This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Competitiveness Maximization on Complex Networks

Jiuhua Zhao, Qipeng Liu, Lin Wang, *Member, IEEE*, and Xiaofan Wang, *Senior Member, IEEE*

Abstract*—We consider a model of competition on complex networks, in which two competitors are fixed to opposite states while other agents, called normal agents, adjust their states according to a distributed consensus protocol. Suppose that one of the competitors could enhance its influence by creating new links. A natural question is, when the number of new links is limited due to the limited resource, how to add these links so as to maximize the influence of the given competitor over the other one (called *competitiveness*). We consider two competitiveness maximization problems: *Problem 1* tries to maximize the number of supporters of the competitor, while *Problem 2* tries to maximize the total supporting degree of normal agents toward the competitor. We prove that *Problem 1* is NP-hard. We also show that the objective function of *Problem 2* is monotonous and submodular, and hence there exists a polynomial-time greedy algorithm (GA) approximately solving *Problem 2*. Several centrality-based heuristic algorithms of less computational burden are also designed to provide approximate solutions to these two problems. We carry out extensive simulations to check the performances of these algorithms in six real networks. We find that GA always provides the best approximate solution to *Problem 2*, while for *Problem 1*, GA only has the best performance in directed networks. Furthermore, among those heuristic algorithms, an algorithm based on centrality in descending order is better than its counterpart in ascending order in solving *Problem 2* in statistical sense. But for *Problem 1*, the performance of the centrality-based heuristic algorithms is more sensitive to the network structure and the locations of competitors.

***Index Terms*—Algorithm design, competitive dynamics, NP-hard, optimization, submodular.**

I. Introduction

It IS of great scientific interest to study the phenomenon of consensus. The past decades have witnessed a large amount of work on it, ranging from human society [1], biotic community [2] to engineering networks [3]. However, disagreement is also ubiquitous in the real world, such as customers buying different brands of the same kind of product, and people holding different beliefs. Disagreement can be caused by many factors. For instance, antagonistic interactions might lead to bipartite consensus [4]. The presence of multiple leaders might also result in the division of followers [5]. Competition as an important factor for disagreement can be seen almost everywhere in our real world, ranging from competition among languages [6], retailers [7], opinion leaders [8] to online marketing [9]. One of the mathematical models for competition on complex networks was the modified voter model, where zealots with opposite opinions were introduced to the classical voter model [10]. The models describing the competition of innovative products were constructed based on the word-of-mouth propagation model [11]–[13]. Moreover, modified susceptible-infected-susceptible models were proposed to investigate competitions of different kinds of viruses and epidemics [14]–[16]. A common feature of the above models is that agents only have discrete states—supportive or antagonistic, products A or B, susceptible or infected. Very few models with continuous states were proposed to study competing situations. In [17], a gossip model with stubborn agents was investigated, where agents possessed continuous states. Recently, we also proposed a model of competition with continuous state, where two competitors held different fixed states, and other normal agents adjusted their states according to a distributed consensus protocol [18]. We characterized the competition result as a function of the network structure and proposed a simple criterion for predicting the outcome.

Besides modeling competitions and analyzing the result of any given competing situation, we can further consider how to influence the competition result if we have an opportunity to manipulate some conditions of the competition. This might be more appealing to the competitors who are eagerly seeking strategies to win the competition. This problem is quite similar to the influence maximization problem. Roughly speaking, the influence maximization problem considers how to choose a set of nodes in a network as initial active ones so as to maximize the spread of a message or an idea to the rest of the network. Most of the existing works focused on the situation without competition, like the influence maximization problem in the word-of-mouth propagation process [19]–[21], the consensus leader selection problem [22], and the participants selection problem for offline event marketing [23]. In the

Manuscript received July 30, 2016; accepted November 20, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61374176, Grant 61473189, and Grant 61503207, in part by the Natural Science Foundation of Shandong Province under Grant ZR2015PF003, in part by the Project Funded by China Post-Doctoral Science Foundation under Grant 2015M571996, in part by the Qingdao Post-Doctoral Application Research Project under Grant 2015123, and in part by the Science Fund for Creative Research Groups of the National Natural Science Foundation of China under Grant 61221003. This paper was recommended by Associate Editor D. DeLaurentis.

J. Zhao, L. Wang, and X. Wang are with the Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China (e-mail: jiuhuadandan@sjtu.edu.cn; wanglin@sjtu.edu.cn; xfwang@sjtu.edu.cn).

Q. Liu is with the Institute of Complexity Science, Qingdao University, Qingdao 266071, China (e-mail: qipengliu@qdu.edu.cn).

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/TSMC.2016.2636240

competition situation, [24] investigated the problem that, with the knowledge of opponent's decision, how to choose the location of initial active nodes such that a competitor could maximize its influence.

All the above works investigated the influential agents (initial active nodes) placement problem in a given network, where the competitors were always outside the network. In this paper, we consider the circumstance where the locations of the competitors are fixed inside the network and one of the competitors has the opportunity to enhance its influence, even change the competition result, by changing the network's topology. Note that, we assume the competitor has the knowledge about the entire network topology. It is possible for competitors to acquire the structure in many cases, such as viral marketing in online social networks. In this paper, we only consider two competitors and assume that one of the competitors could modify the network topology by adding new links such that the competitor can directly influence more normal agents. Since creating new links is resource-costing, we suppose the number of new links is limited to m. Then, the question is how to add these new links such that the competitor has the maximal influence over the other competitor (called competitiveness). We consider two competitiveness maximization problems: 1) *Problem 1* tries to maximize the number of supports of the competitor and 2) *Problem 2* tries to maximize the total supporting degree of normal agents toward the competitor. We first show that *Problem 1* is NP-hard and *Problem 2* admits a submodular objective function. Then, we design proper greedy algorithms (GAs) to search for approximate solutions to the problems. Due to the computational burden of GAs on large scale networks, we further construct several heuristic algorithms based on classic centrality measures. All algorithms are tested and compared through simulations in real-world social networks.

This paper is organized as follows. The competition model and two optimization problems under different definitions of competitiveness are described in Section II. Section III deals with the complexity analysis of the optimization problems. In Section IV, heuristic algorithms based on centrality measures are proposed. Section V includes simulations to demonstrate the performances of these algorithms. We end this paper with the concluding remarks of Section VI.

II. Problem Description

A. Competition Model [18]

We denote a network by a directed graph $G = (V, E)$ with N nodes and M links, where V represents the set of agents $V = \{1, 2, \ldots, N\}$ and E represents the set of links among agents. A link from agents k to l, denoted by $<k, l>$, means agent k pays attention to and thus is influenced by agent l. We use a coupling matrix $A = (a_{kl})_{N,N}$ to describe the topology of the network: if $<k, l> \in E$, then $a_{kl} = 1$; otherwise, $a_{kl} = 0$. For simplicity, throughout this paper, we only consider two competitors in the network which, without loss of generality, are denoted as agents i and j, respectively. Suppose that each agent in the network has an associated continuous state. Here, the word "state" can refer to many specific terms in the real world. In the context of evaluating a product, the state captures the customer's opinion, which can vary smoothly from "very like" to "very dislike." The state can also be the political position of an individual, which can be located in anywhere from extreme right to extreme left. Suppose that the two competitors have the following fixed states:

$$x_i(t) \equiv +1, x_j(t) \equiv -1, \forall t \geq 0. \tag{1}$$

Every other agent $k \in V \backslash \{i, j\}$, here, we call normal agent, has a random initial state in R which is updated at discrete time steps as follows:

$$x_k(t+1) = x_k(t) + \varepsilon \sum_{l \in N_k} (x_l(t) - x_k(t)) \tag{2}$$

where $x_k(t)$ represents the state of normal agent k at time t; $N_k = \{l \in V | a_{kl} = 1\}$ is the set of neighbors of agent k. Equation (2) indicates that the normal agents try to reduce the state difference from their friends and make themselves more popular in their own social cycles. The time-invariant parameter ε captures the extent to which an individual is influenced by his friends.

We reorder the agents such that the competitors come the last, and thus, the coupling matrix and degree matrix have the following form:

$$A = \begin{bmatrix} \bar{A} & \mathbf{c}_i & \mathbf{c}_j \\ \mathbf{r}_i & * & * \\ \mathbf{r}_j & * & * \end{bmatrix} \quad D = \begin{bmatrix} \bar{D} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & d_i & 0 \\ \mathbf{0} & 0 & d_j \end{bmatrix}$$

where $\bar{A} \in R^{(N-2)\times(N-2)}$ captures the neighboring relationships among normal agents; column $\mathbf{c}_i$ ($\mathbf{c}_j$) $\in R^{N-2}$ and row $\mathbf{r}_i$ ($\mathbf{r}_j$) $\in R^{N-2}$ represent the relationships between normal agents and competitor i (j); $\bar{D} \in R^{(N-2)\times(N-2)}$ denotes the out-degree diagonal matrix of normal agents; d_i and d_j are the out-degrees of competitors i and j, respectively.

Suppose that:

1) each normal agent is connected to at least one competitor by a sequence of links;
2) $0 < \varepsilon < d_{\max}^{-1}$, where $d_{\max}$ is the largest out-degree of agents in the network.

Then, as $t \to \infty$, each normal agent's state eventually reaches a steady value

$$\bar{X} = (\bar{D} - \bar{A})^{-1} \begin{bmatrix} \mathbf{c}_i & \mathbf{c}_j \end{bmatrix} \begin{bmatrix} +1 \\ -1 \end{bmatrix} \tag{3}$$

where $\bar{X} = (\bar{x}_k)_{N-2}$ consists of the steady states of all normal agents. The proof of (3) can be found in our previous work [18]. We say agent k supports competitor i (j), if $\bar{x}_k > 0$ ($\bar{x}_k < 0$). $|\bar{x}_k|$ reflects the degree of supporting. Agent k with $\bar{x}_k = 0$ is a neutral agent.

B. Competitiveness Maximization Problems

To win the competition the competitor will try to possess greater influence than its rival. Here, the influence difference between a competitor and its rival is referred to as *competitiveness* (more precise definition will be provided later), which can be used to represent the competition result. A natural way to increase one competitor's competitiveness is creating new

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

links, which helps the competitor form a denser connection to normal agents. We use a vector $\phi \in R^{N-2}$ to indicate the locations of new links: if a new link is created from normal agent k to competitor i, then the kth element of ϕ, i.e., ϕ_k, equals 1; otherwise, ϕ_k equals 0. After adding all the new links, the coupling matrix and degree matrix become

$$A' = \begin{bmatrix} \bar{A} & \mathbf{c}_i + \phi & \mathbf{c}_j \\ \mathbf{r}_i & * & * \\ \mathbf{r}_j & * & * \end{bmatrix} \quad D' = \begin{bmatrix} \bar{D} + \Phi & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & d_i & 0 \\ \mathbf{0} & 0 & d_j \end{bmatrix}$$

where $\Phi = \text{diag}(\phi)$. According to (3), the steady-state vector can be shown as a function of the indicator vector ϕ

$$f_X(\phi) = (\bar{D} + \Phi - \bar{A})^{-1} \begin{bmatrix} \mathbf{c}_i + \phi & \mathbf{c}_j \end{bmatrix} \begin{bmatrix} +1 \\ -1 \end{bmatrix}. \tag{4}$$

Since creating new links is resource-costing, the competitor cannot indefinitely add new links to improve its competitiveness. Hence, the number of new links is limited to m. Then how to optimally locate these new links to maximize a competitor's competitiveness is the main focus of this paper. Next, we will introduce two definitions of competitiveness, based on which we have two versions of the competitiveness maximization problem.

1) Competitiveness Measured by the Number of Supporters: In some competitions, the relative number of supporters decides the outcome, e.g., the leader election campaign. By this consideration, we define the competitiveness of competitor i against competitor j as $\mathbf{1}^T \text{sgn}(f_X(\phi))$.

Then, we formulate the competitiveness maximization problem for competitor i as a constrained combinational optimization problem

$$\begin{aligned} \textit{Problem 1:}\ \max\quad & f(\phi) = \mathbf{1}^T \text{sgn}(f_X(\phi)) \\ \text{s.t.}\quad & \phi^T \mathbf{c}_i = 0 \\ & \phi^T \mathbf{1} = m \\ & \phi_k \in \{0, 1\}, k \in V \backslash \{i, j\}. \end{aligned} \tag{5}$$

The first constraint implies that only new links are added, i.e., links are created from the agents who are not yet linked to competitor i, and the other two constraints imply that the number of new links is limited to m.

2) Competitiveness Measured by the Supporting Degree: In the product selling competition, the sales volume decides the profit rather than the number of customers. Here, the mount of products each customer buy can be viewed as his supporting degree to the competitor. Thus, we define the competitiveness of competitor i against competitor j as $\mathbf{1}^T f_X(\phi)$.

Based on this definition of competitiveness, the problem of competitiveness maximization can be formulated as

$$\begin{aligned} \textit{Problem 2:}\ \max\quad & g(\phi) = \mathbf{1}^T f_X(\phi) \\ \text{s.t.}\quad & \phi^T \mathbf{c}_i = 0 \\ & \phi^T \mathbf{1} = m \\ & \phi_k \in \{0, 1\}, k \in V \backslash \{i, j\}. \end{aligned} \tag{6}$$

Note that the only difference between (5) and (6) is whether a sign function is applied to the objective function, which, however, dramatically affects the analysis of the two problems and the algorithms to solve them.

III. Theoretical Analysis

This section contains the main theoretical results in this paper. We first prove that *Problem 1* is NP-hard. Then we show that the objective function of *Problem 2* is monotone and submodular, and thus we can find good approximate solutions by the GA.

A. NP-Hardness

We first show that *Problem 1* is NP-hard.

Theorem 1: Problem 1, i.e., finding a subset T of normal agents who are not connected to competitor i, such that adding links from agents in T to competitor i will maximize the competitiveness (relative number of supporters) of competitor i, is NP-hard.

Proof: We show a reduction from a classical NP-hard problem—the "dense m-subgraph problem" [25], i.e., finding a set of m vertices, in an undirected graph $\tilde{G} = (\tilde{E}, \tilde{V})$, which has the maximum average degree in the subgraph induced by this set. Given an instance of the dense m-subgraph problem, we construct an instance $\mathscr{G}$ of *Problem 1* as follows.

1) Create a node u_k for every node $k \in \tilde{V}$.
2) Create a node v_{kl} for every edge $(k, l) \in \tilde{E}$.
3) Create two nodes p with state $+1$ and q with state -1, representing the two competitors.
4) Create two nodes u and v.
5) Create $3N$ nodes pu_τ, $\tau = 1, 2, \ldots, 3N$, and $7N$ nodes qu_τ, $\tau = 1, 2, \ldots, 7N$ for node u.
6) Create $9N$ nodes pv_τ, $\tau = 1, 2, \ldots, 9N$, and $11N$ nodes qv_τ, $\tau = 1, 2, \ldots, 11N$ for node v.
7) Add a directed link $<pu_\tau, p>$ for every pu_τ.
8) Add a directed link $<qu_\tau, q>$ for every qu_τ.
9) Add a directed link $<pv_\tau, p>$ for every pv_τ.
10) Add a directed link $<qv_\tau, q>$ for every qv_τ.
11) Add directed links $<u, pu_\tau>$, $\tau = 1, 2, \ldots, 3N$ and directed links $<u, qu_\tau>$, $\tau = 1, 2, \ldots, 7N$ for node u.
12) Add directed links $<v, pv_\tau>$, $\tau = 1, 2, \ldots, 9N$ and directed links $<v, qv_\tau>$, $\tau = 1, 2, \ldots, 11N$ for node v.
13) Add a directed link $<u_k, u>$ for every u_k.
14) Add directed links $<v_{kl}, v>$, $<v_{kl}, u_k>$, and $<v_{kl}, u_l>$ for every v_{kl}.

Now, we have already created a new connected graph $\mathscr{G}$ in polynomial time. See Fig. 1 for an illustration of the process of constructing an instance of *Problem 1* from an undirected graph.

The steady state of each normal agent before adding new links can be obtained by (3). The steady states of nodes pu_t ($t = 1, 2, \ldots, 3N$) and pv_t ($t = 1, 2, \ldots, 9N$) are $+1$. The steady states of nodes qu_t ($t = 1, 2, \ldots, 7N$) and qv_t ($t = 1, 2, \ldots, 11N$) are -1. The steady states of node u and nodes u_k, $\forall k$ are -0.4. The steady state of node v is -0.1. The steady states of nodes v_{kl}, $\forall k, l$ are -0.3.

Then, we consider *Problem 1* on this new graph $\mathscr{G}$, i.e., finding a subset of normal agents of size m which are not connected to p, such that adding links from these agents to p will maximize the number of non-negative normal agents. It is easy to see that in a directed network the state change

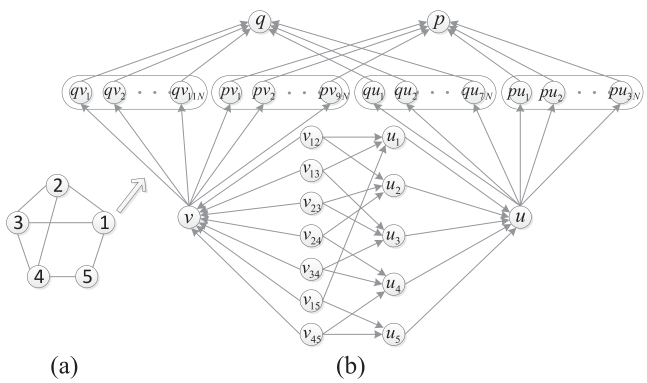


Fig. 1. Illustration of the process of constructing an instance of *Problem 1* from an undirected graph. (a) Simple undirected graph. (b) Resulting graph.

in the upstream nodes will influence the downstream agents, while the change in the downstream nodes will not influence the steady state of the upstream agents. Hence, we start from the upstream agents. At least $4N$ links from nodes of type qu_t to p are needed to make node u's steady state change from negative to non-negative. Since $m < N$, adding m links from nodes of type qu_t will only make these m nodes' steady states change to zero, and increase f by m. For the same reason, adding links from nodes of type qv_t also only increase f by m. The secondary upstream nodes are u and v. Adding a link from u (v) cannot change the sign of its steady state, and hence there is no influence in the sign of the downstream nodes' steady states and no change in f. Adding a link from u_k (or v_{kl}) to p will change the node's state from negative to positive. Then adding m links from nodes of type u_k (or v_{kl}) will at least increase f by $2m$.

The above analysis indicates that it is better to add links from nodes of type u_k or v_{kl} to p. Note that nodes of type v_{kl} are in the downstream of nodes of type u_k. Hence, adding links from nodes of type u_k is optimal. In particular, adding links from nodes of type u_k to p will always increase the steady state of the former from -0.4 to 0.3 and have a total increment of $2m$ on nodes of type u_k, but the increment on the downstream nodes of type v_{kl} will be different from case to case. Therefore, we only need to consider the change of steady state of node of type v_{kl}:

1) from $-3/10$ to $1/6$ and increase f by 2—if both u_k and u_l are in the chosen set;
2) from $-3/10$ to $-1/15$ and no increase in f—if only one of u_k and u_l is in the chosen set;
3) no change and no increase in f—if both u_k and u_l are not in the chosen set.

Hence, to maximize f, we should maximize the number of nodes of type v_{kl}, which have the property that both u_k and u_l are in T. This is a typical dense m-subgraph problem. Since solving *Problem 1* on the created graph $\mathscr{G}$ implies solving the NP-hard problem on the original graph $\tilde{G}$, and we reduce the dense m-subgraph problem to *Problem 1* in polynomial-time, *Problem 1* is NP-hard.

As for *Problem 2*, we conjecture that it is also NP-hard, although we cannot provide a rigorous proof yet. ∎

B. Monotonicity and Submodularity

Since the optimal solution is quite hard to obtain, we try to design algorithms to search for acceptable approximate solutions. We first prove some good properties of objective functions that allow us to construct efficient algorithms. Theorem 2 shows that the objective functions f and g are monotone, and Theorem 3 further shows that g is also submodular. Combining these results, we have the conclusion that, by the GA, we can find very good approximate solutions to *Problem 2*.

Lemma 1: Adding links to competitor i will not decrease the steady state of any normal agent, and adding links to competitor j will not increase the steady state of any normal agent.

Proof: The second statement is a trivial counterpart of the first statement, and hence we only need to prove the first one.

By (3), one has

$$(\bar{D} - \bar{A})\bar{X} = \mathbf{c}_i - \mathbf{c}_j. \tag{7}$$

After adding new links to the original network according to indicator vector ϕ, the new steady-state vector becomes $f_X(\phi)$. By (4), one also has

$$(\bar{D} + \Phi - \bar{A})f_X(\phi) = \mathbf{c}_i + \phi - \mathbf{c}_j. \tag{8}$$

Subtracting (7) from (8) yields

$$(\bar{D} - \bar{A})(f_X(\phi) - \bar{X}) = -\Phi f_X(\phi) + \Phi\mathbf{1} = \Phi(\mathbf{1} - f_X(\phi)).$$

Since $(\bar{D} - \bar{A})$ is nonsingular, we have

$$f_X(\phi) - \bar{X} = (\bar{D} - \bar{A})^{-1}\Phi(\mathbf{1} - f_X(\phi)). \tag{9}$$

$(\bar{D} - \bar{A})$ is also an M-matrix, and thus, $(\bar{D} - \bar{A})^{-1}$ is non-negative [26], [27]. Φ is non-negative by its definition. Furthermore, as stated in [18], all the entries in $f_X(\phi)$ are not greater than one which implies that $\mathbf{1} - f_X(\phi)$ is non-negative. Therefore, $f_X(\phi) - \bar{X}$ is non-negative, which leads to the conclusion that adding links to competitor i will not decrease the steady state of any normal agent. ∎

Then, we introduce the definition of monotonicity and submodularity as preparation for further analysis.

Definition 1 [28]: A set function f is monotonically increasing, if for all set S_1 and S_2 such that $S_1 \subseteq S_2$, one has $f(S_1) \le f(S_2)$.

Definition 2 [28]: A set function f is submodular, if for all sets S_1 and S_2 such that $S_1 \subseteq S_2$ and every $s \notin S_2$, one has $f(S_1 \cup s) - f(S_1) \ge f(S_2 \cup s) - f(S_1)$.

Next, we prove that the objective functions of the proposed two problems are monotonically increasing (but not necessarily strictly increasing), which implies that the objective functions of these two problems will not decrease as we continuously adding new links to competitor i.

Theorem 2: The objective functions $f(\phi)$ in *Problem 1* and $g(\phi)$ in *Problem 2* are monotonically increasing.

Proof: Let W_1 and W_2 be two sets of normal agents with $W_1 \subseteq W_2$. After adding new links from all agents in W_1 to competitor i, where the corresponding indicator vector is denoted by ϕ_1, the steady-state vector of normal agents becomes $f_X(\phi_1)$, and the resulting new network is G_1.

Similarly, for the set W_2, we get ϕ_2, $f_X(\phi_2)$, and G_2. Since $W_1 \subseteq W_2$, network G_2 can be viewed as the result of adding new links from the set $W_2 - W_1$ to competitor i in G_1. Therefore, according to Lemma 1, we have $f_X(\phi_2) \geq f_X(\phi_1)$ in element-wise, and hence $\mathrm{sgn}(f_X(\phi_2)) \geq \mathrm{sgn}(f_X(\phi_1))$ in element-wise. Then $f(\phi_2) - f(\phi_1) = \mathbf{1}^T\mathrm{sgn}(f_X(\phi_2)) - \mathbf{1}^T\mathrm{sgn}(f_X(\phi_1)) \geq 0$, and $g(\phi_2) - g(\phi_1) = \mathbf{1}^T f_X(\phi_2) - \mathbf{1}^T f_X(\phi_1) \geq 0$. Hence, the objective functions $f(\phi)$ and $g(\phi)$ are monotonically increasing.

We next show that the function $g(\phi)$ is submodular. That is to say, for the networks G_1 and G_2 mentioned in Theorem 2, if we add new links from nodes in $\Delta W \subseteq \{k \in V \backslash \{i, j, W_2\}\}$ to competitor i, the increment of competitor i's competitiveness in G_1 is greater than that in G_2. ■

Theorem 3: The function $g(\phi)$ in *Problem 2* is submodular.

Proof: We consider G_1 as the original graph with corresponding $\bar{A}_1$, $\bar{D}_1$, and the steady state is $f_X(\phi_1)$. After adding links from nodes in ΔW to competitor i in G_1, the steady-state vector becomes $f_X(\phi_1 + \Delta\phi)$, where $\Delta\phi$ denotes the indicator vector corresponding to the set ΔW, with $\Delta\Phi = \mathrm{diag}(\Delta\phi)$. By (9), the increment of steady-state vector after adding new links is

$$\begin{aligned}\Delta f_X(\phi_1) &= f_X(\phi_1 + \Delta\phi) - f_X(\phi_1)\\ &= (\bar{D}_1 - \bar{A}_1)^{-1}\Delta\Phi(\mathbf{1} - f_X(\phi_1 + \Delta\phi)). \quad (10)\end{aligned}$$

Similarly, G_2 owns the corresponding $\bar{A}_2$, $\bar{D}_2$. After adding links from ΔW, the increment of steady-state vector is

$$\begin{aligned}\Delta f_X(\phi_2) &= f_X(\phi_2 + \Delta\phi) - f_X(\phi_2)\\ &= (\bar{D}_2 - \bar{A}_2)^{-1}\Delta\Phi(\mathbf{1} - f_X(\phi_2 + \Delta\phi)). \quad (11)\end{aligned}$$

By Lemma 1 and $W_1 \subseteq W_2$, we have $\Delta f_X(\phi_2 + \Delta\phi) \geq \Delta f_X(\phi_1 + \Delta\phi)$. Subtracting (10) from (11) yields

$$\begin{aligned}\Delta f_X(\phi_2) &- \Delta f_X(\phi_1)\\ &\leq \left[(\bar{D}_2 - \bar{A}_2)^{-1} - (\bar{D}_1 - \bar{A}_1)^{-1}\right]\Delta\Phi(\mathbf{1} - f_X(\phi_1 + \Delta\phi))\\ &= -(\bar{D}_2 - \bar{A}_2)^{-1}(\bar{D}_2 - \bar{A}_2 - \bar{D}_1 + \bar{A}_1)(\bar{D}_1 - \bar{A}_1)^{-1}\\ &\quad \times \Delta\Phi(\mathbf{1} - f_X(\phi_1 + \Delta\phi)).\end{aligned}$$

Since $W_1 \subseteq W_2$, the network G_2 can be viewed as the result of adding new links from set $W_2 - W_1$ to competitor i in the network G_1. Then, $\bar{A}_2 = \bar{A}_1$ and $(\bar{D}_2 - \bar{D}_1) \geq 0$. Hence, we have

$$\begin{aligned}\Delta f_X(\phi_2) - \Delta f_X(\phi_1) &\leq -(\bar{D}_2 - \bar{A}_2)^{-1}(\bar{D}_2 - \bar{D}_1)\\ &\quad \times (\bar{D}_1 - \bar{A}_1)^{-1}\Delta\Phi(\mathbf{1} - f_X(\phi_1 + \Delta\phi)).\end{aligned}$$

Since $(\bar{D}_1 - \bar{A}_1)^{-1}$ and $(\bar{D}_2 - \bar{A}_2)^{-1}$ are non-negative, $\Delta\Phi \geq 0$, and $\mathbf{1} - f_X(\phi_1 + \Delta\phi) \geq 0$, we have $\Delta f_X(\phi_2) - \Delta f_X(\phi_1) \leq 0$ in element-wise.

The increment of competitor i's competitiveness in G_1 is denoted by

$$\begin{aligned}\Delta g(\phi_1) &= g(\phi_1 + \Delta\phi) - g(\phi_1)\\ &= \mathbf{1}^T f_X(\phi_1 + \Delta\phi) - \mathbf{1}^T f_X(\phi_1) = \mathbf{1}^T \Delta f_X(\phi_1).\end{aligned}$$

Similarly, we have $\Delta g(\phi_2) = \mathbf{1}^T \Delta f_X(\phi_2)$. Then

$$\Delta g(\phi_2) - \Delta g(\phi_1) = \mathbf{1}^T(\Delta f_X(\phi_2) - \Delta f_X(\phi_1)) \leq 0.$$

Hence, the function $g(\phi)$ is submodular. ■

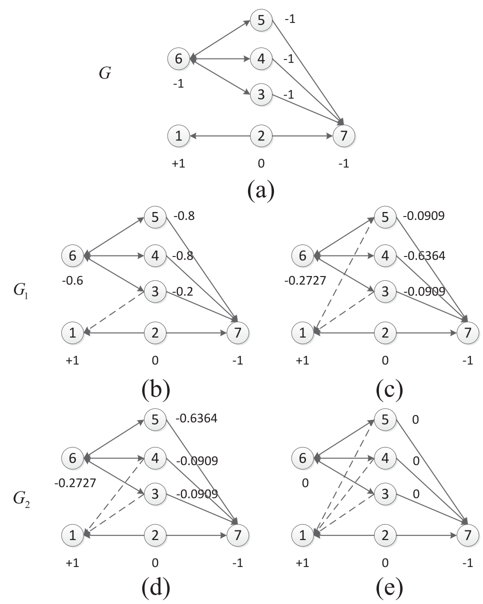


Fig. 2. Counterexample to demonstrate that $f(\phi)$ is not submodular. (a) The original graph G. (b) New link $< 3, 1 >$ is added. (c) New links $< 3, 1 >$ and $< 5, 1 >$ are added. (d) New links $< 3, 1 >$ and $< 4, 1 >$ are added. (e) New links $< 3, 1 >$, $< 4, 1 >$, and $< 5, 1 >$ are added.

Remark 1: The function $f(\phi)$ in *Problem 1* is not submodular. We provide a counterexample to demonstrate this property. As shown in Fig. 2, G is the original network, with two competitors $x_1 \equiv +1$ and $x_7 \equiv -1$. By adding links from $W_1 = \{3\}$ to competitor 1, we get network G_1 with $f(\phi_1) = -4$ shown in Fig. 2(b). If adding new links from $W_2 = \{3, 4\}$, we get G_2 with $f(\phi) = -4$ shown in Fig. 2(d). Let $\Delta W = \{5\}$. After adding links to 1 in G_1 we get network Fig. 2(c) with $f(\phi_1 + \Delta\phi) = -4$. Similarly, by adding links to competitor 1 in G_2 we get network Fig. 2(e) with $f(\phi_2 + \Delta\phi) = 0$. The fact that $\Delta f(\phi_2) > \Delta f(\phi_1)$ is inconsistent with the definition of submodular, and therefore function $f(\phi)$ is not submodular.

Remark 2: Since the objective function $g(\phi)$ is monotone and submodular, we can find a $(1 - 1/e)$-approximation solution to *Problem 2* by designing a proper GA [29]. That is to say, if $\bar{f}$ is the solution obtained by the GA and f^* is the best solution, then $\bar{f} \geq (1 - 1/e)f^*$. Since the objective function $f(\phi)$ in *Problem 1* is not submodular, we conjecture that the GA may not work well for it, which will be checked in Section V.

IV. Algorithm Design

We provide two categories of algorithms. The first one is the GA which has been proved to be a commendable approximation algorithm for *Problem 2*. The other category consists of several heuristic algorithms, which are simply constructed based on network centralities, and hence timesaving when confronting large-scale networks.

A. Greedy Algorithm

The GA searches a solution set T as follows. Initially, the solution set is empty, denoted by T^0, and the set of nodes which are not connected to competitor i is denoted by S^0. At the tth iteration ($t > 0$), the algorithm extends the solution set T^{t-1} by adding the node $k \in S^{t-1}$ which maximizes $\mathrm{Func}(\phi^t)$,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

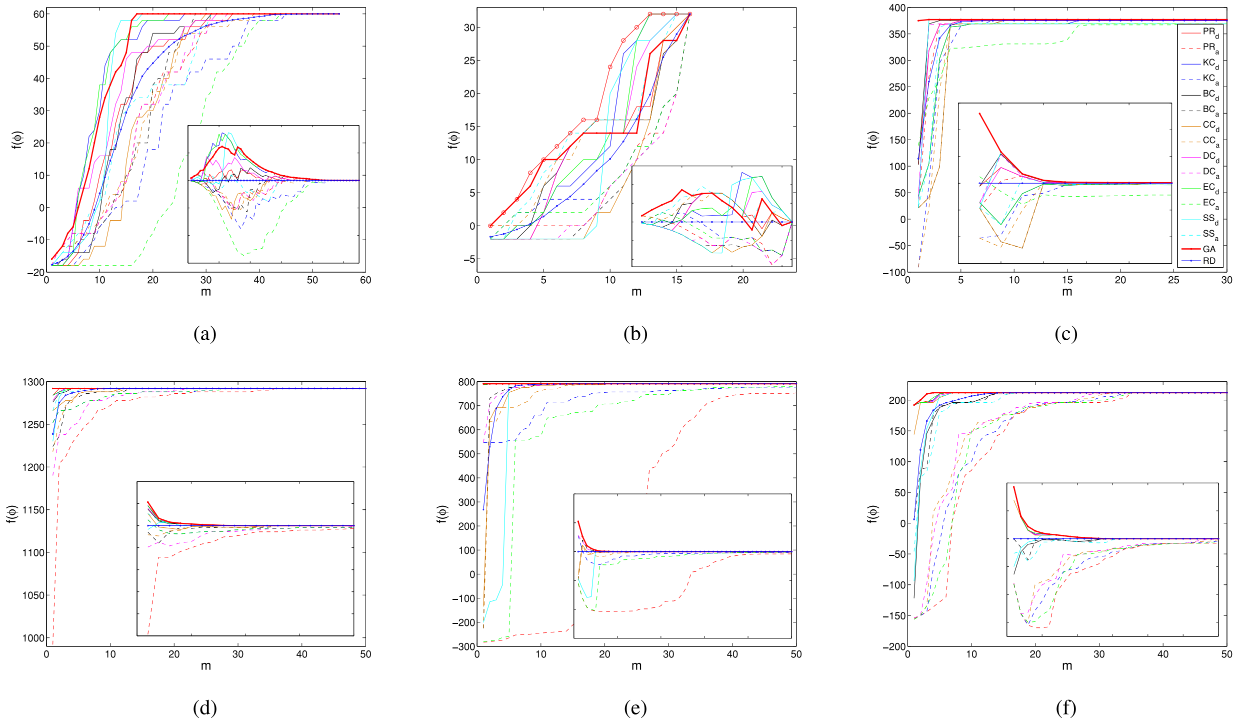


Fig. 3. Performances of algorithms in solving *Problem 1* on six real social networks. The inset shows the performance difference between RD and each other algorithm. (a) Dolphin (undirected). (b) Karate (undirected). (c) Netsci (undirected). (d) Online (directed). (e) Political blogs (directed). (f) Residence (directed).

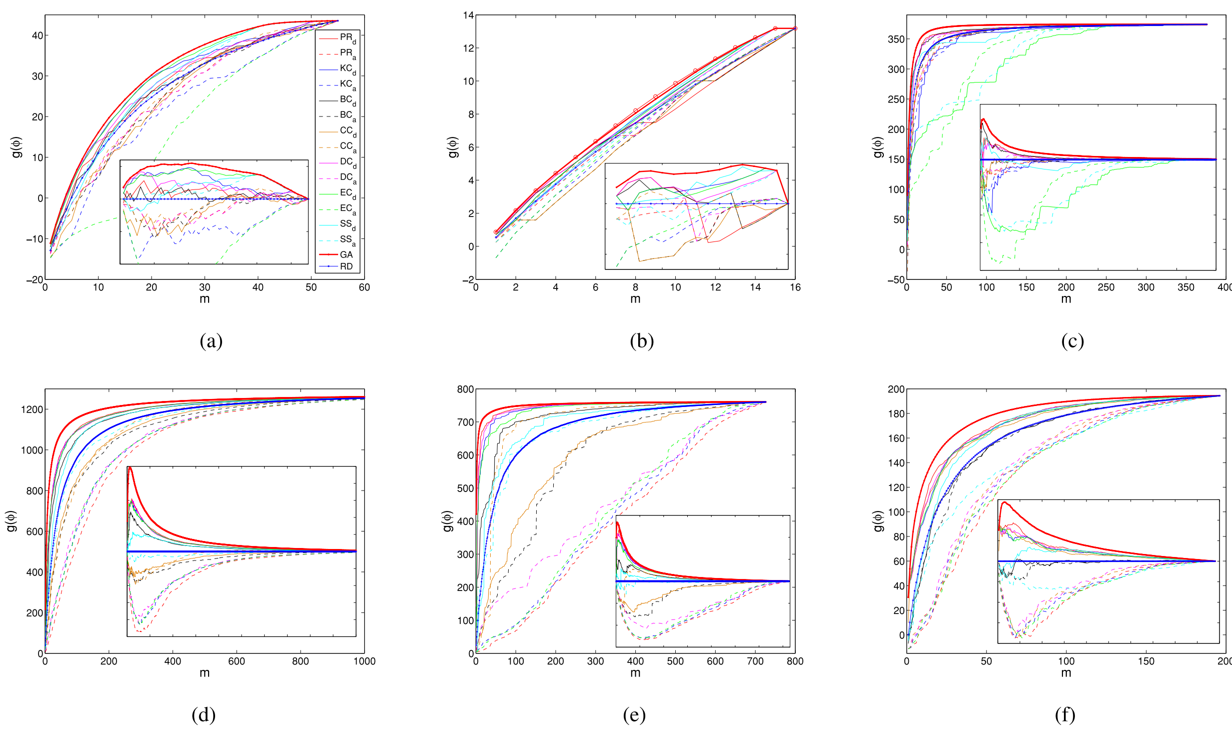


Fig. 4. Performances of algorithms in solving *Problem 2* on six real social networks. The inset shows the performance difference between RD and each other algorithm. (a) Dolphin (undirected). (b) Karate (undirected). (c) Netsci (undirected). (d) Online (directed). (e) Political blogs (directed). (f) Residence (directed).

where ϕ^t is the indicator vector corresponding to set T^t and Func$(\cdot)$ is the objective function of the considered problem, i.e., $f(\cdot)$ in *Problem 1* or $g(\cdot)$ in *Problem 2*, respectively.

B. Heuristic Algorithms

Although the GA is faster than exhaustive enumeration, it is still time-consuming when facing large-scale networks, because we have to compute the objective function (involving matrix inversion) for any single candidate normal agent, add the optimal agent to the solution set, and repeat the process until the whole solution set is constructed. Hence, we next try to construct several heuristic algorithms based on centrality measures to reduce the computational burden. Though some of the centrality measures are also relatively time-consuming, we only need to compute them once in the whole solution seeking process, and hence, their computational burden is much lower than the GA.

Over the years, a number of centrality measures have been proposed to characterize the "importance" of a node in a network. However, one difficulty in applying these centrality measures is that it is often unclear which of the many measures should be used in a particular circumstance. We consider and compare the following six commonly used centrality measures: 1) PageRank value (PR); 2) Katz centrality (KC); 3) betweenness centrality (BC); 4) closeness centrality (CC); 5) degree centrality (DC); and 6) eigenvector centrality (EC), which are shown to be good indicators in predicting the winner of the competition [18]. The heuristic algorithms are designed as follows. Let $S = \{k| < k, i >\notin E\}$ denotes the set of agents which are not connected to competitor i. Then a solution set T is constructed by selecting m agents in set S according to a certain centrality measure. Note that, all the new links are added simultaneously rather than one by one. Thus, there is no need to recalculate the centrality measures. Since there is no obvious evidence whether adding a link from an agent of higher centrality value is better than from an agent of lower centrality value, for each centrality measure, we add links from normal agents both in descending order and ascending order. For the descending order, we get heuristic algorithms PR_d, KC_d, BC_d, CC_d, DC_d, and EC_d. Similarly, for the ascending order, we get heuristic algorithms PR_a, KC_a, BC_a, CC_a, DC_a, and EC_a.

Besides the centrality measures, we also use the agent's steady state to capture the normal agents' importance. Hence, we further propose two more heuristic algorithms based on normal agents' steady states from two aspects: 1) selecting strong negative states first (adding the rival's strong supporters first)—SS_d and 2) selecting weak negative states first (adding the rival's weak supporters first)—SS_a.

At the same time, for comparison purpose, we provide a random link-adding algorithm named as RD. In the RD algorithm, when the number of new links m is given, we randomly add m links from candidate normal agents to competitor i. The result is averaged over 100 realizations.

V. Simulations

Simulations are provided in this section to demonstrate the performances of all the proposed algorithms. Since the competing scenario considered in this paper mainly happens in social networks, we choose real-world social networks of moderate size for simulations. Networks such as neuronal networks, electronic circuits, and regulatory networks, where competitions rarely happen, are temporarily out of consideration. We select six networks of different size, including three undirected networks and three directed networks (see Table I). To guarantee the convergence of states of normal agents in our model, we choose the largest (strongly) connected component of each network, the size of which is shown in Table I.

TABLE I

Properties of Six Real Social Networks. The Number of Agents and the Number of Links Are Denoted by N and M, Respectively. Agents i and j Are One Pair of Randomly Chosen Competitors

Style	Name	N	M	i	j
undirected	dolphin [30]	62	159	8	15
	karate [31]	34	78	1	34
	netsci [32]	379	914	29	54
directed	online [33]	1294	19026	158	1252
	political blogs [34]	793	15781	193	335
	residence [35]	215	2658	59	76

A. Comparison Among All the Proposed Algorithms

The performance of a given algorithm can be captured by the values of f and g resulted by running the algorithm as the number of links m varies. As a benchmark, the brute-force method is also adopted for searching the optimal solutions to the problems. Considering the computational burden of the brute-force method, we only use it in the smallest network: Zachary's karate club network (the red line with circles). The comparisons of algorithms in solving *Problems 1* and *2* under a pair of randomly chosen competitors are shown in Figs. 3 and 4, respectively.

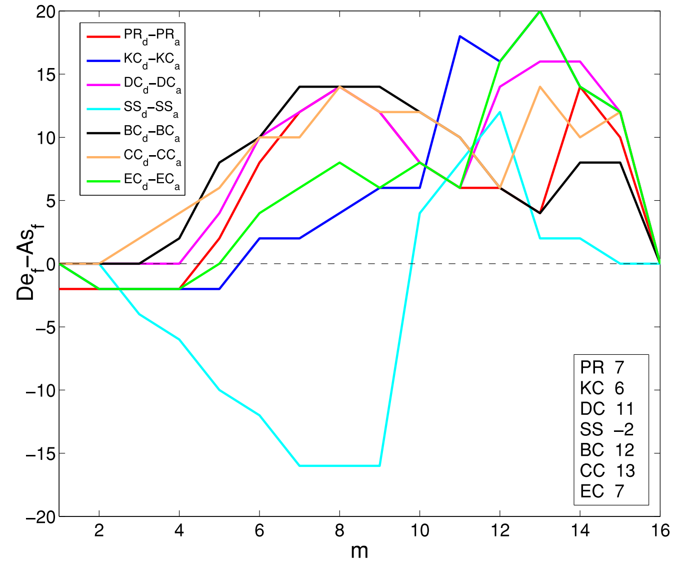
As shown in Fig. 4, the GA has the best performance in all networks. Especially, in Fig. 4(b) the results of the GA are quite close to, sometime even equal to, the ones obtained by the brute-force method for different values of m. This is consistent with our theoretical result in Section III that the GA can provide a very good approximate solution to *Problem 2*. It is also shown in Fig. 4 that some heuristic algorithms have almost the same performances as the GA. This result suggests that, if we are confronted with an extremely large network, those timesaving heuristic algorithms might be good choices to solve *Problem 2*. For *Problem 1*, as shown in Fig. 3, the GA performs better than other algorithms in directed networks, but no single algorithm performs the best in undirected networks. The poor performance might result from the nonsubmodularity of *Problem 1*. For both problems the curves of heuristic algorithms go across each other as the value of m and the network topology change. Therefore, we could not say which centrality measure should be used for a particular network. Comprehensively understanding the relationship among these factors will be a goal in our future work.

To further compare the heuristic algorithms with the RD algorithm, we use the results of RD as the benchmark, and by subtracting the results of RD from the results of other algorithms we get those insets in Figs. 3 and 4. Clearly, a curve above the horizontal zero line implies that the corresponding algorithm is better than the RD. The simulation shows that descending order heuristic algorithms perform better than the RD algorithm in most cases. This may be interpreted as that the descending-order heuristic algorithms are better than their ascending-order counterparts. But it is not always true. In Figs. 3 and 4, there are also some dashed-lines (representing the algorithms in ascending order) above the horizontal zero line, while the solid lines of the same colors are under horizontal zero line. Which order is better in the statistical sense? This question inspires us to further study the heuristic algorithms in the next section.

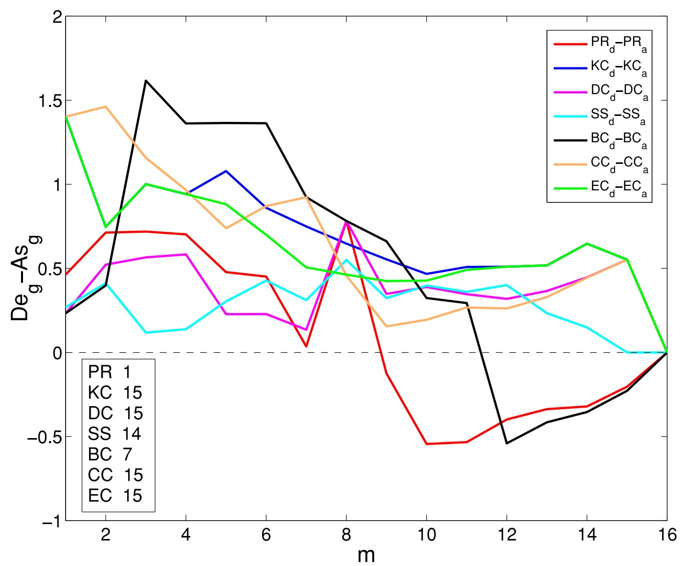
B. Comparison Between the Heuristic Algorithms

In existing works on influence maximization (e.g., [20] and [24]), heuristic algorithms based on centrality measures were always in descending order. That is to say, influential agents are the first choice to affect. However, it is also plausible to argue that an agent of low centrality might be much easier to be affected and then becomes a support of one competitor than an influential hub agent, and thus, we should choose agents according to a centrality measure in ascending order. We are curious about which order is better. To further investigate this problem, for each given m we subtract the result of a given ascending algorithm from the result of the corresponding descending algorithm. If the difference value is positive (negative), we say that the heuristic algorithm in the descending (ascending) order is better under the given value of m. More specifically, let De_f (De_g) denote the vector composed by the results under different numbers of m in solving *Problem 1* (*Problem 2*) by the algorithm based on a given centrality measure in descending order. Similarly, let As_f (As_g) denote the vector composed by the results under different number of m in solving *Problem 1* (*Problem 2*) by the algorithm based on a specific centrality measure in ascending order. Then, $(\mathrm{De}_f - \mathrm{As}_f)_l > 0$ ($(\mathrm{De}_g - \mathrm{As}_g)_l > 0$) means heuristic algorithm based on this centrality measure in the descending (ascending) order is better when $m = l$, where $(\mathrm{De}_f - \mathrm{As}_f)_l$ represents the lth element in vector $\mathrm{De}_f - \mathrm{As}_f$. We report the difference values of each centrality measure as a function of the number of links m in karate club network with competitors 1 and 34 (see Fig. 5). We can see for most centrality measures the difference values in most values of m are positive.

To evaluate the performance of the proposed algorithms under all possible values of m, we define the following parameter: $P_f = \mathbf{1}^T \mathrm{sign}(\mathrm{De}_f - \mathrm{As}_f)$ ($P_g = \mathbf{1}^T \mathrm{sign}(\mathrm{De}_g - \mathrm{As}_g)$) for *Problem 1* (*Problem 2*). For a given centrality measure, if P_f (P_g) is positive (negative), we say that heuristic algorithm in the descending (ascending) order is better under a given pair of competitors for *Problem 1* (*Problem 2*). The values of P_f or P_g for all centrality measures are shown in the text boxes inserted in Fig. 5. We can see that for *Problem 1* all the values of P_f are positive except the one of SS, and for *Problem 2* all the values of P_g are positive, which suggests that for most centrality measures the proposed heuristic algorithms constructed in descending order perform better than their ascending-order counterparts in solving both problems on the karate club network with agents 1 and 34 as the competitor i and j.



(a)



(b)

Fig. 5. Difference values obtained by descending algorithms minus ascending algorithms in solving (a) *Problem 1* and (b) *Problem 2* on the karate club network.

To see whether the above result still holds for different pairs of competitors in all networks considered here, we randomly choose 1000 pairs of competitors in each network (for karate network which has less than 1000 pairs of competitors we choose all of them). We compute P_f and P_g for each pair of competitors and each heuristic algorithm, and obtain the percentage of positive values, negative values, and zeros.

In Fig. 6, though the percentage of $P_f > 0$ is greater than 50% for all centrality measures in all networks, the percentage of $P_f < 0$ cannot be ignored. This phenomenon means that the heuristic algorithms in descending order are not always better than their ascending-order counterpart in solving

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

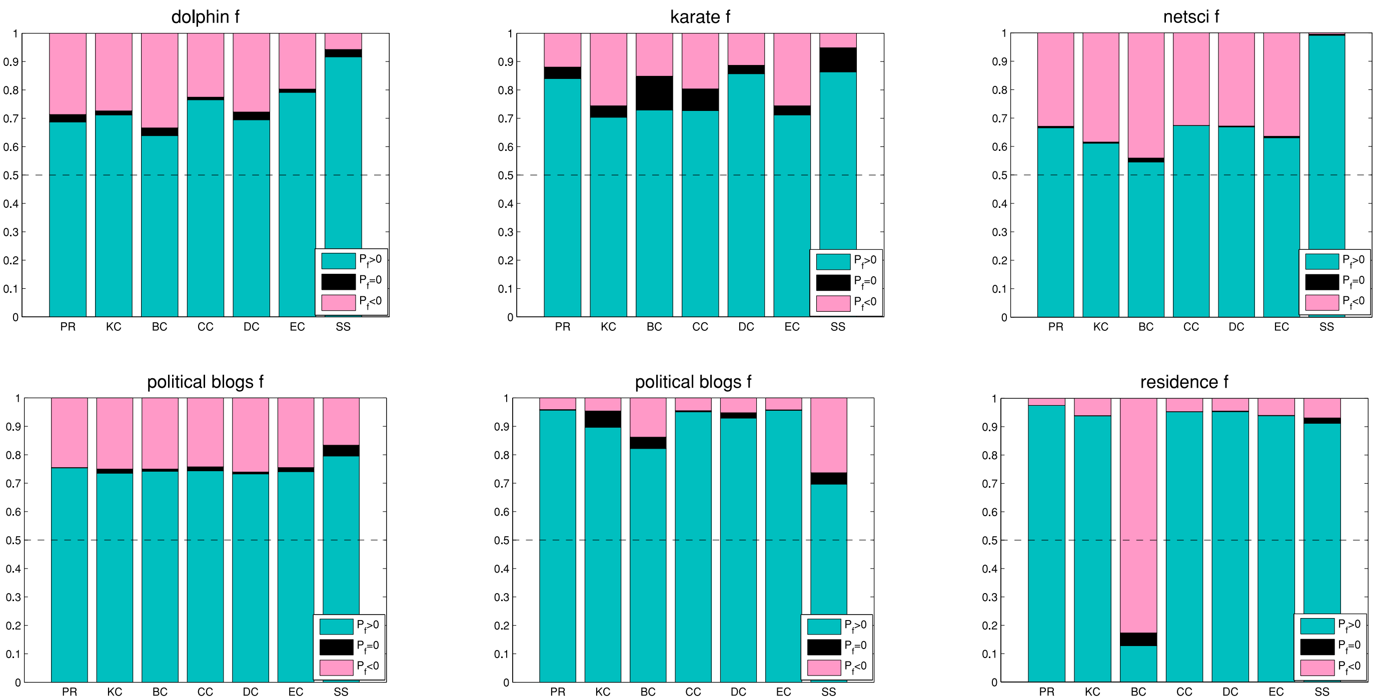


Fig. 6. Comparison between heuristic algorithms in solving *Problem 1* on six real social networks.

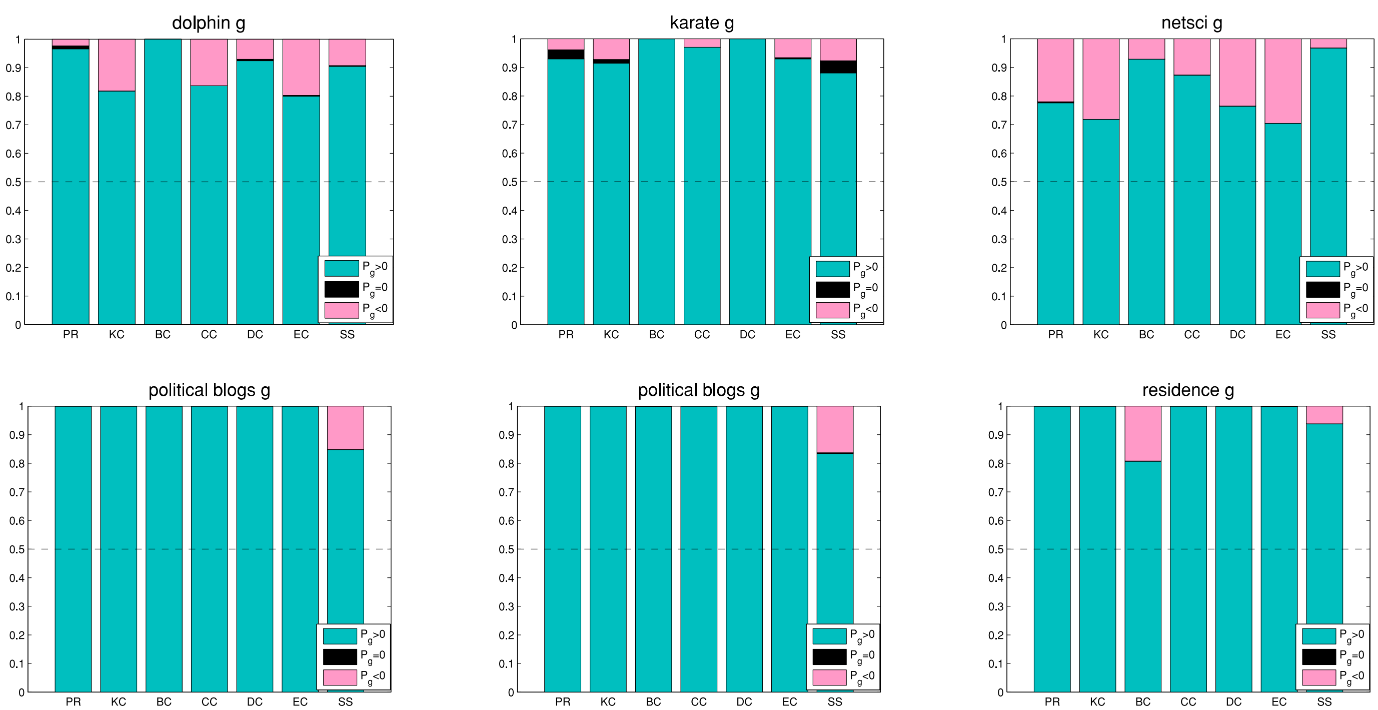


Fig. 7. Comparison between heuristic algorithms in solving *Problem 2* on six real social networks.

Problem 1, which is a surprising result regarding the works in [20] and [24].

In contrast, as shown in Fig. 7, the percentage of $P_g > 0$ is much greater than the percentage of $P_g < 0$ for all networks and all algorithms. Especially in the directed networks the percentage of $P_g > 0$ is close to 100%. This means that the descending order heuristic algorithms is very suitable in solving *Problem 2* in these networks.

VI. Conclusion

In this paper, we investigate how to enhance a competitor's competitiveness by adding new links in networks. We propose two competitiveness maximization problems: *Problem 1* aims to maximize the number of supporters of a competitor, while *Problem 2* tries to maximize the supporting degree toward the competitor. We prove that *Problem 1* is NP-hard. The objective function of *Problem 2* is monotonous and submodular, and hence it can be approximatively solved by the GA in polynomial time. We also construct several heuristic algorithms based on centrality measures for the purpose of reducing computing complexity. By simulations, we find that no single algorithm performs the best in all networks for *Problem 1*, while the GA always provides the best approximate solution in directed networks to *Problem 1* and in all networks to *Problem 2*. Furthermore, among those centrality-based heuristic algorithms, an algorithm based on centrality in descending order is always better then its counterpart in ascending order for *Problem 2* in statistical sense. But for *Problem 1* the performances of the heuristic algorithms in descending order is more sensitive to the network structure and the locations of competitors. This phenomenon may result from the nonsubmodularity of *Problem 1*.

This paper can be extended in several aspects. First, the problem discussed in this paper is elementary in the sense that we only consider one active competitor and leave the other one fixed. As a future research direction, we will consider the situation where both of the competitors have the opportunity to add new links to maximize their competitiveness. In the situation where two competitors add new links in turn, each competitor in its own turn is in fact facing the exact same situation considered in this paper, and thus, could adopt the proposed algorithms here. Another interesting situation is that two competitors move simultaneously, in which case each competitor needs to consider its opponent's possible actions, and thus, game theory is more suitable for the study. Second, in this paper, there is an inherent assumption that the competitor has the knowledge about the entire network topology. In the situation where the network topology is hard to obtain, new algorithms dealing with imperfect topological information should be developed. Third, we only consider two competitors here, as for multicompetitor scenario, the basic settings of the competition problem should be modified. The multiple competitors fixed state may be located in some places inside the interval $[-1, +1]$, rather than holding the two extreme values. An alternative method is extending the state to be multidimensional and keeping the states of competitors at peaks of a regular polygon. A competitor's supporters are those normal agents whose states are closest to this competitor. At last but not least, the cost of adding new links can be considered such that the optimal number of new links should be investigate to achieve a tradeoff between the cost of adding new links and the benefit of higher competitiveness.

References

[1] I. J. Perez, F. J. Cabrerizo, S. Alonso, and E. Herrera-Viedma, "A new consensus model for group decision making problems with non-homogeneous experts," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 4, pp. 494–498, Apr. 2014.

[2] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," *Nature*, vol. 433, no. 7025, pp. 513–516, 2005.

[3] H. M. La, W. Sheng, and J. Chen, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 1–12, Jan. 2015.

[4] C. Altafini, "Consensus problems on networks with antagonistic interactions," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 935–946, Apr. 2013.

[5] 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.

[6] M. Patriarca, X. Castelló, J. R. Uriarte, V. M. Eguíluz, and M. S. Miguel, "Modeling two-language competition dynamics," *Adv. Complex Syst.*, vol. 15, nos. 3–4, 2012, Art. no. 1250048.

[7] Z. He, T. C. E. Cheng, J. Dong, and S. Wang, "Evolutionary location and pricing strategies in competitive hierarchical distribution systems: A spatial agent-based model," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 7, pp. 822–833, Jul. 2014.

[8] A. Chatterjee, M. Mitrović, and S. Fortunato, "Universality in voting behavior: An empirical analysis," *Sci. Rep.*, vol. 3, p. 1049, Jan. 2013.

[9] Y. Yang *et al.*, "A budget optimization framework for search advertisements across markets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 5, pp. 1141–1151, Sep. 2012.

[10] M. Mobilia, A. Petersen, and S. Redner, "On the role of zealotry in the voter model," *J. Stat. Mech. Theory Exp.*, vol. 2007, no. 8, 2007, Art. no. P08029.

[11] A. Borodin, Y. Filmus, and J. Oren, "Threshold models for competitive influence in social networks," in *Internet and Network Economics*. Heidelberg, Germany: Springer, 2010, pp. 539–550.

[12] K. R. Apt and E. Markakis, "Diffusion in social networks with competing products," in *Algorithmic Game Theory*. Heidelberg, Germany: Springer, 2011, pp. 212–223.

[13] A. Fazeli and A. Jadbabaie, "Game theoretic analysis of a strategic model of competitive contagion and product adoption in social networks," in *Proc. IEEE 51st Annu. Conf. Decis. Control (CDC)*, 2012, pp. 74–79.

[14] F. Uekermann and K. Sneppen, "Spreading of multiple epidemics with cross immunization," *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, vol. 86, no. 3, 2012, Art. no. 036108.

[15] A. Beutel, B. A. Prakash, R. Rosenfeld, and C. Faloutsos, "Interacting viruses in networks: Can both survive?" in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Beijing, China, 2012, pp. 426–434.

[16] B. A. Prakash, A. Beutel, R. Rosenfeld, and C. Faloutsos, "Winner takes all: Competing viruses or ideas on fair-play networks," in *Proc. 21st Int. Conf. World Wide Web*, Lyon, France, 2012, pp. 1037–1046.

[17] 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.

[18] J. Zhao, Q. Liu, and X. Wang, "Competitive dynamics on complex networks," *Sci. Rep.*, vol. 4, p. 5858, May 2014.

[19] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Washington, DC, USA, 2003, pp. 137–146.

[20] A. Gionis, E. Terzi, and P. Tsaparas, "Opinion maximization in social networks," in *Proc. SIAM Int. Conf. Data Min.*, Austin, TX, USA, 2013, pp. 387–395.

[21] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, pp. 65–68, 2015.

[22] F. Lin, M. Fardad, and M. R. Jovanović, "Algorithms for leader selection in stochastically forced consensus networks," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1789–1802, Jul. 2014.

[23] Z. Yu, D. Zhang, Z. Yu, and D. Yang, "Participant selection for offline event marketing leveraging location-based social networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 6, pp. 853–864, Jun. 2015.

[24] T. Carnes, C. Nagarajan, S. M. Wild, and A. Van Zuylen, "Maximizing influence in a competitive social network: A follower's perspective," in *Proc. 9th Int. Conf. Elect. Commer.*, Minneapolis, MN, USA, 2007, pp. 351–360.

[25] U. Feige, D. Peleg, and G. Kortsarz, "The dense K-subgraph problem," *Algorithmica*, vol. 29, no. 3, pp. 410–421, 2001.

[26] W. Ren and R. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control: Theory and Applications*. London, U.K.: Springer, 2007.

[27] G. Poole and T. Boullion, "A survey on M-matrices," *SIAM Rev.*, vol. 16, no. 4, pp. 419–427, 1974.

[28] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*. Heidelberg, Germany: Springer, 2003.

[29] 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.

[30] D. Lusseau *et al.*, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405, 2003.

[31] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, 1977.
[32] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, vol. 74, no. 3, 2006, Art. no. 036104.
[33] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Soc. Netw.*, vol. 31, no. 2, pp. 155–163, 2009.
[34] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *Proc. 3rd Int. Workshop Link Disc.*, Chicago, IL, USA, 2005, pp. 36–43.
[35] L. C. Freeman, C. M. Webster, and D. M. Kirke, "Exploring social structure using dynamic three-dimensional color images," *Soc. Netw.*, vol. 20, no. 2, pp. 109–118, 1998.

Jiuhua Zhao was born in Taizhou, China, in 1988. She received the B.S. degree in control theory and control engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2010. She is currently pursuing the Ph.D. degree in control theory and control engineering with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China.

Her research interests include the competitive dynamics on complex networks, social learning, design and analysis of algorithms, and controllability of complex networks.

Qipeng Liu received the B.S. degree in automation from Jinan University, Guangzhou, China, in 2007, the M.S. degree in control theory and control engineering from the Jiangsu University of Science and Technology, Zhenjiang, China, in 2009, and the Ph.D. degree in control theory and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014.

Since 2014, he has been an Assistant Professor with the Institute of Complexity Science, Qingdao University, Qingdao, China. His current research interests include consensus and information aggregation in social networks and cooperative control of multiagent systems.

Lin Wang (M'11) received the B.S. and M.S. degrees from the School of Mathematical Sciences, Shandong Normal University, Jinan, China, in 2003 and 2006, respectively, and the Ph.D. degree in operations research and control theory from the Academy of Mathematics and Systems Science, CAS, Beijing, China, in 2009.

She is currently an Associate Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. Her research interests include analysis and control of multiagent systems, adaptive complex networks, and coordination of multiple manipulators.

Xiaofan Wang (SM'05) received the Ph.D. degree from Southeast University, Nanjing, China, in 1996.

He has been a Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, since 2002 and a Distinguished Professor, since 2008. He has coauthored four books and over 70 papers. His current research interest includes analysis and control of complex dynamical networks.

Dr. Wang was a recipient of the 2002 National Science Foundation for Distinguished Young Scholars of China, the 2005 Guillemin–Cauer Best Transactions Paper Award from the IEEE Circuits and Systems Society, the 2008 Distinguished Professor of the Chang Jiang Scholars Program, Ministry of Education, and the 2010 Peony Prize for Natural Science Researchers in Shanghai. He is the Chair of the IFAC Technical Committee on Large-Scale Complex Systems.