Distributed estimation and control of node centrality in undirected asymmetric networks

Eduardo Montijano, Gabriele Oliva and Andrea Gasparri

Abstract—Measures of node centrality that describe the importance of a node within a network are crucial for understanding the behavior of social networks and graphs. In this paper, we address the problems of distributed estimation and control of node centrality in undirected graphs with asymmetric weight values. In particular, we focus our attention on $\alpha$-centrality, which can be seen as a generalization of eigenvector centrality, particularly suitable for graphs with asymmetric interactions. In this setting, our contribution is twofold: first we derive a distributed protocol where agents can locally compute their $\alpha$-centrality index by means of local interactions; then, we focus our attention on the problem of controlling the weight matrix of the graph in such a way that the network reaches a desired $\alpha$-centrality vector. The interest of our solution lies in obtaining the control objective with minimum changes in the original influence matrix. Moreover, with our algorithm every agent is able to reach the desired value locally and in finite-time. The two algorithms are then applied to two problems of interest in real life. The estimation method is used together with a consensus algorithm to achieve a consensus value weighted by the influence of each node in the network. The control algorithm is exploited to protect the most valuable nodes in a network against a targeted attack, by making every node in the network equally important in terms of $\alpha$-centrality. Simulations results are provided to corroborate the theoretical findings.

I. INTRODUCTION

In graph theory and network analysis, identifying the most central nodes, i.e., the most important nodes within a graph, has been a very important research topic for a long time, see [1]–[3]. Notions of centrality were first developed in social network analysis, as witnessed by the sociological origin of several terms used to measure centrality [4]. Applications involving centrality concepts include, among others, identifying the most influential person(s) in a social network, finding key infrastructure nodes in the Internet or urban networks, and pinpointing super-spreaders of disease. Depending on the specific domain of interest, a variety of metrics have been proposed to measure the centrality of nodes in a network, ranging from node degree [5], eccentricity [6], closeness [7] and betweenness [8] to eigenvector centrality [9] and $\alpha$-centrality [10].

Recently, a few works which attempt to compute node centrality in a distributed fashion have been presented to the research community. In [11], a framework for the calculation of the betweenness-centrality is proposed. In particular, the authors demonstrate that their approach is asymptotically optimal in its communication complexity and that its time complexity scales with the diameter of the network. In [12], a distributed method is given to assess network closeness-centrality based only on localized information restricted to a given neighborhood around each node. In [13], [14], different distributed algorithms to compute betweenness and closeness centrality in a tree graph are proposed. In [15], the authors extend their previous results on closeness centrality to the case of general graphs, by formulating a set of linear inequality and equality constraints, which are distributed in nature. The distributed estimation of betweenness centrality is exploited in [16] to design an efficient routing protocol. The authors of [17] present a distributed method for the estimation of the Harmonic influence centrality [18], defined in the context of opinion dynamics.

One of the most famous measurements of centrality in networks is the PageRank, which is a modified version of eigenvector-centrality, of special interest in web ranking [19]. In [20], the authors propose a distributed algorithm for computing the eigenvector centrality, which accounts for both the lack of synchronicity and heterogeneity of agents in terms of clock rates. In [21], the authors propose deterministic finite-time algorithms for measuring degree, closeness, and betweenness centrality, along with a randomized algorithm for computing the PageRank. Recently, distributed Page-Rank estimation was computed by means of the Power method in [22]. It is worth mentioning that all these methods focus on the estimation of the centrality, but none of them considers the problem of controlling it, i.e., introducing control mechanisms to drive the centrality value to a specific state.

In this work, we focus our attention on the $\alpha$-centrality [10], which can be seen as a generalization of eigenvector centrality that is particularly suitable for networks with asymmetric interactions. Briefly, $\alpha$-centrality measures the total number of paths from a node, exponentially attenuated by their length, where the parameter $\alpha$ sets the length scale of interactions. Compared to other centrality metrics, an interesting property of $\alpha$-centrality is that it can be a tool to discriminate between locally and globally connected nodes; by locally connected nodes we mean nodes that are part of a community, in that their neighbors exhibit a large degree of mutual interconnection, while by globally connected nodes we mean nodes that interconnect poorly connected groups of nodes. Notably, studies on human beings [23] and animals [24] have provided evidence that these latter nodes, often recognized as “bridges” or “brokers”, play a crucial role in the information flow cohesiveness of the entire group.

Our contribution is then threefold: i) we design a distributed
protocol where agents, by means of local interactions, locally compute their \( \alpha \)-centrality index; ii) we introduce and solve an optimization problem that allows to control the value of the \( \alpha \)-centrality to a desired state using only local information and with minimum variation effort. To the best of our knowledge, this is the first work where the problem of controlling centrality is addressed; iii) we present two application scenarios where we use our methods to address two important topics in social networks, weighted consensus and vulnerability to external opinions. A preliminary version of this paper appeared in [25]. Compared to it, we have included additional details on the estimation procedure, as well as lifted an assumption on the associated application. The control solution, together with its application, have not been published before.

The rest of the paper is organized as follows. In Section II some background notions are provided. In Section III the proposed distributed algorithm to compute the \( \alpha \)-centrality index through local interactions is described. In Section IV we characterize the proposed control objective in terms of an optimization problem and we provide a local algorithm to allow the agents reach a desired \( \alpha \)-centrality. Two application scenarios of our methods are described in Section V. In Section VI we present simulation results. Finally, in Section VII we present the main conclusions of this work.

II. PRELIMINARIES

A. Notation

We denote matrices and vectors by boldface uppercase and lowercase letters, respectively. We refer to the \( i \)-th entry of a vector \( x \) by \( x_i \) and to the \( (i,j) \)-th entry of a matrix \( A \) by \( a_{ij} \). We represent by \( 0_n \) and \( I_n \) vectors with \( n \) components, all equal to zero and to one, respectively. Moreover, we denote by \( 0_{n \times n} \) and by \( I_n \) the \( n \times n \) zero and identity matrices, respectively. Given two vectors \( a,b \in \mathbb{R}^n \) we write \( a \geq b \) (respectively, \( a > b \)) if \( a_i \geq b_i \) (respectively, \( a_i > b_i \)) for all \( i \in \{1,\ldots,n\} \); similarly, we write \( A \geq B \) (\( A > B \)) if \( a_{ij} \geq b_{ij} \) (\( a_{ij} > b_{ij} \)) for all \( i,j \). Let \( H \) be an \( n \times m \) matrix; we denote \( \|H\|_F \) the Frobenius norm of \( H \). Finally, a saturation with upper and lower limits \( b \) and \( a \geq a \) is a function \( \text{sat} : \mathbb{R} \rightarrow \mathbb{R} \) defined as follows:

\[
\text{sat}(x,a,b) = \begin{cases} a & \text{if } x \leq a \\ b & \text{if } x \geq b \\ x & \text{otherwise}; \end{cases}
\]

B. Convex Constrained Optimization

We now briefly review the first-order Karush-Kuhn-Tucker (KKT) necessary and sufficient optimality conditions [26]. These conditions will prove useful later in the paper to develop a necessary and sufficient optimality condition for the control problem at hand. Based on our needs, we review only the special case where the objective function is convex and constraints are linear, i.e., we consider a minimization problem having the following structure:

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0, \quad \forall i \in \{1,\ldots,q\} \\
& \quad h_i(x) = 0, \quad \forall i \in \{q+1,\ldots,s\}.
\end{align*}
\]

where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is a convex function and all \( g_i : \mathbb{R}^n \rightarrow \mathbb{R} \) and \( h_i : \mathbb{R}^n \rightarrow \mathbb{R} \) are linear.

Theorem 1 (KKT Optimality Conditions). Consider a constrained optimization problem as in Eq. (1) and let the Lagrange function be defined as follows:

\[
\mathcal{L}(x,\zeta) = f(x) + \sum_{i=1}^{q} \zeta_i g_i(x) + \sum_{i=q+1}^{s} \zeta_i h_i(x) \tag{2}
\]

where \( \zeta = [\zeta_1,\ldots,\zeta_s]^T \) collects the Lagrange multipliers. A necessary and sufficient condition for a point \( x^* \in \mathbb{R}^n \) to be a global minimum is that there is \( \zeta^* \in \mathbb{R}^s \) such that

1) \( \nabla_x \mathcal{L}(x,\zeta)|_{x=x^*,\zeta=\zeta^*} = 0; \)
2) \( \zeta^*_i g_i(x^*) = 0, \quad \forall i = 1,\ldots,q; \)
3) \( g_i(x^*) \leq 0, \quad \forall i = 1,\ldots,q; \)
4) \( h_i(x^*) = 0, \quad \forall i = q+1,\ldots,s; \)
5) \( \zeta_i^* \geq 0, \quad \forall i = 1,\ldots,q. \)

C. Node centrality

Let us consider a network of \( N \) nodes labeled by \( i \in V \). The nodes exchange information with each other following a fixed undirected communication graph \( G = (V,E) \), where \( E \subseteq V \times V \) represents the edge set. In this way, nodes \( i \) and \( j \) can communicate if and only if \( (i,j) \in E \). We assume that the communication graph is connected. The set \( N_i \) of neighbors of node \( i \in V \) is the subset of nodes that can directly communicate with it, i.e., \( N_i = \{ j \in V \mid (i,j) \in E \} \). In addition, let us define \( \hat{N}_i = N_i \cup \{i\} \). Finally, we assume that the graph does not contain self loops.

Given a graph \( G = \{V,E\} \), let us define the set of matrices compatible with \( G \) as

\[
\mathcal{A}_G = \left\{ W \in \mathbb{R}^{|V|\times|V|} \mid w_{ij} = 0, \quad \forall (i,j) \notin E, i \neq j \right\}.
\]

In particular, we define \( W \in \mathcal{A}_G \) as the influence matrix associated to the network, where \( w_{ij} \geq 0 \) represents the influence that agent \( j \)'s information has for agent \( i \). Note that the influence matrix can be asymmetric, to model the fact that two neighboring agents can place different importance on the information provided by each other. The influence matrix can also contain values equal to zero between neighbors, meaning that agent \( j \) is completely disregarded by agent \( i \), even when they talk to each other. Finally, non-neighboring agents have mutual zero influence by construction, since they do not communicate.

As mentioned in the introduction, the notion of node centrality in graph theory is built around the idea of measuring how important a particular vertex is over a certain graph structure. This is typically expressed in terms of a function, \( \rho(W(G)) : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^+ \), that computes a vector where the
Assumption 2. \( \alpha > 0 \) is such that \( \| \alpha \mathbf{W} \| < 1 \).

Remark 1. Assumptions 1 and 2 are required to prove convergence of the proposed distributed protocol, as discussed in Section III. Note that, compared to previous literature (e.g., see [10]), where, in order to guarantee the existence of the inverse and non-negativity of the components of \( \rho_\alpha \), the parameter \( \alpha \) is required to be smaller that the reciprocal of the spectral radius of \( \mathbf{W} \), here due to the asymmetric nature of \( \mathbf{W} \), a more restrictive assumption on the choice of \( \alpha \) is required.

In the next section, we present the proposed distributed algorithm to estimate the nodes’ \( \alpha \)-centrality. After presenting the algorithm, we will discuss the implications of a particular choice of \( \alpha \), as well as a distributed procedure to choose a value of \( \alpha \) such that Assumption 2 is satisfied.

III. DISTRIBUTED ESTIMATION OF NODE \( \alpha \)-CENetRALITY

Algorithm 1 Proposed Algorithm to compute \( \alpha \)-centrality (from the point of view of agent \( i \))

\textbf{procedure} DISTRIBUTED\textbf{ALPHA\textbf{C}ENTRALITY} \\
\textbf{▷} Initial Condition \\
c\(_i\)(0) ← \( z_i \) \\
\( \Delta c_i(0) \leftarrow z_i \) \\
\textbf{▷} Synchronous Iteration \\
c\(_i\)(t + 1) = c\(_i\)(t) + \( \alpha \sum_{j \in \mathcal{N}_i} w_{ji} \Delta c_j(t) \) \\
\( \Delta c_i(t + 1) = c_i(t + 1) - c_i(t) \) \\
\textbf{end procedure}

In this section we present a distributed linear iteration protocol that allows each agent to compute its own value of \( \alpha \)-centrality; the pseudocode of the proposed protocol is given in Algorithm 1. Specifically, let \( c_i(t) \) be the estimation that agent \( i \) has of the \( i \)-th component of \( \rho_\alpha \) at iteration \( t \), which is initialized as \( c_i(0) = z_i \). Then, at each iteration, each agent executes the following update rule:

\[
c_i(t + 1) = c_i(t) + \alpha \sum_{j \in \mathcal{N}_i} w_{ji} \Delta c_j(t),
\]

for any \( t \geq 0 \), where

\[
\Delta c_i(t + 1) = c_i(t + 1) - c_i(t),
\]

represents the increments in the local estimations and \( \Delta c_i(0) = z_i \).

It is noteworthy that, compared to other typical linear protocols, Eq. (5) uses \( w_{ji} \) instead of \( w_{ij} \). This is the underlying motivation of requiring an undirected communication graph, even when \( \mathbf{W} \) is asymmetric, so that nodes can obtain these values by an initial communication round. This change is also imposed by the transpose of \( \mathbf{W} \) in Eq. (3).

Another difference with respect to standard linear iterations is the information exchanged by the agents at each communication round. Our algorithm exchanges variations of the estimation, i.e., \( \Delta c_i(t) \), rather than the actual estimation \( c_i(t) \). Nonetheless, in terms of bandwidth the algorithm has an equivalent cost, requiring the transmission of one scalar per communication round and agent.

The protocol in Eqs. (5)-(6) can be expressed in a compact form as

\[
c(t + 1) = c(t) + \alpha \mathbf{W}^T \Delta c(t),
\]

\[
\Delta c(t + 1) = c(t + 1) - c(t),
\]

which will prove useful for the subsequent convergence analysis. Additionally, in order to characterize the convergence properties of the algorithm, let

\[
e_c(t) = \| c(t) - \rho_\alpha \|
\]

be the estimation error for the whole centrality vector at iteration \( t \). With these definitions we can present the first result of the paper, that establishes the convergence of our algorithm and provides bounds on the estimation error at each iteration.

Theorem 2. For any matrix, \( \mathbf{W} \), satisfying Assumption 2 and any value of \( \alpha \) that satisfies Assumption 2, the execution of Algorithm 1 asymptotically leads to the distributed computation of node \( \alpha \)-centrality, i.e.,

\[
\lim_{t \to \infty} c(t) = \rho_\alpha.
\]

Besides, the estimation error, \( e(t) \), is upper-bounded by

\[
e_c(t) \leq \frac{\kappa^{t+1}}{1 - \kappa} \| z \|,
\]

with \( \kappa = \| \alpha \mathbf{W} \| \).

Proof. In order to demonstrate this result, we start by showing that the following equality holds,

\[
c(t) = \sum_{k=0}^{t} (\alpha \mathbf{W}^T)^k c(0).
\]

We prove this equality via induction. Clearly Eq. (12) holds...
for \( t = 0 \), since \( o^0(W)^0c(0) = c(0) \). Now, assume it is also true up to some general round \( t \) and let us show that it is also true for \( t + 1 \). Using Eq. (7) and Eq. (12)

\[
e(t + 1) = e(t) + αW^T δe(t)
\]

(13)

Developing the second term on the right-hand side of the last equality, using again Eq. (12), we obtain

\[
αW^T δe(t) = αW^T (c(t) − c(t − 1))
\]

(14)

\[
= αW^T \left( \sum_{k=0}^{t} (αW^T)^k c(0) − \sum_{k=0}^{t-1} (αW^T)^k c(0) \right)
\]

\[
= αW^T (αW^T)^t c(0) = (αW^T)^{t+1} c(0).
\]

Plugging this result into the previous equation, Eq. (12) holds for \( t + 1 \), concluding the induction.

Now, by Assumption 2 we can assert that the series in Eq. (12) is convergent, because \((αW^T)^t\) vanishes as \( t \) approaches to infinity. In particular,

\[
\lim_{t \to \infty} \sum_{k=0}^{t} (αW^T)^k = (I − αW^T)^{-1},
\]

(15)

which together with Eq. (12) and the fact that \( c(0) = z \) leads to Eq. (10), thus proving the convergence of our algorithm.

Regarding the error, firstly, recalling that \( c(0) = z \), using again Eq. (12) and the properties of geometric series let us develop the expression of the error,

\[
e_e(t) = \|c(t) − ρ_α\| = \left\| \sum_{k=0}^{t} (αW^T)^k c(0) − ρ_α \right\|.
\]

(16)

At this point, since

\[
\sum_{k=0}^{t} (αW^T)^k = \sum_{k=0}^{∞} (αW^T)^k - \sum_{k=t+1}^{∞} (αW^T)^k
\]

(17)

\[
= \sum_{k=0}^{∞} (αW^T)^k - (αW^T)^{t+1} \sum_{k=0}^{∞} (αW^T)^k
\]

\[
= (I - (αW^T)^{t+1}) (I - αW^T)^{-1},
\]

we have that

\[
e_e(t) = \left\| (I - (αW^T)^{t+1}) ρ_α - ρ_α \right\|
\]

(18)

\[
= \left\| (αW^T)^{t+1} ρ_α \right\|
\]

\[
\leq \left\| (αW^T)^{t+1} \right\| \|ρ_α\| ≤ \|αW^t+1\| \|ρ_α\|.
\]

Using Eq. (3), we can bound the norm of the \( α \)-centrality vector by

\[
\|ρ_α\| ≤ \|(I - αW^T)^{-1}\| \|z\|
\]

\[
= \sum_{i=0}^{∞} (αW^T)^i \|z\|
\]

\[
≤ \sum_{i=0}^{∞} \|αW^i\| \|z\| = \frac{1}{1 - \|αW\|} \|z\|;
\]

where the last equality follows from Assumption 2. At this point, by plugging the above result into Eq. (17), we get Eq. (11). In addition, by Assumption 2 it holds \( αW^t \|z\| < 1 \). Therefore, as \( t \) tends to infinity, \( e_e(t) \) converges to zero, with geometric convergence rate \( κ \).

Let us now discuss the influence of the parameter \( α \) on the estimation procedure.

On the one hand, according to Assumption 2 it follows that the parameter \( α \) needs to be “small enough” to ensure the convergence of the estimation algorithm. In order to obtain a distributed procedure for computing such parameter \( α \), it should be noticed that it holds

\[
\|W\| ≤ \sqrt{\|W\|_1},
\]

therefore, by choosing

\[
α < 1/(\sqrt{\|W\|_1} √{\|W\|_∞})
\]

(19)

it follows that

\[
\|αW\| = \|W\| < \frac{\|W\|}{\sqrt{\|W\|_1} √{\|W\|_∞}} < 1.
\]

thus Assumption 2 holds by construction. At this point, let us notice that the communication graph is undirected and the agents have knowledge of the entries \( w_{ij} \) and \( w_{ji} \) corresponding to their neighbors. Therefore, both the one norm and the infinity norm of \( W \) can be easily computed in finite time via a max-consensus protocol [23]. As a consequence, if needed, the network can agree in a distributed fashion upon a suitable value of the parameter \( α \).

On the other hand, the parameter \( α \) has a strong influence on the convergence rate of the algorithm. The lower this parameter, the faster the algorithm converges, as stated in Eq. (14). Since Assumption 2 only imposes an upper-bound on \( α \), if there are no additional constraints on this parameter, we can achieve arbitrarily fast convergence rate for any network. On the other hand, it should be noted that this value also affects the value of the centrality vector, approximating it to \( z \) as \( α \) tends to zero.

To conclude the section, note that, by using the algorithm with \( w_{ij} \) instead of \( w_{ji} \), we would obtain a measurement of how influenced a node is by the rest of the network. This could also pave the way for future research in better understanding how vulnerable an individual is to external points of view.

IV. DISTRIBUTED CONTROL OF NODE \( α \)-CENTRALITY

There are other applications involving networks, where rather than estimating the current centrality measure of each node, we are interested in controlling its value. In this section
we present a distributed algorithm that allows each agent in the network to perform minimum variations on the influence matrix to achieve this objective in finite time with only one round of communication.

In particular, let \( \rho^*_\alpha \) be the desired \( \alpha \)-centrality vector for a given graph \( G \) with initial influence matrix \( W \) and let \( \tilde{W} \in \mathcal{A}_G \) and \( W \in \mathcal{A}_G \) with \( \tilde{W} \geq W \geq W \). Similarly to the definition of \( w_{ij} \), we denote by \( x_{ij} \) the amount of variation applied on the influence of agent \( j \)'s information for agent \( i \), and \( X = [x_{ij}] \) the matrix containing all the changes in the original influence matrix. Our goal in this section is then to find the matrix \( X^* \in \mathcal{A}_G \) that solves

\[
\min_{X \in \mathcal{A}_G} \frac{1}{2} \|X\|_F^2 \\
\text{subject to} \quad W + X \leq \tilde{W} \\
\quad (I_n - \alpha(W + X)^T)\rho^*_\alpha = z.
\]

(20)

The above optimization problem represents that we want minimum effort variation on the matrix \( W \); in this view, we are interested in scenarios where the nodes aim at slightly modifying how their neighbors perceive their importance in the network while changing the \( \alpha \)-centrality to the desired value. The first two constraints are included to model the fact that every influence value cannot change more than an arbitrary amount, encoded by the matrices \( \tilde{W} \) and \( W \). The last constraint in Eq. (20) imposes that the new influence matrix, \( W + X \), has to yield the desired centrality value.

Notably, the idea of solving a distributed optimization problem with minimum effort has been applied also in [29], where an optimal weight balancing problem is solved by first providing a necessary and sufficient optimality condition and then by resorting to dynamical equations that asymptotically converges to an optimal solution. In this paper we also provide a necessary and sufficient optimality condition, but we develop a finite time algorithm to compute it, which relies on a single communication round, while all further computations are performed locally at each agent.

In the further developments of this section, we assume that the problem in Eq. (20) admits at least one feasible solution. To this end, it is convenient to provide a feasibility condition, as done in the next proposition.

**Proposition 1.** Suppose \( W \geq 0_{n \times n} \) and \( \rho^*_\alpha \geq z \). The problem in Eq. (20) admits a feasible solution if and only if it holds

\[
\begin{align*}
(I_n - \alpha(W + X)^T)\rho^*_\alpha &\leq z \\
(I_n - \alpha(W + X)^T)\rho^*_\alpha &\geq z
\end{align*}
\]

(21)

**Proof.** The equation

\[
(I_n - \alpha(W + X)^T)\rho^*_\alpha = z
\]

is equivalent to

\[
(W + X)^T\rho^*_\alpha = \frac{\rho^*_\alpha - z}{\alpha}.
\]

Since \( W \geq 0_{n \times n} \), we have that any admissible \( X \) satisfies \( W + X \geq 0_{n \times n} \), i.e., the entries of \( W + X \) are nonnegative. Therefore, since also the entries of \( \rho^*_\alpha \) are nonnegative, it holds

\[
W\rho^*_\alpha \leq (W + X)^T\rho^*_\alpha \leq W\rho^*_\alpha,
\]

and thus, the problem admits a feasible solution if and only if

\[
\begin{align*}
\left\{ \begin{array}{l}
\rho^*_\alpha - z \\
\rho^*_\alpha - z
\end{array} \right\} \leq W^T\rho^*_\alpha \\
\left\{ \begin{array}{l}
\rho^*_\alpha - z \\
\rho^*_\alpha - z
\end{array} \right\} \geq W^T\rho^*_\alpha
\end{align*}
\]

which is equivalent to Eq. (21). The proof is complete.

Let us now provide a necessary and sufficient optimality condition for Problem (20), under the assumptions given in Proposition 1. To this end, let us introduce some useful notation. Specifically, consider a vector \( \lambda = [\lambda_1, \ldots, \lambda_n] \) and let \( P(\lambda) \in \mathcal{A}_G \) be the \( n \times n \) matrix such that

\[
p_{ij}(\lambda) = \left\{ \begin{array}{ll}
sat(w_{ij} - \lambda_j \rho^*_\alpha, w_{ij}, \bar{w}_{ij}) & \text{if } j \in \tilde{N}_i \\
0 & \text{otherwise}
\end{array} \right.
\]

(22)

Moreover, let us define \( f : \mathbb{R}^n \to \mathbb{R}^n \) as

\[
f(\lambda) = P^T(\lambda)\rho^*_\alpha - \frac{\rho^*_\alpha - z}{\alpha}
\]

(23)

**Theorem 3.** Under the assumptions given in Proposition 1, the problem in Eq. (20) admits a global minimum \( X^* \in \mathcal{A}_G \) in the form

\[
X^* = P(\lambda^*) - W,
\]

(24)

where the vector \( \lambda^* \in \mathbb{R}^n \) satisfies

\[
f(\lambda^*) = 0_n.
\]

(25)

**Proof.** By construction, the objective function of Problem (20) is differentiable and convex, while all its constraints are linear; moreover, by assumption, Problem (20) is feasible. Therefore the KKT necessary and sufficient conditions in Theorem 1 apply.

Let us consider Lagrange multipliers \( \Phi \in \mathcal{A}_G, \Psi \in \mathcal{A}_G \) and \( \lambda \in \mathbb{R}^n \), that correspond to the first, second and third sets of constraints in Eq. (20), respectively. The Lagrange function \( \mathcal{L}(X, \Phi, \Psi, \lambda) \) associated to the Problem in Eq. (20) is

\[
\mathcal{L}(\cdot) = \frac{1}{2} \sum_{(i,j) \in E} x_{ij}^2 + \sum_{(i,j) \in E} \phi_{ij}(w_{ij} + x_{ij} - \bar{w}_{ij}) \\
+ \sum_{(i,j) \in E} \psi_{ij}(-w_{ij} - x_{ij} - \bar{w}_{ij}) \\
+ \sum_{i=1}^n \lambda_i \left( \sum_{j \in \tilde{N}_i} (w_{ji} + x_{ji}) \rho^*_\alpha - \frac{1}{\alpha} \rho^*_\alpha \alpha + \frac{1}{\alpha} z_i \right).
\]

By Theorem 1, \( X^* \) is the global minimum if and only if there are \( \Phi^* \in \mathcal{A}_G, \Psi^* \in \mathcal{A}_G \) and \( \lambda^* \in \mathbb{R}^n \) such that the following conditions hold true for all \( (i,j) \in E \)

\[
\frac{\partial \mathcal{L}(\cdot)}{\partial X_{ij}} \bigg|_{X=X^*, \Phi=\Phi^*, \Psi=\Psi^*, \lambda=\lambda^*} = 0,
\]

(26)

\[
w_{ij} + x_{ij} \geq \bar{w}_{ij},
\]

(27)

\[
w_{ij} + x_{ij} \leq \bar{w}_{ij},
\]

(28)
\begin{align}
(I_n - \alpha (W + X)^T) \rho^*_n &= z \\
\phi^*_{ij}(wij + x_{ij} - \overline{wij}) &= 0, \\
\psi^*_{ij}(wij + x_{ij} - \overline{wij}) &= 0, \\
\psi^*_{ij} &\geq 0, \\
\phi^*_{ij} &\geq 0.
\end{align}

In particular, the condition in Eq. \((26)\) corresponds to

\begin{equation}
x^*_{ij} = \psi^*_{ij} - \phi^*_{ij} - \lambda^*_j \rho^*_{ai},
\end{equation}

where we refer to the \(i\)-th component of the vector \(\rho_n\) as \(\rho^*_{ai}\). At this point, we plug Eq. \((34)\) into Eqs. \((32)\) and \((33)\), and we obtain, respectively,

\begin{align}
\phi^*_{ij}(wij + \psi^*_{ij} - \phi^*_{ij} - \lambda^*_j \rho^*_{ai} - \overline{wij}) &= 0 \\
\psi^*_{ij}(wij + \psi^*_{ij} - \phi^*_{ij} - \lambda^*_j \rho^*_{ai} - \overline{wij}) &= 0.
\end{align}

We observe that, when \(\phi^*_{ij} > 0\) it must hold \(wij + x_{ij} = \overline{wij}\) and thus \(wij + x_{ij} \neq wij\), which implies that \(\psi^*_{ij} = 0\); as a consequence, when \(\phi^*_{ij} > 0\), by Eq. \((35)\) we have that

\begin{equation}
\phi^*_{ij} = wij - \lambda^*_j \rho^*_{ai} - \overline{wij}.
\end{equation}

Since \(\phi^*_{ij} \geq 0\), we conclude that

\begin{equation}
\phi^*_{ij} = \max\{0, wij - \lambda^*_j \rho^*_{ai} - \overline{wij}\}.
\end{equation}

With a similar reasoning, we have that

\begin{equation}
\psi^*_{ij} = \max\{0, -wij + \lambda^*_j \rho^*_{ai} + \overline{wij}\}.
\end{equation}

Therefore, it holds

\begin{equation}
\psi^*_{ij} - \phi^*_{ij} = \begin{cases} 
-wij + \lambda^*_j \rho^*_{ai} + \overline{wij} & \text{if } wij - \lambda^*_j \rho^*_{ai} < \overline{wij} \\
-wij + \lambda^*_j \rho^*_{ai} + \overline{wij} & \text{if } wij - \lambda^*_j \rho^*_{ai} > \overline{wij} \\
0 & \text{otherwise}
\end{cases}
\end{equation}

The above expression is equivalent to

\begin{equation}
\psi^*_{ij} - \phi^*_{ij} = -wij + \lambda^*_j \rho^*_{ai} + \text{sat}(wij - \lambda^*_j \rho^*_{ai}, \overline{wij}, \overline{wij}) = -wij + \lambda^*_j \rho^*_{ai} + pij(\lambda^*).
\end{equation}

Plugging the above equation into Eq. \((34)\), we get

\begin{equation}
x^*_{ij} = p_{ij}(\lambda^*) - wij \Rightarrow X^* = P(\lambda^*) - W.
\end{equation}

Plugging Eq. \((37)\) in Eq. \((29)\), we conclude that

\begin{equation}
(I_n - \alpha P^T(\lambda^*)) \rho^*_n = z
\end{equation}

by which Eq. \((25)\) follows. This completes our proof. \(\square\)

\section{Finite-time Computation of the Optimal Solution}

In order to calculate the global minimum of Problem \((20)\), we seek \(\lambda^* \in \mathbb{R}^n\) that satisfies \(f(\lambda^*) = 0_n\). Specifically, for all \(i \in \{1, \ldots, n\}\), we have that

\begin{equation}
f_i(\lambda^*) = \sum_{j \in \mathcal{N}_i} \text{sat}\left(\frac{w_{ji} - \lambda^*_j \rho^*_{aj}}{\overline{wij}}, \overline{wij}, \overline{wij}\right) \rho^*_{ao} - \frac{1}{\alpha} (\rho^*_{ai} - z_i).
\end{equation}

From the above equation, it is evident that \(f_i(\lambda^*)\) is indeed a function of just the \(i\)-th component \(\lambda^*_i\), i.e., \(f_i(\lambda^*) = f_i(\lambda^*_i)\).

As a consequence, we have that if each agent knows the coefficients \(w_{ji}, \overline{wij}, \overline{w_{ji}}\), \(\rho^*_{ao}\), associated to its neighbors—such an information can be obtained via a single communication round—it is able to compute its own \(\lambda^*_i\) without further communication. In other words, the problem of finding the optimal \(\lambda^*\) can be decomposed into \(n\) independent tasks that can be solved locally (except for one preliminary communication round) at each agent.

We exploit this idea to develop an algorithm to let each agent \(i\) independently compute its associated \(\lambda^*_i\) and the terms \(p_{ij}(\cdot)\). This computation is done in finite time and with just one communication round, in which the agents share information with their neighbors in order to know the entries \(w_{ji}, \overline{wij}, \overline{w_{ji}}\).

Notice that, if needed, each agent \(i\) is able to obtain the terms \(p_{ij}(\cdot)\) from its neighbors via an additional communication round at the end of the procedure.

Within the proposed algorithm, we aim at finding \(\lambda^*_i\) by decomposing \(\widehat{\mathcal{N}}_i\) into three mutually exclusive and possibly empty sets

\begin{equation}
\widehat{\mathcal{N}}_i^+ = \{ j \in \widehat{\mathcal{N}}_i \big| w_{ji} - \lambda^*_i \rho^*_{aj} \geq \overline{wij}\},
\end{equation}

\begin{equation}
\widehat{\mathcal{N}}_i^- = \{ j \in \widehat{\mathcal{N}}_i \big| w_{ji} - \lambda^*_i \rho^*_{aj} \leq \overline{wij}\},
\end{equation}

\begin{equation}
\widehat{\mathcal{N}}_i^0 = \{ j \in \widehat{\mathcal{N}}_i \big| w_{ji} - \lambda^*_i \rho^*_{aj} \notin \overline{wij}\}.
\end{equation}

Notice that \(\widehat{\mathcal{N}}_i^+\) is the subset of \(\widehat{\mathcal{N}}_i\) for which the argument \(w_{ji} - \lambda^*_i \rho^*_{aj}\) of the saturation function that defines \(p_{ij}(\cdot)\) belongs to the interval \((\overline{wij}, \overline{w_{ji}})\), while \(\widehat{\mathcal{N}}_i^+, \widehat{\mathcal{N}}_i^-\) represent the subsets of \(\widehat{\mathcal{N}}_i\) for which the saturation saturates below and above, respectively.

Let us define the following Boolean variables for all \(j \in \widehat{\mathcal{N}}_i\)

\begin{equation}
x^+_j = \begin{cases} 
1 & \text{if } j \in \widehat{\mathcal{N}}_i^+, \\
0 & \text{otherwise},
\end{cases}
\end{equation}

\begin{equation}
x^-_j = \begin{cases} 
1 & \text{if } j \in \widehat{\mathcal{N}}_i^-, \\
0 & \text{otherwise},
\end{cases}
\end{equation}

\begin{equation}
x^0_j = \begin{cases} 
1 & \text{if } j \in \widehat{\mathcal{N}}_i^0, \\
0 & \text{otherwise}.
\end{cases}
\end{equation}

Considering these definitions, our goal is to compute the optimal \(\lambda^*_i\) and \(x^+_j, x^-_j, x^0_j \in \{0, 1\}\) for all \(j \in \widehat{\mathcal{N}}_i\) that satisfy the following constraints

\begin{equation}
\lambda^*_i \sum_{j \in \mathcal{N}_i} x^+_j (\rho^*_{ao}) = \sum_{j \in \mathcal{N}_i} \left( x^+_j w_{ji} + x^-_j \overline{wij} + x^0_j \overline{w_{ji}} \right) \rho^*_{ao} + \frac{z_i - \rho^*_{ai}}{\alpha},
\end{equation}

\begin{equation}
x^+_j + x^0_j + x^-_j = 1, \quad \forall j \in \mathcal{N}_i,
\end{equation}

\begin{equation}
w_{ji} - \lambda^*_i \rho^*_{aj} \in (\overline{wij}, \overline{w_{ji}}), \quad \forall j \in \widehat{\mathcal{N}}_i | x^+_j = 1,
\end{equation}

\begin{equation}w_{ji} - \lambda^*_i \rho^*_{aj} \leq \overline{wij}, \quad \forall j \in \widehat{\mathcal{N}}_i | x^-_j = 1,
\end{equation}

\begin{equation}w_{ji} - \lambda^*_i \rho^*_{aj} \geq \overline{wij}, \quad \forall j \in \widehat{\mathcal{N}}_i | x^0_j = 1.
\end{equation}

Let us first show that solving the above problem for all \(i \in \{1, \ldots, n\}\) yields a \(\lambda^* = [\lambda^*_1, \ldots, \lambda^*_n]^T\) that corresponds
to the global optimal solution to Problem (20).

**Proposition 2.** Let \( \lambda^*_i \in \mathbb{R} \) and \( x^*_j \in \{0,1\} \) for all \( j \in \bar{N}_i \) be a solution that satisfies the constraints in Eqs. (39a)–(39e). It holds \( f_i(\lambda^*_i) = 0 \), where \( f_i(\cdot) \) is defined as in Eq. (38).

**Proof.** By using the definition of \( f_i(\cdot) \) given in Eq. (38) and by explicitly considering the sets \( \bar{N}_i^{\leftrightarrow}, \bar{N}_i^{\uparrow}, \bar{N}_i^{\downarrow} \), we can express \( f_i(\lambda^*_i) \) as follows:

\[
f_i(\lambda^*_i) = \sum_{j \in \bar{N}_i^{\leftrightarrow}} w_{ji} \rho^*_{\alpha j} - \lambda^*_i \sum_{j \in \bar{N}_i^{\uparrow}} (\rho^*_{\alpha j})^2 + \sum_{j \in \bar{N}_i^{\downarrow}} w_{ji} \rho^*_{\alpha j} + \frac{1}{\alpha} \sum_{j \in \bar{N}_i^{\uparrow}} \rho^*_{\alpha j} + \frac{1}{\alpha} z_i = 0. \tag{40}\]

Since by hypothesis the variables \( x_j^+, x_j^-, x_j^\uparrow, x_j^\downarrow \) satisfy the constraints in Eqs. (39b)–(39e), they define consistent sets \( \bar{N}_i^{\leftrightarrow}, \bar{N}_i^{\uparrow}, \bar{N}_i^{\downarrow} \), hence, we have that Eq. (40) is equivalent to Eq. (38). The thesis follows.

The following corollaries are fundamental in order to calculate \( \lambda^*_i \) in finite time.

**Corollary 1.** The vector \( \lambda^* \) that is the stack of the solutions \( \lambda^*_i \) is a vector of Lagrange multipliers that satisfy the necessary and sufficient condition for Problem (20) given in Eq. (25) of Theorem 3.

**Proof.** The proof follows directly from Eq. (40).

**Corollary 2.** Let \( x_j^+, x_j^-, x_j^\uparrow, x_j^\downarrow \in \{0,1\} \) for all \( j \in \bar{N}_i \) be given variables that satisfy the constraint in Eq. (39b). If it holds \( \sum_{j \in \bar{N}_i} x_j^+ > 0 \), then the conditions in Eqs. (39a)–(39e) are satisfied if and only if, choosing

\[
\lambda^*_i = \frac{\alpha}{\sum_{j \in \bar{N}_i} x_j^+} \sum_{j \in \bar{N}_i} \left( x_j^+ w_{ji} + x_j^\uparrow w_{\alpha j} + x_j^\downarrow w_{\alpha j} \right) \rho^*_{\alpha j} + z_i - \rho^*_{\alpha i}, \tag{41}\]

we have that it holds:

\[
\begin{align*}
\lambda^*_i &\geq \max_{j \in \bar{N}_i} \frac{w_{ji} - w_{\alpha j}}{\rho^*_{\alpha j}}, \\
\lambda^*_i &\leq \min_{j \in \bar{N}_i} \frac{w_{\alpha j} - w_{ji}}{\rho^*_{\alpha j}}, \\
\lambda^*_i &\geq \min_{j \in \bar{N}_i} \frac{w_{\alpha j} - w_{ji}}{\rho^*_{\alpha j}}, \\
\lambda^*_i &\leq \max_{j \in \bar{N}_i} \frac{w_{\alpha j} - w_{ji}}{\rho^*_{\alpha j}}.
\end{align*} \tag{42}\]

Conversely, when \( \sum_{j \in \bar{N}_i} x_j^+ = 0 \), the conditions in Eqs. (39a)–(39e) are satisfied if and only if

\[
\min_{j \in \bar{N}_i} \frac{w_{ji} - w_{\alpha j}}{\rho^*_{\alpha j}} \leq \max_{j \in \bar{N}_i} \frac{w_{ji} - w_{\alpha j}}{\rho^*_{\alpha j}}, \tag{43}\]

and

\[
\sum_{j \in \bar{N}_i} \left( x_j^+ w_{ji} + x_j^\uparrow w_{\alpha j} \right) \rho^*_{\alpha j} + \frac{z_i - \rho^*_{\alpha i}}{\alpha} = 0. \tag{44}\]

In this case, the problem is solved by choosing any

\[
\lambda^*_i \in \left[ \min_{j \in \bar{N}_i} \frac{w_{ji} - w_{\alpha j}}{\rho^*_{\alpha j}}, \max_{j \in \bar{N}_i} \frac{w_{ji} - w_{\alpha j}}{\rho^*_{\alpha j}} \right]. \tag{45}\]

**Remark 2.** A consequence of Corollaries 1 and 2 is that each agent can compute \( \lambda^*_i \) in finite time. Indeed, it is sufficient to enumerate all ways to partition the elements in \( \bar{N}_i \) into three disjoint and possibly empty sets \( \bar{N}_i^{\leftrightarrow}, \bar{N}_i^{\uparrow}, \bar{N}_i^{\downarrow} \) that correspond to the given variables \( x_j^+, x_j^-, x_j^\uparrow, x_j^\downarrow \).

**Algorithm 2 Proposed Algorithm to compute \( \lambda^*_i \)**

```plaintext
procedure OPTIMAL_LAGRANGE_MULTIPLIER
for all \( \{x_j^+, x_j^-, x_j^\uparrow, x_j^\downarrow\} \) do
if \( \sum_{j \in \bar{N}_i} x_j^+ > 0 \) then
choose \( \lambda^*_i \) as in Eq. (41)
if \( \lambda^*_i \) satisfies Eq. (42) then
return \( \lambda^*_i \)
else if Eqs. (43) and (44) are satisfied then
choose any \( \lambda^*_i \) that satisfies Eq. (45)
end
end procedure
```

Algorithm 2 summarizes the procedure outlined in Remark 2 while the next proposition establishes its correctness and finite-time convergence.

**Proposition 3.** Algorithm 2 is correct, i.e., it finds a solution \( \lambda^*_i \) that satisfies \( f_i(\lambda^*_i) = 0 \).

**Proof.** By Proposition 2 it holds \( f_i(\lambda^*_i) = 0 \) if and only if Eqs. (39a)–(39e) are satisfied. At this point we observe that, within Algorithm 2 we enumerate all choices of variables \( x_j^+, x_j^-, x_j^\uparrow, x_j^\downarrow \in \{0,1\} \) that satisfy Eq. (39b), and for each of these choices we follow the approach in Corollary 2 to either compute \( \lambda^*_i \) or discard the choice if the conditions in Corollary 2 are not satisfied. In conclusion, Algorithm 2 correctly finds the optimal \( \lambda^*_i \).

Let us now inspect the computational complexity of Algorithm 2.

**Proposition 4.** Algorithm 2 has a computational complexity \( O(3|\bar{N}_i|) \).

**Proof.** In order to prove the statement, let us calculate the number of solutions that must be enumerated within Algo-
algorithm To this end, let us consider the Stirling numbers of the second kind \(^{[30]}\), which are defined as

\[
S(n, k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{n-j} \binom{n}{j} j^n
\]

and correspond to the number of unordered partitions of \(n\) elements into \(k\) nonempty sets (e.g., \(S(3, 2) = 3\), since \(\{x, y, z\}\) can be partitioned into sets \(\{\{x\}, \{y, z\}\}\), \(\{\{y\}, \{x, z\}\}\) or \(\{\{z\}, \{x, y\}\}\)). Within Algorithm \(^{[2]}\) we enumerate all possible ordered partitions. We point out that, by definition, the number of ways to partition \(n\) elements into \(k\) nonempty ordered partitions is given by the Stirling number multiplied by the number of possible permutations of the \(k\) sets, i.e.,

\[
\hat{S}(n, k) = k! S(n, k) = \sum_{j=0}^{k} (-1)^{n-j} \binom{n}{j} j^n.
\]

Notice that, from the definition of Stirling numbers, we have that \(S(n, 1) = 1\), \(S(n, 2) = 2^{n-1} - 1\) and \(S(n, 3) = \frac{1}{6}(3^n - 3 2^n + 3)\); hence, \(\hat{S}(n, 1) = 1\), \(\hat{S}(n, 2) = 2^n - 2\) and \(\hat{S}(n, 3) = 3^n - 3 2^n + 3\).

At this point we observe that, since \(G\) is undirected and connected by hypothesis \(^{[2]}\) it must hold \(|\hat{N}_i| \geq 2\). In particular, if \(|\hat{N}_i| = 2\) we have to inspect a number \(\nu_i\) of ordered partitions given by

\[
\nu_i = 3\hat{S}(2, 1) + 3\hat{S}(2, 2) = 3 + 3(2^2 - 2) = 9 = 3|\hat{N}_i|;
\]

in fact we can assign all the two element to each of the three sets, or we can chose any pair of sets (we have three pairs) and partition the two elements in such two sets.

Similarly, when \(|\hat{N}_i| > 2\) we also have the option to partition the elements into three nonempty sets, hence \(\nu_i\) becomes

\[
\nu_i = 3\hat{S}(|\hat{N}_i|, 1) + 3\hat{S}(|\hat{N}_i|, 2) + S(|\hat{N}_i|, 3)
= 3 + 3(2|\hat{N}_i| - 2) + 3|\hat{N}_i| - 3|\hat{N}_i| + 3
= 3|\hat{N}_i|.
\]

The proof is completed by noting that, for all choices of the variables \(\{x_j^+, x_j^-, x_j^+ | j \in \hat{N}_i\}\), computing \(\lambda_i^*\) and checking for satisfaction of the constraints has complexity \(O(1)\), as discussed in Remark \(^{[2]}\).

The following corollary is a consequence of the exhaustive exploration of all possible partitions, which are finite and specifically amount to \(3|\hat{N}_i|\) as per Proposition \(^{[2]}\):

**Corollary 3.** Algorithm \(^{[2]}\) computes a value of \(\lambda_i^*\) that satisfies \(f_i(\lambda_i^*) = 0\) in finite time.

Notice that the complexity of Algorithm \(^{[2]}\) is exponential in the number of neighbors of the node executing it. However, in typical applications, the neighborhood is likely to be well below the number of nodes. For instance, it can be easily shown that in random graphs (e.g., see \([31]\)) the average degree is \(np\), where \(n\) is the number of nodes and \(p\) the probability to create a link between two nodes; however, in several practical cases (e.g., while considering percolation phenomena \([32]\)), we are interested in sparse random networks where \(p \approx d/n\) for some fixed value \(d\); hence, the average degree is \(d\). Other relevant examples include lattices, where the degree is constant and typically small, and planar graphs, where the average degree is upper bounded by six (see, for instance, \([33]\)).

**V. APPLICATIONS**

In this section we present two application scenarios where our estimation and control algorithms can be useful tools for networks of agents. In particular, we first present a novel consensus algorithm that is able to reach a weighted consensus on some initial conditions accounting for the influence that each node has in the network. A peculiarity of our algorithm is that it does not require to know beforehand the actual network \(\alpha\)-centrality. Instead, the algorithm is able to correct the consensus value in parallel to the distributed estimation of the \(\alpha\)-centrality. Secondly, we exploit the centrality control algorithm to design the network weights that make every agent in the network equally important. This method can be used in cyber-security applications, implementing protection mechanisms where the importance of the most valuable nodes is reduced in order to discourage targeted attacks. Designing networks where there are no vulnerable agents against external influences.

**A. Influenced-based weighted consensus**

Let \(x_i(0)\) be the initial condition of agent \(i\) to be incorporated in the consensus iteration and \(x(0)\) the concatenation of the initial conditions of all the agents in vector form. Compared to the typical consensus problem of reaching the average of the initial conditions, our aim is to compute in a distributed fashion the following quantity,

\[
x^* = \frac{\rho^x x(0)}{\rho^x_0 1},
\]

which is a weighted average based on the global influence that each agent has in the network, according to the \(\alpha\)-centrality vector. In this way, more influential agents will have larger weight in the final consensus value than those with less influence power. While there exist algorithms that, knowing \(\rho_\alpha\), are able to compute Eq. (46) (e.g., see \([34]\)), our objective is to compute this value without prior knowledge of the centrality vector by the network.

In order to do this, let us start by defining the Perron matrix, \(Q = [q_{ij}]\), associated to the Laplacian matrix \(L\) of \(G\), i.e.,

\[
Q = I - \varepsilon L(G),
\]

with \(\varepsilon < 2/\lambda_L\), being \(\lambda_L\) the largest eigenvalue of \(L\), to guarantee that \(Q\) is a symmetric and doubly stochastic matrix, with largest eigenvalue equal to one and the second largest (in absolute value) denoted by \(\lambda_Q < 1\). For the sake of
completeness, recall that, under the above assumptions on $\varepsilon$, the classical linear iteration
\begin{equation}
    x(t+1) = Qx(t),
\end{equation}
asymptotically converges to the average of the initial conditions, $x(0)$.

The high level idea of our algorithm consists in applying an exogenous input, $\gamma_i$, to the classical consensus algorithm, such that the new final value corresponds to Eq. (46),
\begin{equation}
    x^* = \frac{1}{N} \sum_{i \in V} (x_i(0) + \gamma_i).
\end{equation}
Combining Eq. (46) and Eq. (49), the value of this input needs to be
\begin{equation}
    \gamma_i = \left(\frac{N \rho_i}{\sum_{j=1}^{N} \rho_j} - 1\right) x_i(0) = \left(\frac{\rho_i}{\bar{\rho}_i} - 1\right) x_i(0)
\end{equation}
where $\rho_i$ represents the $i$-th component of $\rho_{\alpha}$ and $\bar{\rho}_i$ represents the average of the influence weights $\rho_{\alpha}$, that is
\begin{equation}
    \bar{\rho}_i = \frac{1}{N} \sum_{j=1}^{N} \rho_{\alpha,j}.
\end{equation}
However, note that, as mentioned before, agents do not have the knowledge of $\rho_{\alpha}$ nor of its average, $\bar{\rho}_i$. Thus, our proposed algorithm consists in the following cascading update rules,
\begin{align}
    c_i(t+1) &= c_i(t) + \alpha \sum_{j \in \mathcal{N}_i} w_{ji} \Delta c_j(t), \tag{51a} \\
    \Delta c_i(t+1) &= c_i(t+1) - c_i(t), \tag{51b} \\
    \bar{c}_i(t+1) &= \sum_{j \in \mathcal{N}_i} q_{ij} \bar{c}_j(t) + \Delta c_i(t+1), \tag{51c} \\
    y_i(t+1) &= \left(\frac{c_i(t+1)}{\bar{c}_i(t+1)} - 1\right) x_i(0), \tag{51d} \\
    \Delta y_i(t+1) &= y_i(t+1) - y_i(t), \tag{51e} \\
    x_i(t+1) &= \sum_{j \in \mathcal{N}_i} q_{ij} x_j(t) + \Delta y_i(t+1), \tag{51f}
\end{align}
with initial conditions $c_i(0) = \Delta c_i(0) = \bar{c}_i(0) = z_i$, $y_i(0) = 0$ and $x_i(0)$ the initial consensus value of agent $i$ as defined at the beginning of the section. The same algorithm can be expressed in vectorial form by,
\begin{align}
    c(t+1) &= c(t) + \alpha W^T \Delta c(t), \tag{52a} \\
    \Delta c(t+1) &= c(t+1) - c(t), \tag{52b} \\
    \bar{c}(t+1) &= Q \bar{c}(t) + \Delta c(t+1), \tag{52c} \\
    y(t+1) &= \text{diag}\left(\frac{c_i(t+1)}{\bar{c}_i(t+1)} - 1\right) x(0), \tag{52d} \\
    \Delta y(t+1) &= y(t+1) - y(t), \tag{52e} \\
    x(t+1) &= Qx(t) + \Delta y(t+1). \tag{52f}
\end{align}
The intuition behind each rule is the following: Eqs. (52a) and (52b), equivalent to Eqs. (7)–(8), are used for the distributed computation of $\rho_{\alpha}$ and are included here for completeness of the rule. Eq. (52c) intends to compute the average value of $\rho_{\alpha}$, estimated in $c(t)$. The addition of $\Delta c(t+1)$ is necessary to account for the estimation error made in $c(t+1)$. The vector $y(t)$ aims at computing Eq. (52d). However, since the convergence to the correct value is asymptotic with $Q$, instead of applying this input at once, we apply it incrementally at each communication round in Eq. (52f), similarly to what was done in [36] to compute the average in an unbalanced digraph.

Before analyzing the convergence properties of the cascade system, we introduce the following Lemma, to handle the possible case of iterations where, $\bar{c}_i(t) = 0$.
\begin{lemma}
Suppose that all the entries of the vector $z$ are non-negative and at least one is strictly positive. Then there exists some $t^*$ such that for all $t > t^*$ all the components in $\bar{c}(t)$ are strictly positive.
\end{lemma}
\begin{proof}
First of all, note that $\Delta c_i(t)$ is not negative, which means that this term in Eq. (52c) is only additive. This means that if the claim is true without considering this term, then it will also hold including it. Thus, let us assume that $\Delta c_i(t) = 0$ for all $i$ and all $t$. This implies that Eq. (52c) becomes a classic averaging rule as in Eq. (48).
Denote
\begin{equation}
    \bar{z} = \frac{1}{N} \sum_{i} z_i.
\end{equation}
Since all the elements in $z$ are non-negative and at least one is positive we can assert that $\bar{z} > 0$. Now, we know that for all $\Delta c_i(t) = 0$, Eq. (52c) will converge to the average of the initial condition $c(0)$ which in this case is equal to $z$, so $\bar{c}(t)$ will converge asymptotically to $\bar{z} 1_n$. This implies that for any arbitrarily small $\varepsilon > 0$ we can find a $t^*$ such that for all $t > t^*$, for all $i$ it holds that $|\bar{c}_i(t) - \bar{z}| < \varepsilon$.

Consequently, there will be a time, $t^*$ such that for all $t > t^*$, all the components $\bar{c}(t)$ will be strictly positive, completing the proof.
\end{proof}

It should be noticed that Lemma 1 is necessary to provide an algorithmic implementation of the proposed protocol. As a matter of fact, by looking at the cascade of update rules given in Eqs. (51a)–(51f), it can be noticed that if $\bar{c}_i(t) = 0$, then Eq. (51d) is not defined. In order to overcome this issue, Eq. (51d) can be replaced as
\begin{equation}
    y_i(t) = \begin{cases} y_i(t-1) & \text{if } \bar{c}_i(t) = 0 \\
    \left(\frac{c_i(t+1)}{\bar{c}_i(t+1)} - 1\right) x_i(0) & \text{otherwise} \end{cases}
\end{equation}
As it will be shown later in Theorem 4 this change does not affect the overall convergence of the algorithm. Intuitively, this can be explained by the fact our goal is to apply the total input by means of a sequence of increments, where at each iteration we compensate for the error in the estimation of the centrality. Therefore, by adding and subtracting the same quantity we do not modify the total input while avoiding the division by zero in Eq. (52c). For the sake of clarity, the equivalent vectorial version of Eq. (52c) based on Eq. (54) is here omitted.

Let us now review an auxiliary result used to prove the convergence of our algorithm.
Lemma 2 (Lemma 3.2 in [17]). Let $0 \leq \lambda < 1$ and $\{\beta(t)\}$ a bounded sequence such that $\lim_{t \to \infty} \beta(t) = 0$. Then
\[ \lim_{t \to \infty} \sum_{j=0}^{t} \lambda^{t-j} \beta(j) = 0. \]

The following Theorem demonstrates convergence to the desired result.

**Theorem 4.** Assume the conditions in Theorem 2 hold; then, the dynamical system in Eq. (52) converges to
\[
\begin{align*}
\lim_{t \to \infty} c(t) &= \rho, \\
\lim_{t \to \infty} \Delta c(t) &= 0, \\
\lim_{t \to \infty} e(t) &= \rho, \\
\lim_{t \to \infty} y(t) &= \text{diag}(\gamma) 1, \\
\lim_{t \to \infty} \Delta y(t) &= 0, \\
\lim_{t \to \infty} x(t) &= x^* 1.
\end{align*}
\]

**Proof.** The limit in Eq. (55a) was already demonstrated in Theorem 2 and the limit in Eq. (55b) comes naturally from it. Let us define now the average of the centrality estimation increments,
\[
\Delta \bar{c}(t) = \frac{1}{N} \sum_{i \in V} \Delta c_i(t).
\]

The average of the centrality vector can be expressed as an infinite sum, and, using Eq. (55a), it holds
\[
\begin{align*}
\sum_{t=0}^{\infty} \Delta \bar{c}(t) &= \sum_{t=0}^{\infty} \frac{1}{N} \sum_{j=1}^{N} \Delta c_j(t) = \frac{1}{N} \sum_{j=1}^{N} \sum_{t=0}^{\infty} \Delta c_j(t) \\
&= \frac{1}{N} \sum_{j=1}^{N} \left( \lim_{t \to \infty} c_j(t) - \bar{c}_j(0) \right) \\
&= \frac{1}{N} \sum_{j=1}^{N} \rho \alpha_j = \rho, \\
\end{align*}
\]
where we used the fact $\bar{c}_j(0) = 0$ for all $j$. In addition, the value of $c(t)$ can be put as a sum by
\[
\bar{c}(t) = \sum_{j=0}^{t} Q^{t-j} \Delta c(j).
\]
Let us define now the difference, $e_c(t) = \| \bar{c}(t) - \rho \alpha 1_n \|$, and compute its limit when the time goes to infinity,
\[
\begin{align*}
\lim_{t \to \infty} e_c(t) &= \lim_{t \to \infty} \| \bar{c}(t) - \rho \alpha 1_n \| \\
&= \lim_{t \to \infty} \left\| \sum_{j=0}^{t} (Q^{t-j} \Delta c(j) - \Delta \bar{c}(j) 1_n) \right\| \\
&\leq \lim_{t \to \infty} \sum_{j=0}^{t} \| Q^{t-j} \Delta c(j) - \Delta \bar{c}(j) 1_n \|,
\end{align*}
\]
where the second line comes from replacing Eq. (57) and Eq. (56) and the third one is by direct application of norm inequalities. Before proceeding, we recall that, for all $t \geq 0$, it holds
\[
Q^t = \frac{1}{N} 1_n 1_n^T + \sum_{i=2}^{N} \lambda_{Q,i} v_{Q,i} v_{Q,i}^T
\]
where $\lambda_{Q,i}$ is the eigenvalue of $Q$ with $i$-th largest magnitude and $v_{Q,i}$ is the associated eigenvector. Since $Q$ is symmetric, we also know that $v_{Q,i}^T 1_n = 0$, $i \geq 2$. Thus, following the development of Eq. (58),
\[
\lim_{t \to \infty} e_c(t) \leq \lim_{t \to \infty} \sum_{j=0}^{t} \| Q^{t-j} (\Delta c(j) - \Delta \bar{c}(j) 1_n) \| \\
&\leq \lim_{t \to \infty} \sum_{j=0}^{t} \left\| \sum_{i=2}^{N} \lambda_{Q,i} v_{Q,i} v_{Q,i}^T \Delta c(j) \\
&+ \frac{1}{N} 1_n 1_n^T \Delta \bar{c}(j) 1_n \right\| \\
&\leq \lim_{t \to \infty} \sum_{j=0}^{t} \gamma Q \sum_{i=2}^{N} \lambda_{Q,i} \| \Delta c(j) \|,
\]
with $\gamma Q = \max_{i \neq 1} \| v_{Q,i} v_{Q,i}^T \|$ a constant. Finally, using Theorem 2 we know that $\| \Delta c(j) \|$ is bounded and converges to zero as $j$ goes to infinity. Additionally, we know that $0 \leq \lambda_{Q,2} < 1$. Thus, using Lemma 2 we can assert that $e_c$ converges to zero, showing that Eq. (55c) is true.

Once we have established convergence of $c(t)$, combining this limit with Theorem 2 together with Eq. (50) and Lemma 1, the limit presented in Eq. (55d) follows up straightforwardly and, consequently, so does the one in Eq. (55e).

In order to prove Eq. (55f), let us notice that by combining Eq. (49) together with Eq. (50) and recalling that $y_i(0) = 0$, $i \in V$, we obtain
\[
x^* = \frac{1}{N} \sum_{i \in V} \left( x_i(0) + \lim_{t \to \infty} y_i(t) \right) \\
= \frac{1}{N} \sum_{i \in V} \left( x_i(0) + \sum_{t=0}^{\infty} (y_i(t) - y_i(t-1)) \right) \\
= \frac{1}{N} \sum_{i \in V} x_i(0) + \frac{1}{N} \sum_{i \in V} \sum_{t=1}^{\infty} (y_i(t) - y_i(t-1)),
\]

where, similarly to the case in Eq. (55f), the following definition has been used
\[
\Delta \bar{y}(t) = \frac{1}{N} \sum_{i \in V} \Delta y_i(t).
\]

At this point, we observe that $x(t)$ can be written as
\[
x(t) = Q^t x(0) + \sum_{j=1}^{t} Q^{t-j} \Delta y(j);
\]
hence, it follows that
\[
\lim_{t \to \infty} x(t) = \lim_{t \to \infty} \left( Q^T x(0) + \sum_{j=1}^{t} Q^{t-j} \Delta y(j) \right) \\
= \lim_{t \to \infty} Q^T x(0) + \lim_{t \to \infty} \sum_{j=1}^{t} Q^{t-j} \Delta y(j) \\
= \frac{1}{N} \sum_{n=1}^{N} \lambda_n \Delta y(n) \\
+ \lim_{t \to \infty} \sum_{j=1}^{N} \Delta y(j) = x^* \mathbf{1}_n,
\]

where it should be noticed that according to Lemma 3 the last term vanishes to zero at \( t \) approaches infinity. Therefore, Eq. (64) becomes
\[
\lim_{t \to \infty} x(t) = \frac{1}{N} \sum_{n=1}^{N} \lambda_n \Delta y(n) \\
+ \sum_{j=1}^{\infty} \Delta y(j) \mathbf{1}_n = x^* \mathbf{1}_n,
\]

thus concluding the proof.

To conclude the section, we briefly analyze some of the properties and requirements of the proposed algorithm.

**Remark 3 (Communication demands).** Our algorithm requires the exchange of three values per agent and communication round. On one hand, each agent sends each time the value of \( \Delta c_i(t) \). This value is the same one used in the previous section to estimate \( \rho_\alpha \). On the other hand, there are two sums in Eq. (51) over the set of neighbors of each agent, one that requires \( \hat{c}_i(t) \) to compute the average of \( \rho_\alpha \) and the other one that requires \( x_j(t) \) to compute the weighted consensus.

**B. Attack protection mechanisms**

As noted in early studies in complex network theory (e.g., [38], [39]) and confirmed in more recent works [40]–[44], attacks dealt to the nodes of a network (e.g., disrupting the target nodes along with their incident edges) may have severe effects in terms of disruption of the residual network connectivity, especially when the attacker selects the target nodes based on their importance, according to topological features (e.g., degree, centrality, etc.). In order to implement effective protection strategies, minimizing the risk that the most valuable nodes become the attacker’s target, a typical approach is to prioritize the protection of the most important nodes, with the aim to make all nodes equally valuable for the attacker (see, among other works, the approach in [45]), where each node is associated an attack cost and the protection strategies aims at raising the attack cost of the most valuable nodes). However, to the best of our knowledge, most of such protection strategies are implemented in a centralized way, while it would be beneficial to let the nodes autonomously modify their importance in the network.

Algorithm 2 represents a valuable tool to achieve the above objective with respect to \( \alpha \)-centrality. In this view, in order to be protected against an attacker, the nodes aim at hiding their true \( \alpha \)-centrality value, obtaining the same \( \alpha \)-centrality for all nodes. Within this scenario, the agents aim at finding the minimum effort matrix \( X \in \mathbb{R}^G \) (i.e., the one that requires the minimum weight variation, in a least squares sense) that satisfies the problem in Eq. (20) with respect to a target \( \alpha \)-centrality that is proportional to 1. Notice that Algorithm 2 requires just one round of communication at the beginning; moreover, each node can compute off-line the values \( X_{ij} \) that are required to transform their \( \alpha \)-centrality into equal values, and such values can be used in case of emergency at run-time during the network operation in order to modify the weights and reduce their importance in the network, with the aim to render all agents equally appealing for an attacker, thus making the attacker less interested in performing targeted attacks.

**VI. SIMULATIONS**

**A. Centrality estimation**

In the first simulation we are going to show an example of the distributed estimation of the \( \alpha \)-centrality of a network and its application to weighted consensus with 15 nodes and topology shown in Fig. 1. We have numbered and assigned colors to each node to better highlight the centrality properties in the simulations.

![Network topology](image.png)

Firstly, we consider the distributed computation of \( \rho_\alpha \) associated to the adjacency matrix, i.e., \( W = A \), with a uniform initial importance vector, \( z = 1 \). This way, there is a direct relationship between the centrality value and the connectivity of each node, resulting in node 5, in green, having the highest centrality value and nodes 1, 10 and 12 having the lowest values.

The evolution of \( c(t) \) is depicted in Fig. 2(a). The parameter \( \alpha \) has been chosen in the simulation equal to 0.8/\( \| W \| \), so that the error is reduced by a factor of 0.8 at each communication round. Considering that in this particular case \( \| z \|/(1-\kappa) \approx 19 \), our analytic bound states that the algorithm should reach an accuracy below 0.1 in approximately 24
communication rounds which is consistent with the plot. The difference between the actual estimation error, $e_p(t)$ and the theoretical bound in Eq. (11) is shown in Fig. 2(b), where we can see that this difference is not only positive for all $t$, but also close to zero.

Secondly, we combine the centrality estimation method together with the influence-based consensus proposed in Section V. To highlight the practical implications of having some $z_i = 0$ we consider now an important vector such that $z_5 = 0$ and $z_i = 1$ for every other node $i \neq 5$. In Figure 3 we show the evolution of the four variables analyzed in Theorem 4 that do not represent increments, i.e., $c(t), c(t), y(t)$ and $x(t)$.

The top left plot in Figure 3 shows the new estimation of the centrality vector. The difference in the initial importance vector leads to a different final centralities. Setting $z_5 = 0$ we observe a decrease in the final centrality value of node 5, from slightly less than 8 in Fig 2(a) to slightly less than 6 in the new simulation. In the top right plot in Figure 3 we can observe how all the nodes in the network reach asymptotically the average of $\rho_0$, shown as grey dashed line. The convergence speed is slower than for the computation of $c(t)$ due to the slower convergence rate of the powers of $Q$.

The bottom left plot in Figure 3 shows the convergence of $y(t)$ to the desired exogenous input in Eq. (50). Note how this input is positive for the most influential nodes, like node 5 (green) in Fig. 1(a), whereas is negative for the least influential nodes, like node 12 (purple). This is consistent with the idea of giving more weight to the values of the most influential nodes, which in our setup is transformed into increasing their initial condition for the consensus algorithm.

Finally, the bottom right plot in Figure 3 shows the consensus evolution of the initial conditions, assigned randomly. For the sake of visualization, we have also included in the plot the value of the average (black dotted line), to better visualize that our algorithm does not converge to this value but to the weighted average (red dashed line) in Eq. (46).

**B. Centrality control**

In this subsection we provide an example of application of the proposed centrality control scheme. Specifically, we consider the network reported in Figure 4a. We consider a weighted average (red dashed line) in Eq. (46).

In particular, we show in red dotted lines the links where $p_{ij}(\cdot) = w$. It can be noted that one saturation to the upper bound and two saturations to the lower bound occur. Notice that Algorithm 2 is indeed able to modify the $\alpha$-centrality, since we obtain a numerical value of $\lambda^*$ such that $\|f(\lambda^*)\| = 7.94 \times 10^{-15}$, which, by Theorem 3 implies that the solution $X^*$ found essentially the global optimal one. As for the effort required for the variation of weights, we obtain $\|X^*\|_2^2/2 = 14.4224$. Notice further that the maximum number of evaluations done within Algorithm 2 are 7 while, according to Proposition 3 the theoretical maximum is $3^{N_5} = 81$. In Figure 5 we provide numerical evidence of the effectiveness of Algorithm 2. Specifically, we consider the above example and we show the results of a Monte Carlo simulation campaign where we generate $10^7$ random candidate solutions, each consisting of a matrix having entries that are sampled uniformly at random from the interval $[w_{i,j}, w_{i,j}]$, i.e., the candidate solutions satisfy the inequality constraints in Problem (20). In the figure we show the Monte Carlo candidate solutions via red crosses and, for each candidate solution, we report the value of the objective function and the sum of the absolute values of the violations of the constraints. Moreover, we show via a blue star the solution found by our algorithm.

We point out that, in our experiment, none of the Monte Carlo candidate solutions is admissible to Problem (20); this is due to the presence of an equality constraint, which has the effect to greatly reduces the set of admissible solutions. In spite of the difficulty of finding a feasible solution at random, the proposed algorithm succeeds in finding a feasible solution which satisfies the conditions in Theorem 4 and is thus the global optimal solution.

Let us now suppose that, for security reasons, the network in Figure 4b needs to modify the $\alpha$-centrality values in order to reduce the importance of the most valuable nodes, by making all nodes have equal $\alpha$-centrality, and specifically $\alpha$-centrality equal to 1. To this end, we consider an initial value $z$ that is proportional to the $\alpha$-centrality given in Eq. (66) and, specifically,

$$z = \frac{\rho^*_i}{1^T \rho^*}.$$  

It can be noted that the problem admits a feasible solution as per Proposition 1. In Figure 6 we report the resulting $\lambda^*_t$ and $p_{ij}(\cdot)$. Also in this case, the procedure is successful, being $\|f(\lambda^*)\| = 2.17 \times 10^{-15}$; moreover we obtain $\|X^*\|_2^2/2 = 4.6742$, i.e., we are able to make all nodes equal with a remarkably small variation of the weights.

**VII. CONCLUSIONS**

In this work, the problems of distributed node centrality identification and control have been addressed. We have developed a protocol for the distributed computation of $\alpha$-centrality, which is particularly suitable for networks with asymmetric interactions. We have also discussed a local solution for the computation of minimum variation of weights such that the network yields a desired centrality value. In addition, motivated by studies on social networks, we have proposed a novel consensus-based algorithm which runs in parallel to the $\alpha$-centrality estimation and achieves a weighted consensus,
where the weights are given precisely by the values of the \( \alpha \)-centrality. The control algorithm has also been applied to the problem of minimizing agents’ vulnerability to external influences.

REFERENCES

[1] L. C. Freeman, “Centrality in social networks conceptual clarification,” Social Networks, vol. 1, no. 3, pp. 215 – 239, 1978.
[2] S. P. Borgatti, K. M. Carley, and D. Krackhardt, “On the robustness of centrality measures under conditions of imperfect data,” Social Networks, vol. 28, no. 2, pp. 124 – 136, 2006.
[3] T. Opsahl, F. Agneessens, and J. Skvoretz, “Node centrality in weighted networks: Generalizing degree and shortest paths,” Social Networks, vol. 32, no. 3, pp. 245 – 251, 2010.
[4] M. Newman, Networks: An Introduction. New York, NY, USA: Oxford University Press, Inc., 2010.
[5] S. Wasserman and K. Faust, Social network analysis: Methods and applications. Cambridge university press, 1994, vol. 8.
[6] G. Oliva, R. Setola, and C. N. Hadjicostis, “Distributed finite-time calculation of node eccentricities, graph radius and graph diameter,” Systems & Control Letters, vol. 92, pp. 20–27, 2016.
[7] A. Bavelas, “Communication Patterns in Task-Oriented Groups,” The Journal of the Acoustical Society of America, vol. 22, no. 6, pp. 725–730, Nov. 1950.
[8] L. C. Freeman, “A Set of Measures of Centrality Based on Betweenness,” Sociometry, vol. 40, no. 1, pp. 35–41, Mar. 1977.
[9] P. Bonacich, “Factoring and weighting approaches to status scores and clique identification,” The Journal of Mathematical Sociology, vol. 2, no. 1, pp. 113–120, 1972.
[10] P. Bonacich and P. Lloyd, “Eigenvector-like measures of centrality for asymmetric relations,” Social Networks, vol. 23, no. 3, pp. 191 – 201, 2001.
[11] K. Anna Lehmann and M. Kaufmann, “Decentralized algorithms for evaluating centrality in complex networks,” Wilhelm-Schickard-Institut, Tech. Rep. WSI-2003-10, October 2003.
[12] K. Wehmuth and A. Ziviani, “Distributed assessment of the closeness centrality ranking in complex networks,” in Fourth Annual Workshop on Simplifying Complex Networks for Practitioners, 2012, pp. 43–48.
[13] W. Wang and C. Y. Tang, “Distributed computation of node and edge betweenness on tree graphs,” in 52nd IEEE Conf. on Decision and Control, Dec 2013, pp. 43–48.
[14] ——, “Distributed computation of classic and exponential closeness on tree graphs,” in 2014 American Control Conf., June 2014, pp. 2090–2095.
[15] ——, “Distributed estimation of closeness centrality,” in 54th IEEE Conference on Decision and Control, Dec 2015, pp. 4860–4865.
[16] L. Maccari, L. Ghiro, A. Guerrieri, A. Montresor, and R. L. Cigno, “On the distributed computation of load centrality and its application to dv routing,” in IEEE International Conference on Computer Communications, 2018.
[17] W. S. Rossi and P. Frasca, “On the convergence of message passing computation of harmonic influence in social networks,” IEEE Transactions on Network Science and Engineering, pp. 1–1, 2018, to appear.
[18] ——, “An index for the local influence in social networks,” in European Control Conference, June 2016, pp. 525–530.
[19] H. Ishii and R. Tempo, “The pagerank problem, multiagent consensus, and web aggregation: A systems and control viewpoint,” IEEE Control Systems Magazine, vol. 34, no. 3, pp. 34–53, June 2014.
[20] T. Charalambous, C. N. Hadjicostis, M. G. Rabbat, and M. Johansson, “Totally asynchronous distributed estimation of eigenvector centrality in digraphs with application to the pagerank problem,” in 55th IEEE Conf. on Decision and Control, Dec 2016, pp. 25–30.
[21] K. You, R. Tempo, and L. Qiu, “Distributed algorithms for computation of centrality measures in complex networks,” IEEE Trans. on Automatic Control, vol. 62, no. 5, pp. 2080–2094, May 2017.
[22] A. Suzuki and H. Ishii, “Distributed randomized algorithms for pagerank based on a novel interpretation,” in American Control Conference, June 2018, pp. 472–477.
[23] M. S. Granovetter, “The strength of weak ties,” American Journal of Sociology, vol. 78, no. 6, pp. 1360–1380, 1973.
[24] D. Lusseau and M. E. J. Newman, “Identifying the role that animals play in their social networks,” Proc. of the Royal Society of London B: Biological Sciences, vol. 271, no. Suppl 6, pp. S477–S481, 2004.
[25] L. Katz, “A new status index derived from sociometric analysis,” Psychometrika, pp. 39–43, 1953.
[26] N. A. Lynch, Distributed algorithms. Elsevier, 1996.
[27] G. Oliva, A. I. Rikos, C. N. Hadjicostis, and A. Gasparri, “Distributed flow network balancing with minimal effort,” IEEE Transactions on Automatic Control, 2019 (Early Access).
Figure 3. Influence-based consensus with influence matrix equal to the adjacency matrix and $z_5 = 0$. Panels (a)–(d) show the evolution of $c(t)$, $\bar{c}(t)$, $y(t)$, and $x(t)$, respectively.

[30] M. Abramowitz, “Stirling numbers of the second kind,” Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, pp. 824–825, 1972.
[31] A. Blum, J. Hopcroft, and R. Kannan, “Foundations of data science,” Vorabversion eines Lehrbuchs, 2016.
[32] S. Janson et al., “On percolation in random graphs with given vertex degrees,” Electronic Journal of Probability, vol. 14, pp. 86–118, 2009.
[33] T. Nishizeki and N. Chiba, Planar graphs: Theory and algorithms. Elsevier, 1988, vol. 32.
[34] S. Sundaram and C. N. Hadjicostis, “Distributed function calculation and consensus using linear iterative strategies,” IEEE Journal on Selected Areas in Communications, vol. 26, no. 4, pp. 650–660, 2008.
[35] F. Bullo, J. Cortés, and S. Martínez, Distributed Control of Robotic Networks, ser. Applied Mathematics Series. Princeton University Press, 2009, electronically available at http://coordinationbook.info.
[36] A. Priolo, A. Gasparri, E. Montijano, and C. Sagues, “A distributed algorithm for average consensus on strongly connected weighted digraphs,” Automatica, vol. 50, no. 3, pp. 946–951, 2014.
[37] E. Montijano, J. I. Montijano, C. Sagues, and S. Martínez, “Robust discrete time dynamic average consensus,” Automatica, vol. 50, no. 12, pp. 3131–3138, 2014.
[38] R. Albert, H. Jeong, and A.-L. Barabási, “Error and attack tolerance of complex networks,” nature, vol. 406, no. 6794, pp. 378–382, 2000.
[39] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, “Attack vulnerability of complex networks,” Physical Review E, vol. 65, no. 5, p. 056109, 2002.
Figure 4. Example with $n = 6$ nodes. Panel (a) reports the initial coefficients $p_{ij}$ for each edge. Panel (b) shows the resulting Lagrange multiplier $\lambda^*_i$ for each node and the value of $p_{ij}(\cdot)$ for each edge, when the desired $\alpha$-centrality is $\rho_\alpha = [8, 7, 6, 5, 4, 3]^T$. We show in red dotted lines the links where $p_{ij}(\cdot) = \pi$ and in blue dashed lines the links for which $p_{ij}(\cdot) = \infty$.

Figure 5. Montecarlo simulation to show that the algorithm finds the global optimal solution.

Figure 6. Example where the network in Figure 4 modifies the importance of the nodes by making their $\alpha$-centralities all equal to one. Specifically, the figure reports the resulting values $\lambda^*_i$ and $p_{ij}(\cdot)$. We show in red dotted lines the links where $p_{ij}(\cdot) = \pi$.

[40] J. Wu, H.-Z. Deng, Y.-I. Tan, and D.-Z. Zhu, “Vulnerability of complex networks under intentional attack with incomplete information,” Journal of Physics A: Mathematical and Theoretical, vol. 40, no. 11, p. 2665, 2007.

[41] X. Huang, J. Gao, S. V. Buldyrev, S. Havlin, and H. E. Stanley, “Robustness of interdependent networks under targeted attack,” Physical Review E, vol. 83, no. 6, p. 065101, 2011.

[42] S. Shao, X. Huang, H. E. Stanley, and S. Havlin, “Percolation of localized attack on complex networks,” New Journal of Physics, vol. 17, no. 2, p. 023049, 2015.

[44] Y. Berezin, A. Bashan, M. M. Danziger, D. Li, and S. Havlin, “Localized attacks on spatially embedded networks with dependencies,” Scientific Reports, vol. 5, 2015.

[45] L. Faramondi, G. Oliva, S. Panzieri, F. Pascucci, M. Schlueter, M. Munte tomoh, and R. Setola, “Network structural vulnerability: A multiobjective attacker perspective,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, no. 99, pp. 1–14, 2018.