

Shifting Opinions in a Social Network Through Leader Selection

Yuhao Yi , Timothy Castiglia, and Stacy Patterson , *Member, IEEE*

Abstract—We study the French–DeGroot opinion dynamics in a social network with two polarizing parties. We consider a network in which the leaders of one party are given, and we pose the problem of selecting the leader set of the opposing party so as to shift the average opinion to a desired value. When each party has only one leader, we express the average opinion in terms of the transition matrix and the stationary distribution of random walks in the network. The analysis shows the balance of influence between the two leader nodes. We show that the problem of selecting at most k absolute leaders to shift the average opinion is NP-hard. Then, we reduce the problem to a problem of submodular maximization with a submodular knapsack constraint and an additional cardinality constraint, and propose a greedy algorithm with upper bound search to approximate the optimum solution. We also conduct experiments in random networks and real-world networks to show the effectiveness of the algorithm.

Index Terms—Approximation algorithm, balance of opinions, French–DeGroot model, optimization, social network.

I. INTRODUCTION

SOCIAL networks have become increasingly influential in shaping public opinions. Within this field, the problem of designing mechanisms to effectively shift opinions in a social network has received great interest in the last two decades [1]–[6]. Much of the existing work studies the problem of choosing individuals in the network to be opinion leaders so as to maximize the influence of a particular opinion, i.e., to shift the average opinion of the network to an extreme opinion. However, fine-grained optimization of the average opinion has not been well studied.

In this article, we study the problem of shifting the average opinion of a network to a given value, which generalizes the intensely studied influence maximization problem. We consider a continuous-time French–DeGroot opinion model with two polarizing parties. The French–DeGroot model [7], [8] is one of the most popular models for opinion dynamics. In the model, the

social network is represented by a graph, with nodes corresponding to individuals. Each node has a real scalar-valued state that represents the individual’s opinion. Each node updates its state continuously by comparing its state with that of its neighbors. We consider a variation on this model where the nodes consists of leaders nodes, defined as the nodes with external reference values, and follower nodes, defined as those without external information.

We assume that there are two opposing sources of opinion, 1 and 0. These could represent, for example, support for Party A (1) or Party B (0), or these sources could represent support and opposition to an event or outcome. In the model we adopt, all members of the social network take opinion values in the interval $[0,1]$. A firm supporter of Party A (or proevent individual) has an opinion close to 1, and an unquestioned supporter of Party B (or antievent individual) holds an opinion close to 0. Individuals with opinion $1/2$ are considered as neutral.

Each party controls a set of nodes as their opinion leaders. We consider two types of opinion leaders: *absolute leaders* and *influenced leaders*. An absolute leader has a constant opinion (1 or 0) that represents its party which never changes. We call a system with such leaders, an *absolute leader system*. An influenced leader receives a constant input from the corresponding party as a reference value, and it adjusts its state according to the reference value and the states of its neighbors. One can think of these leaders as being influenced through a relationship with an individual that is a direct source of the opinion but is not a part of the network, i.e., an external party leader. We refer to a system with such leaders as an *influenced leader system*.

We consider two leader selection problems, one for each type of system. In both cases, we assume that the network already has a leader set for party 0, and our goal is to identify the leader set for party 1 so as to shift the average opinion of the social network toward a target value. In the *absolute leader system*, individuals are selected in the network to act as *absolute leaders*, by hiring them into the party. In the *influenced leader system*, relationships are formed between the identified set of influenced leaders within the network and an external party leader with opinion 1.

Our problem formulation is related to the well-studied problem of influence maximization [2]–[5], i.e., maximizing the average opinion of the network by choosing leaders for party 1, while the leaders for party 0 are fixed. Influence maximization corresponds to the case in our problem where the target value is 1. However, there are cases where maximizing the average opinion is not beneficial. It has been observed that people can form polarizing opinions through interactions in a social group,

Manuscript received May 15, 2020; revised May 20, 2020, November 4, 2020, and December 19, 2020; accepted January 18, 2021. Date of publication February 10, 2021; date of current version September 17, 2021. This work was supported by NSF under Grant CNS-1553340 and Grant CNS-1816307. Recommended by Associate Editor G. Como. (Corresponding author: Yuhao Yi.)

The authors are with the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: yuhaoyi@kth.se; castit@rpi.edu; sep@cs.rpi.edu).

Digital Object Identifier 10.1109/TCNS.2021.3058620

and the problem of depolarizing opinions has received interest in social psychology [9]. If a social group has formed an extreme opinion, one may seek to balance the network opinion around a target value of $\frac{1}{2}$ to encourage a balanced discussion on a particular topic. Moreover, the opinion of an individual relates to his or her behaviors [10], [11]. In particular, it can be viewed as the probability that a user adopts a behavior. From this perspective, party 1 can achieve a desired level of participation in a voluntary activity in a large network by shifting the average opinion to a certain target value.

We begin by analyzing the two proposed models, and we propose the concept of domination score to characterize the balance of influence between leaders of two parties. This analysis relates the models to properties of random walks in a network. We also identify the optimal solution to the leader selection problem for each model when a single leader is chosen for each party. Next, we study the general problem of choosing a leader set for party 1 with a given cardinality, when the leader set for party 0 is already identified. For absolute leader systems, we prove the NP-hardness of the problem. Further, we propose an algorithm for the leader selection problems with provable approximation guarantees. Our algorithm finds an appropriate upper bound for a greedy routine that approximately solves a submodular cost submodular knapsack (SCSK) problem with an additional cardinality constraint. We are not aware of any previous work on SCSK problems with cardinality constraints.

Related Work: In the last two decades, many works considered the French–DeGroot model with leaders accessing the same reference value [12]–[16]. In such systems, leader selection problems have been formulated for different objectives such as minimizing the convergence error [16] or minimizing the total deviation from the reference value of the system in the presence of additional noise on followers [13]–[15]. These combinatorial optimization problems are often intractable. For example, the leader selection problem proposed in [13] has been proven to be NP-hard [17]. Various approaches have been proposed to address these problems, including convex relaxation heuristics [14] and greedy algorithms [16], [18] with constant approximation ratios.

Another line of works considers leaders with different reference values, in particular two groups of leaders with polarizing opinions. In this case, the steady-state opinions of all nodes fall into the interval of leader states [19], [20]. In such systems, different leader selection problems have also been studied. The work [3] proposed a message passing algorithm to place a single absolute leader to maximize its influence in an undirected network. A recent paper [21] extended the results to bidirectional networks with possibly nonsymmetric weights. Grabisch *et al.* [22] studied a zero-sum game in which each party strategically chooses an influenced leader.

Friedkin [1] studied the k absolute leader selection problem with $k > 1$ and proved the submodularity of the average opinion as a function of the leader set. The article [2] studied the influence maximization problem in the Friedkin–Johnsen model, which is related to a French–DeGroot opinion network with absolute leaders in special cases but not equivalent, in general. This work proved the submodularity of the average opinion and the NP-hardness of the average opinion maximization problem. Mai

and Abed [5] studied the problem of maximizing influence of the leader set in a French–DeGroot model where leaders have specified stubbornness, and Hunter and Zaman [4] investigated a similar maximization problem. Both works proved the monotonicity and submodularity of the average opinion in a French–DeGroot opinion network with influenced leader dynamics. Typical greedy algorithms were applied to these problems due to submodularity of the objective functions. In contrast, our work studies the problem of shifting the average opinion of the network to any specified value. Our problem thus includes the influence maximization problem as a special case. In addition, we show that our problem cannot be directly treated as submodular maximization problem with a cardinality constraint. Thus, a more sophisticated optimization algorithm is needed.

Article Outline: The remainder of the article is organized as follows. In Section II, we introduce basic notations and concepts. In Section III, we present the system model and the problem formulation. In Section IV, we give an explicit form of the steady-state opinion vector using the Laplacian of an augmented graph, and we show how this relates to the balance of the leader nodes' influence in a network. In Section V, we propose a greedy algorithm with an upper bound search and provide provable bounds on the approximation ratio of the algorithm. Section VI presents experimental results. Finally, Section VII concludes this article.

II. PRELIMINARIES

In this section, we introduce the notation of a graph and its matrix representations. Further, we review the concepts of hitting time, commute time, resistance distance, and information centrality, which are used as analytical tools in this article.

Vectors and Matrices: We use e_u to denote the u th canonical basis vector of $\mathbb{R}^n$. Vector $\mathbf{b}_{u,v}$ is defined as $\mathbf{b}_{u,v} \stackrel{\text{def}}{=} e_u - e_v$. $\mathbf{1}_n$ represents the all-one vector with length n , and $\mathbf{0}_n$ ($\mathbf{0}_{p \times q}$) represents the all-zero vector (or matrix) with length n (or size $p \times q$). We also use these notations without specifying the sizes if they are implied in context. Apart from these exceptions (e_u , $\mathbf{b}_{u,v}$, $\mathbf{1}_n$, and $\mathbf{0}_n$), a vector or matrix with subscripts denotes the vector or submatrix with indices specified by the subscripts. For example, given a vector $\mathbf{x}$, x_i is its i th entry and $\mathbf{x}_{\mathcal{I}}$ is a vector consisting of entries x_i for all $i \in \mathcal{I}$. For a matrix $\mathbf{X}$, $X_{i,j}$ is the (i,j) th entry of $\mathbf{X}$ and $\mathbf{X}_{\mathcal{I},\mathcal{J}}$ is the submatrix of $\mathbf{X}$ consisting of the entries of $\mathbf{X}$ whose rows are in $\mathcal{I}$ and columns are in $\mathcal{J}$. In addition, we use $\mathbf{I}$ to denote the identity matrix, and we use $\mathbf{X}^\dagger$ to denote the Moore Penrose pseudoinverse of the matrix $\mathbf{X}$.

Graphs and Their Matrix Representations: We denote a directed graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, where $\mathcal{V}$ and $\mathcal{E}$ are the node set and edge set of the graph, respectively, with $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$. An undirected graph can be viewed as a symmetrically coupled bidirectional graph in this context. We let $e = (u, v) \in \mathcal{E}$ represent an edge from nodes u to node v , and $w : \mathcal{E} \rightarrow \mathbb{R}^+$ is the edge weight function. We denote $\mathcal{N}_v^\downarrow$ as the set of in-neighbors of v ($u \in \mathcal{N}_v^\downarrow$ iff $(u, v) \in \mathcal{E}$), and $\mathcal{N}_v^\uparrow$ as the set of out-neighbors of v ($u \in \mathcal{N}_v^\uparrow$ iff $(v, u) \in \mathcal{E}$). In addition, for a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, and a subset of nodes $V \subseteq \mathcal{V}$, we denote

the subgraph supported on V as $\mathcal{G}[V] = (V, E, \omega)$, where $E = \{e = (u, v) \in \mathcal{E} : u, v \in V\}$ and $\omega(e) = w(e)$ for all $e \in E$. Further, we define the plus operation on graphs as follows. For two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, w_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, w_2)$, let $\mathcal{H} = (\mathcal{U}, \mathcal{M}, \omega) = \mathcal{G}_1 + \mathcal{G}_2$ be a new graph with $\mathcal{U} = \mathcal{V}_1 \cup \mathcal{V}_2$, $\mathcal{M} = \mathcal{E}_1 \cup \mathcal{E}_2$, and $\omega : \mathcal{M} \rightarrow \mathbb{R}^+$ the new edge weight function defined as $\omega(e) = w_1(e)$ if $e \in (\mathcal{E}_1 \setminus \mathcal{E}_2)$, $\omega(e) = w_2(e)$ if $e \in (\mathcal{E}_2 \setminus \mathcal{E}_1)$, and $\omega(e) = w_1(e) + w_2(e)$ if $e \in (\mathcal{E}_2 \cap \mathcal{E}_1)$.

The weighted Laplacian matrix of a graph is defined as $\mathbf{L} \stackrel{\text{def}}{=} \mathbf{D} - \mathbf{A}$, where $\mathbf{A}$ is the adjacency matrix with $A_{u,v} = w(e)$ for $e = (u, v) \in \mathcal{E}$ and $A_{u,v} = 0$ for $(u, v) \notin \mathcal{E}$, and $\mathbf{D}$ is the out-degree diagonal matrix, where $D_{u,u} = \sum_v A_{u,v}$ and $D_{u,v} = 0$ if $u \neq v$. From the definition, it is clear that $\mathbf{L} = \sum_{(u,v) \in \mathcal{E}} w(u, v) \mathbf{b}_{u,v} \mathbf{e}_u^\top$.

For a matrix (vector, scalar) associated with a graph, we sometimes use a superscript to explicitly show that it corresponds to the graph. For example, $\mathbf{L}^{\mathcal{G}}$ is the Laplacian matrix of graph $\mathcal{G}$.

Random Walks on Graphs: We define $\mathbf{W} \stackrel{\text{def}}{=} \mathbf{A}^\top \mathbf{D}^{-1}$ as the random walk transition matrix of graph $\mathcal{G}$. A random walker has a probability $W_{i,j} = \frac{A_{j,i}}{D_{j,j}}$ to transition from vertex j to vertex i . When the graph is strongly connected, there exists a positive vector (unique up to scaling) such that $\boldsymbol{\pi} = \mathbf{W} \boldsymbol{\pi}$. When the vector is scaled such that $\sum_v \pi_v = 1$, $\boldsymbol{\pi}$ is called the *stationary distribution* of the random walk defined by $\mathbf{W}$. We define $\boldsymbol{\Pi}$ as a diagonal matrix in which $\Pi_{v,v} \stackrel{\text{def}}{=} \pi_v$ for all vertex $v \in \mathcal{V}$. We note that $\mathbf{L} = \mathbf{D}(\mathbf{I} - \mathbf{W}^\top)$ and $\mathbf{L}\mathbf{1} = \mathbf{0}$.

In a connected graph $\mathcal{G}$, the *hitting time* from vertex u to v is the expected number of steps that a random walker, starting from vertex v , takes until it hits u for the first time. We denote by $H_{u,v}$ the hitting time from u to v .

Lemma II.1 (Hitting Time [23]): In a strongly connected graph $\mathcal{G}$

$$H_{u,v}^{\mathcal{G}} = \mathbf{b}_{u,v}^\top (\mathbf{I} - (\mathbf{W})^\top)^\dagger \boldsymbol{\Pi}^{-1} (\boldsymbol{\pi} - \mathbf{e}_v).$$

If $\mathcal{G}$ is an undirected graph, $H_{u,v}^{\mathcal{G}} = 2m \cdot \mathbf{b}_{u,v}^\top \mathbf{L}^\dagger (\boldsymbol{\pi} - \mathbf{e}_v)$.

The *commute time* $C_{u,v}$ is defined as $C_{u,v} \stackrel{\text{def}}{=} H_{u,v} + H_{v,u}$.

Lemma II.2 (Commute Time [23]): In a strongly connected graph $\mathcal{G}$

$$C_{u,v}^{\mathcal{G}} = \mathbf{b}_{u,v}^\top (\mathbf{I} - (\mathbf{W})^\top)^\dagger \boldsymbol{\Pi}^{-1} \mathbf{b}_{u,v}.$$

If $\mathcal{G}$ is an undirected graph, $C_{u,v}^{\mathcal{G}} = 2m \cdot \mathbf{b}_{u,v}^\top \mathbf{L}^\dagger \mathbf{b}_{u,v}$.

Effective Resistance and Information Centrality: Given an undirected graph $\mathcal{G}$, we define an *electrical network* $\bar{\mathcal{G}}$. In $\bar{\mathcal{G}}$, every edge e of $\mathcal{G}$ is replaced by a resistor of resistance $1/w(e)$, and the resistors are connected if the edges are incident. Then, the *effective resistance* between node u and v in graph $\mathcal{G}$ (or electrical graph $\bar{\mathcal{G}}$) is defined as the voltage difference between vertices u and v in $\bar{\mathcal{G}}$ when unit current is injected from u and extracted from v . We recall the following lemma relating to effective resistance.

Lemma II.3 (Effective Resistance [24]): In a connected undirected electrical network defined by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, the effective resistance between nodes u and v is

$$R_{u,v}^{\mathcal{G}} = (\mathbf{L}^\dagger)_{v,v} - 2(\mathbf{L}^\dagger)_{v,u} + (\mathbf{L}^\dagger)_{u,u}.$$

Effective resistance is a global distance metric defined in undirected networks for node pairs. $R_{u,v}^{\mathcal{G}} > 0$ if there is a path between u and v . We further recall the definition of information centrality.

Definition II.4 (Information Centrality): In a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, the information centrality of a vertex u is defined by

$$\theta^{\mathcal{G}}(u) = \frac{n}{\sum_{v \in \mathcal{V}} R_{u,v}^{\mathcal{G}}}.$$

From Lemma II.3, we obtain

$$\sum_{u \in \mathcal{V}} R_{u,v}^{\mathcal{G}} = n \cdot (\mathbf{L}^\dagger)_{v,v} + \text{Tr}(\mathbf{L}^\dagger).$$

Information centrality makes use of all paths between a node and other nodes in a social network and give them relative weighting [25].

III. PROBLEM FORMULATION

We consider a directed strongly connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. Nodes represent individuals in the social network, and an edge $(u, v) \in \mathcal{E}$ models a social link from node u to node v , indicating that node u follows node v , or node v exerts influence on node u . Edge weights represent the strengths of the social links. Each node v has a scalar-valued state $\mathbf{x}_v \in \mathbb{R}$, which represents its opinion. The node set can be divided into a leader set S and a follower set F . The set S can be further divided into two disjoint sets S_0 and S_1 , which are sets of nodes controlled by two parties, namely party 0 and party 1. All nodes in S_0 have access to reference value 0, and all nodes in S_1 have access to reference value 1. Nodes in F update their states according to a diffusion law.

A. System Dynamics

We consider the French–DeGroot opinion model with absolute leaders and a variation of this model with influenced leaders that are connected to external absolute sources of information. The two models differ in how the leaders use their reference values.

In the absolute leader system, leaders initialize their states with 0 (for $v \in S_0$) or 1 (for $v \in S_1$), and their states remain unchanged over time. The dynamics of a leader node v is characterized by $\dot{\mathbf{x}}_v(t) = 0$. A follower node v begins with an arbitrary initial state $\mathbf{x}_v(0) = \mathbf{x}_v^0$, and it updates its state by the dynamics

$$\dot{\mathbf{x}}_v(t) = - \sum_{u \in \mathcal{N}_v^\dagger} w(v, u) (\mathbf{x}_v(t) - \mathbf{x}_u(t)).$$

We partition the state vector $\mathbf{x}$ as

$$\mathbf{x} = (\mathbf{x}_S^\top \mathbf{x}_F^\top)^\top$$

where $\mathbf{x}_S$ is associated with the leaders and $\mathbf{x}_F$ is associated with the followers. Similarly, we partition the Laplacian matrix

L^G and adjacency matrix A into blocks as

$$L^G = \begin{pmatrix} L_{S,S} & L_{S,F} \\ L_{F,S} & L_{F,F} \end{pmatrix}.$$

Then, the dynamics of the leaders and the followers can be written as

$$\dot{x}_S(t) = 0 \quad (1)$$

$$\dot{x}_F(t) = -L_{F,F}x_F - L_{F,S}x_S. \quad (2)$$

In the system described by (1) and (2), the steady-state values of the leader nodes are

$$\hat{x}_S = x_S^0 \quad (3)$$

for $v \in S$. Since $-L_{F,F}$ is Hurwitz for a nonempty leader set S [26], the system converges to a single stable steady state [27]. Letting $\dot{x}_F(t) = 0$, we obtain the steady state of the followers

$$\hat{x}_F = -(L_{F,F})^{-1}L_{F,S}\hat{x}_S. \quad (4)$$

We note that $L_{F,S}\hat{x}_S$ can be viewed as the sum of columns of $L_{F,S}$ that correspond to 1-leaders (columns of 0-leaders are weighted by 0).

In the influenced leader system, disjoint subsets of nodes S_0 and S_1 are influenced by two external party leaders with opinions 0 and 1, respectively. These external nodes are not part of the graph $\mathcal{G}$, and further, they do not change their opinions. Each of the influenced leaders in $S_0 \cup S_1$ updates its state according to its current state, the states of its neighbors, and the reference value from its external leader, 0 for nodes in S_0 and 1 for nodes in S_1 .

The system can start from any initial state and the dynamics is given by

$$\dot{x}_v = -\sum_{u \in \mathcal{N}_v^\uparrow} w(v, u)(x_v(t) - x_u(t)) + \kappa_v(0 - x_v(t)), v \in S_0$$

$$\dot{x}_v = -\sum_{u \in \mathcal{N}_v^\uparrow} w(v, u)(x_v(t) - x_u(t)) + \kappa_v(1 - x_v(t)), v \in S_1$$

$$\dot{x}_v = -\sum_{u \in \mathcal{N}_v^\uparrow} w(v, u)(x_v(t) - x_u(t)), v \in F$$

where the value κ_v is the weight that the influenced leader puts on its reference value. We also refer to it as the *stubbornness* of the node. The dynamics can be expressed more compactly as

$$\dot{x} = -(L^G + E^S K)x + E^{S_1} K \mathbf{1} \quad (5)$$

where E^S is the diagonal matrix with $E_{v,v}^S = 1$ for $v \in S$ and $E_{v,u}^S = 0$ otherwise; E^{S_1} is defined similarly with nonzero entries for $v \in S_1$. The matrix K is diagonal with $K_{v,v} = \kappa_v$, the stubbornness of vertex v if chosen as an influenced node.

For system (5), $-(L^G + E^S K)$ is Hurwitz for a nonempty leader set S , so the system converges to a single steady state. We let $\dot{x}(t) = 0$ and obtain the steady-state values of all nodes

$$\hat{x} = (L^G + E^S K)^{-1} E^{S_1} K \mathbf{1}. \quad (6)$$

In this article, we study the average opinion of all nodes in the network.

Definition III.1: In both the absolute and influenced leader systems, given the leader set S_0 , the *average opinion* μ of a network as a function of leader set S_1 is defined as

$$\mu(S_1) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{v \in \mathcal{V}} \hat{x}_v. \quad (7)$$

Besides the above definition, $\mu(S_1)$ has an interesting interpretation in an opinion-behavior model based on the French-DeGroot model. We can model the opinion-behavior linkage in the system by treating $\hat{x}_v$ as the success probability of a Bernoulli random variable X_v of taking the value 1. In the social network, $X_v = 1$ indicates the event that node (individual) v takes an action, and $X_v = 0$ indicates the event that v does not take an action. We recall that $n = |\mathcal{V}|$ for both the absolute and influenced leader systems. We then assume that after reaching a steady state of opinions, nodes in the network choose their actions independently. Then, $X_1, X_2, \dots, X_v, \dots, X_n$ are treated as n mutually independent Bernoulli random variables associated with corresponding nodes in the network. In particular, X_v is defined by

$$\Pr(X_v = 1) = \hat{x}_v$$

$$\Pr(X_v = 0) = 1 - \hat{x}_v$$

for all $v \in \mathcal{V}$, and therefore $E[X_v] = \hat{x}_v$.

We are interested in the fraction of nodes that take an action. We define the random variable $\bar{X} := \frac{1}{n} \sum_v X_v$. Since X_v are independent bounded random variables, $\bar{X}$ concentrates at

$$\mu(S_1) = \frac{1}{n} \sum_v \hat{x}_v. \quad (8)$$

According to Hoeffding's inequality,

$$\Pr\left(|\bar{X} - \mu(S_1)| \geq \sqrt{\frac{\ln n}{n}}\right) \leq \frac{2}{n^2} \quad (9)$$

which indicates that $\mu(S_1)$ determines the fraction of the population that take an action in a large network, with a diminishing error bound and a diminishing probability that this bound is violated. Therefore, a party can control the fraction of population that take part in an activity or event by shifting the average opinion of the network to a certain value.

B. Leader Selection Problems

In a system where set S_0 is given, we define the problem of choosing at most k leaders for set S_1 , such that the average opinion of all nodes (including leaders and followers) $\mu(S_1)$ is closest to a given value α . Specifically, we are interested in minimizing the following objective function:

$$f(P, \alpha) \stackrel{\text{def}}{=} |\mu(P) - \alpha|. \quad (10)$$

We first formally define the problem for the absolute leader system.

Problem 1 (Absolute Leader Selection): In an absolute leader system, given a strongly connected directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, an opinion 0 absolute leader set $S_0 \neq \emptyset$, a specified

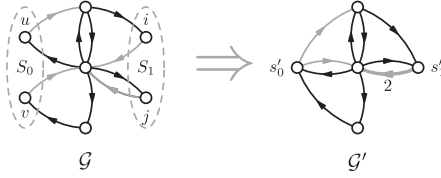


Fig. 1. Example of constructing a leader-equivalent graph for an absolute leader system. Nodes u and v in $\mathcal{G}$ become the leader s'_0 in $\mathcal{G}'$, and nodes i and j in $\mathcal{G}$ become the leader s'_1 in $\mathcal{G}'$. Edges without labels are weighted 1; otherwise, edges are labeled with their weights.

value $\alpha \in [0, 1]$, a candidate set $Q \subseteq \mathcal{V} \setminus S_0$, $|Q| = q$, and an integer $1 \leq k \leq q$, find the node set $S_1 \subseteq Q$, $|S_1| \leq k$ such that

$$S_1 \in \arg \min_{P \subseteq Q, |P| \leq k} f(P, \alpha). \quad (11)$$

We define a similar problem for the influenced leader system.

Problem 2 (Influenced Leader Selection): In an influenced leader system, given a strongly connected directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, an opinion 0 leader set $S_0 \neq \emptyset$, a stubbornness function of 0 leader nodes $\kappa_0 : S_0 \rightarrow \mathbb{R}^+$, a specified value $\alpha \in [0, 1]$, a candidate set $Q \subseteq \mathcal{V} \setminus S_0$, $|Q| = q$, another stubbornness function $\kappa_1 : Q \rightarrow \mathbb{R}^+$, and an integer $1 \leq k \leq q$, find the node set $S_1 \subseteq Q$, $|S_1| \leq k$ such that

$$S_1 \in \arg \min_{P \subseteq Q, |P| \leq k} f(P, \alpha). \quad (12)$$

We note that for both Problems 1 and 2, influence maximization corresponds to the degenerate case of $\alpha = 1$.

IV. ANALYSIS

In this section, we give analytical solutions for Problems 1 and 2 for the case where $k = 1$. We also present hardness results for the case where $k > 1$.

Our analysis utilizes a leader-equivalent graph to give analytical expressions for the average opinion of the network. Furthermore, for a network with a single leader for each party, we express the average opinion using the transition matrix and the stationary distribution of random walks in the network.

A. Opinions in Leader-Equivalent Systems

We note that the dynamics of both the absolute leader system and the influenced leader system can be fully characterized by a system defined in a *leader-equivalent graph*. For these two different kinds of systems, we construct the corresponding leader-equivalent graphs in different ways.

The system described by (1) and (2) is equivalent to a system in which all nodes in S_0 are identified as a single absolute leader s'_0 , and all nodes in S_1 are identified as a single absolute leader node s'_1 . We denote the contracted graph by $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', w')$, where $\mathcal{V}' = F \cup \{s'_0\} \cup \{s'_1\}$, $\mathcal{E}' = \{(u, v) : u, v \in F\} \cup \{(u, s'_0) : (\mathcal{N}_u \cap S_0) \neq \emptyset\} \cup \{(u, s'_1) : (\mathcal{N}_u \cap S_1) \neq \emptyset\}$, and $w'(u, v) = w(u, v)$ if $u, v \in F$, $w'(u, s'_0) = \sum_{v \in (S_0 \cap \mathcal{N}_u)} w(u, v)$, and $w'(u, s'_1) = \sum_{v \in (S_1 \cap \mathcal{N}_u)} w(u, v)$. In addition, we define $S' = \{s'_0, s'_1\}$ and $F' = \mathcal{V}' \setminus S'$. Note that $F' = F$ in this case. Fig. 1 shows an example of constructing a leader-equivalent graph for an absolute leader system.

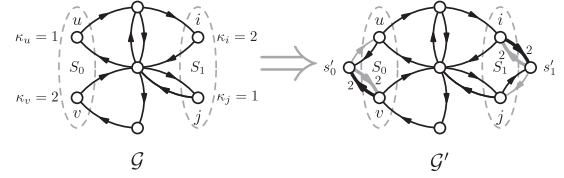


Fig. 2. Example of constructing a leader-equivalent graph from an influenced leader system.

We denote the Laplacian matrix of $\mathcal{G}'$ as $\mathbf{L}^{\mathcal{G}'}$. Then, the dynamics of F' in the system defined on the leader-equivalent graph is expressed by

$$\dot{\mathbf{x}}_{F'}(t) = -\mathbf{L}^{\mathcal{G}'}_{F', F'} \mathbf{x}_{F'} - \mathbf{L}^{\mathcal{G}'}_{F', \{s'_1\}}. \quad (13)$$

The influenced leader system described by (5) is equivalent to a system in which two virtual absolute leaders s'_0 and s'_1 are added to the graph, and all nodes in the original network $\mathcal{G}$ are treated as followers. We define the augmented graph as $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', w')$, where $\mathcal{V}' = \mathcal{V} \cup \{s'_0\} \cup \{s'_1\}$, and $\mathcal{E}' = \mathcal{E} \cup \{(u, s'_0) : u \in S_0\} \cup \{(u, s'_1) : u \in S_1\} \cup \{(s'_0, u) : u \in S_0\} \cup \{(s'_1, u) : u \in S_1\}$, $w'(u, v) = w(u, v)$ if $(u, v) \in \mathcal{E}$, $w'(u, s'_0) = w'(s'_0, u) = \kappa_u$ if $u \in S_0$, and $w'(u, s'_1) = w'(s'_1, u) = \kappa_u$ if $u \in S_1$. We again define $S' = \{s'_0, s'_1\}$ and $F' = \mathcal{V}' \setminus S'$; in this case, $F' = \mathcal{V}$. Fig. 2 shows an example of constructing a leader-equivalent graph for an influenced leader system. With this augmented graph, the dynamics of the influenced leader system is also described by (13).

By constructing the corresponding leader-equivalent graphs, we can study both absolute and influenced leader systems using a unified framework. We remark that this does not mean the systems are equivalent. Choosing leaders in different system models leads to different leader-equivalent graphs and hence different steady states, although system (5) approaches system (2) as $\mathbf{K}_{v,v} \rightarrow +\infty$ for all $v \in (S_0 \cup S_1)$.

For both the absolute and influenced leader systems, the nodes s'_1 and s'_0 are the only nodes that directly use reference values as their states in the leader-equivalent graph. Their steady states are $\mathbf{x}_{s'_0} = 0$ and $\mathbf{x}_{s'_1} = 1$. The steady states of all remaining nodes satisfy

$$\mathbf{L}^{\mathcal{G}'}_{F', F'} \hat{\mathbf{x}}_{F'} + \mathbf{L}^{\mathcal{G}'}_{F', \{s'_1\}} = \mathbf{0}. \quad (14)$$

We note that the edges from s'_0 or s'_1 to other nodes are not used according to the dynamics. We deliberately add these edges to make the graph strongly connected, which facilitates our analysis.

Let $(\mathbf{A}^{\mathcal{G}'})^\top (\mathbf{D}^{\mathcal{G}'})^{-1}$ be the random walk matrix of a leader-equivalent graph $\mathcal{G}'$. Then, we define the following matrices for $\mathcal{G}'$:

$$\mathcal{L}^{\mathcal{G}'} \stackrel{\text{def}}{=} \mathbf{\Pi} (\mathbf{I} - (\mathbf{W}^{\mathcal{G}'})^\top) \quad (15)$$

$$\mathcal{R}^{\mathcal{G}'} \stackrel{\text{def}}{=} (\mathbf{I} - (\mathbf{W}^{\mathcal{G}'})^\top)^\dagger \mathbf{\Pi}^{-1}. \quad (16)$$

In general, $(\mathcal{L}^{\mathcal{G}'})^\dagger \neq \mathcal{R}^{\mathcal{G}'}$, but for any $\mathbf{p} \perp \mathbf{1}$, $\mathbf{q} \perp \mathbf{1}$, $\mathbf{p}^\top (\mathcal{L}^{\mathcal{G}'})^\dagger \mathbf{q} = \mathbf{p}^\top \mathcal{R}^{\mathcal{G}'} \mathbf{q}$. For more details, we refer the readers to [23]. For an undirected graph, $\mathcal{L}^{\mathcal{G}'} = \frac{1}{2-m} \mathbf{L}^{\mathcal{G}'}$.

Proposition IV.1: For either an absolute leader system or an influenced leader system, we consider its leader-equivalent graph $\mathcal{G}'$. For any node $v \in \mathcal{V}'$, the steady state value $\hat{x}_v$ is given by

$$\hat{x}_v = \frac{\mathbf{b}_{v,s'_0}^\top \mathcal{R}^{\mathcal{G}'} \mathbf{b}_{s'_1,s'_0}}{\mathbf{b}_{s'_1,s'_0}^\top \mathcal{R}^{\mathcal{G}'} \mathbf{b}_{s'_1,s'_0}} = \frac{\mathbf{b}_{v,s'_0}^\top (\mathcal{L}^{\mathcal{G}'})^\dagger \mathbf{b}_{s'_1,s'_0}}{\mathbf{b}_{s'_1,s'_0}^\top (\mathcal{L}^{\mathcal{G}'})^\dagger \mathbf{b}_{s'_1,s'_0}}. \quad (17)$$

When $\mathcal{G}$ is an undirected graph, the expression degenerates to

$$\hat{x}_v = \frac{\mathbf{b}_{v,s'_0}^\top (\mathcal{L}^{\mathcal{G}'})^\dagger \mathbf{b}_{s'_1,s'_0}}{\mathbf{b}_{s'_1,s'_0}^\top (\mathcal{L}^{\mathcal{G}'})^\dagger \mathbf{b}_{s'_1,s'_0}}. \quad (18)$$

The construction of solution (17) is given in Appendix B. Since $\mathbf{L}$ and $\mathbf{L}^T$ have different kernels, the construction of the solution is different to the undirected case. The uniqueness of the solution is guaranteed by the fact that $\mathbf{L}_{F',F'}^{\mathcal{G}'}$ is full rank and $\mathbf{E}^{S_1} \mathbf{K}' \mathbf{1}$ is nonzero.

The value of $\hat{x}_v$ is, in fact, the *escape probability* of node v , which is defined as the probability that a random walker starting from vertex v , reaches node s'_1 before it reaches node s'_0 . We note that expression (18) was given in [20] in a different context. The paper [20] studied an opinion dynamics model where the sum of differences between the states of a node and its neighbors is divided by the out-degree of the node before it is applied as a negative feedback to the state of the node. If the leaders take the same values, the system studied in [20] has a different convergence rate than the absolute leader system but shares the same steady-state values.

B. Single Leader for Each Party

For absolute leader systems, if $|S_1| = |S_0| = 1$, the leader-equivalent graph $\mathcal{G}'$ is the same as the original graph $\mathcal{G}$. We let the leaders in $\mathcal{G}$ be denoted s_0 and s_1 for parties with opinion 0 and 1, respectively. Then, by Proposition IV.1,

$$\mu(S_1) = \frac{(\mathcal{L}^\dagger)_{s_0,s_0} - (\mathcal{L}^\dagger)_{s_0,s_1}}{(\mathcal{L}^\dagger)_{s_0,s_0} - (\mathcal{L}^\dagger)_{s_0,s_1} + (\mathcal{L}^\dagger)_{s_1,s_1} - (\mathcal{L}^\dagger)_{s_1,s_0}}. \quad (19)$$

Intuitively, we can view this expression as the influence of node s_1 to node s_0 , normalized by the sum of their mutual influence. We quantify this influence with the following definition.

Definition IV.2: In a strongly connected directed graph $\mathcal{G}$, the *domination score* of node u over v is defined as

$$D_{u,v}^{\mathcal{G}} = (\mathcal{L}^\dagger)_{v,v} - (\mathcal{L}^\dagger)_{v,u}. \quad (20)$$

We provide two physical interpretations for $D_{u,v}^{\mathcal{G}}$ in special cases. The first interpretation is that in a balanced regular (directed or undirected) graph, $D_{u,v}^{\mathcal{G}}$ is the hitting time $H_{u,v}^{\mathcal{G}}$. A larger $H_{u,v}^{\mathcal{G}}$ indicates that a random walker, starting from node u , spends more time in the network before it reaches v , therefore, exerting greater influence in the network. The second interpretation is that in an undirected graph $\mathcal{G}$ and its induced electrical network $\bar{\mathcal{G}}$, $\frac{1}{2-m} D_{u,v}^{\mathcal{G}}$ is the average voltage value of all nodes in the electrical network when unit current is injected at u and extracted from v , and v is grounded (s_0 has voltage 0).

From the definition of domination score and the expression of commute time in Lemma II.2, we immediately obtain

$$\mu(S_1) = \frac{D_{s_1,s_0}^{\mathcal{G}}}{C_{s_0,s_1}^{\mathcal{G}}} = \frac{D_{s_1,s_0}^{\mathcal{G}}}{D_{s_0,s_1}^{\mathcal{G}} + D_{s_1,s_0}^{\mathcal{G}}}. \quad (21)$$

As for the deviation of the average opinion from the given value α , we give its expression the following theorem.

Theorem IV.3: For absolute leader systems, if $|S_0| = |S_1| = 1$,

$$f(S_1, \alpha) = \frac{|(1 - \alpha)D_{s_1,s_0}^{\mathcal{G}} - \alpha D_{s_0,s_1}^{\mathcal{G}}|}{D_{s_0,s_1}^{\mathcal{G}} + D_{s_1,s_0}^{\mathcal{G}}}. \quad (22)$$

The proof of Theorem IV.3 follows directly from (19) and Definition IV.2. The numerator is the absolute value of a weighted average of $D_{s_1,s_0}^{\mathcal{G}}$ and $-D_{s_0,s_1}^{\mathcal{G}}$. Therefore, Theorem IV.3 shows a weighted balance between the domination score of s_0 over s_1 and the domination score of s_1 over s_0 , which decides the deviation of the average opinion from α . Theorem IV.3 indicates that for Problem 1, if $|S_0| = |S_1| = 1$, given the leader s_0 , it suffices to find a node s_1 such that $(1 - \alpha)D_{s_1,s_0}^{\mathcal{G}} = \alpha D_{s_0,s_1}^{\mathcal{G}}$ to shift the average opinion to α .

For influenced leader systems, the vector $\hat{x}$ is given by (6). We do not apply the leader-equivalent graph analysis in this case because $\mathcal{G}' \neq \mathcal{G}$. We instead interpret $\hat{x}$ using properties of $\mathcal{G}$. Fortunately, when we choose one leader for each party, $\mathbf{E}^{S_1}$ is a rank-1 matrix, and $\mathbf{E}^S = \mathbf{E}^{S_0} + \mathbf{E}^{S_1}$ is a rank-2 matrix. Applying the rank-1 update of matrices twice leads to the following theorem.

Theorem IV.4: For influenced leader systems, if $|S_0| = |S_1| = 1$, we obtain

$$f(S_1, \alpha) = \frac{|(1 - \alpha)(\frac{\mathbf{d}_{s_0}}{\kappa_0 \pi_{s_0}} + D_{s_1,s_0}^{\mathcal{G}}) - \alpha(\frac{\mathbf{d}_{s_1}}{\kappa_1 \pi_{s_1}} + D_{s_0,s_1}^{\mathcal{G}})|}{(\frac{\mathbf{d}_{s_0}}{\kappa_0 \pi_{s_0}} + D_{s_1,s_0}^{\mathcal{G}}) + (\frac{\mathbf{d}_{s_1}}{\kappa_1 \pi_{s_1}} + D_{s_0,s_1}^{\mathcal{G}})}$$

where the entries of vector $\mathbf{d}$ are defined as $\mathbf{d}_v = \mathbf{D}_{v,v}$, $\forall v \in \mathcal{V}$.

We defer the proof of Theorem IV.4 to Appendix C. As observed in Theorem IV.3 for absolute leader systems, for influenced leader systems, Theorem IV.4 also shows the balancing behavior of domination scores in the social network, which decides the deviation of the average opinion from α . In addition, Theorem IV.4 indicates that for Problem 2, if $|S_0| = |S_1| = 1$, given the leader s_0 , it suffices to find a node s_1 such that $(1 - \alpha)(\frac{\mathbf{d}_{s_0}}{\kappa_{s_0} \pi_{s_0}} + D_{s_1,s_0}^{\mathcal{G}}) = \alpha(\frac{\mathbf{d}_{s_1}}{\kappa_{s_1} \pi_{s_1}} + D_{s_0,s_1}^{\mathcal{G}})$ to shift the average opinion to α . Assuming $\kappa_1 = \kappa_2$, and they both approach infinity, then the condition is the same as what we have derived in the absolute leader system.

The balancing behaviors shown in Theorems IV.3 and IV.4 exhibit interesting results when $\mathcal{G}$ is undirected and $\alpha = 1/2$. In particular, Theorems IV.3 and IV.4 imply the following corollaries.

Corollary IV.5: For absolute leader systems, when $\mathcal{G}$ is undirected, $\alpha = 1/2$, and $|S_0| = |S_1| = 1$,

$$f(S_1, 1/2) = \frac{|\theta^{\mathcal{G}}(s_0)^{-1} - \theta^{\mathcal{G}}(s_1)^{-1}|}{2R_{s_0,s_1}^{\mathcal{G}}}. \quad (23)$$

Corollary IV.6: For influenced leader systems, when $\mathcal{G}$ is undirected, $\alpha = 1/2$, and $|S_0| = |S_1| = 1$,

$$f(S_1, 1/2) = \frac{|\theta^{\mathcal{G}}(s_0)^{-1} + 1/\kappa_0 - \theta^{\mathcal{G}}(s_1)^{-1} - 1/\kappa_1|}{2(R_{s_0, s_1}^{\mathcal{G}} + 1/\kappa_0 + 1/\kappa_1)}. \quad (24)$$

These corollaries show the role of information centrality of leader nodes in an undirected network when the objective is to balance the opinions in the network. If s_1 has the same information centrality as s_0 (assuming $\kappa_1 = \kappa_0$ for influenced leader systems), then $\mu(S_1) = \frac{1}{2}$, and so the opinion network is balanced. If there is no such s_1 , then it is beneficial to find a node s_1 such that $|\theta^{\mathcal{G}}(s_1) - \theta^{\mathcal{G}}(s_0)|$ is small while $R_{s_0, s_1}^{\mathcal{G}}$ is relatively large.

C. Hardness of Choosing Optimal k Leaders

Next, we show that Problem 1 is NP-hard. The hardness of Problem 2 remains an open question.

Theorem IV.7: The Absolute Leader Selection problem for shifting social opinion, described in Problem 1, is NP-hard.

The proof of Theorem IV.7 is given in a technical report [29]. We note that in both Problems 1 and 2, $\mu(S_1)$, as a function of S_1 , is monotone and submodular.

Theorem IV.8: For both absolute and influenced leader systems, the set function $\mu(S_1)$ is monotone and submodular.

The monotonicity and submodularity of $\mu(S_1)$ for absolute leader systems is shown in [1]. The same properties for influenced leader systems follow in a straightforward manner from results in [4], [5].

V. ALGORITHM

In this section, we present an algorithm for selecting a set of nodes to be leaders in S_1 , given set of leaders S_0 , to shift the average opinion as close as possible to a given value α .

A. Algorithm Intuition

It is well known that greedy algorithms give a $(1 - 1/e)$ approximation for monotone submodular maximization problems with cardinality constraints [30]. According to Theorem IV.8, for either Problem 1 or 2, a greedy algorithm provides a $(1 - 1/e)$ approximation for the problem when $\alpha = 1$. However, for other values of α , the problems are not trivial to solve. We observe that if $\mu(S_1) \leq \alpha$ always holds, we have a submodular maximization problem with cardinality constraint; if $\mu(S_1) \geq \alpha$ always holds, the problem is a submodular minimization problem with the same cardinality constraint. However, we do not know the value of $\mu(S_1)$ beforehand. Therefore, we need to design a more sophisticated algorithm to approximately solve Problems 1 and 2.

The intuition behind our algorithm is to consider these problems as submodular cost submodular knapsack (SCSK) constraint maximization problems [32], [33]. An SCSK constrained maximization problem is defined as

$$\text{maximize } f(X) \quad \text{subject to } g(X) \leq b.$$

for submodular functions f and g , and upper bound $b \in \mathbb{R}$. Problems 1 and 2 can be interpreted as special cases of SCSK with additional cardinality constraints

$$\begin{aligned} &\text{maximize } \mu(S_1) \\ &\text{subject to: } S_1 \subseteq Q, \mu(S_1) \leq b, |S_1| \leq k. \end{aligned} \quad (25)$$

Our algorithm is motivated by an approach in [33] for the general SCSK problem. We approximate the optimum $\mu(S_1)$ for Problem 1 or 2 by imposing an upper bound for the submodular function μ and then applying a submodular maximization algorithm to the bounded problem. Specifically, we find an appropriate upper bound constraint $\mu(S_1) \leq b$, such that a greedy algorithm for maximizing $\mu(S_1)$ with upper bound b leads to an approximation algorithm for optimum solution S^* , where

$$S^* \in \arg \min_{P \subseteq Q, |P| \leq k} |\mu(P) - \alpha|$$

is an optimal solution for Problem 1 or 2, respectively.

We apply a greedy algorithm to problem (25). For an upper bound b , the algorithm Greedy returns a solution S_b . We can compare different upper bounds by the solutions Greedy returns. The bound b_1 is a better upper bound than b_2 if $|\mu(S_{b_1}) - \alpha| < |\mu(S_{b_2}) - \alpha|$. We further define the best upper bound input for algorithm Greedy as b^* , or formally,

$$b^* \in \arg \min_{b \in [\alpha, 1]} |\mu(S_b) - \alpha|. \quad (26)$$

We use a modified binary search to converge to the best upper bound b^* for Greedy. In the next section, we describe both the bound search algorithm and the routine Greedy.

B. Bounded Search Approximation Algorithm

We first define the algorithm in terms of Problem 1. We describe the changes of the algorithm in order to solve Problem 2 in the end of the section.

Our algorithm, BoundSearch, is given in Algorithm 1. Algorithm takes as input a graph $\mathcal{G}$, a candidate vertex set Q , an objective opinion α , a cardinality constraint k , and a precision parameter δ for binary search. It returns a set of nodes P , which is a subset of Q satisfying $|P| \leq k$.

The bound $\hat{b}$ is initialized with value 1, and the algorithm searches for b^* in the interval $[b_{\min}, b_{\max}]$ that might include a better upper bound than $\hat{b}$, the current best bound found by the algorithm that leads to the smallest $|\mu(S_b) - \alpha|$. We update $b_{\min}$ and $b_{\max}$ until $b_{\min} \approx_{\delta} b_{\max}$, and $b^*, \hat{b}, \tilde{b} \in [b_{\min}, b_{\max}]$. We obtain $\hat{b} \approx_{\delta} b^*$. Since $\hat{b}$ is the current best upper bound found by the algorithm, for any $b \notin [b_{\min}, b_{\max}]$, $|S_{\hat{b}} - \alpha| \leq |S_b - \alpha|$.

Before analyzing Algorithm 1, we recall the concept of ϵ -approximation [31]:

Definition V.1: Given two numbers $a, b \in \mathbb{R}$, $a, b \geq 0$, if

$$\exp(-\epsilon)a \leq b \leq \exp(\epsilon)a$$

then a is an ϵ -approximation of b , denoted by $a \approx_{\epsilon} b$.

Note that $a \approx_{\epsilon} b$ if and only if $b \approx_{\epsilon} a$.

In Algorithm 2, we present the greedy routine $P = \text{Greedy}(\mathcal{G}, Q, \hat{b}, k)$ for the constrained submodular maximization described in (25). The algorithm takes as input a graph $\mathcal{G}$,

Algorithm 1: $P = \text{BoundSearch}(\mathcal{G}, Q, \alpha, k, \delta)$.

```

1  $P \leftarrow \emptyset$ 
2  $b_{\min} \leftarrow \alpha$ ;  $b_{\max} \leftarrow 1$ 
3  $\hat{b} \leftarrow 1$  // current upper bound
4  $\check{b} \leftarrow \hat{b}$  // current best upper bound
5  $d_{\min} \leftarrow \alpha$  // minimal  $|\mu - \alpha|$  so far
6  $t \leftarrow 0$ 
7 do
8    $t \leftarrow t + 1$ 
9    $(S, \mu) = \text{Greedy}(\mathcal{G}, Q, \hat{b}, k)$ 
10   $\hat{d} \leftarrow |\mu - \alpha|$ 
11  if  $\hat{d} < d_{\min}$  then
12     $P \leftarrow S$ ;  $d_{\min} \leftarrow \hat{d}$ ;  $\check{b} \leftarrow \hat{b}$ 
13  if  $\mu > \alpha$  then
14    while  $\mu \leq (b_{\min} + b_{\max})/2$  do
15       $b_{\max} \leftarrow (b_{\min} + b_{\max})/2$ 
16  else
17     $b_{\min} \leftarrow \hat{b}$ 
18    if  $(\alpha + \hat{d}) < b_{\max}$  then
19       $b_{\max} \leftarrow (\alpha + \hat{d})$ 
20       $\hat{b} \leftarrow b_{\max}$ ; continue
21   $\hat{b} \leftarrow (b_{\min} + b_{\max})/2$ 
22 while  $\frac{b_{\max}}{b_{\min}} > \exp(\delta)$ 
23 return  $P$ 
    
```

Algorithm 2: $(P, \mu) = \text{Greedy}(\mathcal{G}, Q, \hat{b}, k)$.

```

1  $P \leftarrow \emptyset$ 
2 while  $|Q| > 0$  and  $|P| < k$  do
3    $s \leftarrow \arg \max_{u \in Q} \mu(P \cup \{u\})$ 
4   if  $\mu(P \cup \{s\}) \leq \hat{b}$  then
5      $P \leftarrow (P \cup \{s\})$ 
6    $Q \leftarrow (Q \setminus \{s\})$ 
7  $\gamma \leftarrow \mu(P)$ 
8 return  $(P, \gamma)$ 
    
```

a candidate set Q , an SCSK upper bound $\hat{b}$, and an integer k for the cardinality constraint. It returns a set of nodes P , which is a subset of Q satisfying $|P| \leq k$ and $\mu(P) \leq \hat{b}$. The algorithm chooses the node that most increases $\mu(P)$ without violating the upper bound from the candidate set in each iteration, deletes it from the candidate set, and adds it to the current leader set.

To analyze Algorithm 2, we introduce the concept of the minimum cover number.

Definition V.2: The *minimum cover number* $k_{\mu, b}$ for set function $\mu(S)$, $S \subseteq Q$, and $b \in \mathbb{R}$ is defined as

$$k_{\mu, b} \stackrel{\text{def}}{=} \min\{|S| : \mu(S) \geq b\}$$

if there exists S satisfying $\mu(S) \geq b$, otherwise $k_{\mu, b} \stackrel{\text{def}}{=} +\infty$.

Then, we obtain the approximation ratio of BoundSearch.

Theorem V.3: Consider a graph $\mathcal{G}$, a candidate set Q , an objective α , a cardinality constraint $|P| \leq k$, and a precision parameter $\delta > 0$. Let S^* be an optimal solution for Problem 1 for

TABLE I

AVERAGE OPINION IN AN ABSOLUTE LEADER SYSTEM

α	Optimum	DS	CT	Random
0.25	0.249830	0.250214	0.000083	0.248751
0.50	0.499699	0.500495	0.000083	0.122848
0.75	0.750014	0.750014	0.000083	0.002391
1.00	0.999645	0.999645	0.000083	0.576522

Note: The graph is the Twitter retweet network *rt-higgs* with a fixed s_0 and a node s_1 chosen via various methods.

these parameters. Algorithm $P = \text{BoundSearch}(\mathcal{G}, Q, \alpha, k, \delta)$ returns a node set P such that $\mu(P) \approx_{\sigma} \mu(S^*)$, in which $\sigma = -\ln(1 - \zeta) + \delta$, and $\zeta \stackrel{\text{def}}{=} \max(1/e, 1/k_{\mu, \alpha})$.

We defer the proof of Theorem V.3 to Appendix D.

The guarantee given in Theorem V.3 can also be written as

$$(1 - \zeta) e^{-\delta} \mu(S^*) \leq \mu(P) \leq (1 - \zeta)^{-1} e^{\delta} \mu(S^*).$$

The BoundSearch algorithm can be applied to Problem 2 with the same approximation guarantee with the only difference that the stubbornness function κ is an input of the algorithm. The stubbornness function is also passed into Greedy to calculate the average opinion. Theorem V.3 holds for the corresponding algorithm $P = \text{BoundSearch}(\mathcal{G}, Q, \alpha, k, \kappa, \delta)$, which calls $\text{Greedy}(\mathcal{G}, Q, \hat{b}, k, \kappa)$.

C. Complexity Analysis

A naive implementation of the proposed algorithm runs in $O(kqn^3 \log \frac{1}{\delta})$ time, which is expensive for large graphs. Using blockwise inversion and rank-1 update of matrices, we can improve the running time of BoundSearch to $O(n^3 \log \frac{1}{\delta})$.

Theorem V.4: There exists an implementation of Algorithm 1 for a graph with n nodes that has running time $O(n^3 \log \frac{1}{\delta})$.

The proof of Theorem V.4 is given in Appendix 5.

VI. EXPERIMENTS

In this section, we present experiments to highlight the analytical results and to show the effectiveness of the proposed algorithm.

We first study the properties of $\mu(S_1)$ when $|S_0| = |S_1| = 1$ in absolute and influenced leader systems for $\alpha = 0.25, 0.5, 0.75$, and 1. The leader s_0 is chosen uniformly at random. We run experiments on a directed and weighted social network. We utilize the largest strongly connected component of the Twitter retweet network with the keyword “higgs,” which we refer to as *rt-higgs* [34]. The edges are weighted by the number of retweets to a user. The network has 13 086 nodes and 63 537 edges.

For the absolute leader system, we find the average opinion of the network for the optimal solution to Problem 1, i.e., the optimal s_1 as given by Theorem IV.3. We also show the average opinion when s_1 is chosen using heuristics motivated by the theorem. The first heuristic, DS, is based on the domination score; we find the s_1 such that the resulting $\mu(\{s_1\})$ minimizes the numerator of (22). We also use a heuristic based on commute time (CT); here, s_1 is chosen so as to maximize the denominator of (22). Finally, we compute the average opinion for a randomly chosen s_1 . The results of this experiment are shown in Table I.

TABLE II
AVERAGE OPINION IN AN INFLUENCED LEADER SYSTEM.

α	Optimum	DS&K	CT	Random
0.25	0.250010	0.250010	0.000028	0.327045
0.50	0.500010	0.500010	0.000028	0.345662
0.75	0.750111	0.750111	0.000028	0.305945
1.00	0.997124	0.997124	0.000028	0.000000

Note: The Graph is the twitter retweet network *rt-higgs* with a fixed s_0 and a node s_1 chosen via various methods.

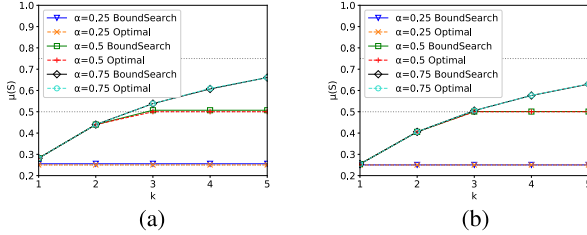


Fig. 3. Average opinion of Optimum average opinion of BoundSearch in an Erdős-Rényi graph with 30 nodes and connecting probability 0.1. S_0 leader sets are chosen randomly, and S_1 leader sets are chosen by brute-force search and BoundSearch with different k and α values. (a) Absolute leader system. (b) Influenced leader system.

We also conduct an experiment for an influenced leader system using the *rt-higgs* network. Influenced leaders have uniform stubbornness $\kappa = 1$, and the other parameters are the same as the experiment for the absolute leader system. We find the optimal s_1 as well as the s_1 chosen by heuristics motivated by the numerator (DS&K) and denominator (CT) of the result given in Theorem IV.4. We note that the s_1 that minimizes denominator of the result in Theorem IV.3 also minimizes denominator of the result in Theorem IV.4. The results are shown in Table II.

Tables I and II show that when $|S_0| = |S_1| = 1$, the domination score well captures the behavior of $\mu(\{s_1\})$. We have observed similar results in various Erdős-Rényi graphs with different choices of a single leader s_0 .

Next, to show the effectiveness of our leader selection algorithm, we compare the result returned by our algorithm BoundSearch with the optimal value returned by brute-force search. We use an unweighted undirected Erdős-Rényi graph with 30 nodes and connecting probability 0.1. We choose an S_0 leader set of size 3 at random. We run the BoundSearch algorithm for both absolute leader system and influenced leader systems with $\alpha \in \{0.25, 0.50, 0.75\}$. Influenced leaders use uniform stubbornness $\kappa = 1$. The results are shown in Fig. 3. In all cases, BoundSearch returns nearly optimal results.

We next run our leader selection algorithm on *rt-higgs* with $|S_0| = 100$ and $k = 100$. We compare the result produced by our algorithm BoundSearch with a heuristic we call the Propositional Domination Score and with a randomly select set. To calculate the Proposition Domination Score, for each leader in S_0 , we choose a leader for S_1 according to Theorem IV.3 for absolute leaders and Theorem IV.4 for influenced leaders. For all influenced leaders, $\kappa = 1$. Tables III and IV show that our algorithm converges to the desired value and outperforms the heuristic in all tested cases.

TABLE III
AVERAGE OPINION IN AN ABSOLUTE LEADER SYSTEM ON THE *RT-HIGGS* NETWORK WITH $|S_0| = 100$ AND $k = 100$

α	BoundSearch	PDS	Random
0.25	0.250730	0.237610	0.428106
0.50	0.500975	0.565233	0.206765
0.75	0.750976	0.843537	0.495980
1.00	1.000000	1.000000	0.466443

TABLE IV
AVERAGE OPINION IN AN INFLUENCED LEADER SYSTEM ON THE *RT-HIGGS* NETWORK WITH $|S_0| = 100$ AND $k = 100$

α	BoundSearch	PDS&K	Random
0.25	0.250732	0.252612	0.640516
0.50	0.500976	0.499114	0.418073
0.75	0.750976	0.748867	0.455767
1.00	0.975863	0.973205	0.533567

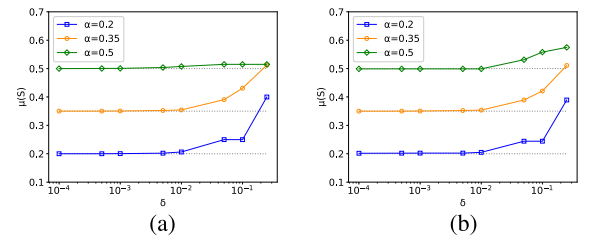


Fig. 4. Effect of varying δ on BoundSearch at $\alpha \in \{0.20, 0.35, 0.50\}$. Experiment uses the Haggles graph. (a) Absolute leader system. (b) Influenced leader system.

Finally, we explore the effect of varying the δ parameter in BoundSearch. We run BoundSearch on the Haggles [35] social contact graph. The Haggles graph is a multigraph, which we turn it into an undirected simple graph by deleting all duplicate edges. We use the largest connected component of the graph that has 274 nodes and 2124 edges. All edges have unit edge weight. We set $k = 15$ and $\alpha \in \{0.2, 0.35, 0.5\}$. We vary δ from 0.0001 to 0.25. For the absolute leader system, we have $|S_0| = 80$, and for the influenced leader system, we have $|S_0| = 15$. Influenced leaders use uniform stubbornness $\kappa = 1$. The results are shown in Fig. 4. We observe that as δ decreases, the results from BoundSearch converge to a value close to α .

VII. CONCLUSION

We studied two French-DeGroot opinion dynamics models where leaders have polarizing opinions. For both models, we showed expressions for the steady-state opinion using the Laplacian matrix of a leader-equivalent graph. For the single leader case, we gave an explicit expression for the steady-state opinion vector and analyzed the average opinion based on the expression. Then, we studied the problem of shifting the average steady-state opinion to a given value by selecting an opposing leader set with a cardinality constraint. We gave both a hardness result for this problem and an algorithm with provable approximation ratio. We also presented experiments showing that our algorithm returns results close to optimal in practice. Future work will focus on algorithms with better approximation ratios and

running time and the hardness of the influenced leader selection problem.

APPENDIX

A. Some Useful Matrix Identities

We introduce some matrix identities. Proofs for the following lemmas are given in our technical report [29].

Lemma A.1: For any $\mathbf{p} \perp \mathbf{1}$, $\mathbf{r} \perp \mathbf{1}$

$$\mathbf{p}^\top \mathcal{L}^\dagger \mathbf{r} = \mathbf{p}^\top \mathcal{R} \mathbf{r}.$$

Lemma A.2:

$$\begin{aligned} \mathbf{I} - \mathbf{L}\mathbf{L}^\dagger &= \frac{1}{\|\mathbf{D}^{-1}\boldsymbol{\pi}\|^2} \mathbf{D}^{-1} \boldsymbol{\pi} \boldsymbol{\pi}^\top \mathbf{D}^{-1} \\ \mathbf{I} - \mathbf{L}^\dagger \mathbf{L} &= \frac{1}{n} \mathbf{1}\mathbf{1}^\top. \end{aligned}$$

Lemma A.3: For any $\mathbf{p} \perp \mathbf{1}$, $\mathbf{r} \perp \mathbf{1}$

$$\mathbf{p}^\top \mathbf{L}^\dagger \mathbf{D} \boldsymbol{\Pi}^{-1} \mathbf{r} = \mathbf{p}^\top \mathcal{L}^\dagger \mathbf{r}.$$

Lemma A.4: For any $\mathbf{y} \perp \mathbf{1}$

$$(\boldsymbol{\Pi}(\mathbf{I} - \mathbf{W}^\top)) ((\mathbf{I} - \mathbf{W}^\top)^\dagger \boldsymbol{\Pi}^{-1}) \mathbf{y} = \left(\mathbf{I} - \frac{1}{n} \mathbf{1}\mathbf{1}^\top \right) \mathbf{y}.$$

B. Proof of Proposition IV.1

Proof: We can express (1) and (2) in the following form:

$$\begin{pmatrix} \dot{\mathbf{x}}_S(t) \\ \dot{\mathbf{x}}_F(t) \end{pmatrix} = - \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{F,S} & \mathbf{L}_{F,F} \end{pmatrix} \begin{pmatrix} \mathbf{x}_S(t) \\ \mathbf{x}_F(t) \end{pmatrix}.$$

When the equilibrium is reached and

$$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{F,S} & \mathbf{L}_{F,F} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}}_S(t) \\ \hat{\mathbf{x}}_F(t) \end{pmatrix} = \mathbf{0}.$$

Since $\mathbf{L} = \mathbf{D}(\mathbf{I} - \mathbf{W}^\top)$, this is equivalent to solving

$$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ [\mathbf{I} - \mathbf{W}^\top]_{F,S} & [\mathbf{I} - \mathbf{W}^\top]_{F,F} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}}_S(t) \\ \hat{\mathbf{x}}_F(t) \end{pmatrix} = \mathbf{0}.$$

When $S_0 = \{s'_0\}$ and $S_1 = \{s'_1\}$, $\mathbf{x}_S(t) = (1 \ 0)^\top$; it suffices to solve

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ [x\mathbf{I} - \mathbf{W}^\top]_{F,s'_0} [I - \mathbf{W}^\top]_{F,s'_1} [I - \mathbf{W}^\top]_{F,F} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ \hat{\mathbf{x}}_F(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \quad (27)$$

By solving

$$(\mathbf{I} - \mathbf{W}^\top)(\mathbf{z}_S^\top \mathbf{z}_F^\top)^\top = (-\boldsymbol{\pi}_{s'_0}^{-1} \boldsymbol{\pi}_{s'_1}^{-1} \mathbf{0}^\top)^\top \quad (28)$$

we obtain a $\mathbf{z}_F$ that satisfies the latter $n - 2$ equations in (27). We note that (28) has solutions because $(-\boldsymbol{\pi}_{s'_0}^{-1} \boldsymbol{\pi}_{s'_1}^{-1} \mathbf{0}^\top)^\top \in \ker(\mathbf{I} - \mathbf{W}^\top)$. Since the rank of $(\mathbf{I} - \mathbf{W}^\top)$ is $n - 1$ and $(\mathbf{I} - \mathbf{W}^\top)\mathbf{1} = \mathbf{0}$, for any $\mathbf{z}$ satisfying the system of equations (28), $\mathbf{y} = \mathbf{z} + \gamma\mathbf{1}$ also satisfies (28), where γ can be any real number.

We observe that $\mathbf{z} = (\mathbf{I} - \mathbf{W}^\top)^\dagger \boldsymbol{\Pi}^{-1} \mathbf{b}_{s'_1, s'_0}$ is a solution of (28). This can be verified by plugging it into (28)

$$\begin{aligned} &(\mathbf{I} - \mathbf{W}^\top)(\mathbf{I} - \mathbf{W}^\top)^\dagger \boldsymbol{\Pi}^{-1} \mathbf{b}_{s'_1, s'_0} \\ &= \boldsymbol{\Pi}^{-1} (\boldsymbol{\Pi}(\mathbf{I} - \mathbf{W}^\top)) ((\mathbf{I} - \mathbf{W}^\top)^\dagger \boldsymbol{\Pi}^{-1}) \mathbf{b}_{s'_1, s'_0} \\ &= \boldsymbol{\Pi}^{-1} \left(\mathbf{I} - \frac{1}{n} \mathbf{1}\mathbf{1}^\top \right) \mathbf{b}_{s'_1, s'_0} = \boldsymbol{\Pi}^{-1} \mathbf{b}_{s'_1, s'_0}. \end{aligned}$$

The second equality follows from Lemma A.4. Then, we further set $\mathbf{z}' = \mathbf{z} - \mathbf{z}_{s'_0} \mathbf{1}$ to make $\mathbf{z}'_{s'_0} = 0$. Now, we have found $\mathbf{z}'$ which satisfies (27) except for the second equation. We note that by multiplying a factor β to $\mathbf{z}'$, the other $n - 1$ equations are still satisfied. So we let $\mathbf{y}' = (\mathbf{z}_u - \mathbf{z}_v)^{-1} \mathbf{z}'$. Then, $\hat{\mathbf{x}} = \mathbf{y}'$ is the solution of (27). ■

C. Proof of Theorem IV.4

Proof: According to the Sherman–Morrison formula

$$\begin{aligned} \hat{\mathbf{x}}_v &= \mathbf{e}_v^\top \left((\mathbf{L} + \mathbf{E}^{s_0} \kappa_0)^{-1} \right. \\ &\quad \left. - \frac{\kappa_1 (\mathbf{L} + \mathbf{E}^{s_0} \kappa_0)^{-1} \mathbf{E}^{s_1} (\mathbf{L} + \mathbf{E}^{s_0} \kappa_0)^{-1}}{1 + \kappa_1 \mathbf{e}_{s_1}^\top (\mathbf{L} + \mathbf{E}^{s_0} \kappa_0)^{-1} \mathbf{e}_{s_1}} \right) \mathbf{e}_{s_1} \kappa_1. \end{aligned} \quad (29)$$

Let us then consider $(\mathbf{L} + \mathbf{E}^{s_0} \kappa_0)^{-1}$. Since $\mathbf{L}$ is a singular matrix, the Sherman–Morrison formula cannot be applied in this case. Instead, we apply the rank-1 update given by [36, Th. 1]. By further applying some matrix identities discussed in Appendix 1, we obtain

$$\begin{aligned} (\mathbf{L} + \mathbf{E}^{s_0} \kappa_0)^{-1} &= \mathbf{L}^\dagger - \frac{1}{\mathbf{q}_{s_0}} \cdot (\mathbf{L}^\dagger \mathbf{e}_{s_0}) \mathbf{q}^\top - \mathbf{1} (\mathbf{e}_{s_0}^\top \mathbf{L}^\dagger) \\ &\quad + (1/\kappa_0 + \mathbf{e}_{s_0}^\top \mathbf{L}^\dagger \mathbf{e}_{s_0}) \cdot \frac{1}{\mathbf{q}_{s_0}} \cdot \mathbf{1} \mathbf{q}^\top \end{aligned} \quad (30)$$

where $\mathbf{q} = \mathbf{D}^{-1} \boldsymbol{\pi}$. Plugging (30) into (29), we arrive at

$$\hat{\mathbf{x}}_v = \frac{\frac{1}{\kappa_0 \mathbf{q}_{s_0}} + \mathbf{b}_{v, s_0}^\top \mathbf{L}^\dagger \mathbf{D} \boldsymbol{\Pi}^{-1} \mathbf{b}_{s_1, s_0}}{\frac{1}{\kappa_0 \mathbf{q}_{s_0}} + \frac{1}{\kappa_1 \mathbf{q}_{s_1}} + \mathbf{b}_{s_1, s_0}^\top \mathbf{L}^\dagger \mathbf{D} \boldsymbol{\Pi}^{-1} \mathbf{b}_{s_1, s_0}}. \quad (31)$$

We further note that for any $\mathbf{p} \perp \mathbf{1}$ and $\mathbf{r} \perp \mathbf{1}$, $\mathbf{p}^\top \mathbf{L}^\dagger \mathbf{D} \boldsymbol{\Pi}^{-1} \mathbf{r} = \mathbf{p}^\top \mathcal{L}^\dagger \mathbf{r}$ (see Appendix 1 for details). Then, we obtain

$$\mu(S_1) = \frac{\frac{1}{\kappa_0 \mathbf{q}_{s_0}} + \mathbf{e}_{s_0}^\top \mathcal{L}^\dagger \mathbf{b}_{s_0, s_1}}{\frac{1}{\kappa_0 \mathbf{q}_{s_0}} + \frac{1}{\kappa_1 \mathbf{q}_{s_1}} + \mathbf{b}_{s_1, s_0}^\top \mathcal{L}^\dagger \mathbf{b}_{s_1, s_0}}$$

which directly leads to the desired result. ■

D. Proof of Theorem V.3

Proof: We let $\tilde{b}$ be the best bound found by the algorithm with the smallest $|\mu(S_{\tilde{b}}) - \alpha|$. And, Greedy with the best upper bound b^* returns the result $\mu(S_{b^*})$. b^* is given by (26).

If $\mu(P \cup \{s\}) \leq \alpha$ is always satisfied during the execution, then $\mu(P \cup \{s\}) \leq \tilde{b}$ is also always satisfied. Then, the returned $S_{\tilde{b}}$ is the same as what we get from a greedy algorithm that adds the element with largest marginal gain to the current set

in each iteration until the cardinality constraint is violated. We further define $\tilde{S} \in \arg \max_{T \subseteq Q, |T| \leq k} \mu(T)$; therefore, by the result in [30], we obtain $\mu(\tilde{S}) \geq \mu(S_b) \geq (1 - \frac{1}{e})\mu(\tilde{S})$.

If $\alpha \geq \mu(\tilde{S})$, then $\mu(\tilde{S}) = \mu(S^*)$, we attain the guarantee $\mu(S_b) \approx_\gamma \mu(\tilde{S})$, where $e^{-\gamma} = (1 - 1/e)$. If $\mu(S_b) \leq \alpha \leq \mu(\tilde{S})$, then $\mu(S^*) \in [\mu(S_b), \mu(\tilde{S})]$, which implies $\mu(S_b) \approx_\gamma \mu(S^*)$, where $e^{-\gamma} = (1 - 1/e)$.

If $\mu(P \cup \{s\}) \leq \alpha$ is first violated when we add the $(t+1)$ th node, we define P_t as the set of chosen nodes of size t in Greedy; therefore, $|P_t| = t$. We further define $\rho(s_{t+1}) = \mu(P_t \cup \{s_{t+1}\}) - \mu(P_t)$. From the submodularity of $\mu(S)$, we know that $\rho(s_{t+1}) \leq \frac{1}{t+1}\mu(P_t \cup \{s_{t+1}\})$ holds for the Greedy algorithm. Then, $\mu(P_t) = \mu(P_t \cup \{s_{t+1}\}) - \rho(s_{t+1}) \geq \frac{t}{t+1}\mu(P_t \cup \{s_{t+1}\}) \geq \frac{k_{\mu,\alpha}-1}{k_{\mu,\alpha}}\mu(P_t \cup \{s_{t+1}\})$. By letting $\bar{b} = \mu(P_t \cup \{s_{t+1}\})$ (then by definition $\bar{b} = \mu(S_b) = \mu(P_t \cup \{s_{t+1}\})$), we obtain $\mu(S_\alpha) \geq (1 - \frac{1}{k_{\mu,\alpha}})\mu(S_b)$. We further attain $\mu(S^*), \mu(S_{b^*}) \in [\mu(S_\alpha), \mu(S_b)]$, and $b^* \in [\alpha, \bar{b}]$. Since $\bar{b}, b^* \in [b_{\min}, b_{\max}]$ and $b_{\min} \approx_\delta b_{\max}$, $\bar{b} \approx_\delta b^*$, we obtain $\bar{b} \in [\alpha, e^\delta \bar{b}]$, and therefore, $\mu(S_b) \in [\mu(S_\alpha), e^\delta \mu(S_b)]$, so $\mu(S_b) \approx_\gamma \mu(S^*)$, where $e^{-\gamma} = (1 - 1/k_{\mu,\alpha})e^{-\delta}$.

E. Proof of Theorem V.4

Proof: We take the algorithm for the absolute leader selection problem as an example. In each execution of Line 3 of Algorithm 2, we need to calculate the sum of steady states of followers given by $-1^\top (L_{F,F})^{-1} L_{F,S} x_S$, for all $P \cup \{u\}$, $u \in Q$. P and Q are the current leader set of opinion 1 and the current candidate set. Calculating $(L_{F,F})^{-1}$ when $S_1 = \emptyset$ takes $O(n^3)$ running time. L_{FF} can be updated at iteration $t+1$ by deleting the row and column associated with candidate node u . From block matrix inversion, we obtain that its inverse can be updated by

$$\begin{aligned} (L_{(F(t) \setminus \{u\}), (F(t) \setminus \{u\})})^{-1} &= \left((L_{F(t), F(t)})^{-1} \right. \\ &\quad \left. - \frac{(L_{F(t), F(t)})^{-1} e_u e_u^\top (L_{F(t), F(t)})^{-1}}{e_u^\top (L_{F(t), F(t)})^{-1} e_u} \right)_{(F(t) \setminus \{u\}), (F(t) \setminus \{u\})}. \end{aligned}$$

To calculate $\mu(P_t)$, we do not need to find $(L_{(F(t) \setminus \{u\}), (F(t) \setminus \{u\})})^{-1}$ explicitly. It suffices to compare $-1^\top (L_{(F(t) \setminus \{u\}), (F(t) \setminus \{u\})})^{-1} L_{(F(t) \setminus \{u\}), (S(t) \cup \{u\})} x_{S \cup \{u\}}$, for all u in the current candidate set. We note that e_u ($e_u^\top$) takes a column (row) of $(L_{F(t), F(t)})^{-1}$, and $L_{(F(t) \setminus \{u\}), (S(t) \cup \{u\})} x_{S \cup \{u\}}$ is a column vector. By the associative law, we compute the vector inner product first and find the updated $\mu(S_1)$ for at most n candidates in $O(n^2)$ total running time. The operations of taking the submatrices do not change the complexity because for any candidate u , these operations only take $O(|\mathcal{N}_u^\uparrow| + |\mathcal{N}_u^\downarrow|)$ running time. So, in each execution of Line 3 of Algorithm 2, these operations can be done in $O(m)$ total running time, where m is the number of edges in the graph. After we find the best choice s_{t+1} in step $t+1$, we update $(L_{F,F})^{-1}$ explicitly, which takes additional $O(n^2)$ time. Therefore, execution of Line 3 of Algorithm 2 takes $O(n^2)$ time. By using this simple acceleration, the complexity of Algorithm 2

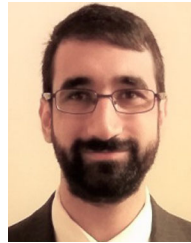
is improved to $O(n^3 + kn^2) = O(n^3)$. Algorithm 1 calls Greedy $O(\log \frac{1}{\delta})$ times until $b_{\max} \approx_\delta b_{\min}$. Since $b_{\max} - b_{\min}$ decreases geometrically in Algorithm 1, the total running time of BoundSearch is $O(n^3 \log \frac{1}{\delta})$.

For the influenced leader selection problem, the rank-1 update is obtained using the Sherman–Morrison formula. The running time of the Greedy routine is also $O(n^3)$ by a similar implementation. We omit the details of the analysis. ■

REFERENCES

- [1] M. E. Yildiz, A. E. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione, "Binary opinion dynamics with stubborn agents," *ACM Trans. Econ. Comput.*, vol. 1, no. 4, pp. 19:1–19:30, 2013.
- [2] A. Gionis, E. Terzi, and P. Tsaparas, "Opinion maximization in social networks," in *Proc. 13th SIAM Int. Conf. Data Mining*, 2013, pp. 387–395.
- [3] L. Vassio, F. Fagnani, P. Frasca, and A. E. Ozdaglar, "Message passing optimization of harmonic influence centrality," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 109–120, Mar. 2014.
- [4] D. S. Hunter and T. Zaman, "Optimizing opinions with stubborn agents under time-varying dynamics," 2019, *arXiv:1806.11253v3*.
- [5] V. S. Mai and E. H. Abed, "Optimizing leader influence in networks through selection of direct followers," *IEEE Trans. Automat. Control*, vol. 64, no. 3, pp. 1280–1287, Mar. 2019.
- [6] E. Mackin and S. Patterson, "Maximizing diversity of opinion in social networks," in *Proc. Amer. Control Conf.*, 2019, pp. 2728–2734.
- [7] J. R. French Jr., "A formal theory of social power," *Psychol. Rev.*, vol. 63, no. 3, p. 181, 1956.
- [8] M. H. DeGroot, "Reaching a consensus," *J. Amer. Stat. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.
- [9] C. R. Sunstein, "The law of group polarization," *J. Political Philos.*, vol. 10, no. 2, pp. 175–195, Apr.–Jun. 2002.
- [10] N. E. Friedkin, "The attitude-behavior linkage in behavioral cascades," *Social Psychol. Quart.*, vol. 73, no. 2, pp. 196–213, 2010.
- [11] N. E. Friedkin, "The problem of social control and coordination of complex systems in sociology: A look at the community cleavage problem," *IEEE Control Syst. Mag.*, vol. 35, no. 3, pp. 40–51, Jun. 2015.
- [12] P. Barooah and J. P. Hespanha, "Graph effective resistance and distributed control: Spectral properties and applications," in *Proc. 45th IEEE Conf. Decis. Control*, 2006, pp. 3479–3485.
- [13] S. Patterson and B. Bamieh, "Leader selection for optimal network coherence," in *Proc. 49th IEEE Conf. Decis. Control*, 2010, pp. 2692–2697.
- [14] F. Lin, M. Fardad, and M. R. Jovanovic, "Algorithms for leader selection in stochastically forced consensus networks," *IEEE Trans. Automat. Control*, vol. 59, no. 7, pp. 1789–1802, Jul. 2014.
- [15] A. Clark, L. Bushnell, and R. Poovendran, "A supermodular optimization framework for leader selection under link noise in linear multi-agent systems," *IEEE Trans. Automat. Control*, vol. 59, no. 2, pp. 283–296, Feb. 2014.
- [16] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, "Minimizing convergence error in multi-agent systems via leader selection: A supermodular optimization approach," *IEEE Trans. Automat. Control*, vol. 59, no. 6, pp. 1480–1494, Jun. 2014.
- [17] H. Li, R. Peng, L. Shan, Y. Yi, and Z. Zhang, "Current flow group closeness centrality for complex networks" in *Proc. World Wide Web Conf.*, 2019, pp. 961–971.
- [18] E. Mackin and S. Patterson, "Submodular optimization for consensus networks with noise-corrupted leaders," *IEEE Trans. Automat. Control*, vol. 64, no. 7, pp. 3054–3059, Jul. 2019.
- [19] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar, "Opinion fluctuations and disagreement in social networks," *Math. Oper. Res.*, vol. 38, no. 1, pp. 1–27, 2013.
- [20] G. Como and F. Fagnani, "From local averaging to emergent global behaviors: The fundamental role of network interconnections," *Syst. Control Lett.*, vol. 95, pp. 70–76, 2016.
- [21] W. S. Rossi and P. Frasca, "On the convergence of message passing computation of harmonic influence in social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 2, pp. 116–129, Apr.–Jun. 2019.
- [22] M. Grabisch, A. Mandel, A. Rusinowska, and E. Tanimura, "Strategic influence in social networks," *Math. Oper. Res.*, vol. 43, no. 1, pp. 29–50, 2018.

- [23] M. B. Cohen, J. Kelner, J. Peebles, R. Peng, A. Sidford, and A. Vladu, "Faster algorithms for computing the stationary distribution, simulating random walks, and more," in *Proc. 57th IEEE Annu. Symp. Found. Comput. Sci.* 2016, pp. 583–592.
- [24] D. J. Klein and M. Randić, "Resistance distance," *J. Math. Chem.*, vol. 12, no. 1, pp. 81–95, 1993.
- [25] K. Stephenson and M. Zelen, "Rethinking centrality: Methods and examples," *Social Netw.*, vol. 11, no. 1, pp. 1–37, 1989.
- [26] Y. Cao, W. Ren, and M. Egerstedt, "Distributed containment control with multiple stationary or dynamic leaders in fixed and switching directed networks," *Automatica*, vol. 48, no. 8, pp. 1586–1597, 2012.
- [27] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010, vol. 33.
- [28] G. Fricke, S. T. Hedetniemi, and D. P. Jacobs, "Independence and irredundance in k-regular graphs," *Ars Combinatoria*, vol. 49, pp. 271–279, 1998.
- [29] Y. Yi, T. Castiglia, and S. Patterson, "Shifting opinions in a social network through leader selection," 2020, *arXiv:1910.13009v2*.
- [30] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.
- [31] R. Peng and D. A. Spielman, "An efficient parallel solver for SDD linear systems," in *Proc. 46th Symp. Theory Comput.*, 2014, pp. 333–342.
- [32] A. Atamtürk and V. Narayanan, "The submodular knapsack polytope," *Discrete Optim.*, vol. 6, no. 4, pp. 333–344, 2009.
- [33] R. K. Iyer and J. A. Bilmes, "Submodular optimization with submodular cover and submodular knapsack constraints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2436–2444.
- [34] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 4292–4293.
- [35] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 606–620, Jun. 2007.
- [36] C. D. Meyer Jr., "Generalized inversion of modified matrices," *SIAM J. Appl. Math.*, vol. 24, no. 3, pp. 315–323, 1973.



Timothy Castiglia received the B.S. degree in computer science in 2017 from the Rensselaer Polytechnic Institute, Troy, NY, USA, where he is currently working toward the Ph.D. degree in the computer science.

His research interests include distributed algorithms, network science, and machine learning.

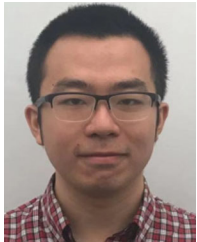


Stacy Patterson (Member, IEEE) received the M.S. and Ph.D. degrees in computer science from UC Santa Barbara, Santa Barbara, CA, USA, in 2003 and 2009, respectively.

She is an currently Associate Professor with the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, USA. From 2009 to 2011, she was a Postdoctoral Scholar with the Center for Control, Dynamical Systems and Computation, UC Santa Barbara. From 2011 to 2013, she was a Postdoctoral Fellow

with the Department of Electrical Engineering at Technion – Israel Institute of Technology, Haifa, Israel. Her research interests include distributed systems and algorithms, cloud computing, and the Internet of Things.

Dr. Patterson was the recipient of the Viterbi Postdoctoral Fellowship, the IEEE CSS Axelby Outstanding Paper Award, and an NSF CAREER award.



Yuhao Yi received the B.S. and Ph.D. degrees in computer science from Fudan University, Shanghai, China, in 2013 and 2019, respectively.

From 2016 to 2017, he was a Visiting Scholar at the Rensselaer Polytechnic Institute, Troy, NY, USA, where he is currently a Postdoctoral Researcher. His research interests include network science and distributed control, and analyzing and optimizing the performance of averaging consensus dynamics in large-scale networks.