A resilient continuous-time consensus method using a switching topology⋆

Guilherme Ramos[a,*,1], Daniel Silvestre[b] and A. Pedro Aguiar[c]

[a]LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal
[b]Department of Electrical and Computer Engineering, Faculty of Sciences and Technology, NOVA University of Lisbon, Portugal
[c]Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Portugal

ARTICLE INFO

Keywords:
Agents and autonomous systems
consensus methods
switching systems
reputation systems
resilient systems

ABSTRACT

This paper addresses the design problem of a resilient consensus algorithm for agents with continuous-time dynamics. The main proposal is that by incorporating a switching mechanism selecting the network topology to avoid malicious nodes from communicating, the remaining nodes will converge to a value closer to the original steady-state without the attacker being present. The switching occurs at discrete-time steps where each node evaluates the reputation score of the neighbors and deactivates/ignores edges in the network. We explore the proposed method with illustrative examples ranging from static topologies to dynamic ones, considering directed and undirected graphs, presenting several attacking scenarios that are successfully mitigated with our method. Finally, we compare the best undetectable attacking strategy and the commonly used approach named MSR, highlighting the advantages of our method.

1. Introduction

The consensus problem is a central block in many networked multi-agent systems. The issue is to have an iterative algorithm such that the set of agents can agree upon a value via local interactions through a communication network. From another perspective, the problem consists of the design of a distributed procedure where the communication between entities with computational power is restricted by the network topology.

Examples of applications where consensus has a core role are optimization [1, 2], motion coordination tasks [3, 4], rendezvous problems [5, 6, 7], resource allocation in computer networks [8], desynchronization of transmitters at the MAC layer in sensor networks [9], and measuring the relative importance of web pages with PageRank-like algorithms [10]. In [11], the authors proposed a distributed Kalman Filter based on two consensus systems to estimate the 2D motion of a target, and they experimentally assessed it in [12] to estimate the motion of a real robot.

⋆This work was supported by FCT through the LASIGE Research Unit, ref. UIDB/00408/2020 and ref. UIDP/00408/2020, by projects: RELIABLE (PTDC/EEI-AUT/3522/2020) funded by FCT/MCTES; and DynamiCITY (NORTE-01-0145-FEDER-000073), funded by NORTE2020/PORTUGAL2020, through the European Regional Development Fund. D. Silvestre work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through Institute for Systems and Robotics (ISR), under Laboratory for Robotics and Engineering Systems (LARSyS) project UIDB/50009/2020, through project PCIF/MPG/0156/2019 FirePuma and through COPELABS, University Lusófona project UIDB/04111/2020.

*Corresponding author
ghramos@fc.ul.pt (G. Ramos); dsilvestre@isr.tecnico.ulisboa.pt (D. Silvestre); pedro.aguiar@fe.up.pt (A.P. Aguiar)
ORCID(s): 0000-0001-6104-8444 (G. Ramos); 0000-0002-8097-0626 (D. Silvestre); 0000-0001-7105-0505 (A.P. Aguiar)

The importance of consensus and its underlying communication aspect makes it crucial to delve into the resilience aspects, i.e., the capability of overcoming abnormal situations. Therefore, an agent in a consensus network should effectively and efficiently identify neighbors that are sharing false information. By filtering out corrupted state values, the normal agents aim to converge to a steady-state as close as possible to the true value.

Resilient consensus. In [13], the authors study the continuous-time consensus problem in the presence of adversaries. They modeled the network of multi-agents as a switched system, where the normal agents have integrator dynamics, and the switching signal defines the network topology. Under the assumption that, at most, a fraction of the neighbors of any normal agent may be attacked, they presented a novel graph-theoretic metric, called fractional robustness, to analyze the network topologies where the set of normal agents reach consensus.

The work in [14] considered resilient consensus and synchronization of identical agents following a continuous-time LTI system model. The authors devised a resilient consensus protocol called ARC-P, which has a robustness parameter f, together with resilient control laws for the synchronization purposes. Also, they presented necessary and sufficient requirements such that the distributed control laws accomplish their goal for time-invariant and time-varying networks.

In [15], the resilient consensus of switched multi-agent systems is studied. The author proposes a switched filtering strategy that can cope with a subset of malicious nodes in directed networks under arbitrary switching rules. As generalizations, the authors addressed both the resilient scaled consensus and the resilient scaled formation generation problems for multi-agent switched systems.

The work in [16] addressed the general problem of reaching resilient consensus among a set of agents in the presence of faulty nodes. The authors developed a general method that requires, as input, a consensus algorithm and the robustness parameter f. The method is an extension of [17] and is suitable for both discrete-time and continuous-time consensus, letting the normal agents identify the set of attacked nodes and correct the consensus value by ignoring the faulty nodes.

In [18], resilience in a type of consensus system was achieved by computing the variance of the received state values of the neighbors. In a sense, the variance can be thought as providing a relative reputation of the node, with the agent with the largest value being an attacker in that case given the properties of a specific type of consensus dynamics.

In this work, in contrast with the general method proposed in [16], each agent does not require to keep in memory an exponential number of variables. Here, each agent only needs to store its state and Boolean values for each neighbor. These Boolean values result from the reputation score that the agent computes for each neighbor, and correspond to an activation or not of their connection. Additionally, the method that we propose has the advantage of not requiring, as input, a parameter defining the maximum number of attacked agents.

Reputation systems. The notion of an entity's *reputation* is an impression about that entity that naturally appears from evaluating it using a set of criteria. Often, we cast reputations on persons, corporations, services, and many other entities. Usually, we compare the properties of an entity with other related entities. Thus, we create a reputation based on what we would expect of an entity, comparing it with others. Therefore, the concept of reputation is omnipresent and yields a powerful tool of social control in a plethora of areas. For example, areas as natural societies, business, education, social networks use the reputation concept.

Due to its importance and ubiquity, a galore of computer science applications adopted this concept to develop efficient, effective, and robust methods. For example, in [19, 20, 21, 22, 23], the authors develop methods using reputation as a central concept to mitigate the effect of attacks and bribery.

Furthermore, reputation is also a relevant measure to be evaluated in the field of Social Networks as in [24, 25, 26]. In [27], the authors propose a method to address the problem of determining a degree of reputation for agents behaving as assistants to the members of an electronic community. The seminal work in [28] introduced a discrete-time reputation-based consensus method, where each agent assigns a reputation to each neighbor to filter neighbors with low reputation. We refer the reader to the surveys in [24, 29] and references therein to a more detailed overview of reputation systems.

The **main contribution** of this paper is the development of a reputation-based consensus method via a switching system, where the agents' states evolve in continuous-time, but with discrete-time switching of the network topology that depends on the reputation assigned among neighbors. Our method limits what an attacker can do in the best undetectable attacking strategy and is more robust than the commonly used methods.

Paper structure: In Section 1.1, we define the adopted notation. In Section 2, we present the main results of the paper, providing a reputation-based resilient continuous-time consensus method. Section 3 shows illustrative examples of the proposed method, highlighting in particular the higher robustness of the proposed algorithm in comparison with typically used methods. Finally, Section 4 closes the paper and points future research directions.

1.1. Notation

A *directed graph* or *digraph* is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a nonempty set of *nodes*, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of *edges*. An edge is an ordered pair encoding a relationship of accessibility between nodes. In other words, if $u, v \in \mathcal{V}$ and $(u, v) \in \mathcal{E}$ then the node v directly accesses information of node u. Given a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if $(u, v) \in \mathcal{E}$ *if and only if* $(v, u) \in \mathcal{E}$ (if there is an edge starting in u and ending in v there is also the reciprocate edge), then we say that the digraph is undirected or simply a *graph*. In the scope of consensus methods, we also refer to a digraph (or graph) as a *network*, and we refer to nodes as network *agents*.

Given a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if for any $u, v \in \mathcal{V}$, with $u \neq v$ we have that $(u, v) \in \mathcal{E}$, then we say that $\mathcal{G}$ is a *complete digraph* or *complete network*. Additionally, for an agent $v \in \mathcal{V}$, the set of nodes that v can directly access information in the network is defined as $\mathcal{N}_v = \{v\} \cup \{u : (u, v) \in \mathcal{E}\}$, and these agents are called the *neighbors* of v. The *in-degree* of $v \in \mathcal{V}$, d_v is the number of proper neighbors of v, $d_v = |\mathcal{N}_v \backslash \{v\}|$. Analogously, the *out-degree* of a node $v \in \mathcal{V}$, o_v, is the number of nodes that have v has a neighbor, $o_v = |\{u : v \in \mathcal{N}_u \setminus \{u\}\}|$. If the digraph is a graph, then the in-degree is the same as the out-degree, and we refer to either as the node degree. A *path* in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a sequence of distinct nodes $(v_1, v_2, \dots v_k)$, with $(v_i, v_{i+1}) \in \mathcal{E}$ for all $i = 1, \dots, k-1$. We denote by $\mathcal{G} \equiv \mathcal{G}_t = (\mathcal{V}, \mathcal{E}^{(t)})$ a network with $|\mathcal{V}|$ nodes (the nodes are fixed) such that the edges $\mathcal{E}^{(t)} \subset \mathcal{V} \times \mathcal{V}$ can vary with time (t).

A common representation of a digraph with n nodes is via its adjacency matrix $A \in \mathbb{R}^{n \times n}$, where $A_{u,v} = 1$ if $(u, v) \in \mathcal{E}$, and $A_{u,v} = 0$, otherwise. A *subdigraph* or a *subnetwork* $\mathcal{H} = (\mathcal{V}', \mathcal{E}')$ of a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a digraph such that $\mathcal{V}' \subset \mathcal{V}$, and $\mathcal{E}' \subset \left(\mathcal{V}' \times \mathcal{V}' \cap \mathcal{E}\right)$. By $\mathcal{G} \setminus \mathcal{A}$, with $\mathcal{A} \subset \mathcal{V}$, we denote the subdigraph $\mathcal{H} = (\mathcal{V} \setminus \mathcal{A}, \mathcal{E}')$, where $\mathcal{E}' = \{(u, v) \in \mathcal{E} : u, v \notin \mathcal{A}\}$.

Last, we use the ceiling function $\lceil \cdot \rceil : \mathbb{R} \to \mathbb{Z}$ that can be defined as $\lceil x \rceil = \arg\min_{y \in \mathbb{Z}} y$ such that $x \leq y$.

2. A reputation-based resilient continuous-time consensus method using a switching topology

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{(t)})$ be a network with n agents. The agents can reach consensus following a distributed and linear algorithm with dynamics given by:

$$\dot{x}_i(t) = -\sum_{j\in\mathcal{N}_i} w_{ij}(x_i(t) - x_j(t)), \tag{1}$$

where w_{ij} is 0 if agent j does not communicate with agent i, and $w_{ij} > 0$ otherwise, and $x_i(0) = x_i^0$.

Suppose now that we may have a set of agents, $\mathcal{A} \subset \mathcal{V}$, that can be attacked or malfunctioning. In this situation, it is desirable that the remaining agents in $\mathcal{V} \setminus \mathcal{A}$ are able to disclose which of their neighbor agents are not following the consensus protocol and discard the erroneous information.

In this paper, we tackle the aforementioned problem under the next assumption, which states that more than half of a normal agent neighbors are also normal, and the network of normal agents is connected (if undirected) or strongly connected (if directed).

Assumption $\mathbf{A_1}$: For every agent $u \in \mathcal{V} \setminus \mathcal{A}$ that is not malfunctioning, we have that $|\mathcal{N}_u \setminus \mathcal{A}| \geq |\mathcal{N}_u|/2$. Additionally, we have that $\mathcal{G} \setminus \mathcal{A}$ is connected (if $\mathcal{G}$ is undirected) or strongly connected (if $\mathcal{G}$ is directed). ⋄

2.1. The attacker model

In this work, we consider an attacker that may corrupt the states of a subset of agents $\mathcal{A}$ using an unbounded signal. We assume that the attacker cannot corrupt the communication between agents and send distinct messages to different neighbors. Note that this assumption permits the attacker to change the state of a subset of agents to (possibly) distinct values. For instance, in a network with a dozen agents, the attacker may modify the states of agents 1 and 2, independently. However, it cannot alter the network communication scheme and have two neighbors of an agent receiving different value. Such a scenario is expected in a wireless medium.

Moreover, the attacker cannot create artificial nodes nor change the network topology. Notice that if a malicious entity could create nodes in the network, it would be impossible to deter, as the attacked nodes could become the majority.

Additionally, we do not allow the always undetectable scenario where the attacker targets the initial state of an agent, see *Definition 3 (Undetectable Input)* of [30]. Observe that the state evolution of the network agents in a scenario where the initial state of an agent is changed follows precisely the same execution trace as the same scenario where the actual initial state of the agents is the same value as the attacker changed.

2.2. The proposed method

Now, we propose a switching system that uses the notion of reputation to achieve resilient consensus. The continuous-time dynamics of the method consists in each mode of the switching system, i.e., when agents states are evolving according to (1). The discrete-time behaviour happens at sampling times when each agent calculates a reputation score for each of its neighbors and uses is to (possibly) change its communication with the neighbors, i.e., to switch the network topology by (possibly) removing a set of edges corresponding to those with poor reputation. In fact, each agent considers only the information of the neighbor (or the neighbors) with the maximum reputation (up to $\varepsilon > 0$ but $\varepsilon \approx 0$).

The dynamics of each agent in a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is given as

$$\dot{x}_i(t) = -\sum_{j\in\mathcal{N}_i} c_{ij}(\sigma(t))w_{ij}(x_i(t) - x_j(t)), \tag{2}$$

where $\sigma : \mathbb{R}_0^+ \to \{1, \dots, m, \dots\}$ is a piece-wise constant signal that only switches once in a given dwell-time, and $c_{ij}(\sigma(t))$ is the reputation that agent $i \in \mathcal{V}$ assigns to agent $j \in \mathcal{N}_i$ at time $t \in \mathbb{R}_0^+$, computed as we next detail.

Given the dwell-times where the function σ is constant, $\{[t_0, t_1), \dots, [t_k, t_{k+1}), \dots\}$, with $t_0 = 0$, we compute $c_{ij}(\sigma(t))$, for $t \in [t_k, t_{k+1})$, using Algorithm 1.

Algorithm 1 Node's neighbors reputation assignment

1: **input**: function σ, time $t \in [t_k, t_{k+1})$, agent $i \in \mathcal{V}$ and its neighbors $\mathcal{N}_i$ states $\{x_i(t_k)\} \cup \left(\{x_j(t_k)\}_{j\in\mathcal{N}_i}\right)$, and the smallest floating point number that we can represent with a selected precision, ε.
2: **output**: reputations $c_{ij}(\sigma(t)) \equiv c_{ij}(\sigma(t_k))$, for all $j \in \{i\} \cup \mathcal{N}_i$ and for all $t \in [t_k, t_{k+1})$
3: **if** $t_k = 0$ **then**
4: **set**

$$c_{ij}(t_k) = 1,\ \forall j \in \{i\} \cup \mathcal{N}_i$$

5: **else**
6: **set** ▷ Reputation computation

$$\tilde{c}_{ij}(\sigma(t_k)) = -\sum_{v\in\mathcal{N}_i\cup\{i\}} \frac{|x_j(t_k) - x_v(t_k)|}{|\mathcal{N}_i|},\ \forall j \in \mathcal{N}_i$$

7: **set** ▷ Normalized Reputation update

$$\tilde{\tilde{c}}_{ij}(\sigma(t_k)) = \frac{\tilde{c}_{ij}(\sigma(t_k)) - \min\limits_{v\in\mathcal{N}_i} \tilde{c}_{iv}(\sigma(t_k))}{\max\limits_{v\in\mathcal{N}_i} \tilde{c}_{iv}(\sigma(t_k)) - \min\limits_{v\in\mathcal{N}_i} \tilde{c}_{iv}(\sigma(t_k))},\ \forall j \in \mathcal{N}_i$$

8: **set** ▷ Final Reputation update

$$c_{ij}(\sigma(t_k)) = \begin{cases} 1, & \text{if } i = j \vee \tilde{\tilde{c}}_{ij}(\sigma(t_k)) \geq 1 - \varepsilon \\ 0, & \text{otherwise} \end{cases},\ \forall j \in \{i\} \cup \mathcal{N}_i$$

9: **end if**

Furthermore, it is worth noticing that the derived consensus value of normal agents has the usual guarantees for resilient consensus, i.e., the final consensus lies within the convex hull of the initial agents' states [31, 32, 33]. This result is a consequence of having a continuous-time consensus

method with switching network topology (3). In fact, what we propose, in this work, is a reputation computation step which triggers the switching of the network topology.

The next result shows that the normal agents are able to correctly identify the attacked ones when each attacked agent aims to drive (in the limit) the consensus value to a common value (different from the non attacked scenario). Other scenarios are illustrated in Section 3. Note that the required normalization and "binarization" (Step 8) key steps in Algorithm 1 make a general proof hard, being out of scope of this paper.

Theorem 1. *Consider a network of agents $\mathcal{V}$ with a subset of agents $\mathcal{A} \subset \mathcal{V}$, satisfying assumption $\mathbf{A_1}$, that do not behave as normal agents. If, by using (2) with Algorithm 1, for $a \in \mathcal{A}$ $\lim_{t\to\infty} x_a(t) = x_a$ and for $v \in \mathcal{V} \setminus \mathcal{A}$ $\lim_{t\to\infty} x_v(t) = x_\infty \neq x_a$ then the agents behaving normally identify the attacked agents, i.e., assign them reputation equal to zero.* ∘

Proof. Suppose, by contradiction, that the conditions of the theorem hold but the agents behaving normally do not identify the attacked agents, i.e., assigning reputation of 1. The previous implies there is a normal agent that assigns a reputation to an attacked neighbor agent greater or equal to the normal neighbor agents. In other words, for each $a \in \mathcal{A} \cap \mathcal{N}_i$ and $j \in \mathcal{N}_i \setminus \mathcal{A}$, we have that

$$\lim_{t\to\infty} \tilde{c}_{ia}(\sigma(t)) \geq \lim_{t\to\infty} \tilde{c}_{ij}(\sigma(t)),$$

which is equivalent to

$$\lim_{k\to\infty} 1 - \sum_{v\in\bar{\mathcal{N}}_i} \frac{|x_a(t_k) - x_v(t_k)|}{|\mathcal{N}_i|} \geq \lim_{k\to\infty} 1 - \sum_{v\in\bar{\mathcal{N}}_i} \frac{|x_j(t_k) - x_v(t_k)|}{|\mathcal{N}_i|}.$$

Now, the previous equation is equivalent to

$$\lim_{k\to\infty} \sum_{v\in\bar{\mathcal{N}}_i} |x_a(t_k) - x_v(t_k)| \leq \lim_{k\to\infty} \sum_{v\in\bar{\mathcal{N}}_i} |x_j(t_k) - x_v(t_k)|,$$

which we can re-write as

$$\lim_{k\to\infty} \sum_{\substack{v\in\bar{\mathcal{N}}_i \\ v\notin\mathcal{A}}} |x_a(t_k) - x_v(t_k)| + \lim_{k\to\infty} \sum_{\substack{v\in\bar{\mathcal{N}}_i \\ v\in\mathcal{A}}} |x_a(t_k) - x_v(t_k)| \leq \lim_{k\to\infty} \sum_{\substack{v\in\bar{\mathcal{N}}_i \\ v\notin\mathcal{A}}} |x_j(t_k) - x_v(t_k)| + \lim_{k\to\infty} \sum_{\substack{v\in\bar{\mathcal{N}}_i \\ v\in\mathcal{A}}} |x_j(t_k) - x_v(t_k)|,$$

equivalent to

$$|\mathcal{N}_i \setminus \mathcal{A}||x_a - x_\infty| + |\mathcal{N}_i \cap \mathcal{A}||x_a - x_a| \leq |\mathcal{N}_i \setminus \mathcal{A}||x_\infty - x_\infty| + |\mathcal{N}_i \cap \mathcal{A}||x_\infty - x_a|,$$

and, finally, the same as $|\mathcal{N}_i \setminus \mathcal{A}| \leq |\mathcal{N}_i \cap \mathcal{A}|$, which contradicts assumption $\mathbf{A_1}$. □

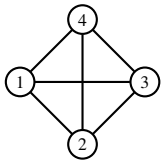
Remark: For each agent $i \in \mathcal{V}$, the computational cost of the discrete-time step of the consensus, i.e., the cost of Algorithm 1, is $\mathcal{O}(|\mathcal{N}_i|^2)$. This cost corresponds to step 7, the step of maximum cost. In this step, the agent i computes $\tilde{c}_{ij}(\sigma(t))$ for each $j \in \mathcal{N}_i$ with cost $\mathcal{O}(|\mathcal{N}_i|)$. ⋄

Next, we illustrate the use of the proposed method with several examples.

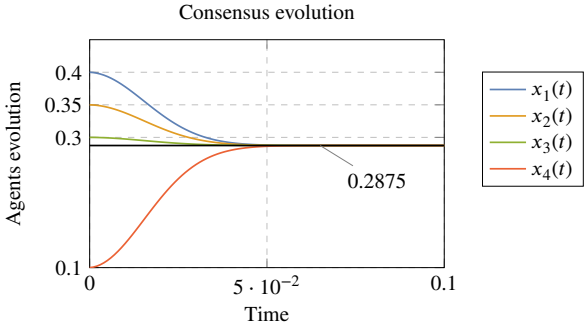
3. Illustrative examples

We start to explore the scenario where the attacked agents behave as stubborn agents, i.e., always sharing the same value. For all the experiments, we fixed $\varepsilon = 10^{-16}$ in Algorithm 1.

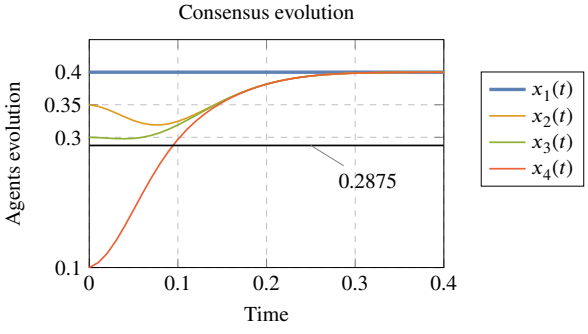
3.1. Attacked agents that share a constant value – undirected network



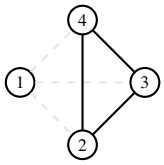
(a) Network topology graph $\mathcal{G}_1$.



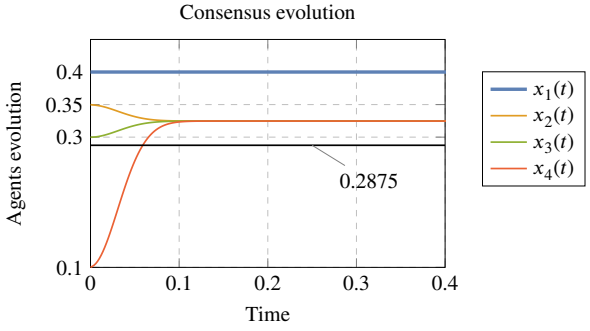
(b) State evolution without an attacker, i.e., $\mathcal{A} = \emptyset$.



(c) State evolution with an attacker, i.e., $\mathcal{A} = \{1\}$, and without resilience.



(d) Network resulting from (2) with Algorithm 1 for $\mathcal{G}_1$ with attacked nodes $\mathcal{A} = \{1\}$.



(e) State evolution with an attacker, $\mathcal{A} = \{1\}$, and (2) with Algorithm 1.

Figure 1: State evolution of agents from network $\mathcal{G}_1$ with initial states $x(0)$ for different configurations of the attacked agent set $\mathcal{A}$.

In the first example, we consider a complete network of 4 agents, $\mathcal{G}_1$, as depicted in Figure 1 (a), and initial state $x(0) = \begin{bmatrix}0.4 & 0.35 & 0.3 & 0.1\end{bmatrix}^\intercal$.

In Figure 1 (b), we show the agents' states evolution when there are not attacked agents. Subsequently, we consider that agent 1 is attacked and always communicates the same value. In this scenario, using a continuous-time consensus that is not resilient to attacks, the evolution of the agents' sates is depicted in Figure 1 (c), and the attack takes effect. Using (2) with Algorithm 1, we obtain the evolution depicted in Figure 1 (e) and the attack is, therefore, mitigated. The final communication network considered by the agents is depicted in Figure 1 (d).

Subsequently, consider the network of 10 agents $\mathcal{G}_2$, depicted in Figure 2 (a) and initial state

$$x(0) = [\,0.4\ 0.35\ 0.3\ 0.1\ -0.2\ -1.4\ 2.3\ 1\ 0.8\ 0.5\,]^\intercal .$$

Next, using (2) with Algorithm 1 and attackers $\mathcal{A} = \{4\}$, we obtain the evolution depicted in Figure 2 (b), and the attack is successfully mitigated. Furthermore, using (2) with Algorithm 1, we obtain the evolution depicted in Figure 2 (c) and the attack is successfully mitigated, when $\mathcal{A} = \{4, 8\}$.

3.2. Attacked agents that share a constant value - directed network

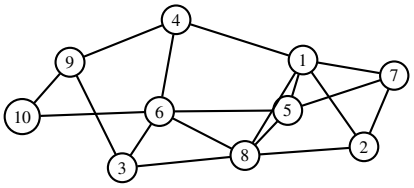
The following example illustrates the scenario where the network of agents is directed. We consider a network of 8 agents, $\mathcal{G}_3$, as depicted in Figure 3 (a), with initial state

$$x(0) = [\,0.4\ 0.35\ 0.3\ 0.1\ -0.2\ -1.4\ 2.3\ 1\ 0.8\ 0.5\,]^\intercal .$$

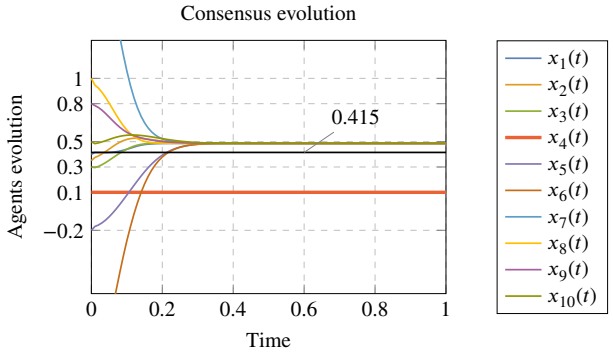
Next, using (2) with Algorithm 1, we obtain the evolution depicted in Figure 3 (b) and the attack is, therefore, mitigated.

3.3. Attacked agents that do not share a constant value

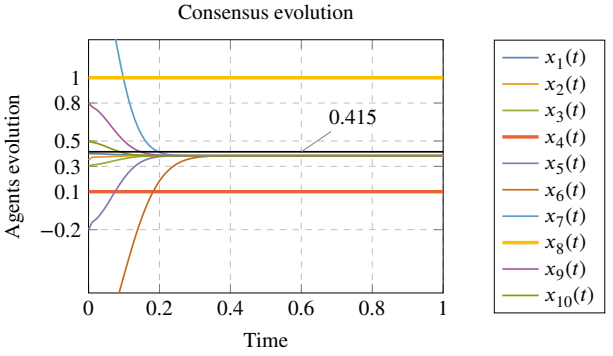
Here, we explore examples where the attacked agents do not behave as stubborn agents, i.e., that share a non-constant



(a) Network topology graph $\mathcal{G}_2$.



(b) State evolution using (2) with Algorithm 1, with attacker set $\mathcal{A} = \{4\}$.



(c) State evolution using (2) with Algorithm 1, with attacker set $\mathcal{A} = \{4, 8\}$.

Figure 2: State evolution of agents from network $\mathcal{G}_2$ with initial states $x(0)$, with attacked nodes $\mathcal{A}$.

value. In the first example, we use the network of agents $\mathcal{G}_1$, depicted in Figure 1 (a), set of attacked agents $\mathcal{A} = \{1\}$, and initial state $x(0) = \begin{bmatrix}0.4 & 0.35 & 0.3 & 0.1\end{bmatrix}^\intercal$. The state of agent 1 evolves as $x_1(t) = \frac{0.05\sin(40t)}{t+0.1} + 0.1$. In Figure 4, we depict the evolution of the agents' states using (2) with Algorithm 1, and the attack is successfully deterred.

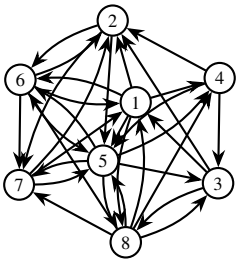
In the second example, we use the network of agents $\mathcal{G}_4$, depicted in Figure 5 (a), set of attacked agents $\mathcal{A} = \{4, 8\}$, and initial state

$$x(0) = [\,0.4\ 0.35\ 0.3\ 0.1\ -0.2\ -1.4\ 2.3\ 1\ 0.8\ 0.5\,]^\intercal .$$

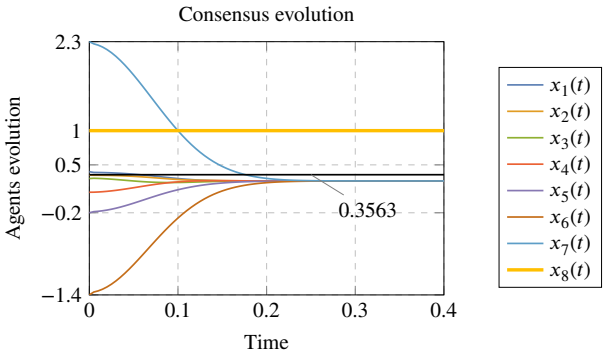
In Figure 5 (b), we depict the evolution of the agents' states using (2) with Algorithm 1, and the attack is, again, successfully deterred.

3.4. Influence of function σ

A question that emerges is if the role of the dwell-time of σ is crucial in the detection of attacked agents. We explore the



(a) Directed network $\mathcal{G}_3$



(b) $\mathcal{A} = \{8\}$

Figure 3: State evolution of agents from network $\mathcal{G}_3$ with initial states $x(0)$, with attacked nodes $\mathcal{A}$.

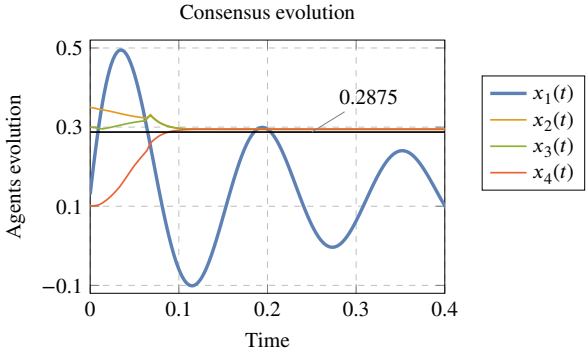
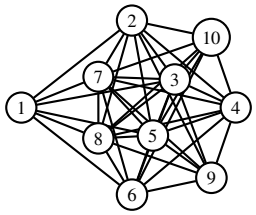


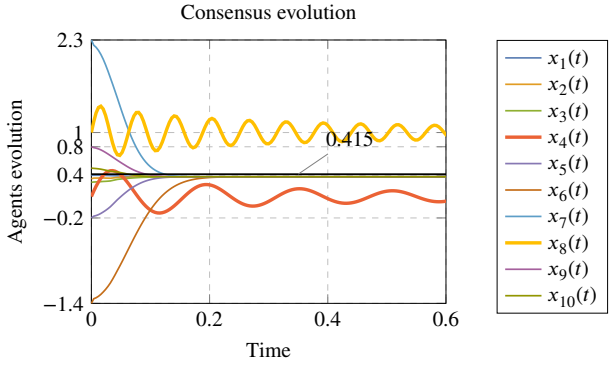
Figure 4: State evolution of agents from network $\mathcal{G}_1$ with initial states $x(0)$, with attacked nodes $\mathcal{A} = \{1\}$ and $x_1(t) = \frac{0.05\sin(40t)}{t+0.1} + 0.1$.

example of network $\mathcal{G}_1$, with attacked nodes $\mathcal{A} = \{1\}$, and initial state $x(0) = \begin{bmatrix}0.4 & 0.35 & 0.3 & 0.1\end{bmatrix}^\intercal$. In Figure 1 (e), this scenario was explored when $\sigma(t) = \lceil 100t \rceil$. Hence, we illustrate two extreme scenarios, where $\sigma(t) = \lceil 5t \rceil$ (depicted in Figure 6) and $\sigma(t) = \lceil 500t \rceil$, presented in Figure 7 (a) and (b), respectively.

We observe that in the three cases, the attack was successfully deterred. Notwithstanding, when the dwell-time function σ has a smaller number of discontinuities, the convergence is slower and the state evolution of the agents presents, consequently, more evident jumps.



(a) Network $\mathcal{G}_4$



(b) $\mathcal{A} = \{4, 8\}$

Figure 5: State evolution of agents from network $\mathcal{G}_4$ with initial states $x(0)$, with attacked nodes $\mathcal{A}$, $x_4(t) = \frac{0.05\sin(40t)}{t+0.1} + 0.1$, and $x_8(t) = \frac{0.08\sin(100t)}{t+0.2} + 1$.

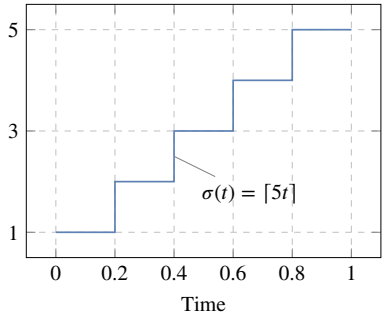


Figure 6: Plot of the function $\sigma(t) = \lceil 5t \rceil$.

3.5. Switching network topology

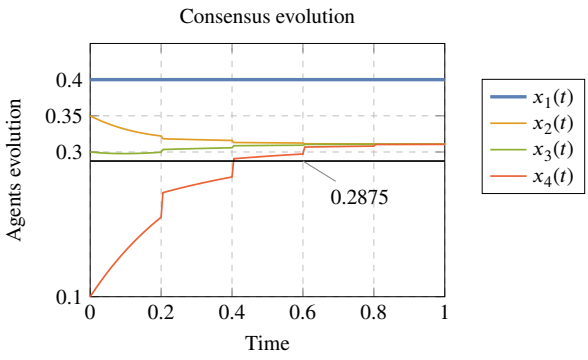
Finally, we explore how the presented method behaves if the consensus network has a switching topology (same number of agents but the edges may switch with time). Therefore, the agents' evolution is governed by the following:

$$\dot{x}_i(t) = -\sum_{j \in \mathcal{N}_i} w(t)_{ij}(x_i(t) - x_j(t)), \tag{3}$$

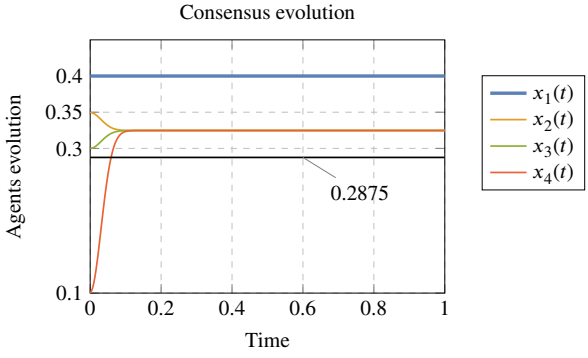
where $w(t)_{ij}$ is a piece-wise constant function such that $w(t)_{ij}$ is 0 if agent j does not communicate with agent j at time t, and $w(t)_{ij} > 0$ otherwise, and $x_i(0) = x_i^0$.

We consider the network with switching topology given as $\mathcal{G}(t) = \begin{cases} \mathcal{G}_{t\leq 0.4} & \text{if } t \leq 0.4 \\ \mathcal{G}_{t>0.4} & \text{if } t > 0.4 \end{cases}$, depicted in Figure 8 (a) and (b), and initial state

$$x(0) = [\,0.4\ 0.35\ 0.3\ 0.1\ -0.2\ -1.4\ 2.3\ 1\ 0.8\ 0.5\,]^\intercal.$$

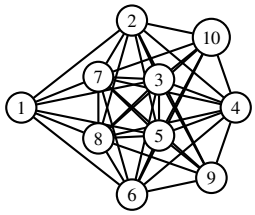


(a) $\sigma(t) = \lceil 5t \rceil$.

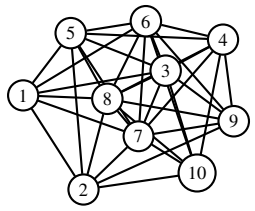


(b) $\sigma(t) = \lceil 500t \rceil$.

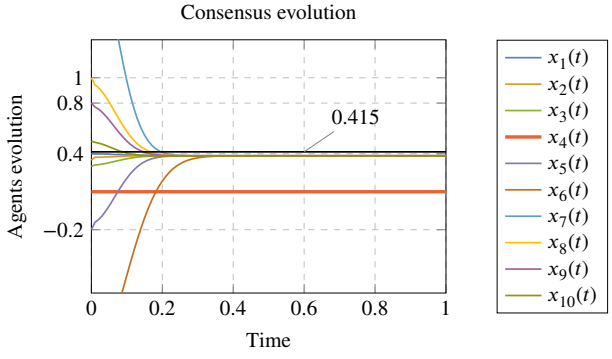
Figure 7: Influence of function σ in the example of network $\mathcal{G}_1$, with attacked nodes $\mathcal{A} = \{1\}$.



(a) $\mathcal{G}_{t\leq 0.4}$



(b) $\mathcal{G}_{t>0.4}$



(c) $\mathcal{A} = \{4\}$

Figure 8: State evolution of agents from network switching topology networks $\mathcal{G}(t)$, with initial states $x(0)$, and attacked nodes $\mathcal{A}$.

Also in this setup, as envisioned, the proposed method is able to mitigate the attack, see Figure 8 (c).

3.6. Finding the best attacking strategy

We explore the best strategy that an attacker can follow to maximize the change of the consensus value without being detected. This scenario helps to understand the limits of both the proposed method and the limits of an attacking strategy. That said, we assume that an attacker has full knowledge of the system at any time.

3.6.1. Best attacking strategy for our approach

Using the full knowledge of the system that evolves according to (3), with c_{ij} updated according to Algorithm 1, with aimed final consensus value $\bar{x}$, an attacker can select its state $x_a(t_k) = z^*(t_k)$, to share in each mode of the σ function, by solving the following optimization

$$\begin{aligned} z^*(t_k) \quad &= \quad \arg\max_{z\in\mathbb{R}} (z-\bar{x})^2 \\ \text{s.t.} \quad &\text{for some } i, \text{ with } a \in \mathcal{N}_i, \\ &\tilde{c}_{ia}(\sigma(t_k)) \geq \tilde{c}_{ij}(\sigma(t_k)),\ \forall j \in \mathcal{N}_i \setminus \{a\}. \end{aligned}$$

Intuitively, the attacker wants to maximize its influence on the final consensus value while remaining undetected by at least one of the agents to which it communicates. This strategy implies that the attacker must have a reputation as good as the best reputation of that agent's neighbors. Also, intuitively, we must select a σ function that initially changes with a large frequency, and that changes a few times after that. This selection reflects the fact that as the agents' states are evolving to consensus, the maximum influence on the consensus outcome that the attacker can do without being detected is going to zero.

3.6.2. Best attacking strategy for approaches that eliminate extreme values

Now, we explore the best attacking scenario for the previous typically used approaches. In the literature, whenever designing a resilient consensus system, the concept of mean subsequence reduce (MSR) is often used to remove the extreme points from the ones received by all the neighbors [34, 35, 36]. Specifically, for each mode of the system, an agents sorts the values received by its neighbors and discard the f highest and the f smallest values (where f is a robustness parameter). Therefore, we consider each node to apply the aforementioned filtering in each discrete-time transition of the system.

It is easy to see that if $f = 1$ then an attacking strategy where the attacked node chooses its value to be $\epsilon > 0$ smaller than the maximum value of all the nodes or $\epsilon > 0$ greater than the minimum value of all the nodes will make the attacker undetectable for some agent, yielding the best attacking strategy with the goal of being undetectable for at least one agent.

3.6.3. *Example of a best attacking strategy for the two previous scenarios*

We start by fixing a network of agents $\mathcal{G}$ (Figure 9) and a set of initial values

$$x(0) = [\,1\ 4\ 9\ 16\ 25\ 36\ 49\ 64\ 81\ 100\,]^\intercal$$

and deploy the best above mentioned attacking strategies considering two σ functions. The first is $\sigma_1 = \lceil 25t \rceil$, and the second is $\sigma_2 = \lceil 40t \rceil$.

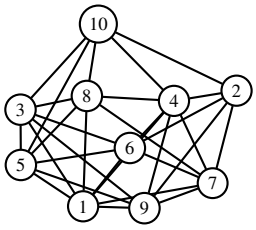


Figure 9: Network $\mathcal{G}$.

We can see, from Figures 10 and 11 that the best undetectable attack is able to change the final consensus value more when using MSR than with our proposed method. In other words, the strategy that we propose further limits the attacking strategy when the attacked agent wants to be undetected to some other agent.

3.6.4. *A comparison between the approaches*

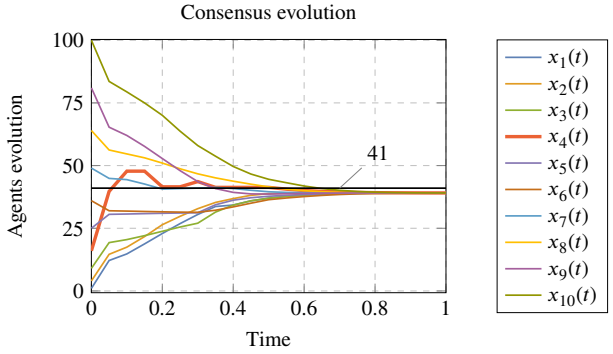
Finally, we fix the initial states of the agents as

$$x(0) = [\,1\ 2.297\ 3.737\ 5.278\ 6.899\ 8.586\ 10.33\ 12.126\ 13.967\ 15.849\,]^\intercal,$$

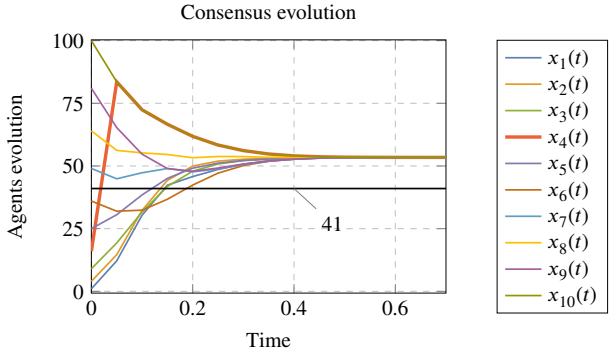
we generate 30 networks of 10 agents, apply the best attacking strategy for the MSR and compute the absolute error between the final true consensus value and the final consensus obtained with the proposed approach (ours) and the MSR, with $f = 1$. In Figure 12, we present a box-whisker chart of the absolute error distribution when using both approaches that illustrates the advantage of the proposed method.

4. Conclusion

In this paper, we addressed the problem of a set of agents achieving resilient continuous-time consensus. To this end, we developed a method that consists of a switching system, which switches between network topologies (in discrete-time), and, in each mode of the system, the agents follow a



(a) Best attacking strategy for **our approach**, with $\mathcal{A} = \{4\}$ and $\sigma \equiv \sigma_1$.



(b) Best attacking strategy for the MSR, with $\mathcal{A} = \{4\}$ and $\sigma \equiv \sigma_1$.

Figure 10: Consensus evolution using the best (undetectable) attacking strategy using the proposed approach vs. the MSR, with $\sigma \equiv \sigma_1$.

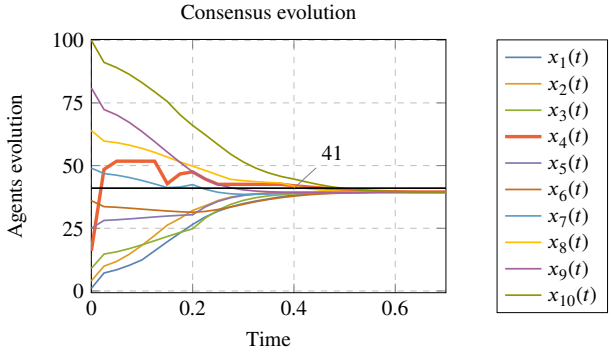
typical continuous-time consensus using the network topology of that mode. Moreover, the discrete-time part consists of each agent computing a reputation score that it assigns to each neighbor. This reputation score reflects the mean error of that neighbor's state regarding other neighbors' states. Then, the score is used to exclude a subset of suspicious neighbors, changing the network topology. Lastly, we explore the proposed method with illustrative examples, and compare with the MSR method, illustrating the advantages of our approach.

CRediT authorship contribution statement

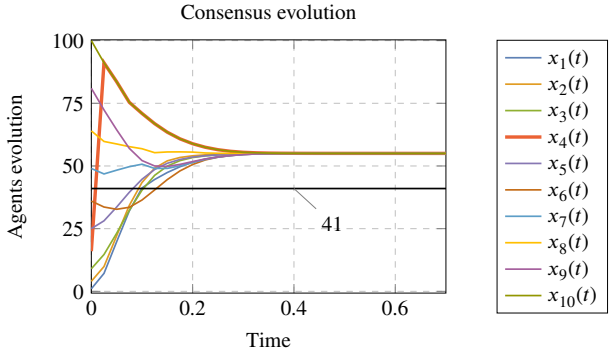
Guilherme Ramos: Conceptualization, Writing - original draft, Investigation, Methodology, Software, Writing - review & editing, Validation, Formal analysis. **Daniel Silvestre:** Supervision, Writing - review & editing. **A. Pedro Aguiar:** Supervision, Writing - review & editing, Funding acquisition.

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) Best attacking strategy for **our approach**, with $\mathcal{A} = \{4\}$ and $\sigma \equiv \sigma_2$.



(b) Best attacking strategy for the MSR, with $\mathcal{A} = \{4\}$ and $\sigma \equiv \sigma_2$.

Figure 11: Consensus evolution using the best (undetectable) attacking strategy using the proposed approach vs. the MSR, with $\sigma \equiv \sigma_2$.

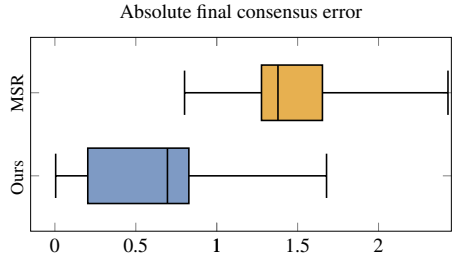


Figure 12: Box-whisker chart of the absolute error distribution when using both approaches with the best attacking strategy for the MSR.

References

[1] J. Tsitsiklis, D. Bertsekas, M. Athans, Distributed asynchronous deterministic and stochastic gradient optimization algorithms, IEEE Transactions on Automatic Control 31 (9) (1986) 803–812. doi:10.1109/TAC.1986.1104412.

[2] 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. doi:10.1109/CDC.2008.4739339.

[3] A. Jadbabaie, J. Lin, A. S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Transactions on automatic control 48 (6) (2003) 988–1001.

[4] A. Alessandretti, A. P. Aguiar, An optimization-based cooperative path-following framework for multiple robotic vehicles, IEEE Transactions on Control of Network Systems 7 (2) (2019) 1002–1014.

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

[6] R. Ribeiro, D. Silvestre, C. Silvestre, A rendezvous algorithm for multi-agent systems in disconnected network topologies, in: 2020 28th Mediterranean Conference on Control and Automation (MED), 2020, pp. 592–597. doi:10.1109/MED48518.2020.9183093.

[7] R. Ribeiro, D. Silvestre, C. Silvestre, Decentralized control for multi-agent missions based on flocking rules, in: J. A. Gonçalves, M. Braz-César, J. P. Coelho (Eds.), CONTROLO 2020, Springer International Publishing, Cham, 2021, pp. 445–454.

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

[9] D. Silvestre, J. Hespanha, C. Silvestre, Desynchronization for decentralized medium access control based on gauss-seidel iterations, in: 2019 American Control Conference (ACC), 2019, pp. 4049–4054. doi:10.23919/ACC.2019.8814471.

[10] D. Silvestre, J. Hespanha, C. Silvestre, A pagerank algorithm based on asynchronous gauss-seidel iterations, in: American Control Conference (ACC), 2018, pp. 484–489. doi:10.23919/ACC.2018.8431212.

[11] R. Olfati-Saber, Distributed Kalman filtering for sensor networks, in: 46th IEEE Conference on Decision and Control (CDC), 2007, pp. 5492–5498. doi:10.1109/CDC.2007.4434303.

[12] P. Alriksson, A. Rantzer, Experimental evaluation of a distributed kalman filter algorithm, in: 46th IEEE Conference on Decision and Control (CDC), 2007, pp. 5499–5504. doi:10.1109/CDC.2007.4434590.

[13] H. J. LeBlanc, H. Zhang, S. Sundaram, X. Koutsoukos, Resilient continuous-time consensus in fractional robust networks, in: 2013 American Control Conference, IEEE, 2013, pp. 1237–1242.

[14] H. J. LeBlanc, X. Koutsoukos, Resilient first-order consensus and weakly stable, higher order synchronization of continuous-time networked multiagent systems, IEEE Transactions on Control of Network Systems 5 (3) (2017) 1219–1231.

[15] Y. Shang, Resilient consensus of switched multi-agent systems, Systems & Control Letters 122 (2018) 12–18.

[16] G. Ramos, D. Silvestre, C. Silvestre, General resilient consensus algorithms, International Journal of Control 0 (0) (2020) 1–15.

[17] G. Ramos, D. Silvestre, C. Silvestre, A general discrete-time method to achieve resilience in consensus algorithms, in: 2020 59th IEEE Conference on Decision and Control (CDC), 2020, pp. 2702–2707. doi:10.1109/CDC42340.2020.9304107.

[18] D. Silvestre, J. P. Hespanha, C. Silvestre, Resilient desynchronization for decentralized medium access control, IEEE Control Systems Letters 5 (3) (2021) 803–808. doi:10.1109/LCSYS.2020.3005819.

[19] R.-H. Li, J. Xu Yu, X. Huang, H. Cheng, Robust reputation-based ranking on bipartite rating networks, in: Proceedings of the 2012 SIAM international conference on data mining, SIAM, 2012, pp. 612–623.

[20] J. Saúde, G. Ramos, L. Boratto, C. Caleiro, A robust reputation-based group ranking system and its resistance to bribery, ACM Transactions on Knowledge Discovery from Data (TKDD) 16 (2) (2021) 1–35.

[21] 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.

[22] G. Ramos, L. Boratto, Reputation (in)dependence in ranking systems: Demographics influence over output disparities, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, 2020, p. 2061–2064.

[23] D. Silvestre, Reputation-based method to deal with bad sensor data, IEEE Control Systems Letters (2020).

[24] A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decision support systems 43 (2) (2007) 618–644.

[25] C. Zhu, H. Nicanfar, V. C. Leung, L. T. Yang, An authenticated trust and reputation calculation and management system for cloud and sensor networks integration, IEEE Transactions on Information Forensics and Security 10 (1) (2015) 118–131.

[26] G. Ramos, D. Silvestre, C. Silvestre, Node and network resistance to bribery in multi-agent systems, Systems & Control Letters 147 (2021) 104842.

[27] J. M. Pujol, R. Sangüesa, J. Delgado, Extracting reputation in multi agent systems by means of social network topology, in: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, ACM, 2002, pp. 467–474.

[28] G. Ramos, D. Silvestre, C. Silvestre, A discrete-time reputation-based resilient consensus algorithm for synchronous or asynchronous communications, arXiv preprint arXiv:2107.00431 (2021).

[29] F. Hendrikx, K. Bubendorfer, R. Chard, Reputation systems: A survey and taxonomy, Journal of Parallel and Distributed Computing 75 (2015) 184–197.

[30] F. Pasqualetti, A. Bicchi, F. Bullo, Consensus computation in unreliable networks: A system theoretic approach, IEEE Transactions on Automatic Control 57 (1) (2011) 90–104.

[31] J. Usevitch, D. Panagou, Resilient leader-follower consensus to arbitrary reference values in time-varying graphs, IEEE Transactions on Automatic Control 65 (4) (2019) 1755–1762.

[32] F. Tan, D. Liu, X. Guan, Consensus value of multi-agent networked systems with time-delay, in: 2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, IEEE, 2009, pp. 179–184.

[33] M. Franceschelli, A. Giua, A. Pisano, Finite-time consensus on the median value with robustness properties, IEEE Transactions on Automatic Control 62 (4) (2016) 1652–1667.

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

[35] H. J. LeBlanc, X. Koutsoukos, Resilient asymptotic consensus in asynchronous robust networks, in: 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2012, IEEE, 2012, pp. 1742–1749.

[36] S. Sundaram, Ignoring extreme opinions in complex networks: The impact of heterogeneous thresholds, in: 55th IEEE Conference on Decision and Control (CDC), 2016, pp. 979–984. doi:10.1109/CDC.2016.7798395.