, such as connecting people with trust scores [3], identifying the users that maximize the spread of influence [4], or utilizing social relationships in software design for assisting recommender systems [5].

Relationship types in online social groups range from like-minded friends to ideological foes. In contrast, conventional social network analysis often treats all relations as *friendly*. Accordingly, various works have recently emphasized the importance of integrating multiple relationship types in social networks [6]–[10]. Identifying positive and negative relationship types in human communities dates back to balance and status theories in social psychology [11], [12], which provide a graph-theoretic characterization of balanced structures in social communities. Signed links are also utilized in social media to represent the positive and negative relationships in human interactions [13], in which the evolution of the link structures is studied to explore the underlying tendencies of user behavior. Predicting positive and negative relationships in online network data is considered in [14] from a machine-learning framework, in which certain consistencies are observed in the relationship patterns. Key seeds are identified in a signed network for short and long term influence maximization in [9] through random diffusion of information [15]. Reference [10] studies community detection in a signed social network. Bluetooth-enabled mobile phones are used in [16] as wearable sensors for measuring information access to infer social patterns and relationships between persons using proximity, time, and location data. Human interactions are inspected in [17] through social sensors that can detect conversational dynamics automatically. Their real-time speech extracting capability can detect social signals like interest and excitement and capture the amount of influence one person has on another [18]. Recent studies also point out effective directions for turning an unsigned network to a signed one by predicting the positive and negative social ties [19]–[21]. We presume that the relationship type between two persons can be identified by extracting information from various interaction forms such as shared messages, photos and videos. An ideological ally or a foe can be identified by learning one's own ideological standing from the shared or favored media content.

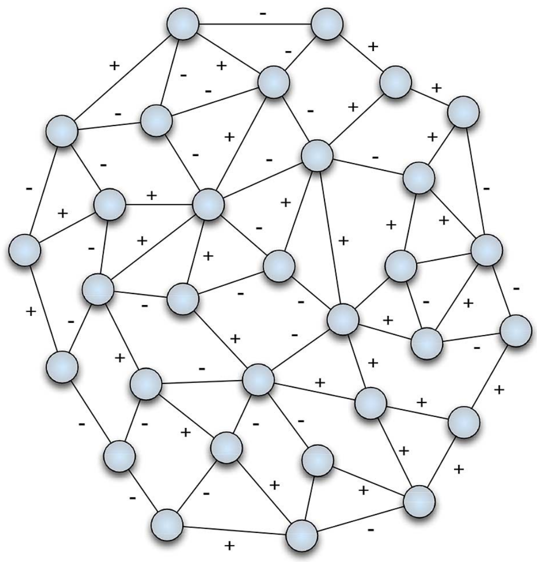


Fig. 1. Signed links in a signed social network.

In this paper, we study a signed social network in which relationships are modeled by positive or negative signs as illustrated in Fig. 1. A positive sign refers to a like-minded neighbor whereas a negative sign represents a neighbor with an opposite world view. We propose to utilize socially aware sensors, which can gather and process data available in social media, for the prediction of friendship versus antagonistic relationship types. Link prediction can be achieved through various data types, such as textual data obtained from the posts and comments in online communities, visual data from the photos shared in a social event, or audio recordings of social conversations [22], [23]. Incorporating social sensors is essential in the design of network propagation schemes that are aware of, learn from and adapt to the structural differences in human relations. Context-aware wireless sensors are used in [24] from mobile phones with access to online published content and create social links. Wireless social sensors are utilized in [25] to understand social behaviors and detect the collective behavior patterns on bird communities. A novel application is introduced in [26] for using classifiers that execute partly on sensor-enabled mobile phones and partly on backend servers to construct a personal sensing system.

Our methods are based on the property that individuals are influenced by their friends and foes differently [27], which is known as the principle of *homophily* [28]. In particular, persons tend to agree with others who are ideologically similar, and oppose to the ideas that come from their ideological foes [29]. As an example, consider an online voting process between two candidates (candidate A and candidate B) with opposite world views. Concurrently, a recommender in social media is suggesting one of the two candidates to the individuals based on the preceding votes and friendship structures. Assume that such a person, *Alice*, has two neighbors *Bob* and *Eve*. Bob has the same world interpretation with Alice, in other words is an ideological ally, whereas Eve has an opposite world view. The recommender, who can observe the individual votes, makes suggestions of type "Bob supports candidate B, do you want to vote for B, too?" Suppose that both Bob and Eve are known by the recommender to support candidate B, and that this information is still unknown to Alice. The recommender can then make one of the two suggestions to Alice, "Bob supports candidate B, do you want to vote for B, too?" or "Eve supports candidate B, do you want to vote for B, too?". In case the first suggestion is made, Alice is likely to support candidate B as Bob is an ideological ally. On the other hand, if Alice sees that Eve supports candidate B as in the latter, she will have a negative opinion about the candidate as she considers Eve as an ideological foe. In effect, the two different recommendations have the possibility of influencing Alice in two opposite directions. To this end, it is essential for the recommendation system to make judicious suggestions by taking into account the interpersonal relationship types.

We posit that persons take sides *in favor of* or *against* an idea, product, candidate, or opinion, based on the information made available to them [30]. While an idea is propagating through the network, one has a tendency to like it if it is supported by a like-minded friend, one that is ideologically similar or shares similar interests. On the other hand, a negative relationship or an antagonistic world view may cause one to act with caution to the idea promoted by the neighbor. The interesting case occurs when a neighbor with an opposite world view is *against* an idea. In such a situation, one has a tendency to go against the neighbor, which results in a positive disposition towards the original idea. While this may appear far-fetched at first glance, it fits well with many observations on various occasions, including the historical details of the European alliances before World War I [31], [32].

Modern applications of our work involve situations that arise out of differing ideas, interpretation of situations, acts, groups, events, or activities, that are led by social media. These include promoting a candidate in a voting process, civil unrest events, conflicts between pro and anti-government groups, recurring incidents of radical acts such as terrorism and violence, or simply rooting for/against a product, sports team etc. The biases of individuals are often observed through the posts shared or pages *liked* enabled by the growing use of social media. Persons whose tendencies are in disagreement can be represented by a negative link, whereas persons with similar tendencies can be considered to have a positive link. The posts from a large number of social sensors often go through a filtering process in modern applications before making their way to our *newsfeed*, or the suggestions section. The central entity who performs the filtering process can control whose posts, acts, or choices are prioritized. Judicious selection of these posts can in turn affect the alignment of a target entity towards a particular act, such as a particular candidate in a voting process, a particular action in a civil event, or to a certain opinion about a situation.

We presume that a social link incurs a cost of propagation, which incorporates a number of social and physical factors such as the propagation delay, interaction frequency, friendship/foe tie strength, or the power of the propagating idea. In doing so, we establish the optimal policies through a policy-free measurement metric. The right metric is often a weighted combination of multiple social and physical conditions and depends on the specific network goal. To this end, it is required to integrate the intended performance metric with a sensor network capable of providing accurate information in a timely manner. As an example, one physical metric we address in this study is the end-to-end delay, which is important for providing the fastest network experience to the user. As such, we study minimizing the end-to-end propagation cost to influence a target node *in favor of* an idea. This represents the fastest policy that influences a target node positively when the cost metric is the propagation delay. The network provider can then accompany these policies with routing schemes with different design goals as required. We expect our study to be useful for network applications in which social relationships cause a significant impact on influence structures.

The proposed scheme applies, including but not limited to, situations open to interpretation, as well as conflicts and crises, the management of which requires a high level of situational awareness. Effectively, such events strongly indicate the need for designing intelligent systems for achieving situational awareness, in which leveraging influence structures in network propagation schemes takes a significant part.

The remainder of the paper is organized as follows: In Section II, we introduce the system model. We present the influence propagation scheme in Section III. We extend the propagation model to account for message deterioration and ignorance in Section IV. Section V investigates how to control the number of negatively influenced users. We elaborate on minimizing the negative influence in Section VI. Sections VII and VIII generalize the propagation schemes to cyclic graphs. Numerical results are given in Section IX. We conclude the paper in Section X.

II. Social Network Model

In this section, we represent the social network with a directed acyclic graph $G = (V, E)$ with $|V|$ nodes. A directed edge exists between nodes u and v if $(u, v) \in E$. The tuple (u_x, u_y) represents the coordinates of a node $u \in V$. In the sequel, we refer to a node by its index and its coordinates interchangeably. The edge (u, v) is labeled with a sign $s_{u,v} \in \{-1, 1\}$ that stands for the relationship type between u and v, which reflects the attitude of one person towards another.

Initially, the source node is activated by an external cue such as a news article, an event, an advertising campaign or a political discussion. This node then passes the information to its neighbor which results in a positive or negative influence. The propagation continues until the message reaches the target node. This model can alternatively be used to characterize a recommendation network, in which a recommender makes suggestions to subscribed users, based on the previous choices of their neighbors. Therefore, a person is likely to be positively influenced by the recommender if the previous neighbor is a friend and is supporting the idea (candidate, product), whereas if the previous contact is an enemy, the person is likely to oppose the idea (candidate, product). Optimal propagation policies with such social structures necessitate the utilization of socially aware sensor networks that have the ability to predict relationship types and make judicious decisions. Unlike conventional routing schemes, these sensors should be able to gather and process both social data measurements such as neighbors with a friendship relation versus ideological foes and physical measurements such as the frequency of interaction or the strength of the propagation channel between the neighbors.

The cost of influence propagation between two nodes is expressed by a nonnegative weight. An example is the *propagation delay* which captures both social and physical environmental factors. From a physical perspective, it assesses the QoS (quality of service) of multi-hop sensor networks, which depends on various quantities such as the bandwidth, load, and physical distance between the travelled links. From a social perspective, it quantifies the impact of one person's actions on influencing another person, in which a smaller delay refers to a quicker response. From yet another perspective, the delay may represent the frequency of interaction between the two persons/sensors.

We focus on a social network with possibly *asymmetric* connectivities (acquaintances), in which establishing a direct link to the destination can be less than prevalent, and at times this may not be an option. For example, the destination could be a public figure *known* or *followed* by a large community such as a politician or an author. We note, however, that the incentives for propagating an idea will be different for various scenarios, and in case the source can connect directly to the destination with little effort, it may be beneficial to do so. We provide a formal definition of the influence propagation problem in the sequel.

III. Socially Constrained Minimum-Cost Propagation

Our focus in this section is on influence propagation with minimum expected end-to-end cost. We represent the source and destination nodes with u_o and u_d, respectively. Our aim is to determine the path and policy with minimum total cost for positively influencing the destination (target person). The propagation cost from u to v is given by $d_{u,v} \geq 0$. The sign of the influence between u and v is given by $s_{u,v}$. We represent the set of all possible paths from the source to the destination by $\mathcal{P}$. Then the minimum-cost positive influence propagation problem is given by:

$$\begin{aligned} \min_{P \in \mathcal{P}} \quad & \sum_{u,v\colon (u,v) \in P} d_{u,v} \\ \text{s.t.} \quad & \prod_{u,v\colon (u,v) \in P} s_{u,v} = +1 \end{aligned} \tag{1}$$

in which the objective function represents the total cost of path P. The multiplicative constraint guarantees that the target node is *positively* influenced. Note that (1) is a dynamic program that can be solved with backward induction. We label the node indices in such a way that for every edge $(u, v) \in E$, $u \leq v$, by noting that such an ordering is feasible for any directed acyclic graph [33].

We posit that the cost $d_{u,v}$ can be utilized to model the degree of alignment, namely positivity or negativity, between two persons. For instance, consider a system in which the cost $d_{u,v}$ is the delay between u initiating an action and its neighbor v reacting to it. If two persons are strongly aligned, either positive or negative, one would expect the reaction time to be low, whereas it may take longer to draw a neighbor's attention who is only weakly aligned, as it may require multiple initiatives, messages, posts, or tweets.

Algorithm 1 Backward Induction Dynamic Programming for Minimum-Cost Influence Propagation

1. Initialize $s_{u,v}$ and $d_{u,v}$ for every link $(u,v) \in E$.
2. Set the boundary conditions from (4) and (5).
3. Starting from the destination node u_d, update the value functions at each node u using
$$S(u,0) = \min_{v:(u,v)\in E}\{d_{u,v} + \delta(s_{u,v}-1)S(v,0) + \delta(s_{u,v}+1)S(v,1)\}$$
$$S(u,1) = \min_{v:(u,v)\in E}\{d_{u,v} + \delta(s_{u,v}-1)S(v,1) + \delta(s_{u,v}+1)S(v,0)\}.$$
4. Calculate the minimum end-to-end cost $S(u_o,0)$ upon reaching the source node u_0.
5. Starting from u_o, determine the optimal decisions $\pi(u)$, $\forall u \in V$ recursively.
6. Determine the optimal path using $\pi(u)$.

The arguments of the problem are the node label $u \in V$ and the parity variable $z \in \{0,1\}$. The case $z = 0$ indicates that the parity from node u to the destination node is even, i.e., the product of the signs from u to the destination is equal to $+1$. Similarly, $z = 1$ refers to an odd parity, i.e., the product of the signs from u to the destination is given by -1. Optimal value function $S(u,z)$ quantifies the minimum total cost of the optimal path from node u to the destination. The optimal policy function $\pi(u)$ defines the optimal decision taken at u which specifies the index of the node to be chosen next. The relations for the even and odd parity paths from node u to the destination are given as follows:

$$S(u,0) = \min_{v:(u,v)\in E}\{d_{u,v} + \delta(s_{u,v}-1)S(v,0) + \delta(s_{u,v}+1)S(v,1)\} \tag{2}$$
$$S(u,1) = \min_{v:(u,v)\in E}\{d_{u,v} + \delta(s_{u,v}-1)S(v,1) + \delta(s_{u,v}+1)S(v,0)\} \tag{3}$$

in which $S(u,0)$ is the even and $S(u,1)$ is the odd parity path. The delta function is given by $\delta(0) = 1$ and $\delta(x) = 0$ for all $x \neq 0$. $S(u_o,0)$ is the minimum total cost for influencing the target (destination) node positively. Lastly, we state the boundary conditions as follows:

$$i)\quad S(u_d,0) = 0, \qquad S(u_d,1) = \infty \tag{4}$$
$$ii)\quad \text{Any direction with no edge has infinite cost}. \tag{5}$$

The pseudo code of the backward induction algorithm that solves (1) is given in Algorithm 1.

IV. Propagation in the Presence of Message Deterioration and Ignorance

A propagating idea often *distorts* as it is repeated, which is known as the "Telephone" effect [34]. As such, individual interpretations or subjective priority assessments may alter the content of the message, news, or an idea, propagating in the social network.

We now elaborate on how to quantify the impact of *message freshness* on influence propagation, by allowing the nodes to ignore an incoming message based on the strength of the link and message freshness. If a node ignores a message, the recommender has to *refresh* the message to reactivate the node with an additional cost, which can be achieved by an advertisement or a promotion. The recommender can also activate a node with a cost even if it is not ignored, solely for refreshing the message. Our problem is now to determine the optimal path with minimum expected cost and the activation sequence, which corresponds to the set of nodes to activate even in the case of no ignorance. We note that if a node ignores a message, which may or may not happen, reactivation is necessary.

In order to reflect the impact of message deterioration and ignorance on network propagation, we incorporate message freshness and the possibility that persons may choose to ignore each other. Specifically, we represent message freshness by the age of a message, k, which is the number of nodes the message has passed through since the last activation. An activation sets the message age to 1. It is required if a node ignores its neighbor. In this case, the cost of activating node u is $\bar{c}_u$. In case the message is not ignored, the recommender may still choose to activate a node to reset the age to 1 and increase the impact of the message; however, there is a cost c_u for activating node u. The maximum message age is K which, when exceeded, requires the next node on the path to be activated.

We denote the cost from node u to node v for a message of age $k_{u,v}$ by the random variable $d_{u,v}(k_{u,v})$. The tendency of ignoring a message increases at each node as the message age increases. The probability that node v will ignore node u, when a message of age $k_{u,v}$ is conveyed from u to v, is given by $p_{u,v}(k_{u,v})$.

The ignorance probability $p_{u,v}$ can be used to define the degree of connectivity between two persons. An intermittent connection or a weak tie can be represented by a large $p_{u,v}$, whereas smaller values can be used for stronger ties.

We presume that the nodes know the message age, which could be included in the message when necessary, in all other applications it can be set to 1. It could also measure how much the message loses its effectiveness, as it passes through hops. For example, in the first hop, the person could be very eager about the message, but when it passes through other nodes, it may lose some of its *content, quality*, and the remaining persons may lose their interest in it accordingly.

The minimum expected cost can be determined as the solution of the following problem:

$$\begin{aligned}\min_{\substack{P\in\mathcal{P},\\ a_{u,v}}} \sum_{(u,v)\in P} &\big\{\mathrm{E}[d_{u,v}(k_{u,v})] + c_v\delta(a_{u,v}-1)(1-p_{u,v}(k_{u,v})) + \bar{c}_v p_{u,v}(k_{u,v})\big\}\\ \text{s.t.} \prod_{(u,v)\in P} &s_{u,v} = 1,\\ &a_{u,v} \in \{0,1\}, \forall (u,v)\in P,\\ &k_{u,v} \in \{1,2,\ldots,K\},\ \forall (u,v)\in P,\\ &k_{v,w} = (k_{u,v}+1)\delta(a_{u,v}),\ \forall (u,v),(v,w)\in P,\\ &k_{u_o,v} = 1,\ \forall (u_o,v)\in P\end{aligned} \tag{6}$$

Algorithm 2 Minimum-Cost Influence Propagation in the Presence of Message Deterioration and Ignorance

1. Initialize $s_{u,v}$ and $d_{u,v}(k_{u,v})$ for each message age $k_{u,v} = 1, \ldots, K$ for every $(u,v) \in E$.
2. Assign the boundary conditions from
$$S(u_d, k, 0) = 0,\ S(u_d, k, 1) = \infty,\ k = 1, \ldots, K.$$
3. Going backwards from the destination u_d, find the value functions $S(u,k,z)$ at each u, k, z from
$$\begin{aligned} S(u,k,z) = \min_v \mathrm{E}[d_{u,v}(k)] & \\ + p_{u,v}(k)(\bar{c}_v + \delta(s_{u,v}-1)S(v,1,z) & \\ + \delta(s_{u,v}+1)S(v,1,\bar{z})) & \\ + (1-p_{u,v}(k))\min\{\delta(s_{u,v}-1)S(v,k+1,z) & \\ + \delta(s_{u,v}+1)S(v,k+1,\bar{z}), \delta(s_{u,v}-1)S(v,1,z) & \\ + \delta(s_{u,v}+1)S(v,1,\bar{z}) + c_v\}. & \end{aligned}$$
4. When the source node u_o is reached, calculate the minimum cost $S(u_o, 1, 0)$.
5. Determine the optimal decisions starting from the source.
6. Find the optimal path through the optimal decisions.

in which we optimize over the path P and the activation sequence $(a_{u,v})$ which is 1 if node u decides to activate node v and 0 otherwise. We use dynamic programming to solve (6). The recursive equations for backward induction are given as:

$$\begin{aligned} S(u,k,z) = \min_v \ \mathrm{E}[d_{u,v}(k)] & \\ + p_{u,v}(k)(\bar{c}_v + \delta(s_{u,v}-1)S(v,1,z) & \\ + \delta(s_{u,v}+1)S(v,1,\bar{z})) & \\ + (1-p_{u,v}(k))\min\{\delta(s_{u,v}-1)S(v,k+1,z) & \\ + \delta(s_{u,v}+1)S(v,k+1,\bar{z}), \delta(s_{u,v}-1)S(v,1,z) & \\ + \delta(s_{u,v}+1)S(v,1,\bar{z}) + c_v\} & \qquad (7) \end{aligned}$$

where $\bar{0} = 1$ and $\bar{1} = 0$, and $S(u,k,z)$ denotes the value function at u with message age $k \in \{1, 2, \ldots, K\}$ and disparity $z \in \{0, 1\}$. The boundary conditions are:

$$S(u_d, k, 0) = 0, S(u_d, k, 1) = \infty,\ \forall k \in \{1, \ldots, K\}. \quad (8)$$

Finally, the answer for the minimum expected cost is $S(u_o, 1, 0)$. Algorithm 2 provides the steps of the proposed method.

V. Limiting the Number of Negative Influences

The propagation schemes proposed in the previous sections focused on influencing a target node positively without taking into account the dispositions of the intermediate nodes. However, real-life scenarios often require avoiding the situations in which a large number of intermediate nodes are influenced negatively. Accordingly, we consider in this section the problem of how to influence a target node positively while limiting the number of negatively influenced intermediate nodes.

We provide a forward induction dynamic programming algorithm to quantify the influence from the source node to the intermediate nodes. We denote P_u as the fragment of the path P that ends at node u. In other words, P_u is a path from u_o to u with the condition that if $(u', v') \in P_u$, then $(u', v') \in P$. The problem can be formally stated as follows:

$$\begin{aligned} \min_{P \in \mathcal{P}} & \sum_{u,v:(u,v)\in P} d_{u,v} \\ \text{s.t.} & \prod_{u,v:(u,v)\in P} s_{u,v} = +1 \\ & \left| \left\{ u : \prod_{u',v':(u',v')\in P_u} s_{u',v'} = -1 \right\} \right| \le Q \end{aligned} \qquad (9)$$

in which Q is the maximum allowed number of negatively influenced intermediate nodes. We consider deterministic costs.

We let $S(u,q,0)$ denote the value of the minimum-cost even-parity path connecting u_o with u when the number of negatively influenced intermediate users are no more than q. $S(u,q,1)$ stands for the minimum-cost for the odd-parity path between u_o and u with at most q negatively influenced intermediate users. The recursive relations for the even and odd parity paths at each node are then given by:

$$S(u,q,0) = \min_{v:(v,u)\in E} \{d_{v,u} + \delta(s_{v,u}-1)S(v,q,0) + \delta(s_{v,u}+1)S(v,q-1,1)\} \quad (10)$$

$$S(u,q,1) = \min_{v:(v,u)\in E} \{d_{v,u} + \delta(s_{v,u}-1)S(v,q-1,1) + \delta(s_{v,u}+1)S(v,q,0)\} \quad (11)$$

for $u \in V$ and $0 \le q \le Q$. The boundary conditions are given as follows:

$$S(u_o, q, 0) = 0, S(u_o, q, 1) = \infty,\ \forall q \in \{0, 1, \ldots, Q\}. \quad (12)$$

The minimum cost to influence the destination positively, with the condition that no more than Q intermediate nodes are affected negatively, is then given by $S(u_d, Q, 0)$. The steps of the forward induction dynamic program is given in Algorithm 3.

Algorithm 3 Forward Induction Dynamic Programming for Limiting the Number of Negative Influences

1. Initialize the sign and cost of every edge.
2. Assign the boundary conditions from
$$S(u_o, q, 0) = 0,\ S(u_o, q, 1) = \infty,\ \forall q \in \{0, 1, \ldots, Q\}.$$
3. Starting from the source node u_o, update the value functions at each node u via
$$\begin{aligned} S(u,q,0) &= \min_{v:(v,u)\in E} \{d_{v,u} + \delta(s_{v,u}-1)S(v,q,0) \\ &\qquad + \delta(s_{v,u}+1)S(v,q-1,1)\} \\ S(u,q,1) &= \min_{v:(v,u)\in E} \{d_{v,u} + \delta(s_{v,u}-1)S(v,q-1,1) \\ &\qquad + \delta(s_{v,u}+1)S(v,q,0)\}. \end{aligned}$$
4. Calculate the minimum end-to-end cost $S(u_d, Q, 0)$ upon reaching the destination u_d.
5. Determine the optimal path from the optimal decisions.

Algorithm 4 Minimize the Total Number of Negatively Influenced Persons

1. Define the positive and negative relationships.
2. Set the boundary conditions from
$$S(u_o, n, 0) = 0,\ S(u_o, n, 1) = \infty,\ n = 0, 1, \ldots, N.$$
3. Start from the source node u_o and update the value functions at each node u according to
$$S(u,n,0) = \min_{v:(u,v)\in E}\{\delta(s_{u,v}-1)S(v,n-1,0) + \delta(s_{u,v}+1)(S(v,n-1,1)+1)\}$$
$$S(u,n,1) = \min_{v:(u,v)\in E}\{\delta(s_{u,v}-1)(S(v,n-1,1)+1) + \delta(s_{u,v}+1)S(v,n-1,0)\}.$$
4. The total minimum number of negatively influenced persons on the path is $S(u_d, N, 0)$.
5. The optimal path can be determined from the optimal decisions.

VI. Minimizing the Number of Negative Influences

In this section, we study the problem of *minimizing the number of negatively influenced users* subject to a maximum number of hops allowed before reaching the target node. We state this problem as follows:

$$\begin{aligned} \min_{P\in\mathcal{P}} \quad & \left|\left\{u : \prod_{u',v':\,(u',v')\in P_u} s_{u',v'} = -1\right\}\right| \\ \text{s.t.} \quad & \prod_{u,v:\,(u,v)\in P} s_{u,v} = +1, \\ & |P| \le N. \end{aligned} \tag{13}$$

We denote $S(u,n,0)$ as the number of negatively influenced users through the even-parity path between u_o and u where no more than n hops are used to reach u. We denote $S(u,n,1)$ as the number of negatively influenced users through the odd-parity path between u_o and u with at most n hops from u_o to u.

The recursive relations for the even and odd parity paths are given as follows:

$$S(u,n,0) = \min_{v:(u,v)\in E}\{\delta(s_{u,v}-1)S(v,n-1,0) + \delta(s_{u,v}+1)(S(v,n-1,1)+1)\} \tag{14}$$

$$S(u,n,1) = \min_{v:(u,v)\in E}\{\delta(s_{u,v}-1)(S(v,n-1,1)+1) + \delta(s_{u,v}+1)S(v,n-1,0)\} \tag{15}$$

where $u \in V$ and $n = 1, \ldots, N$. The maximum number of hops allowed to reach the destination is N. The boundary conditions for this problem can be defined as follows:

$$S(u_o,n,0) = 0,\ S(u_o,n,1) = \infty,\ \forall n \in \{0,1,\ldots,N\}. \tag{16}$$

The solution is given by $S(u_d, N, 0)$, which refers to the even-parity path with the minimum number of negatively influenced users upon reaching the destination with no more than N hops. The steps of the forward induction dynamic program to find the optimal path for influencing a target node positively while minimizing the number of negatively influenced persons on the path is given in Algorithm 4.

Algorithm 5 Minimum-Cost Propagation with Positive Influence for Cyclic Graphs

1. Initialize the sets $N_+ = \{u_o\}$, $N_- = \{u_o\}$.
2. Assign the permanent labels of the source node u_o as $\pi'_+(u_o) = 0$ and $\pi'_-(u_o) = \infty$.
3. Set the temporary labels of remaining nodes $u \in V$ by:
$$\pi_+(u) = \begin{cases} d_{u_o,u} & \text{if } s_{u_o,u} = +1 \\ \infty & \text{o.w.} \end{cases}$$
$$\pi_-(u) = \begin{cases} d_{u_o,u} & \text{if } s_{u_o,u} = -1 \\ \infty & \text{o.w.} \end{cases}$$
where an infinite cost is used whenever no edge exists between nodes u_o and u.
4. Find a node $v \in V$ such that:
$$\pi(v) = \min_{i\in V-N_+,\, j\in V-N_-}\{\pi_+(i), \pi_-(j)\}$$
5. **if** $\pi(v) = \pi_+(v)$
   $\pi'_+(v) = \pi(v)$ and $N_+ = N_+ \cup \{v\}$
6. **else**
   $\pi'_-(v) = \pi(v)$ and $N_- = N_- \cup \{v\}$
7. **if** $N_+ \cap N_- = V$
   STOP
   else
8. Update the temporary labels $\forall (v,u) \in E$:
9. **if** $s_{v,u} = +1$
10. **if** $(\pi(v) = \pi_+(v)) \wedge (u \in N - N_+)$
    $\pi_+(u) = \min(\pi_+(u), \pi'_+(v) + d_{v,u})$
11. **else if** $(\pi(v) = \pi_-(v)) \wedge (u \in N - N_-)$
    $\pi_-(u) = \min(\pi_-(u), \pi'_-(v) + d_{v,u})$
12. **else**
13. **if** $(\pi(v) = \pi_+(v)) \wedge (u \in N - N_-)$
    $\pi_-(u) = \min(\pi_-(u), \pi'_+(v) + d_{v,u})$
14. **else if** $(\pi(v) = \pi_-(v)) \wedge (u \in N - N_+)$
    $\pi_+(u) = \min(\pi_+(u), \pi'_-(v) + d_{v,u})$
15. Go to Step 4.

VII. Minimum-Cost Influence Propagation for Graphs With Cycles

We consider in this section the minimum-cost influence propagation problem from (1) for directed cyclic graphs. We note that the methods introduced in Section III cannot be applied to solve (1) directly, since the graphs we study in this section may involve directed cycles. Therefore, we propose a modified Dijkstra-like algorithm to tackle (1). Initially, we define positive and negative temporary labels $\pi_+(u)$ and $\pi_-(u)$ for each node $u \in V$. Similarly, we denote $\pi'_+(u)$ and $\pi'_-(u)$ as the permanent positive and negative labels for each $u \in V$. The sets of nodes

Algorithm 6 Minimizing the Total Number of Negatively Influenced Persons for Cyclic Graphs

1. Initialize $N_+ = \{u_o\}$, $N_- = \{u_o\}$.
2. Define the permanent labels of the source u_o as $\pi'_+(u_o) = 0$ and $\pi'_-(u_o) = \infty$.
3. Assign the temporary labels of the remaining nodes $u \in V$ by:

$$\pi_+(u) = \begin{cases} 0 & \text{if } s_{u_o,u} = +1 \\ \infty & \text{o.w.} \end{cases}$$
$$\pi_-(u) = \begin{cases} 1 & \text{if } s_{u_o,u} = -1 \\ \infty & \text{o.w.} \end{cases}$$

where an infinite cost represents that no edge exists between u_o and u.

4. Find a node $v \in V$ such that:

$$\pi(v) = \min_{i \in V - N_+, j \in V - N_-} \{\pi_+(i), \pi_-(j)\}$$

5. **if** $\pi(v) = \pi_+(v)$

$$\pi'_+(v) = \pi(v) \text{ and } N_+ = N_+ \cup \{v\}$$

6. **else**

$$\pi'_-(v) = \pi(v) \text{ and } N_- = N_- \cup \{v\}$$

7. **if** $N_+ \cap N_- = V$

   STOP

   else

8. Update the temporary labels $\forall (v, u) \in E$:
9. **if** $s_{v,u} = +1$
10. **if** $(\pi(v) = \pi_+(v)) \wedge (u \in N - N_+)$

$$\pi_+(u) = \min(\pi_+(u), \pi'_+(v))$$

11. **else if** $(\pi(v) = \pi_-(v)) \wedge (u \in N - N_-)$

$$\pi_-(u) = \min(\pi_-(u), \pi'_-(v) + 1)$$

12. **else**
13. **if** $(\pi(v) = \pi_+(v)) \wedge (u \in N - N_-)$

$$\pi_-(u) = \min(\pi_-(u), \pi'_+(v) + 1)$$

14. **else if** $(\pi(v) = \pi_-(v)) \wedge (u \in N - N_+)$

$$\pi_+(u) = \min(\pi_+(u), \pi'_-(v)).$$

15. Go to Step 4.

that are assigned permanent positive/negative labels are represented by N_+ and N_-, respectively. The steps of our solution for cyclic graphs are provided in Algorithm 5.

It is important to note that the optimal path in our model may include a cycle, unlike the generalized shortest path algorithms for cyclic graphs. The intuition behind this idea lies in the fact that traversing a cycle may result in an even parity path with a smaller cost than an acyclic path, due to a sign change through the cycle.

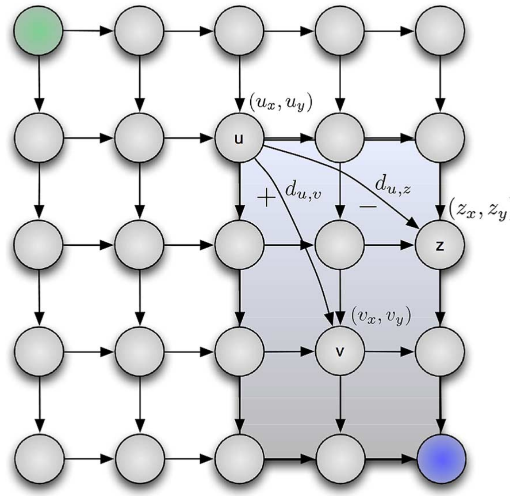


Fig. 2. Grid network structure for the signed social graph.

VIII. Minimizing the Number of Negatively Influenced Persons for Graphs With Cycles

In this part, we introduce a Dijkstra-like algorithm for minimizing the total number of negatively influenced persons while influencing a destination in favor of an idea when the underlying graph contains cycles. From an algorithmic perspective this problem can be formulated in a similar way to Algorithm 5, however, we now update the temporary labels of each node at each iteration to reflect the minimum number of negatively influenced nodes from the source to the node. The steps of this algorithm are provided in Algorithm 6.

IX. Numerical Results

We first consider a small-scale network for our simulations to motivate the propagation model and the optimal policies. Next, we switch to a large-scale network and use the online Epinions dataset to test our findings, and to demonstrate the impact of our results. To this end, we first study a grid network with directed acyclic links as shown in Fig. 2. In order to prevent directed cycles, we focus our attention on grid networks where edge (u, v) exists only if $u_x \leq v_x$, $u_y \leq v_y$, and $u \neq v$. That is, node u can only influence the nodes in the shaded rectangle in Fig. 2. Here, the source node is at the top left corner in green and the destination node is at the bottom right corner in blue.

We consider a random graph where the existence of edge (u, v) is modeled by a Bernoulli random variable where the probability of existence for the edge is monotonically decreasing in the distance between nodes u and v. We presume that edges with a small $||u - v||$ model close neighbors that frequently interact with each other. We note that this differs from the traditional notion of *friendship*, as two individuals may be frequently engaging in social interactions even if they are persons of different ideologies such as political rivals. To this end, we posit that this information is gathered from sensor data that measures the frequency of one person *interacting* with another person through social discussions or debates. This can be obtained by various methods ranging from analyzing the conversations in which one person *mentions* another or processing the textual transactions in social media. Similarly, a large $\|u-v\|$ stands for distant neighbors who know each other at an acquaintance level, and do not interact frequently. The propagation cost between nodes u and v is uniformly distributed over $[0, d_{\max}(u,v)]$:

$$d(u,v) \sim U(0, d_{\max}(u,v)) \tag{17}$$

where $d_{\max}(u,v)$ is a monotonically increasing function of the distance between the two nodes:

$$d_{\max}(u,v) = \beta\|u-v\|^{\alpha} \quad \alpha, \beta \geq 0 \tag{18}$$

where $\|u-v\| \geq 1$ for all $u \neq v$. The parameter α is introduced to capture the impact of social distances on physical costs such as propagation delay. To this end, a large α intensifies the impact of the distance between two neighbors on the propagation cost for the edge between these two neighbors. In effect, many real-world applications suggest that propagating a message through distant neighbors often takes more effort. On the other hand, with a low α, distant neighbors are treated by the network as close contacts as their propagation cost approaches to those. All neighbors, whether socially distant or close, are treated as equals by the network when α is zero, i.e., the distance between nodes has no effect on the propagation cost. Hence, this parameter is termed the *distance impact parameter* throughout our analysis. The coefficient β is a design-specific weight parameter that is equal for all node pairs. We denote the probability of an edge having a positive sign by μ, which refers to a friendship relation between the two nodes. Accordingly, the probability of an edge having a negative label is $\bar{\mu} = 1-\mu$ in which case the two persons experience an antagonistic relationship type. Unless otherwise stated, we choose $\alpha = 1/2, \beta = 1$, and $\mu = 1/2$ as the default values for our simulations.

We demonstrate the optimal policies for Algorithm 1 for a 10-by-10 grid network in Figs. 3(a)–3(c) for various distance impact parameters. We observe from Fig. 3(a) that for a small α, the optimal policy is achieved through distant neighbors as the algorithm utilizes edges with longer distances without incurring a high propagation cost. This is consistent with our intuition of distant neighbors becoming equally efficient as close neighbors when α is decreased. Hence, the algorithm reduces the number of hops in order to lower the end-to-end costs, while satisfying the positive influence constraint. On the other hand, Fig. 3(c) shows that when α is large, propagating through the distant neighbors becomes too costly, and hence the optimal policy is to follow close neighbors with more hops instead of the distant ones.

We next introduce *ignorance* to our simulations through an ignorance probability $p_{u,v}(k_{u,v})$ which is the probability that node v will ignore node u while u is attempting to transmit a message of age $k_{u,v}$ to v. It is defined as a monotonically increasing function of $k_{u,v}$ and the distance between the two nodes. Figs. 4(a)–4(c) show optimal paths for cost minimization with message deterioration and ignorance following the steps in Algorithm 2. By comparing Figs. 4(a) and 4(b), we observe that increasing the activation cost results in a lower number of activations even though older messages are more likely to be ignored. We note that in Fig. 4(a) with a low activation cost, the second node on the path is activated even though the message it receives has age 1. This is done in order to keep the message fresh without incurring a high activation cost, and thus prevent ignorance further down the path. Fig. 4(c) shows the optimal path for the same setup except the costs do not depend on distance. As a result, the optimal path is able to make bigger jumps without incurring additional cost. However, we observe that bigger jumps are more likely to result in ignorance, and therefore a penalty for activation in the total cost.

Figs. 5(a)–5(c) show optimal paths that minimize the total cost while negatively influencing no more than K nodes calculated by Algorithm 3. As can be observed, a lower K limits the feasibility of paths more strictly. Thus, the minimum total cost potentially increases. In addition, we see that the optimal path

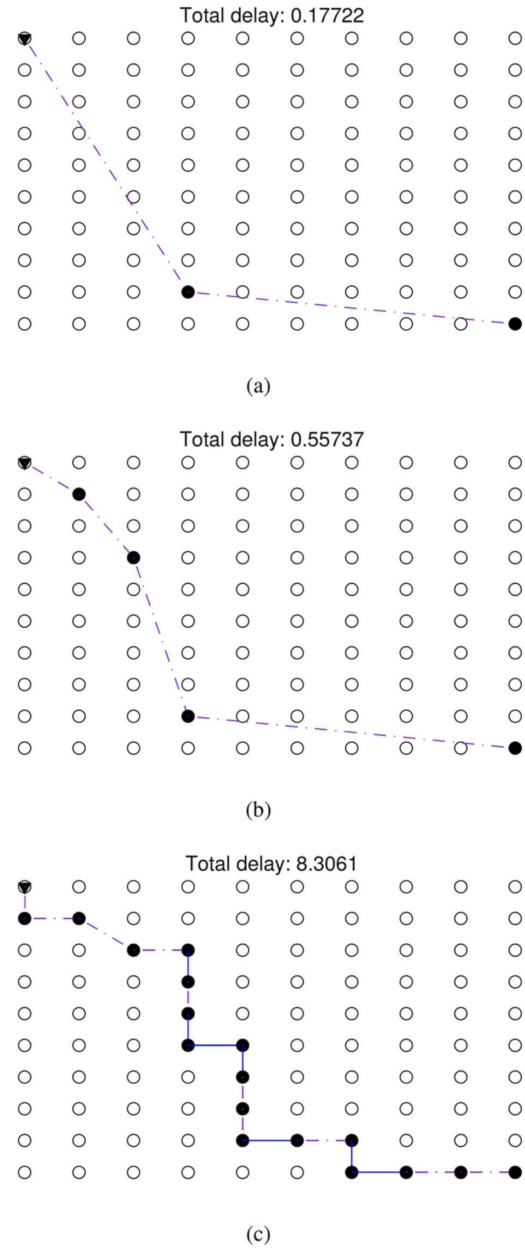


Fig. 3. Simulation results for cost minimization with distance impact parameter (a) $\alpha = 0$, (b) $\alpha = 0.5$, (c) $\alpha = 4$. Solid lines denote edges with a positive sign and dotted lines denote edges with a negative sign. The nodes visited by the optimal path are demonstrated by filled circles.

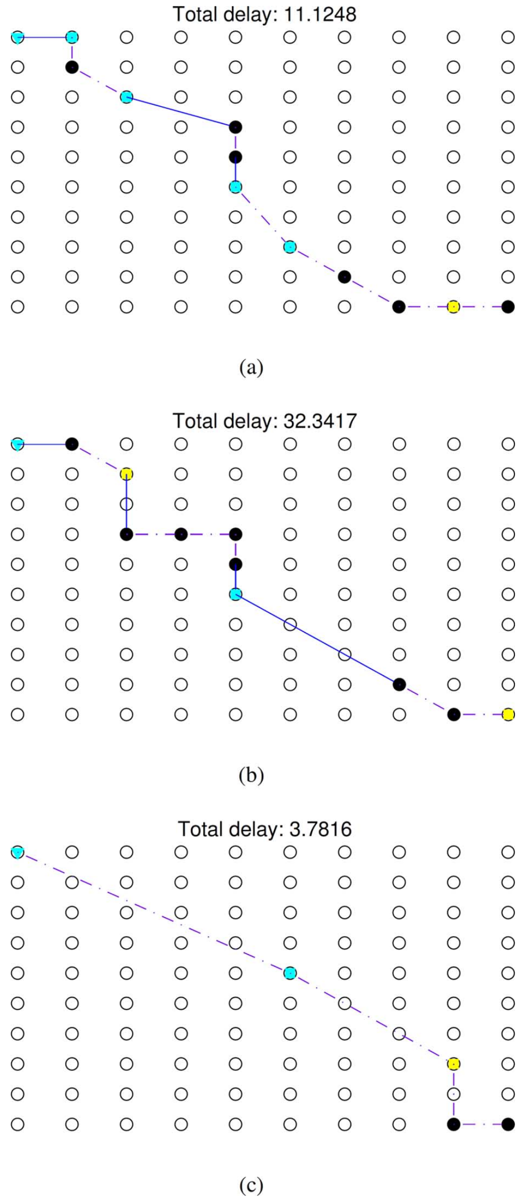


Fig. 4. Simulation results for cost minimization with message deterioration and ignorance with activation cost (a) $c_u = \bar{c}_u = 1\ \forall u$, (b) $c_u = \bar{c}_u = 100\ \forall u$, (c) $c_u = \bar{c}_u = 1\ \forall u$ with distance-independent costs. Solid lines denote edges with a positive sign and dotted lines denote edges with a negative sign. The nodes visited by the optimal path are filled where a yellow filling indicates activation as a result of ignorance, a cyan filling indicates activation without ignorance.

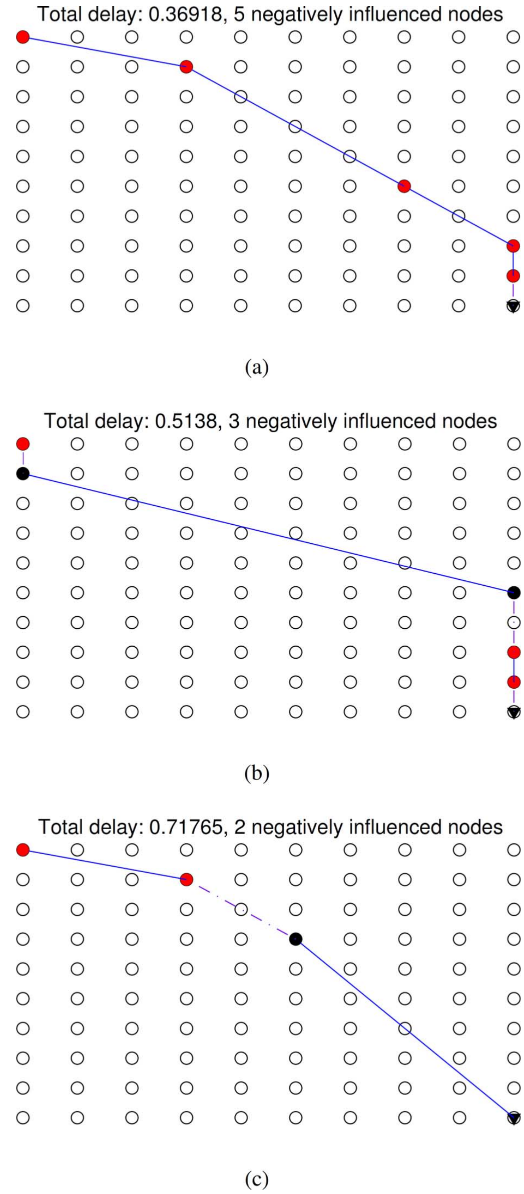


Fig. 5. Simulation results for cost minimization with at most K negatively influenced nodes where (a) $K = 10$, (b) $K = 3$, (c) $K = 2$. Solid lines denote edges with a positive sign and dotted lines denote edges with a negative sign. The nodes visited by the optimal path are filled where a red filling indicates a negatively influenced node.

in Fig. 5(a) negatively influences only 5 nodes when it can actually influence $K = 10$ nodes. This implies that it is not always optimal to negatively influence as many nodes as possible, i.e., increasing K does not always result in a lower total cost.

Next, Figs. 6(a)–6(c) show optimal paths that minimize the number of negatively influenced nodes in at most K hops via Algorithm 4. As in the previous experiment, lowering the value of K results in the elimination of some of the feasible paths, and thus the optimal path is compelled to negatively influence some of the nodes. Another interesting consequence of this limitation is that the optimal policy may require the source node to be seeded with the opposite of an idea, i.e., should be recommended against the idea or should start spreading negative rumors on the original idea in order to influence the destination positively. This in turn allows the network to utilize an odd path to the destination which results in a smaller number of negatively influenced nodes than that results by the even paths.

Finally, we elaborate on the impact of positive and negative edge sign distributions on network propagation. To this end, we study the optimal policies under various edge sign probabilities λ in Figs. 7(a)–7(c) and Figs. 8(a)–8(c). The evaluations are performed for Algorithm 3 in which we observe that the problem formulation is well suited to demonstrate the effect of sign distributions. We present the optimal propagation policy of minimum end-to-end cost for influencing a target node in favor of an idea when no intermediate node is allowed to be influenced negatively in Figs. 7(a)–7(c). We observe that as μ, i.e., the probability of an edge having a positive sign, increases, the

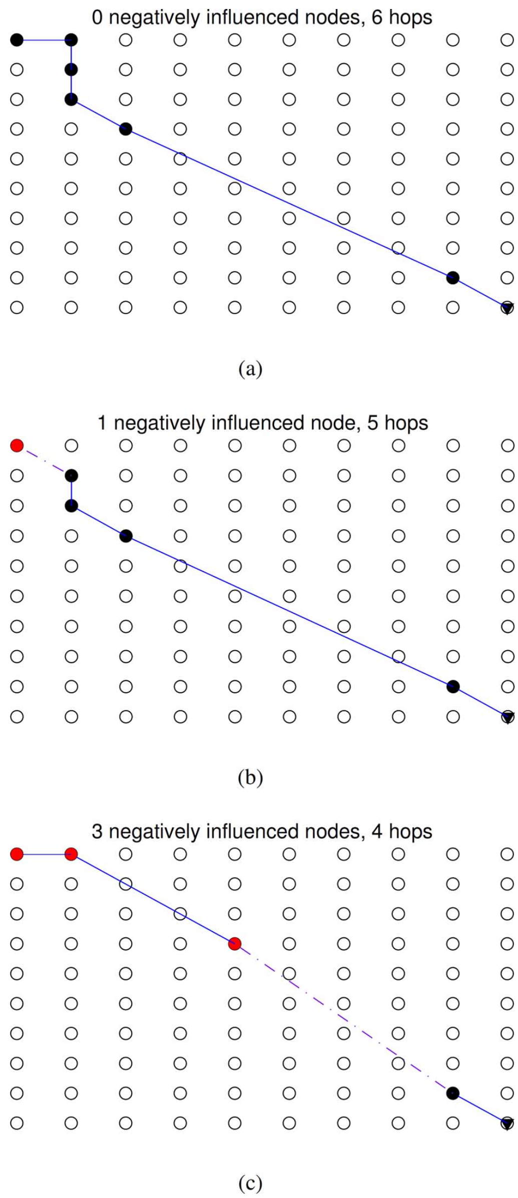


Fig. 6. Simulation results for negative influence minimization with at most K hops where (a) $K = 10$, (b) $K = 5$, (c) $K = 4$. Solid lines denote edges with a positive sign and dotted lines denote edges with a negative sign. The nodes visited by the optimal path are filled where a red filling indicates a negatively influenced node.

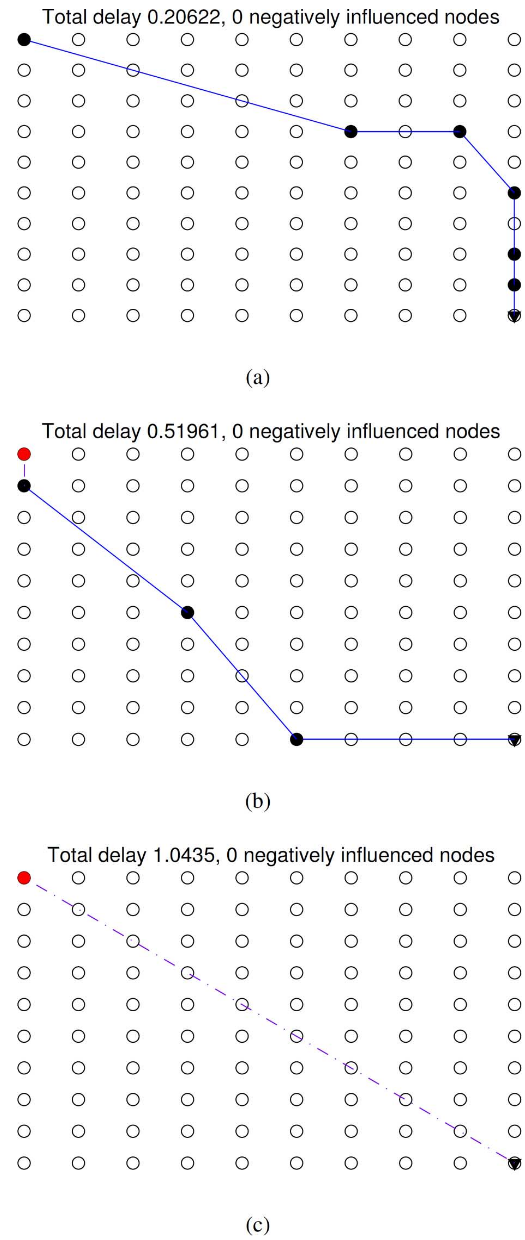


Fig. 7. Simulation results for cost minimization with at most $K = 0$ negatively influenced nodes with positive edge probability (a) 1, (b) 0.5, (c) 0. Solid lines denote edges with a positive sign and dotted lines denote edges with a negative sign. The nodes visited by the optimal path are filled where a red filling indicates a negatively influenced node.

network consists mainly of friendship connections. As a result, the algorithm can choose the best propagation path to minimize the end-to-end cost over a large number of positive paths as in Fig. 7(a). On the other hand when μ is decreased, the number of paths that consists only of positively influenced nodes decreases as well, and the algorithm reduces the number of hops as much as possible, which becomes a single node in Fig. 7(c). The tolerance on the number of negatively influenced intermediate persons is increased in Figs. 8(a)–8(c). This in turn allows for greater flexibility on the feasible paths, and the algorithm can now choose a new optimal path with a lower total propagation cost. Note that the optimal policies for the two problems coincide when $\mu = 1$, which eliminates the negative paths and reduces the problems into conventional minimum delay network propagation. We observe that the optimal policies in Figs. 7(b), 7(c) and Figs. 8(b), 8(c) require initialization with a negative disposition at the source as discussed for the previous algorithms.

In addition to the small-scale simulations, we also perform large-scale evaluations using online data. We chose to use the Epinions social graph [13], a common topology used in the signed networks literature [9], [10], [14], [19], [20] for analyzing friend and foe relationships in addition to trust and distrust. Epinions is a consumer review website where users can indicate their friends and foes based on the opinions of other users. This signed social graph has 131828 nodes and 841372 edges, with a diameter (longest shortest path) of 14. Throughout our evaluations, the source and destination nodes are selected randomly. For every possible source-destination pair, we try to

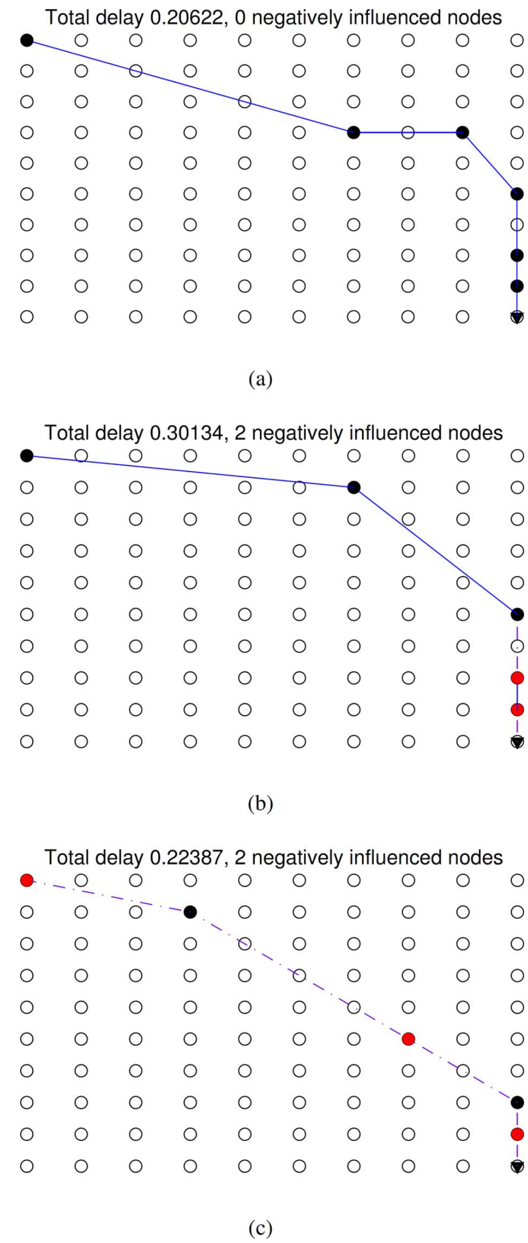


Fig. 8. Simulation results for cost minimization with at most $K = 2$ negatively influenced nodes with positive edge probability (a) 1, (b) 0.5, (c) 0. Solid lines denote edges with a positive sign and dotted lines denote edges with a negative sign. The nodes visited by the optimal path are filled where a red filling indicates a negatively influenced node.

find the optimal path and propagation policy such that the source positively influences the destination.

We compare the results between Dijkstra-type algorithms and a naïve myopic algorithm to find a low cost positive path. The first Dijkstra-type algorithm we implement is Algorithm 5, which finds paths that minimize the sum of the costs on the path between source and destination nodes, where the cost between adjacent nodes i and j is given by $\kappa|i-j|$ where κ is a weight parameter. In our simulations, we select $\kappa = 0.1$. The myopic algorithm is referred to as *shortest DFS*. It is a depth first search algorithm that traverses the graph starting from the source looking for the destination. At each node, it selects the successor with lowest cost, recursively repeating the process until the destination is reached. If a path from a node to the destination is not found, the algorithm selects a successor of the node with higher cost. For computational reasons, we limit the length of the paths to 1500. Finally, we implement Algorithm 6, which seeks to minimize the number of negatively influenced nodes on a path while influencing a destination *in favor of* an idea. This Dijkstra-type algorithm is termed *min negative path* in the sequel.

First, we randomly select 100 sources and 100 destinations. Using Algorithm 5, we find that each of the 100 sources is positively connected to 88.3 destinations on average. The median number of destinations that are positively connected to each source is 96.0. The average (respectively median) path length is 54.45 (respectively 40.0) hops with a variance of 2090.85 hops. The average (respectively median) path cost is 3436.57 (respectively 2488.4) with a variance of 10590519.21. The positive paths found by the *shortest DFS* algorithm have an average (respectively median) length of 660.73 (respectively 638.0) hops with a variance of 160122.99. The average (respectively median) cost of these paths is 17604.64 (respectively 16875.4) with a variance of 140341478.22. It can be observed that on the average, the cost of the paths from Algorithm 5 is less than a fifth of the cost of the paths found by the *shortest DFS* algorithm.

For Algorithm 6 we find that the average (respectively median) path length is 4.023 (respectively 4.0) hops, with a variance of 0.843 hops. Relaxing the constraint to minimize costs implies much shorter paths than what is found by Algorithm 5. The average (respectively median) number of negatively influenced nodes on each path is 0.096 (respectively 0), with a variance of 0.113. This means *at least half of the paths* have no negatively influenced nodes.

Next, we implement the algorithms on a randomly selected 500 sources and 500 destinations. From Algorithm 5, we find that each source is positively connected on average to 436.998 destinations. The median number of destinations positively connected to each source is 490.0. The average (median) path length is 55.15 (respectively 42.0) with variance 2166.64. The average (respectively median) path cost is 3419.91 (respectively 2363.7) with a variance of 10727606.92. On the other hand, with *shortest DFS* the average (median) length of the paths found is 726.95 (respectively 727.0). The average (median) cost of these paths is 19134.18 (respectively 18425.9). Similar to the previous case, the average cost of Algorithm 5 is less than a fifth of the cost of the paths found by *shortest DFS*.

We also find that, for Algorithm 6 the average (median) path length is 4.097 (respectively 4.0) hops, with a variance of 1.044 hops. The average (median) number of negatively influenced nodes on each path is 0.057 (respectively 0), with a variance of 0.058. Again, at least half of the optimal paths have no negatively influenced nodes.

The results for the analysis of Algorithm 5 with the Epinions dataset are provided in Table I for 100, 500 and 10000 sources and destinations, respectively. Similarly, the results of the *shortest DFS* (myopic) algorithm are given in Table II. From comparing the results in Tables I and II, we observe that the average cost of the paths found by the *shortest DFS* algorithm is five times the cost of the paths found by Algorithm 5. Further analysis on the selected 100 sources and 100 destinations show that, among the 10000 possible source-destination pairs, in 8957 cases the destination is not reachable from the source with the

TABLE I
MINIMUM COST ALGORITHM (ALGORITHM 5) RESULTS FOR THE EPINIONS DATASET.

Number of sources	Number of destinations	Total number of paths found	Average path length	Median path length	Average path cost	Median path cost
100	100	8830	54.450	40.0	3436.569	2488.4
500	500	218499	55.148	42.0	3419.907	2363.7
10000	10000	78029370	47.024	30.0	5027.842	4145.1

TABLE II
RESULTS OF THE *shortest DFS* (MYOPIC) ALGORITHM WITH THE EPINIONS DATASET.

Number of sources	Number of destinations	Total number of paths found	Average path length	Median path length	Average path cost	Median path cost
100	100	1041	660.727	638.0	17604.642	16875.4
500	500	27309	726.949	727.0	19134.178	18425.9

TABLE III
RESULTS FOR THE *min negative path* ALGORITHM (ALGORITHM 6) WITH THE EPINIONS DATASET.

Number of sources	Number of destinations	Total number of paths found	Average path length	Median path length	Negatively influenced persons (average)	Negatively influenced persons (median)
100	100	8830	4.023	4.0	0.096	0.0
500	500	218499	4.097	4.0	0.057	0.0
10000	10000	78029370	4.646	5.0	0.123	0.0

shortest DFS algorithm. We note that some of these pairs may in fact be unreachable as a natural result of the graph structure, i.e., the source and the destination may not be connected. However, the same analysis shows that there exist only 1168 cases in which the destination is not reachable from the source with the Dijkstra-type algorithm. Hence, the number of paths discovered by the *shortest DFS* algorithm is only a tenth of the paths found by Algorithm 5. Similarly, for 500 sources and 500 destinations, the number of cases a destination is not reachable from the source with the *shortest DFS* algorithm is as large as 222650, whereas with the Dijkstra-type algorithm this number is 31460. Again, the number of paths found by the *shortest DFS* algorithm is a tenth of Algorithm 5. As importantly, we have observed that the *shortest DFS* algorithm can not terminate within a reasonable amount of computing time for 10000 nodes.

The results for the implementation of the *min negative path* algorithm, i.e., minimizing the total number of negatively influenced persons on the Epinions dataset, are given in Table III for 100, 500 and 10000 sources and destinations.

The time complexity of the *shortest DFS* algorithm is $O(|E|)$, as with a regular DFS. That is because instead of visiting each edge at most once, it can be traversed at most twice, once considering that the end node of the edge is on a positive path and once on a negative path. This yields the same asymptotic complexity. On the other hand, the modified Dijkstra's algorithm has complexity $O((|E|+|V|)\log|V|)$, which is identical to the regular Dijkstra's algorithm. That is because as with *shortest DFS*, each edge may be traversed twice, which leads to the same asymptotic complexity as the regular Dijkstra's algorithm. However, it is important to note that the *shortest DFS* algorithm does not yield an optimal solution, as confirmed by the experimental results. On the other hand, the modified Dijkstra's procedure does.

An important outcome of our evaluations is that, even for a very large number of source and destination pairs, at least half of the paths have *zero* negatively influenced nodes. In addition, the average number of hops in each path is less than 5. This justifies our intuition that, it is actually possible to find a relatively short path from one node to another purely dominated by friendship (homophily) relations. In effect, our findings show that in general any node can influence another node *positively* within a small number of hops.

X. CONCLUSION

We have studied a social network with positive and negative relationship types, in which friends and foes are characterized by positive and negative signs, respectively. We have proposed a propagation scheme to influence a target person in favor of an idea, an action, or a product. Depending on the underlying relationship structure, we presume that persons are influenced in their decisions by the observations made available to them. To this end, our propagation schemes apply to networks with socially aware sensors that can extract information about social and physical phenomena. We have studied the optimal propagation policies by integrating social awareness into network propagation under influence-centric constraints. We have implemented the proposed algorithms under an artificially created setup as well as the Epinions dataset in order to gain an understanding of the optimal policies in both small and large-scale networks. We have observed that the average network propagation cost can be reduced significantly compared to naïve myopic schemes, and that randomly chosen sources can positively influence randomly chosen destinations in over 87% of the cases.

In this paper, we have considered positive and negative relationships, and have assumed the knowledge of the underlying social graph and the corresponding polarities. Future directions include constructing a multilayer influence propagation scheme for signed networks, incorporating multi-level relationship types, multi-modal sensor observations, practical applications in modern social networks, degree of positivity and negativity through threshold based influence patterns, and developing inference methods for enhancing situational awareness at the target node.

REFERENCES

[1] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[2] D. Watts and S. Strogatz, "Collective dynamics of small-world networks," *Nature*, pp. 393–440, 1998.

[3] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proc. 13th Int. Conf. WWW*, New York, NY, USA, 2004, pp. 403–412.

[4] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2009, pp. 199–208.

[5] F. S. F. Walter and S. Battiston, "A model of a trust-based recommendation system on a social network," *Auton. Agents and Multi-Agent Syst.*, vol. 16, pp. 57–74, 2008.

[6] J. Kunegis, A. Lommatzsch, and C. Bauckhage, "The slashdot zoo: mining a social network with negative edges," in *Proc. 18th Int. Conf. WWW*, 2009, pp. 741–750.

[7] T. Hogg, D. M. Wilkinson, G. Szabo, and M. J. Brzozowski, "Multiple relationship types in online communities and social networks," in *Proc. AAAI Spring Symp.: Social Inf. Process.*, 2008, pp. 30–35.

[8] C. A. Lampe, E. Johnston, and P. Resnick, "Follow the reader: filtering comments on slashdot," in *Proc. ACM SIGCHI Conf. Human Factors in Comput. Syst.*, 2007, pp. 1253–1262.

[9] Y. Li, W. Chen, Y. Wang, and Z.-L. Zhang, "Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships," in *Proc. 6th ACM Int. Conf. Web Search and Data Mining*, 2013, pp. 657–666.

[10] P. Anchuri and M. Magdon-Ismail, "Communities and balance in signed networks: A spectral approach," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM'12)*, 2012, pp. 235–242.

[11] F. Heider, "Attitudes and cognitive organization," *J. . Psychol.*, vol. 21, no. 1, pp. 107–112, 1946.

[12] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory," *Psychol. l Rev.*, vol. 63, no. 5, p. 277, 1956.

[13] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proc. SIGCHI Conf. Human Factors in Comput. Syst.*, New York, NY, USA, 2010, pp. 1361–1370.

[14] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proc. 19th Int. Conf. WWW*, Raleigh, NC, USA, 2010, pp. 641–650.

[15] D. Kempe, J. Kleinberg, and É Tarodos, "Maximizing the spread of influence through a social network," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2003, pp. 137–146.

[16] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, 2006.

[17] D. O. Olguín and A. S. Pentland, "Social sensors for automatic data collection," *Proc. AMCIS*, no. 171, 2008.

[18] A. Pentland, "Socially aware, computation and communication," *IEEE Comput.*, vol. 38, no. 3, pp. 33–40, Mar. 2005.

[19] S.-H. Yang, A. J. Smola, B. Long, H. Zha, and Y. Chang, "Friend or frenemy?: predicting signed ties in social networks," in *Proc. 35th Int. ACM SIGIR Conf. Res. and Develop. Inf. Retrieval*, 2012, pp. 555–564.

[20] P. Agrawal, V. K. Garg, and R. Narayanam, "Link label prediction in signed social networks," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2591–2597.

[21] A. Papaoikonomou, M. Kardara, K. Tserpes, and D. Varvarigou, "Predicting edge signs in social networks using frequent subgraph discovery," *IEEE Internet Comput.*, vol. 18, no. 5, pp. 36–43, Sep.Oct. 2014.

[22] A. Tumasjan, T. Sprenger, P. Sandner, and I. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proc. 4th Int. AAAI Conf. Weblogs and Social Media*, 2010, pp. 178–185.

[23] T. Gan, "Social interaction detection using a multi-sensor approach," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 1043–1046.

[24] D. Namiot and M. Sneps-Sneppe, "Wireless networks sensors and social streams," in *Proc. 27th Int. Conf. Adv. Inf. Netw. Applicat. Workshops (WAINA)*, 2013, pp. 413–418.

[25] E. Stattner, M. Collard, P. Hunel, and N. Vidot, "Wireless sensor networks for social network data collection," in *Proc. IEEE 36th Conf. Local Comput. Netw. (LCN)*, 2011, pp. 867–874.

[26] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, 2008, pp. 337–350.

[27] B. Guler, B. Varan, K. Tutuncuoglu, M. Nafea, A. A. Zewail, A. Yener, and D. Octeau, "Optimal strategies for targeted influence in signed networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. and Mining—Workshop Curbing Collusive Cyber-Gossips Social Netw. (ASONAM'14)*, Beijing, China, 2014, pp. 906–911.

[28] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, pp. 415–444, 2001.

[29] M. J. Brzozowski, T. Hogg, and G. Szabo, "Friends and foes: ideological social networking," in *Proc. ACM SIGCHI Conf. Human Factors Comput. Syst.*, New York, NY, USA, 2008, pp. 817–820.

[30] B. Guler, B. Varan, K. Tutuncuoglu, M. Nafea, A. A. Zewail, A. Yener, and D. Octeau, "Communicating in a socially-aware network: Impact of relationship types," in *Proc. IEEE Global Conf. Signal and Inf. Process.—Symp. Netw. Theory (GlobalSIP'14)*, Atlanta, GA, USA, 2014, pp. 956–960.

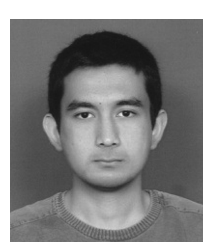
[31] W. L. Langer, *Eur. Alliances and Alignments*. Santa Barbara, CA, USA: Greenwood Press, 1977, pp. 1871–1890.

[32] B. E. Schmitt, "Triple alliance and triple entente," *Amer. Historical Rev. 1902-1914*, vol. 29, no. 3, pp. 449–473, 1924.

[33] S. E. Dreyfus and A. M. Law, *The Art and Theory of Dynamic Programming*. Amsterdam, The Netherlands: Elsevier Science (Mathematics in Science and Engineering), 1977.

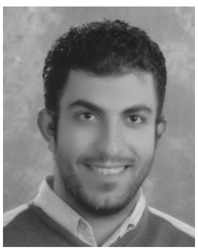
[34] S. Blackmore, *The Meme Machine*. Oxford, UK: Oxford Univ. Press, 2000.

Basak Guler (S'13) received her B.Sc. degree in electrical and electronics engineering from Middle East Technical University (METU), Ankara, Turkey, in 2009 and her M.Sc. degree in electrical engineering from the Wireless Communications and Networking Laboratory, Pennsylvania State University, University Park, PA, in 2012. She is currently pursuing the Ph.D. degree with the Wireless Communications and Networking Laboratory, Pennsylvania State University. Her research interests include semantic communication, source coding and data compression, semantic information theory, social networks, interactive communication, and interference management in heterogeneous wireless networks.



Burak Varan (S'13) received the B.S. degree in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, in 2011. Since 2011, he has been a Graduate Research Assistant with the Wireless Communications and Networking Laboratory at the Pennsylvania State University, University Park, PA, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include green communications and optimal resource allocation in energy harvesting networks under competitive and altruistic communication scenarios.

Kaya Tutuncuoglu (S'08) received his two B.S. degrees in electrical and electronics engineering and in physics from the Middle East Technical University, Ankara, Turkey, in 2008 and 2009, respectively. He is currently pursuing the Ph.D. degree and has been a Graduate Research Assistant with the Department of Electrical Engineering, Pennsylvania State University, University Park, PA, since 2009. His research interests include green communications, energy harvesting and resource allocation for wireless networks. He received the AT&T Graduate Fellowship at the Pennsylvania State University in 2012, and the IEEE Marconi Prize Paper Award in Wireless Communications in 2014.



Mohamed Nafea (S'11) received his B.S. degree in electronics and communications engineering from Alexandria University, Alexandria, Egypt, in 2010. He received his M.S. degree in wireless communications from Wireless Intelligent Networks Center (WINC), Nile University, Giza, Egypt, in 2012. He is currently pursuing his Ph.D. degree and has been a Graduate Research Assistant with the Department of Electrical Engineering, Pennsylvania State University, University Park, PA, since 2012. His research interests include physical layer security, network information theory, and wireless communication theory.

Ahmed A. Zewail (S'07) received his B.Sc. degree in electronics and communications engineering from the University of Alexandria, Alexandria, Egypt, in 2011, and his M.Sc. from Nile University in Wireless Communications, Giza, Egypt, in 2013. He is currently pursuing the Ph.D. degree with the Wireless Communications and Networking Laboratory (WCAN), Pennsylvania State University. His research interests include capacity of relay networks, and information theoretic security.

Aylin Yener (S'91–M'00–SM'13–F'14) received the B.Sc. degree in electrical and electronics engineering, and the B.Sc. degree in physics, from Bogazici University, Istanbul, Turkey; and the M.S. and Ph.D. degrees in electrical and computer engineering from Wireless Information Network Laboratory (WINLAB), Rutgers University, New Brunswick, NJ. Commencing fall 2000, for three semesters, she was a P.C. Rossin Assistant Professor at the Electrical Engineering and Computer Science Department, Lehigh University, PA. In 2002, she joined the faculty of The Pennsylvania State University, University Park, PA, where she was an Assistant Professor, then Associate Professor, and is currently Professor of Electrical Engineering since 2010. During the academic year 2008–2009, she was a Visiting Associate Professor with the Department of Electrical Engineering, Stanford University, CA. Her research interests are in information theory, communication theory and network science. Dr. Yener received the NSF CAREER award in 2003, the best paper award in Communication Theory in the IEEE International Conference on Communications in 2010, the Penn State Engineering Alumni Society (PSEAS) Outstanding Research Award in 2010, the IEEE Marconi Prize paper award in 2014, the PSEAS Premier Research Award in 2014, and the Leonard A. Doggett Award for Outstanding Writing in Electrical Engineering at Penn State in 2014. She is a fellow of the IEEE.

Dr. Yener previously served as a technical program chair or co-chair for various conferences for the IEEE Communications Society, as an associate editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, as an associate editor and an editorial advisory board member for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. She served as the student committee chair for the IEEE Information Theory Society 2007–2011, and was the co-founder of the Annual School of Information Theory in North America co-organizing the school in 2008, 2009, and 2010. Dr. Yener currently serves as a member on the board of governors of the IEEE Information Theory Society, where she was previously the treasurer (2012-2014).

Damien Octeau received his B.Sc. and Master's degrees from Ecole Centrale de Lyon, France, in 2007 and 2010, respectively. He got the M.Sc. and Ph.D. degrees in computer science and engineering from the Pennsylvania State University in 2010 and 2014, respectively. He is currently a Research Associate with a joint appointment at the Department of Computer Sciences at the University of Wisconsin-Madison and at the Department of Computer Science and Engineering at the Pennsylvania State University. His research interests include systems, mobile and software security and program analysis. He was awarded the Best Research Artifact Award at the 2012 International Symposium on the Foundations of Software Engineering (FSE). He also received the 2013 Penn State AT&T Graduate Fellowship.