Systems & Control Letters 147 (2021) 104842

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Node and network resistance to bribery in multi-agent systems☆

Guilherme Ramos [a,*], Daniel Silvestre [b,c], Carlos Silvestre [b,c]

[a] Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Portugal
[b] Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, China
[c] Institute for Systems and Robotics, Instituto Superior Técnico, University of Lisbon, Portugal

ARTICLE INFO

Article history:
Received 19 August 2020
Received in revised form 28 October 2020
Accepted 22 November 2020
Available online 7 December 2020

Keywords:
Bribery resistance
Multi-agent systems
Consensus

ABSTRACT

In this paper, we propose a framework to study the resistance to bribery of nodes in a network via average consensus. We extend the proposed bribery resistance measure to sets of nodes, and networks. The proposed framework evaluates quantitatively how much an external entity needs to drive the state of an agent away from its current state, to change the final consensus value. Subsequently, we illustrate our framework with a set of examples, namely: i) how we can use it to compute the bribing resistance of each node in a network; ii) comparing our measure against metrics from the literature in measuring network bribing resistance; iii) how we may utilize the proposed framework to evaluate the bribing resistance of clusters/groups of nodes in large-scale networks.

1. Introduction

In the last decades, the cyber-security aspects of networked control systems have been emphasized by the research community to provide services with resilience guarantees. In tandem with the growing use of general-purpose networks, malicious entities are paying increasing attention to these sometimes critical networks and are profiting from attacking them. An example of such malicious entities attacks is the Stuxnet incident [1], caused by a malicious computer worm. Other well-known instances are the power system cyber-attacks that occurred in Ukraine [2]. Consequently, the research community has been developing methods to improve the resilience of cyber–physical systems to overcome such abnormal situations. These methods have a central role, because such circumstances may lead the systems to critical stages, which can be irrecoverable or may even, in the worst-case scenario, result in a cyber-war.

Multi-agent systems [3] are a particular case of cyber–physical systems, and they are computerized systems composed of multiple interacting intelligent agents. These systems can solve problems that are hard or impossible for a single agent to resolve. In other words, multi-agent systems may be seen as networks of dynamic agents aiming to achieve a global intent employing local interactions. These systems consist of a research line of paramount importance.

A nuclear problem in multi-agent systems is when all agents in a network seek to agree on a shared value, known as *consensus*. The problem of consensus has numerous applications, making it a research subject of particular interest. For instance, it emerges in areas like distributed optimization [4,5]; tasks of motion coordination, e.g. leader following [6]; rendezvous problems [7]; resource allocation in computer networks [8]; and even in computing relative importance of webpages in the PageRank algorithm [9].

A particular type of consensus problem is the *average consensus*, i.e. designing a linear distributed iterative algorithm that leads agents to compute the arithmetic average of their initial states. This problem has a plethora of applications. It has been studied for both the cases of undirected [6] and directed [10] networks. Further, we may also classify the consensus problems according to the time update domain as discrete-time [11], or as continuous-time [12]. The time-domain can even be hybrid as studied in [13], where resilience consensus is proposed for systems composed of multiple dynamical agents governed by both continuous-time and discrete-time control laws. Further, the network communication between nodes may be synchronous [11], or asynchronous [4,14] and the communication may occur deterministically [15], or stochastically [16]. Lastly, the communication network can be static [15], or dynamic [11].

The growing use of consensus algorithms praises the need to deal with both faults and attacks. In other words, it is crucial to ensure that consensus methods are resilient to agents that

☆ G. Ramos acknowledges the support of Institute for Systems and Robotics (ISR), Instituto Superior Técnico, University of Lisbon, Portugal, through scholarship BL112/2019. This work was partially supported by project MYRG2018-00198-FST of the University of Macau, China, by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through ISR, under Laboratory for Robotics and Engineering Systems (LARSyS) project UIDB/50009/2020 and by FCT, Portugal project POCI-01-0145-FEDER-031411-HARMONY.

* Corresponding author.
E-mail addresses: gramos@fe.up.pt (G. Ramos), dsilvestre@isr.ist.utl.pt (D. Silvestre), csilvestre@umac.mo (C. Silvestre).

https://doi.org/10.1016/j.sysconle.2020.104842
0167-6911/

behave abnormally. In [17], a fault-tolerant algorithm to perform approximate Byzantine consensus in asynchronous networks is introduced. The algorithm requires a topological condition to be successfully used, which is less restrictive than the standard requirements. Their work also applies to synchronous networks and networks with delay on the communication paths, and the authors extended the results to systems with a time-varying underlying graph. In [18], the problem of resilient consensus of sampled-data multi-agent networks with double-integrator dynamics is studied. Under the assumption that we know a bound on the number of malicious agents, a resilient consensus method is proposed by making each agent discard a set neighbor values, which are large and small states.

The work in [19] investigates consensus and clustering of expressed and private opinions against Byzantine individuals in directed and time-varying networks. The author outlines a distributed censoring algorithm to rule the opinion dynamics of private and expressed opinions and renders necessary and sufficient conditions for resilient consensus and clustering based on network robustness.

Much work has been devoted in trying to mitigate the effects of possible nodes being attacked. To help better identify the nodes of a network that are easier (with a smaller cost) to attack and for which, when designing a network, one has to pay special attention to, we develop a measure to quantitatively assess how resilient to being attacked a node in a network is. The proposed methodology may be used to prevent attacks and to identify nodes or sets of nodes that may strategically be more protected due to its criticality in a network. Graph resilience has a plethora of application areas, such as consensus methods, power grids, brain connectivity, social networks, geolocation services, digital multimedia content generation and delivery, argument graphs, logistics and supply chain management, computer, and data communication networks, and chemical engineering. The concept of graph resilience quantifies the ability to find alternative, though generally more costly, paths when edges or nodes with their incident edges are deleted from the graph. In this paper, we explore a novel perspective inspired by average consensus. The proposed view translates in a measure of how easy it is to influence the entire network by changing a particular node value.

In [20], a graph measure that the authors called effective graph resistance, derived from the field of electric circuit analysis, was proposed. They define the effective graph resistance as the accumulated effective resistance between all pairs of vertices. The measure the authors proposed only aims to evaluate the resistance of a graph qualitatively. In contrast, we present, in this paper, a measure that can be computed for a node, a set of nodes, and also used to calculate the resistance of the graph. In [21], the authors proposed a new, generic, and scalable graph resilience metric to the edge removal scenario, relying on the weighted sum of the number of paths that cross specific vertices of extensive communication and structural value, is proposed. Also, in [16], an in-depth study of the design and analysis of gossip algorithms for averaging in an arbitrarily connected network of nodes is studied. The authors show that the averaging time needed to reach consensus depends on the second largest eigenvalue of a doubly stochastic matrix, characterizing the averaging algorithm. They discovered that the smaller this eigenvalue is, the faster the averaging algorithm converges.

In [22], the authors apply a machine learning method to evaluate the robustness of multi-agent systems. They use neural networks (NN) consisting of Multilayer Perceptrons (MLPs) to learn the representation of multi-agent networks, and use softmax as the classifiers. Using multi-agent model, in [23], the authors theorize that network metrics such as average path length, clustering coefficient, size of the largest connected component in the network and the maximum distance between nodes in the largest connected component relate to the robustness of supply networks, and statistically test these hypotheses with several examples.

The work of [24] focuses on improving community robustness to attacks and failures. The authors introduce a robustness measure to assess the community similarity between an original network and a broken network. Finally, they propose a multi-agent genetic algorithm to maximize the community robustness on networks.

In this work, we aim to shed light on the problem from a different point-of-view. Instead of viewing from a defense perspective, we look at the issue of attacking the network at the lowest cost. In other words, we evaluate what is the best attacking strategy, in the case of bribing attacks [25–27], that results in achieving a goal with the lowest cost concerning a given cost function. We seek to find a lower bound on how much cost/energy an attacker has to spend to drive the consensus to the attacker's desired state, defining a measure to evaluate the bribing resistance of nodes and networks. Our approach contrasts with the usual approaches that intuitively quantify how much a network can continue working when nodes are attacked. We aim to intuitively quantify how easily we can deviate a network from its normal function, by attacking some nodes. Our study is, in a sense, dual to the one in [28], where the authors showed how to compute a threshold for the "maximum impact" of an undetected fault/attack in linear consensus and networked physical systems, by solving a non-convex quadratic problem. Resorting to the framework developed in [28], an attacker can estimate the maximum that it can do, without being detected. With the framework that we develop in this work, an attacker may further determine the minimum cost required to drive the system towards the attacker's goal. Combining both results, the intrinsic limitations on the attacker capabilities can be derived, which depend on the network topology and on the assumption that the defender can run non-conservative detection methods.

Main contributions. (i) We propose a framework to quantify the resistance to bribery of nodes and sets of nodes in a network via average consensus; (ii) the proposed framework may be used under a concrete setting for average consensus when we know the initial states of each agent, and we know if the dynamic is discrete-time or continuous-time; (iii) if the only available knowledge is the network topology of agents (structure of the network), a general measure of bribing resistance is also proposed; (iv) we used the setup in (iii) to proposed a measure of network bribing resistance.

2. Preliminaries & notation

We denote by $\mathbb{N}$ the set of non-negative integers. We denote sets by calligraphic letters, e.g. $\mathcal{A}, \mathcal{E}$ and $\mathcal{V}$. We use lowercase letters to refer to elements in sets, e.g. $a \in \mathcal{A}$ and $v \in \mathcal{V}$. Further, we use uppercase letters to refer to matrices, e.g. A, B and W. A *network* can be modeled as a *graph*, which is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ is a set of $n > 0$ *nodes* (or *vertices*) and $\mathcal{E} \subset \{\{u, v\} : u, v \in \mathcal{V}\}$ is a relation of accessibility between nodes, and $e \in \mathcal{E}$ is called an *edge* that is an unordered pair $e = \{u, v\}$, with $u, v \in \mathcal{V}$.

The set of neighbors of a vertex $v \in \mathcal{V}$ are $\mathcal{N}_v = \{u : \{u, v\} \in \mathcal{E}\}$. From this point onwards, we assume to be working with undirected networks. The degree of a node $v \in \mathcal{V}$ is the number of vertices to that have an edge with v, that is, $d(v) = |\mathcal{N}_v|$. The diagonal matrix $D \in \mathbb{R}^{n \times n}$ such that $D_{ii} = d(i)$ is the *degree matrix* of graph $\mathcal{G}$. A *path* p of size $l > 0$ is a set of distinct vertices connected by edges, i.e. $p = (v_1, \dots, n_{l+1})$, where $\{v_i, v_{i+1}\} \in \mathcal{E}$ for $i = 1, \dots, l$. Further, we say that the path p starts in v_1 and

ends in v_{k+1}. A network is *connected* if, for any pair of vertices $u, v \in \mathcal{E}$, there is a path starting in u and ending in v.

A convenient form of representing a graph is by its *adjacency matrix*, i.e., a matrix $A \in \mathbb{R}^{n\times n}$, such that $A_{ii} = 0$, $A_{ij} = 1$ $(i \neq j)$ if there is an edge between vertices i and j and $A_{ij} = 0$ otherwise. Another common representation is via the *Laplacian matrix*, $L \in \mathbb{R}^{n\times n}$, $L = D - A$. In other words, We denote the set of eigenvalues of a matrix $A \in \mathbb{R}^{n\times n}$ by $\sigma(A) = \{\sigma_1, \ldots, \sigma_n\}$. We denote the set of non-zero eigenvalues of A by $\mu(A) = \{\mu_1, \ldots, \mu_k\}$, where $k \leq n$. Further, we denote by $\mathbf{1}_n$ the vector with size $n \in \mathbb{N}$ with all entries equal to one, and we omit the n when the underlying size is evident from the context. Lastly, we denote by e_i^n the n dimensional vector with the ith entry equal to 1 and the remaining entries equal to 0, and we omit n, writing e_i, when it is obvious from the context.

In [29], the authors define the *effective graph resistance* of a graph $\mathcal{G}$ using the Laplacian matrix L as:

$$R_{\mathcal{G}} = N \sum_{i=1}^{N} \frac{1}{\mu_i}, \text{ where } \mu(L) = \{\mu_1, \ldots, \mu_N\}.$$

For discrete-time, $k \in \mathbb{N}$, and synchronous communication, the average consensus algorithm can be modeled as:

$$x^{(k+1)} = Ax^{(k)} \text{ s.t. } \lim_{k\to\infty} x^{(k)} = \frac{\mathbf{1}^\intercal x^{(0)}}{n}\mathbf{1},$$

where $x^{(k)} \in \mathbb{R}^n$ and $x_i^{(k)}$ is the state of agent i, and for a vector w, $w^\intercal$ denotes its transpose.

Our goal is to study the bribing resistance that an agent or a set of agents has in an average consensus network. We may model the influence that an attacker exerts in the nodes through matrix B, where $B \in \mathcal{B}^{n\times p}$ encodes the selection of nodes to be bribed and $\mathcal{B} = \{0, 1\}$, and a sequence of $u^{(k)}$, with $u^{(k)} \in \mathbb{R}^p$ and $k \in \mathbb{N}$. This influence leads to the following dynamics

$$x^{(k+1)} = Ax^{(k)} + Bu^{(k)}.$$

Notice that a bribing attack differs from the so-called Byzantine attacks. In a bribing attack, it is not possible, for instance, to remove communication links, discard messages, make agents send different messages at the same time, or alter the protocol in any way.

Assumption 1. The cost of changing an agent's state by a units at some time instant $k \in \mathbb{N}$ is quadratic, more specifically, the cost is a^2. ◇

Observation. *Assumption 1 is adopted to simplify the analysis in the manuscript. Other cost function may also be adopted and the measures adapted accordingly.*

Under the described setup, we may conceive a trivial and illustrative scenario of a complete network of n agents where each agent's state is 0 ($x_i^{(k)} = 0$), for which the goal is to change the consensus value to 1.

Strategy 1 (*Naïve Strategy*). *A naïve bribing strategy to accomplish the envisioned attack is to select an agent v and, in the first time instant ($k = 0$), change its state to the number of agents in the network, i.e., to n. In other words: $B_{v,v} = 1$, and $B_{i,j} = 0$, otherwise, and $u_i^{(k)} = n$, if $i = v$ and $k = 0$ and $u_i^{(k)} = 0$, otherwise.*

Notice that the naïve strategy aims to change the steady state of the system by an amount of a, by changing (adding to) one agent's initial value an, where n is the number of agents in the network. The original average of the initial states was $x_\infty = \sum_{i=1}^n x_i^{(0)}/n$. When we apply the naïve strategy, we get as the new average $\tilde{x}_\infty = (an + \sum_{i=1}^n x_i^{(0)})/n = x_\infty + a$, which is changed by the amount of a as desired.

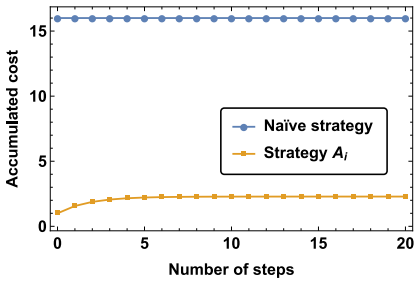


Fig. 1. The structural bribing resistance of nodes (value depicted close to each node) for all the connected graphs with 4 nodes.

Using Strategy 1 (Naïve strategy), we obtain the following

$$x^{(k+1)} = \underbrace{A^k x^{(0)}}_{\text{natural response}} + \underbrace{\sum_{\tau=0}^{k-1} A^{k-1-\tau} B u^{(\tau)}}_{\text{attack response}}. \tag{1}$$

As it is aimed by the attacker, the final consensus value is

$$\begin{aligned}\lim_{k\to\infty} x^{(k+1)} &= \lim_{k\to\infty} A^k x^{(0)} + \sum_{\tau=0}^{k-1} A^{k-1-\tau} B u^{(\tau)} \\ &= \lim_{k\to\infty} A^{k-1} B(n \cdot e_i) = \frac{\mathbf{1}\mathbf{1}^\intercal}{n} B(n \cdot e_i) \\ &= \frac{\mathbf{1}\mathbf{1}^\intercal}{n}(n \cdot e_i) = \mathbf{1}\mathbf{1}^\intercal e_i = \mathbf{1}.\end{aligned}$$

Moreover, we may wonder if it is always the case that to change the final consensus value by 1 unit, changing the state of one agent we always have a cost of n^2 (n the number of agents in the network, using a quadratic cost).

We may think of another bribing strategy deviating the state of agent i by the necessary amount such that its state is 1.

Strategy 2 (*Strategy A_i*). *The bribing strategy A_i consists of adding to the state of agent i the amount such that its state is changed to 1. In other words: $B_{v,v} = 1$, and $B_{i,j} = 0$, otherwise, and*

$$u_i^{(k)} = \begin{cases}(1 - x_i^{(k)}), & \text{if } i = v \\ 0, & \text{otherwise.}\end{cases}$$

For $n = 4$, the accumulated cost of the Strategies 1 and 2 are depicted in Fig. 1.

For the complete network, we have that any node yields the same bribing accumulated cost. Hence, a question that emerges is the following: *Is the cost associated to a bribing strategy as A_i the same for any node in a network?*

Next, we see that the answer is negative. In the following example we consider a path network, with $n = 4$, and we explore the Strategy 2 when selecting an agent to bribe in the extreme of the path (see Strategy A_2 in Fig. 2) or another of the two agents (see Strategy A_3 in Fig. 2). In fact, the cost of a bribing strategy depends on the selected agent, see Fig. 2.

This observation drives the need to build an evaluation framework for the bribing resistance of the various nodes or the entire network. Further, with such a framework in hands, we may design a network selecting a configuration of edges that makes the network (or a set of nodes) to be more resilient to bribing attacks.

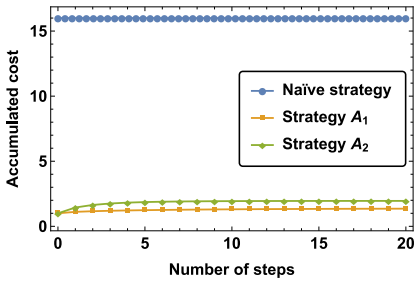


Fig. 2. The structural bribing resistance of nodes (value depicted close to each node) for all the connected graphs with 4 nodes.

3. Discrete-time bribing resistance of nodes

First, we state the assumption required for the framework that we will detail. The assumption is not limiting, as our framework can easily be extend to other setups.

Assumption 2. The network is undirected and connected. ◇

The assumption about connectivity is not a restriction because the analysis of node bribing resistance in a disconnected network may be (trivially) done by analyzing each of its connected subnetworks individually. Observe that the assumption about the edges being undirected is also not very restrictive, not only because several relevant applications use undirected networks, but also because the framework we propose may be easily generalized for the directed network scenario, as long as the used directed network may reach consensus. The difference would be in the update rule of each agent state $x_v^{(k)}$, and the cost computations would follow the same expressions, where the state update is changed according to the utilized consensus algorithm.

To measure the influence a node has in a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we consider the discrete-time average consensus algorithm that, for $k \in \mathbb{N}$, using the Metropolis weights [30]:

$$\begin{cases} x_u^{(0)} &= x_u^0 \\ x_u^{(k+1)} &= \sum_{v\in\mathcal{N}_u} W_{uv} x_v^{(k)}, \end{cases} \text{ where}$$

$$W_{uv} = \begin{cases} (1+\max\{|\mathcal{N}_u|, |\mathcal{N}_v|\})^{-1}, & \text{if } v \in \mathcal{N}_u \text{ and } u \neq v \\ 0, & \text{if } v \notin \mathcal{N}_u \text{ and } u \neq v \\ 1 - \sum_{v\in\mathcal{N}_u} W_{uv}, & \text{if } u = v. \end{cases}$$

Given a network of nodes $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we define the *cost* of changing node $v \in \mathcal{V}$ state at time $k \in \mathbb{N}$ to $a \in \mathbb{R}$ by:

$$c_v^{(k)}(a) = (a - x_v^{(k)})^2.$$

When, at time $k > 0$, the state of each agent $v \in \mathcal{A} \subset \mathcal{V}$ is changed to a_v, the state of agent $u \in \mathcal{V}$, at time $k+1$ is given as:

$$x_u^{(k+1)} = \sum_{w\in\mathcal{N}_u\setminus\mathcal{A}} W_{uw} x_w^{(k)} + \sum_{v\in\mathcal{A}} W_{uv} a_v.$$

We extend the cost definition to establish the cost of changing node $v \in \mathcal{V}$ to $a \in \mathbb{R}$ during $N \in \mathbb{N}$ time steps by:

$$C_v^N(a) = \sum_{k=1}^{N} c_v^{(k)}(a).$$

Analogously, we extend the aforementioned for the case of changing the states of a set of nodes, $\mathcal{A} \subset \mathcal{V}$, as:

$$C_{\mathcal{A}}^N(a) = \sum_{v\in\mathcal{A}} \sum_{k=1}^{N} c_v^{(k)}(a) = \sum_{v\in\mathcal{A}} \sum_{k=1}^{N} C_v^N(a).$$

Given a network of nodes $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set of initial states $\{x_v^{(0)}\}_{v\in\mathcal{V}}$, we define the *bribing resistance* of node v, and a set of nodes $\mathcal{A}$, to be changed to state a, respectively, as

$$R_v(a) = \lim_{N\to\infty} C_v^N(a) \text{ and } R_{\mathcal{A}}(a) = \lim_{N\to\infty} C_{\mathcal{A}}^N(a).$$

We observe that the quadratic cost function that we propose not only allows an easier exposition but also can model scenarios such as a multi-agent system that corresponds to moving agents, where a malicious entity exerts a force in an agent to drive the agent to a certain position. In this case, the cost of doing so is related to the kinetic energy the entity needs to apply, a quadratic function. Next, we show that its value converges.

Proposition 1. *Given a connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}| > 1$ and a set of initial values $\{x_u^{(0)}\}_{u\in\mathcal{V}}$, then the bribing resistance $C_{\mathcal{A}}^N(a)$ converges, i.e, $R_{\mathcal{A}}(a)$ is finite.*

Proof. The difference between two iterations is

$$\begin{aligned} \left|R_{\mathcal{A}}^{N+1}(a) - R_{\mathcal{A}}^N(a)\right| &= \left|\sum_{v\in\mathcal{A}} \left(\sum_{k=1}^{N+1} c_v^{(k)}(a) - \sum_{k=1}^{N} c_v^{(k)}(a)\right)\right| \\ &= \left|\sum_{v\in\mathcal{A}} c_v^{(N+1)}(a)\right| = \sum_{v\in\mathcal{A}} \left|a - x_v^{(N+1)}\right|. \end{aligned}$$

So, we need to show that, for each $v \in \mathcal{A}$, $x_v^{(N+1)}$ converges.

$$\begin{aligned} |x_v^{(N+1)} - x_v^{(N)}| &= \sum_{w\in\mathcal{N}_v\setminus\mathcal{A}} \frac{|x_w^{(N)} - x_w^{(N-1)}|}{1+\max\{|\mathcal{N}_v|, |\mathcal{N}_w|\}} \\ &\leq \lambda_\alpha |x_\alpha^{(N)} - x_\alpha^{(N-1)}|, \end{aligned} \tag{2}$$

where $\lambda_\alpha |x_\alpha^{(N)} - x_\alpha^{(N-1)}| = \max_{w\in\mathcal{N}_v\setminus\mathcal{A}} \frac{|x_w^{(N)} - x_w^{(N-1)}|}{1+\max\{|\mathcal{N}_v|, |\mathcal{N}_w|\}}$. Since $|\mathcal{N}_v|, |\mathcal{N}_\alpha| > 0$ for any $u \in \mathcal{V}$, it follows that $\lambda_\alpha = (1+\max\{|\mathcal{N}_v|, |\mathcal{N}_\alpha|\})^{-1} \leq \frac{1}{2} < 1$. Hence, the method is a contractive map and $x_v^{(N)}$ converges when $N \to \infty$, and $R_{\mathcal{A}}^N(a)$ also converges. □

Next, we compute the convergence rate of the method.

Proposition 2. *Given a connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set of initial values $\{x_u^{(0)}\}_{u\in\mathcal{V}}$, the bribing resistance $C_{\mathcal{A}}^N(a)$ converges with exponential rate $\eta = \frac{1}{2}$.*

Proof. Using (2), it readily follows that

$$\begin{aligned} |x_v^{(N+1)} - x_v^{(N)}| &\leq \lambda_\alpha |x_\alpha^{(N)} - x_\alpha^{(N-1)}| \leq \frac{1}{2} |x_\alpha^{(N)} - x_\alpha^{(N-1)}| \\ &\leq \cdots \leq (1/2)^N |x_\alpha^{(1)} - x_\alpha^{(0)}|. \end{aligned}$$

The method converges with exponential rate $\eta = 1/2$. □

If we want to obtain an approximation up to a certain precision, we need to run the given method the number of steps provided in the following result.

Corollary 1. *Consider a connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set of initial values $\{x_u^{(0)}\}_{u\in\mathcal{V}}$, to obtain an ε-approximation, $\varepsilon > 0$, of $R_{\mathcal{A}}(a)$, we need to compute $C_{\mathcal{A}}^K(a)$, where $K = \log_\eta \varepsilon$ and η is given by Proposition 2.*

Proof. Since by Proposition 2 we have that $|C_{\mathcal{A}}^{K+1}(a) - C_{\mathcal{A}}^K(a)| \leq \eta^K |x_\alpha^{(1)} - x_\alpha^{(0)}|$. Hence, the smallest K that ensures $\eta^K |x_\alpha^{(1)} - x_\alpha^{(0)}| \leq \varepsilon$ is

$$K = \log_{(1/2)} \left(\varepsilon \Big/ \left|x_\alpha^{(1)} - x_\alpha^{(0)}\right|\right). \quad \square$$

3.1. Discrete-time structural bribing resistance of nodes

Next, we develop a measure of bribing resistance based only on the structure of the network, i.e., that does not depend on quantities such as states of nodes of the network [31,32].

To evaluate the *structural bribing resistance* to changes of a node or a set of nodes in the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we set $x_v^0 = 0$ for all $v \in \mathcal{V}$ and, we set $a = 1$ for the set of nodes we are evaluating. Thus, we have that the cost of changing a node $v \in \mathcal{V}$ state at time $k \in \mathbb{N}$ is $c_v^{(k)} = (1 - x_v^{(k)})^2$, and the cost of changing a node $v \in \mathcal{V}$, and set of nodes $\mathcal{A} \subset \mathcal{V}$, for N time steps is, respectively

$$C_v^N = \sum_{k=1}^{N} c_v^{(k)}, \text{ and } C_{\mathcal{A}}^N = \sum_{v \in \mathcal{A}} \sum_{k=1}^{N} C_v^N.$$

Given a network of nodes $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we define the *structural bribing resistance* or *importance* of node v, and of a set of nodes $\mathcal{A} \subset \mathcal{V}$, respectively as

$$\bar{R}_v = \lim_{N \to \infty} C_v^N \text{ and } \bar{R}_{\mathcal{A}} = \lim_{N \to \infty} C_{\mathcal{A}}^N. \tag{3}$$

In this case, we have the following result.

Corollary 2. *Given a network of nodes $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the structural bribing resistance of a set of nodes $\mathcal{A}$, $\bar{R}_{\mathcal{A}}$, converges with exponential rate η, given in Proposition 2, to $\bar{R}_{\mathcal{A}} = \sum_{v \in \mathcal{A}} R_v$.*

Utilizing the previous definitions, we establish a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ bribing resistance as the average:

$$\bar{R}_{\mathcal{G}} = R_{\mathcal{V}} \big/ |\mathcal{V}|.$$

4. Continuous-time bribing resistance of nodes

Depending on the application that we envision to use a network, it may also make sense to study a continuous-time version of the proposed bribing resistance notions. Therefore, alternatively, we may define the previous measures of bribing resistance for the continuous-time average consensus. Here, for a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we use as average consensus:

$$\dot{x}^{(t)} = -Lx^{(t)},$$

where $x^{(t)} = (x_1^{(t)}, \dots, x_n^{(t)})$ and $x_1^{(0)}, \dots, x_n^{(0)} \in \mathbb{R}$.

For a network of nodes $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we define the *cost* of changing node $v \in \mathcal{V}$ state at time $t \in \mathbb{R}$ to $a \in \mathbb{R}$ by:

$$c_v^{(t)}(a) = (a - x_v^{(t)})^2.$$

In this case, whenever the state of each agent $v \in \mathcal{A} \subset \mathcal{V}$ with $|\mathcal{A}| = k$ is constantly changed to a_v, the states of the agents are given by:

$$\dot{x}^{(t)} = -Lx^{(t)} + Bu^{(t)},$$

where $B \in \mathbb{R}^{n \times k}$, $u^{(t)} \in \mathbb{R}^k$, and $B_{ij} = \begin{cases} 1, \text{ if } j \in \mathbb{N}_i \\ 0, \text{ otherwise,} \end{cases}$ and $u_v^{(t)} = a_v - x_v^{(t)}$. Therefore, we have that $c_v^{(t)}(a) = \left(u_v^{(t)}\right)^2$. Now, we may simply extend the cost definition to establish the cost of changing node $v \in \mathcal{V}$, and a set of nodes $\mathcal{A} \subset \mathcal{V}$, to $a \in \mathbb{R}^+$ until time $\tau \in \mathbb{R}$ to be respectively:

$$C_v^\tau(a) = \int_0^\tau \left(u_v^{(t)}\right)^2 dt \text{ and } C_{\mathcal{A}}^\tau(a) = \sum_{v \in \mathcal{A}} \int_0^\tau \left(u_v^{(t)}\right)^2 dt.$$

Given a network of nodes $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set of initial states $\{x_v^{(0)}\}_{v \in \mathcal{V}}$, we define the *bribing resistance* of node v to be changed to state a as

$$R_v(a) = \lim_{\tau \to \infty} C_v^\tau(a) = \int_0^\infty \left(u_v^{(t)}\right)^2 dt.$$

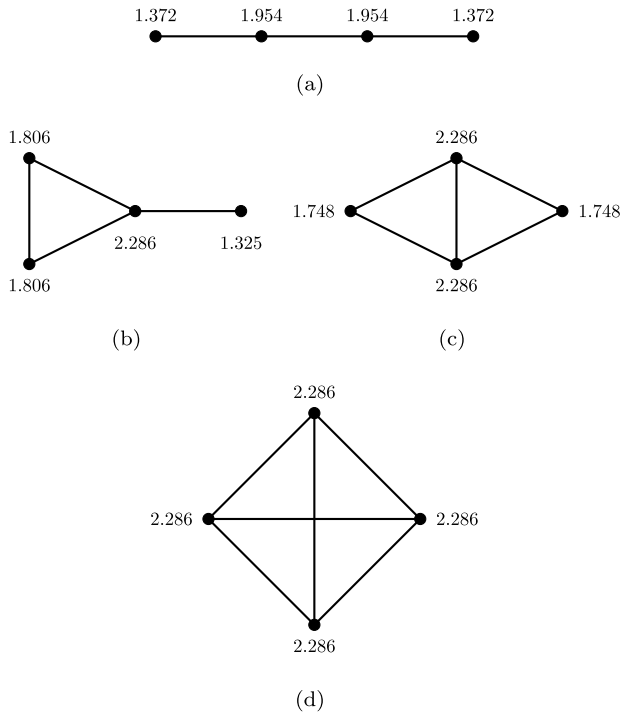


Fig. 3. The structural bribing resistance of nodes (value depicted above each node) for all the connected graphs with 4 nodes.

Similarly, we extend the notion to a set of nodes $\mathcal{A} \subset \mathcal{V}$ to be changed to state a defined as:

$$R_{\mathcal{A}(a)} = \lim_{\tau \to \infty} C_{\mathcal{A}}^\tau(a) = \sum_{v \in \mathcal{A}} \int_0^\infty \left(u_v^{(t)}\right)^2 dt. \tag{4}$$

4.1. Continuous-time structural bribing resistance of nodes

Here, as we did in Section 3.1, we propose a measure of bribing resistance that uses the continuous-time average consensus and that is based only on the structure of the network. Given a network, this measure corresponds to set $a = 1$ and $x_v^{(0)} = 0$ for $v \in \mathcal{V}$. Thus, the cost of changing node $v \in \mathcal{V}$ at time $t \in \mathbb{R}^+$ is $c_v^{(t)} = (1 - x_v^{(t)})^2$. When the state of each agent $v \in \mathcal{A} \subset \mathcal{V}$ with $|\mathcal{A}| = k$ is constantly changed to 1, the states of the agents are given by:

$$\dot{x}^{(t)} = -Lx^{(t)} + Bu^{(t)},$$

where $B \in \mathbb{R}^{n \times k}$, $u^{(t)} \in \mathbb{R}^k$, and $B_{ij} = \begin{cases} 1, \text{ if } j \in \mathcal{A} \\ 0, \text{ otherwise,} \end{cases}$ and $u_v^{(t)} = 1 - x_v^{(t)}$. Since $c_v^{(t)} = \left(u_v^{(t)}\right)^2$, the cost of changing node $v \in \mathcal{V}$, and a set of nodes $\mathcal{A} \subset \mathcal{V}$, to 1 until time $\tau \in \mathbb{R}$ is, respectively:

$$C_v^\tau(a) = \int_0^\tau \left(u_v^{(t)}\right)^2 dt \text{ and } C_{\mathcal{A}}^\tau = \sum_{v \in \mathcal{A}} \int_0^\tau \left(u_v^{(t)}\right)^2 dt.$$

For $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set of initial states $\{x_v^{(0)}\}_{v \in \mathcal{V}}$, we define the *structural bribing resistance* of node v as

$$R_v = \lim_{\tau \to \infty} C_v^\tau(a) = \int_0^\infty \left(u_v^{(t)}\right)^2 dt.$$

Likewise, we extend the previous definition to the structural bribing resistance of a set of nodes $\mathcal{A}$ as

$$R_{\mathcal{A}} = \lim_{\tau \to \infty} C_{\mathcal{A}}^\tau = \sum_{v \in \mathcal{A}} \int_0^\infty \left(u_v^{(t)}\right)^2 dt.$$

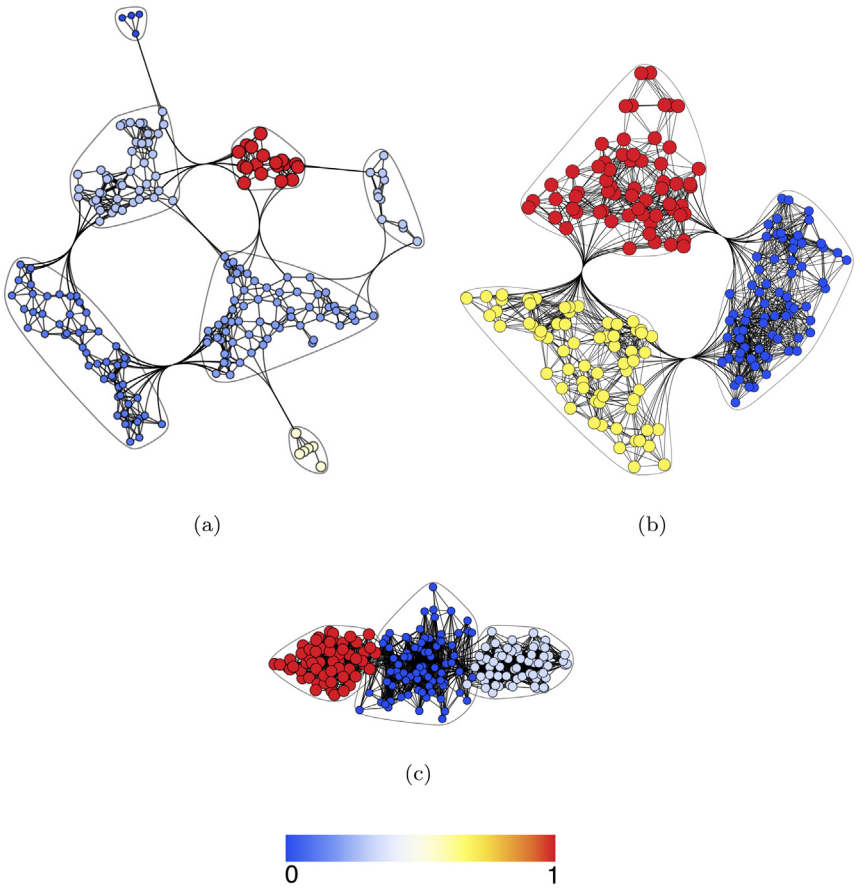


Fig. 4. Normalized bribing resistance of groups/clusters of nodes. The groups/clusters of nodes consist of circumscribed nodes with the same color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Illustrative examples

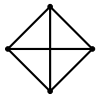
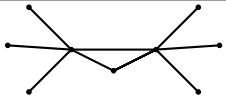
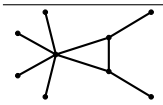
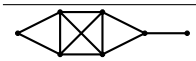
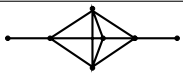
To illustrate the bribing resistance measure based only on the structure of the network, we consider all possible networks with 4 nodes and, for each node, we present the value of its structural resistance above the node in Fig. 3.

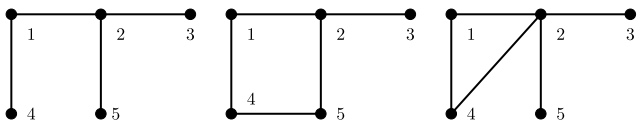
Next, in Table 1, we illustrate the measures of network bribing resistance we propose and compare them with the measures proposed in [29] and [20].

Notice that the proposed methods evaluate network bribing resistance in a different light than the techniques in the literature. For example, when we look at the first three networks of Table 1, the more resilient network depends on the utilized measure. In the context of network controllability or consensus networks, the measures that we introduce find a direct application and have a more intuitive meaning.

Observation. *In Table 1, we can see that for networks with four nodes (the first three rows of the table), the measure $\mathcal{R}_{\mathcal{G}}$ [29] yields that the more resilient networks are the first and the third ones. For the $\mathcal{R}'_{\mathcal{G}}$ [20] measure, we obtain that the more resilient network is the complete network (second row). Intuitively, the measure tells that the complete network can continue to work if we corrupt one node because every other node is connected. The last two columns are the measures that we propose here. In this case, we are intuitively measuring how easy it is on average to corrupt all the network by changing (bribing) only one node. Notice that, with $\bar{\mathcal{R}}^c_{\mathcal{G}}$, the network with smaller bribing resistance is the complete network because if we corrupt one node, it passes information directly to every other node.*

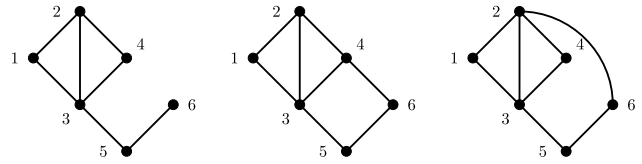
Table 1
Network resilience and bribing resistance of graphs using different metrics. For the two first measures, $\mathcal{R}_{\mathcal{G}}$ [29] and $\mathcal{R}'_{\mathcal{G}}$ [20], small values mean more resilient, whereas for the two proposed in this paper, $\bar{\mathcal{R}}_{\mathcal{G}}$ and $\bar{\mathcal{R}}^c_{\mathcal{G}}$, large values mean more resilient.

Network	$R_{\mathcal{G}}$	$R'_{\mathcal{G}}$	$\bar{R}_{\mathcal{G}}$	$\bar{R}^c_{\mathcal{G}}$
	2	10	1.663	1.235
	4	3	2.286	1.125
	2	9	1.565	2.484
	2.25	64	2.089	12.359
	2.25	64	2.091	11.733
	3.667	19.8	2.710	3.040
	3.667	21	2.712	4.173



(a) $\bar{R}_{\mathcal{G}} \approx 1.917$ and $R'_{\mathcal{G}} = 18$. (b) $\bar{R}_{\mathcal{G}} \approx 2.031$ and $R'_{\mathcal{G}} = 11.5$. (c) $\bar{R}_{\mathcal{G}} \approx 1.829$ and $R'_{\mathcal{G}} \approx 12.667$

Fig. 5. Adding the edge $(4, 5)$ is optimal for $R'_{\mathcal{G}}$ (smaller means more resistant) and also optimal for $\bar{R}_{\mathcal{G}}$ (larger means a larger bribing cost and, thus, more resilient).



(a) $\bar{R}_{\mathcal{G}} \approx 2.077$ and $R'_{\mathcal{G}} = 20.5$. (b) $\bar{R}_{\mathcal{G}} \approx 2.411$ and $R'_{\mathcal{G}} \approx 12.6$. (c) $\bar{R}_{\mathcal{G}} \approx 2.331$ and $R'_{\mathcal{G}} \approx 12.4$

Fig. 6. Adding the edge $(2, 6)$ is optimal for $R'_{\mathcal{G}}$ (smaller means more resistant) and adding the edge $(4, 6)$ is optimal for $\bar{R}_{\mathcal{G}}$ (larger means a larger bribing cost and, thus, more resilient).

Cluster bribing resistance. In the following examples, we illustrate the proposed structural bribing resistance measure applied to groups/clusters of users. Our measure of bribing resistance captures the notion of a *group of nodes relevance* quantitatively in a network, which may be used in several applications to access the importance of clusters of nodes, such as in applications related to social networks. The bribing resistance of groups of nodes may be computed using discrete-time or continuous-time, respectively with (3) or (4). In this case, we utilized the discrete-time case, see Fig. 4. We normalized the bribing resistance values to be in the interval [0, 1].

Improving graph bribing resistance by edge addition. Next, we illustrate how our framework may be used to increase graph bribing resistance by adding edges to a graph. Three examples are shown in Figs. 5–7, where we seek to achieve the optimal selection of one edge to increase network bribing resistance.

5.1. Limitations of the proposed framework

Although the proposed framework sheds light on a different perspective of evaluating network and nodes robustness/resilience, this view comes with a computational cost that can be larger than other approaches.

The computational complexity of the proposed framework may become unfeasible as the size of the networks increases, as it involves computing the cost of bribing each possible node in the network and/or subsets of nodes. For example, for Fig. 4(c) that has 200 nodes, it took 98 seconds to evaluate the resilience of the three groups of nodes. This contrasts with the computational complexity of approaches that solely consist of computing an expression using the eigenvalues of the matrix that represents the network.

6. Conclusions

In this paper, we developed a framework to evaluate the resistance of nodes in a network to attacks via average consensus algorithms. Utilizing the previous measure, we extended it to compute the resistance of a set of nodes or the entire network. It can be applied to the scenario where it is used an average consensus algorithm or in a general setting, where we do not know the values that nodes may share. We further utilized the proposed method to increase network bribing resistance via edges addition. Finally, we showed some examples of networks with groups/clusters of users, where the proposed framework may be used to assess each cluster bribing resistance.

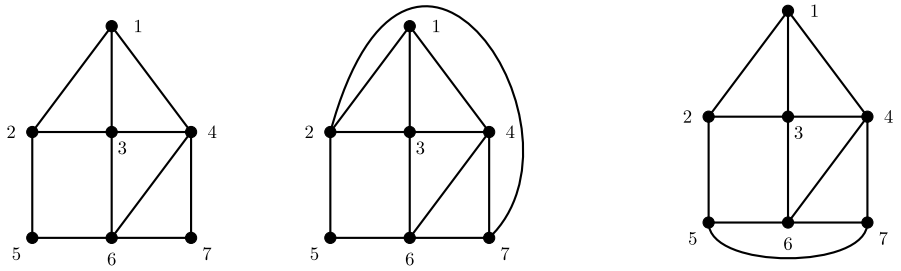
Future work includes generalizing the proposed bribing analysis framework to other types of consensus, for instance using the method in [33,34].

CRediT authorship contribution statement

Guilherme Ramos: Conceptualization, Writing - original draft, Investigation, Methodology, Software, Writing - review & editing, Validation, Formal analysis. **Daniel Silvestre:** Supervision, Writing - review & editing. **Carlos Silvestre:** Supervision, Writing - review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



(a) $\bar{R}_{\mathcal{G}} \approx 2.952$ and $R'_{\mathcal{G}} \approx 14.9$. (b) $\bar{R}_{\mathcal{G}} \approx 3.036$ and $R'_{\mathcal{G}} \approx 12.5$. (c) $\bar{R}_{\mathcal{G}} \approx 3.110$ and $R'_{\mathcal{G}} \approx 12.4$.

Fig. 7. Adding the edge $(5, 7)$ is optimal for $R'_{\mathcal{G}}$ (smaller means more resistant) and also optimal for $\bar{R}_{\mathcal{G}}$ (larger means a larger bribing cost and, thus, more resilient).

References

[1] R. Langner, Stuxnet: Dissecting a cyberwarfare weapon, IEEE Secur. Priv. 9 (3) (2011) 49–51.

[2] D.U. Case, Analysis of the cyber attack on the Ukrainian power grid, Electr. Inf. Shar. Anal. Center (2016).

[3] J. Ferber, G. Weiss, Multi-Agent Systems: an Introduction to Distributed Artificial Intelligence, Vol. 1, Addison-Wesley Reading, 1999.

[4] J. Tsitsiklis, D. Bertsekas, M. Athans, Distributed asynchronous deterministic and stochastic gradient optimization algorithms, IEEE Trans. Automat. Control 31 (9) (1986) 803–812.

[5] B. Johansson, T. Keviczky, M. Johansson, K.H. Johansson, Subgradient methods and consensus algorithms for solving convex optimization problems, in: 47th IEEE Conference on Decision and Control, CDC, 2008, pp. 4185–4190.

[6] A. Jadbabaie, J. Lin, A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Trans. Autom. Control 48 (6) (2003) 988–1001.

[7] J. Cortés, S. Martínez, F. Bullo, Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions, IEEE Trans. Autom. Control 51 (8) (2006) 1289–1298.

[8] M. Chiang, S.H. Low, A.R. Calderbank, J.C. Doyle, Layering as optimization decomposition: A mathematical theory of network architectures, Proc. IEEE 95 (1) (2007) 255–312.

[9] D. Silvestre, J. Hespanha, C. Silvestre, A pagerank algorithm based on asynchronous Gauss-Seidel iterations, in: American Control Conference, ACC, 2018, pp. 484–489.

[10] D. Silvestre, J.P. Hespanha, C. Silvestre, Broadcast and gossip stochastic average consensus algorithms in directed topologies, IEEE Trans. Control Netw. Syst. 6 (2) (2019) 474–486.

[11] M. Zhu, S. Martínez, Discrete-time dynamic average consensus, Automatica 46 (2) (2010) 322–329.

[12] W. Ren, R.W. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, IEEE Trans. Autom. Control 50 (5) (2005) 655–661.

[13] Y. Shang, Consensus of hybrid multi-agent systems with malicious nodes, IEEE Trans. Circuits Syst. II 67 (4) (2019) 685–689.

[14] C. Yu, B.D.O. Anderson, S. Mou, J. Liu, F. He, A.S. Morse, Distributed averaging using periodic gossiping, IEEE Trans. Autom. Control 62 (8) (2017) 4282–4289.

[15] K. Cai, H. Ishii, Average consensus on general strongly connected digraphs, Automatica 48 (11) (2012) 2750–2761.

[16] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, Randomized gossip algorithms, IEEE Trans. Inf. Theory 52 (6) (2006) 2508–2530.

[17] A. Haseltalab, M. Akar, Approximate byzantine consensus in faulty asynchronous networks, in: American Control Conference, ACC, 2015, pp. 1591–1596.

[18] S.M. Dibaji, H. Ishii, Resilient consensus of second-order agent networks: Asynchronous update rules with delays, Automatica 81 (2017) 123–132.

[19] Y. Shang, Consensus and clustering of expressed and private opinions in dynamical networks against attacks, IEEE Syst. J. 14 (2) (2020) 2078–2084.

[20] W. Ellens, F. Spieksma, P. Van Mieghem, A. Jamakovic, R. Kooij, Effective graph resistance, Linear Algebr. Appl. 435 (10) (2011) 2491–2506.

[21] H. Chan, L. Akoglu, Optimizing network robustness by edge rewiring: a general framework, Data Min. Knowl. Discov. 30 (5) (2016) 1395–1425.

[22] G. Wang, M. Xu, Y. Wu, N. Zheng, J. Xu, T. Qiao, Using machine learning for determining network robustness of multi-agent systems under attacks, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2018, pp. 491–498.

[23] A. Nair, J.M. Vidal, Supply network topology and robustness against disruptions – an investigation using multi-agent model, Int. J. Prod. Res. 49 (5) (2011) 1391–1404.

[24] S. Wang, J. Liu, A multi-agent genetic algorithm for improving the robustness of communities in complex networks against attacks, in: 2017 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2017, pp. 17–22.

[25] J. Saúde, G. Ramos, C. Caleiro, S. Kar, Reputation-based ranking systems and their resistance to bribery, in: 2017 IEEE International Conference on Data Mining, ICDM, IEEE, 2017, pp. 1063–1068.

[26] G. Ramos, L. Boratto, C. Caleiro, On the negative impact of social influence in recommender systems: A study of bribery in collaborative hybrid algorithms, Inf. Process. Manag. 57 (2) (2020) 102058.

[27] J. Saude, G. Ramos, L. Boratto, C. Caleiro, A robust reputation-based group ranking system and its resistance to bribery, 2020, arXiv preprint arXiv:2004.06223.

[28] D. Silvestre, P. Rosa, J. ao P. Hespanha, C. Silvestre, Stochastic and deterministic fault detection for randomized gossip algorithms, Automatica 78 (2017) 46–60.

[29] D.J. Klein, M. Randić, Resistance distance, J. Math. Chem. 12 (1) (1993) 81–95.

[30] L. Xiao, S. Boyd, S. Lall, A scheme for robust distributed sensor fusion based on average consensus, in: Fourth International Symposium on Information Processing in Sensor Networks, 2005, IPSN 2005, IEEE, 2005, pp. 63–70.

[31] G. Ramos, A.P. Aguiar, S. Pequito, Structural systems theory: an overview of the last 15 years, 2020, arXiv preprint arXiv:2008.11223.

[32] G. Ramos, S. Pequito, C. Caleiro, On the index of convergence of a class of Boolean matrices with structural properties, Int. J. Control (2019) 1–9.

[33] Y. Shang, Finite-time weighted average consensus and generalized consensus over a subset, IEEE Access 4 (2016) 2615–2620.

[34] Y. Shang, Resilient consensus of switched multi-agent systems, Syst. Control Lett. 122 (2018) 12–18.