Distributed Network Size Estimation and Average Degree Estimation and Control in Networks Isomorphic to Directed Graphs

Iman Shames, Themistoklis Charalambous, Christoforos N. Hadjicostis and Mikael Johansson

Abstract— **Many properties of interest in graph structures are based on the nodes' average degree (i.e., the average number of edges incident to/from each node). In this work, we present asynchronous distributed algorithms, based on ratio consensus, that can be used to accurately estimate the number of nodes in a multi-component system whose communication topology is described by a directed graph. In addition, we describe an asynchronous distributed algorithm that allows each node to introduce or terminate links in order to reach a target average degree in the network. Such an approach can be useful in many realistic scenarios; for example, for the introduction and removal of renewable energy resources in a power network, while maintaining an average degree that fulfils some structural and dynamical properties and/or optimises some performance indicators of the network. The effectiveness of the proposed algorithms is demonstrated via illustrative examples.**

I. Introduction

Driven by the growing ubiquity of mobile devices with a plethora of communication capabilities, distributed systems and novel applications are now within reach. As a result, various strands of research in distributed systems have attracted a lot of attention. Particularly, carrying out distributed computations has been at the forefront of research (see, for example, [1]–[4] and references there-in). Such systems are often modeled by graphs, where each component is represented by a graph vertex, and the sensing and communication links between each pair of components are represented by edges. A particular line of research that has attracted attention concerns problems that aim to achieve an objective by modifying the properties of these graphs; examples include, methodologies to add links in order to optimize the synchronization properties of the network, or to remove links to decrease the rate of propagation of a pathogen in the network, and others. To achieve many of these objectives global information about the properties of the network and its underlying graph is required (e.g., global information about the Laplacian of the graph [5]). In this paper, we focus initially on obtaining such global information relying purely on local measurements. Particularly, we are interested in accurately estimating the number of the nodes, and subsequently the average degree of the graph.

The problem of estimating the number of nodes in a graph has been already considered under the umbrella term of *network size estimation* (see, for example, [6]–[8] and references there-in). However, nearly all the available methods in the literature rely on statistical methods and require nodes to exchange excessive information. For example, one strand of research has concentrated on *random walk strategies* [9], [10] where statistical properties of the graph are used to infer the network size. Another strand of research makes use of *capture-recapture strategies* [11], [12], which are strongly connected to the estimation of total populations using sampling of finite populations. In this work, the exchange of information is limited to vectors of length equal to the valency of each node. The proposed algorithm achieves estimates that asymptotically converge to the correct value for the number of nodes under synchronous or asynchronous communication.

In addition, we are able to estimate a global property of a network that plays an important role in determining the performance of the network in different situations, namely the so-called *average node degree* of the graph, which is the ratio of the number edges over the number of nodes in the network (note that the average number of edges incident into each node is the same as the average number of edges out of each node, and the average node degree is twice this number). The average node degree can be found easily using the algorithm for network size estimation but with different initial conditions. It is known that the average node degree plays a universal role in cooperation in different networks [13]. For example, it can be shown that graphs with certain node degrees are more robust to infection propagation [14]. Moreover, the role of the average degree of the graph of a population on how the distribution of antidotes affects the behaviour of the population has been studied [15]. Additionally, some graphs with average node degree smaller than a threshold are known to be immune to detrimental effects (for example, in the so-called *firefighter problem* [16], [17]). Sensor networks literature has also studied the relationship between the average degree and the k-connectivity of the graph [18]. For more information on how average degree affects different properties of a network the reader may refer to [19] and references there-in.

Finally, we address the problem of controlling the average node degree in a network. The problem of estimation of the average node degree, centralized and otherwise, has been studied in several contexts [20], [21]. However, to the best

Iman Shames was formerly with the Automatic Control Lab, Electrical Engineering Department and ACCESS Linnaeus Center, Royal Institute of Technology (KTH), Stockholm, Sweden. He is currently a Research Fellow at the Department of Electrical and Electronic Engineering, University of Melbourne. E-mail: iman.shames@unimelb.edu.au.

Themistoklis Charalambous and Mikael Johansson are with the Automatic Control Lab, Electrical Engineering Department and ACCESS Linnaeus Center, Royal Institute of Technology (KTH), Stockholm, Sweden. E-mail: {themisc, mikaelj}@kth.se.

Christoforos N. Hadjicostis is with the Department of Electrical and Computer Engineering at the University of Cyprus, Nicosia, Cyprus. Corresponding author's address: 75 Kallipoleos Avenue, P.O.Box 20537, 1678 Nicosia, Cyprus. E-mail: chadjic@ucy.ac.cy.

of our knowledge the problem of controlling the average node degree of a network in a distributed manner has not been considered.

The outline of this paper is as follows. In Section II the notation used throughout the paper is provided. In Section III the model is presented along with background on graph properties and theoretical preliminaries that are necessary for our subsequent development. In Section IV we describe our proposed algorithm for distributed estimation of the number of nodes in a directed graph. We consider both synchronous and asynchronous versions of the algorithm, as well as the estimation of the number of nodes in a time-varying network with a changing number of nodes. Then, in Section V, a distributed algorithm is suggested which allows each node to control its degree. The validity of our theoretical results is demonstrated by examples in Section VI. Finally, Section VII summarizes the results of the paper.

II. Notation

The sets of real, integer and natural numbers are denoted by $\mathbb{R}$, $\mathbb{Z}$ and $\mathbb{N}$, respectively; their positive orthant is denoted by the subscript $+$ (e.g. $\mathbb{R}_+$). Vectors are denoted by small letters whereas matrices are denoted by capital letters. A^T denotes the transpose of matrix A. By $\mathbb{1}$ we denote the all-ones vector and by I we denote the identity matrix (of appropriate dimensions). A matrix whose elements are nonnegative, called nonnegative matrix, is denoted by $A \geq 0$ and a matrix whose elements are positive, called positive matrix, is denoted by $A > 0$.

Let the exchange of information between nodes be modeled by a weighted directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, P)$ of order n ($n \geq 2$), where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} - \{(j, j) \mid j \in \mathcal{V}\}$ is the set of edges, and $P = [p_{ji}] \in \mathbb{R}_+^{n \times n}$ is a weighted $n \times n$ adjacency matrix where p_{ji} are nonnegative elements. A directed edge from node i to node j is denoted by $\varepsilon_{ji} = (j, i) \in \mathcal{E}$, and represents a directed information exchange link from node i to node j, i.e., it denotes that node j can receive information from node i. A directed edge $\varepsilon_{ji} \in \mathcal{E}$ if and only if $p_{ji} > 0$. The graph is undirected if and only if $\varepsilon_{ji} \in \mathcal{E}$ implies $\varepsilon_{ij} \in \mathcal{E}$.

All nodes that can transmit information to node j directly are said to be in-neighbors of node j and belong to the set $\mathcal{N}_j^- = \{i \in \mathcal{V} : \varepsilon_{ji} \in \mathcal{E}\}$. The cardinality of $\mathcal{N}_j^-$, is called the *in-degree* of j and it is denoted by $d_j^- = |\mathcal{N}_j^-|$. The nodes that receive information from node j are called out-neighbors of node j and belong to the set $\mathcal{N}_j^+ = \{\ell \in \mathcal{V} : \varepsilon_{\ell j} \in \mathcal{E}\}$. The cardinality of $\mathcal{N}_j^+$, is called the *out-degree* of j and it is denoted by $d_j^+ = |\mathcal{N}_j^+|$.

A directed graph is called *strongly* connected if there exist a path from each vertex in the graph to every other vertex. This means paths in each direction: a path from i to j and vice versa, for all $i, j \in \mathcal{V}$. In other words, for any $j, i \in \mathcal{V}$, $j \neq i$, one can find a sequence of nodes $i = l_1$, l_2, l_3, …, $l_t = j$ such that link $(l_{k+1}, l_k) \in \mathcal{E}$ for all $k = 1, 2, \ldots, t-1$.

III. Theoretical Preliminaries

In this section, we establish some basic notions that are needed for the development of our algorithms.

Each node j updates and sends its information regarding a variable of interest to its out-neighbors (and also receives similar information from its in-neighbors) at discrete times $t_0, t_1, t_2, \ldots$. We index nodes' information states and any other information at time t_k by k. We use $x_j[k] \in \mathbb{R}$ to denote the information state (or estimate) of node j at time t_k. When there exist no delays in the communication links, each node updates its information state $x_j[k]$ by combining the available information received by its in-neighbors, i.e., the values $\{x_i[k] \mid i \in \mathcal{N}_j^-\}$ using a weighted linear combination. More specifically, the positive weights $p_{ji}[k]$ capture the weight of the information inflow from agent i to agent j at time k (note that unspecified weights in P correspond to pairs of nodes (j, i) that are not connected and are set to zero, i.e., $p_{ji}[k] = 0$, $\forall \varepsilon_{ji} \notin \mathcal{E}$). In this work, we assume that each node j can choose its self-weight and the weights on its d_j^+ out-going links only (i.e., node j chooses the weights $\{p_{\ell j} \mid \ell \in \mathcal{N}_j^+ \cup \{j\}\}$) . Hence, in its general form, each node updates its information state according to the following relation:

$$x_j[k+1] = p_{jj}[k] x_j[k] + \sum_{i \in \mathcal{N}_j^-} p_{ji}[k] x_i[k], \qquad (1)$$

for $k \geq 0$, where $x_j[0] \in \mathbb{R}$ is the initial state of node j. If we let $x[k] = (x_1[k] \;\; x_2[k] \;\; \ldots \;\; x_n[k])^T$ and $P[k] = [p_{ji}[k]] \in \mathbb{R}_+^{n \times n}$, then (1) can be written in matrix form as

$$x[k+1] = P[k] x[k], \qquad (2)$$

where $x[0] = (x_1[0] \;\; x_2[0] \;\; \ldots \;\; x_n[0])^T \equiv x_0$. We first consider a static network and hence the graph remains invariant. In this case, the weights can be chosen to be constant for all times k (i.e., $p_{ji}[k] = p_{ji}$ $\forall k$), and equation (2) can be expressed as $x[k+1] = P x[k]$, where $P[k] = P \in \mathbb{R}_+^{n \times n}$. In this work, we will be employing algorithms in which nodes reach asymptotic average consensus, i.e., are able to calculate (for large k) the average of their initial values. In other words, we would like

$$\lim_{k \to \infty} x_j[k] = \frac{\sum_{l=1}^{n} x_l[0]}{n} \;, \forall j \in \mathcal{V} \; . \qquad (3)$$

The necessary and sufficient conditions for (2) (with $P[k] = P$) to reach average consensus are the following: (a) P has a simple eigenvalue $\lambda_i(P) = 1$ with left eigenvector $\mathbb{1}^T$ and right eigenvector $\mathbb{1}$, and (b) all other eigenvalues of P ($\lambda_j(P), j \neq i$) have magnitude less than 1 ($|\lambda_j(P)| < 1$). If $P \geq 0$ (as in our case), the necessary and sufficient condition is that P is a primitive doubly stochastic matrix. In an undirected graph, assuming each node knows n (or an upper bound $n', n' \geq n$) and the graph is connected, each node can distributively choose the weights on its outgoing links so that the resulting P is primitive doubly stochastic (e.g., [22]). For directed graphs, it is not straightforward to choose weights to make P doubly stochastic.

In [23], an algorithm is suggested that solves the average consensus problem on a directed graph using a column stochastic (but not necessarily row stochastic) matrix. More specifically, asymptotic average consensus is reached by running two consensus algorithms with appropriately chosen initial conditions. We make the following assumption:

(A1) The graph is strongly connected, and the (nonnegative) weights p_{ji} are nonzero for $j = i$ and $(j, i) \in \mathcal{E}$, and satisfy $\sum_{\ell=1}^{n} p_{\ell j} = 1$ for all $j \in \mathcal{V}$ (so that they form a column stochastic matrix P).

The algorithm is as follows.

Lemma 1: [23] Let $\hat{x}_j[k]$ and $\check{x}_j[k]$, $\forall j$ be the result of the iterations

$$\hat{x}_j[k+1] = p_{jj}\hat{x}_j[k] + \sum_{i \in \mathcal{N}_j^-} p_{ji}\hat{x}_i[k] \ , \ \forall j \in \mathcal{V}, \quad (4)$$

$$\check{x}_j[k+1] = p_{jj}\check{x}_j[k] + \sum_{i \in \mathcal{N}_j^-} p_{ji}\check{x}_i[k] \ , \ \forall j \in \mathcal{V}, \quad (5)$$

for $k = 0, 1, 2, \ldots$, initial conditions $\hat{x}[0] = x_0$ and $\check{x}[0] = \mathbb{1}$, and a matrix $P = [p_{ji}]$ that is primitive column stochastic. Then, the solution to the average consensus problem can be asymptotically achieved as

$$\lim_{k \to \infty} \mu_j[k] = \frac{\sum_{\ell=1}^{n} \hat{x}_\ell[0]}{n} \ , \ \ \forall j \in \mathcal{V} \ ,$$

where $\mu_j[k] = \frac{\hat{x}_j[k]}{\check{x}_j[k]}$.

Integer $\tau_{ji}[k] \geq 0$ is used to represent the delay of a message sent from node i to node j at time instant k. Under Assumption (A2), we require that $0 \leq \tau_{ji}[k] \leq \bar{\tau}_{ji} \leq \bar{\tau}$ for all $k \geq 0$ for some finite $\bar{\tau} = \max\{\bar{\tau}_{ji}\}$, $\bar{\tau} \in \mathbb{Z}_+$. Also, we make the reasonable assumption that $\tau_{jj}[k] = 0$, $\forall j \in \mathcal{V}$, at all time instances k (i.e., the own value of a node is always available without delay). In summary, we make the following assumption:

(A2) There exists a finite $\bar{\tau}$ that uniformly bounds the delay terms; i.e. $\tau_{ji}[k] \leq \bar{\tau} < \infty$ for all links $(j, i) \in \mathcal{E}$ at time instant k. In addition $\tau_{jj}[k] = 0$ for all $j \in \mathcal{V}$ and all k.

An adaptation of the above approach to a protocol where each node updates its information state $x_j[k+1]$ by combining the available information (possibly delayed) received by its neighbors $x_i[s]$ ($s \in \mathbb{Z}, s \leq k$, $i \in \mathcal{N}_j^-$) using constant positive weights p_{ji} was developed in [24]. Specifically, each node updates its information state according to the following relation:

$$x_j[k+1] = p_{jj}x_j[k] + \sum_{i \in \mathcal{N}_j^-} \sum_{r=0}^{\bar{\tau}} p_{ji}x_i[k-r]I_{k-r,ji}[r], \quad (6)$$

for $k \geq 0$, where $x_j[0] \in \mathbb{R}$ is the initial state of node j, p_{ji} satisfy Assumption (A1), and

$$I_{k,ji}(\tau) = \begin{cases} 1, & \text{if } \tau_{ji}[k] = \tau, \\ 0, & \text{otherwise.} \end{cases}$$

In the absence of delay, we have $\tau_{ji}[k] = 0$ and the update relation (6) reduces to (1) with constant weights. It is established in [24] that the distributed coordination algorithm leads to asymptotic average consensus, regardless of the nature and order of the delays, as long as they are bounded (and the weight matrix P is primitive column stochastic).

Lemma 2: [24, *Lemma 2*] Let $\hat{x}_j[k]$, $\forall j \in \mathcal{V}$, be the result of iteration (6) with initial conditions $\hat{x}[0] = x_0$, and let $\check{x}_j[k]$, $\forall j \in \mathcal{V}$, be the result of iteration (6) with initial condition $\check{x}[0] = \mathbb{1}$. Then, the solution to the average consensus problem can be obtained as

$$\lim_{k \to \infty} \mu_j[k] = \frac{\sum_{\ell=1}^{n} \hat{x}_\ell[0]}{n}, \quad \forall j \in \mathcal{V} \ ,$$

where $\mu_j[k] = \frac{\hat{x}_j[k]}{\check{x}_j[k]}$.

Note that the two iterations are coupled via $I_{k,ji}(\cdot)$ which are identical for both iterations (i.e., it is assumed that the values on each link for each iteration undergo the same delays). Summarizing, by simultaneously running two iterations $\hat{x}[k]$ and $\check{x}[k]$ as in (6), with initial conditions $\hat{x}[0] = x_0$ and $\check{x}[0] = \mathbb{1}$, respectively, then average consensus is asymptotically reached with the ratio $\hat{x}_j[k]/\check{x}_j[k]$, $\forall j \in \mathcal{V}$, provided the graph characterizing the network is strongly connected, even in the presence of delays in the communication between nodes (as long as these delays are bounded).

IV. Distributed estimation of the number of nodes in a directed graph

We now introduce an algorithm in which the nodes distributively adjust the weights of their outgoing links such that the algorithm converges to $1/|\mathcal{V}|$, thus revealing the number of nodes in the graph. First, we will describe a synchronous algorithm for estimating the number of nodes in a directed graph and subsequently we will extend this approach to an asynchronous algorithm. Finally, we consider the problem of estimating the number of nodes in a time-varying network as nodes join and leave the network.

A. Synchronous estimation of the number of nodes

We assume that each node sets the weights on the edges to its out-neighbors (so that they satisfy Assumption (A1)), and observes but cannot set the edge weights of its in-neighbors. Given a strongly connected digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the distributed algorithm has each node j decide for two initial values, say $\check{x}_j[0] = y_j[0]$ and $\hat{x}_j[0] = z_j[0]$.

Before presenting the general case, a simplified algorithm will be described in which the initial values are set *a priori*. In this simplified algorithm, all nodes have $y_j[0] = 1$ and $z_j[0] = 0$, except one node, say i, that has initial conditions $y_i[0] = z_i[0] = 1$. By running the iteration (4) simultaneously with these initial conditions, for a matrix $P = [p_{ji}]$ that is primitive column stochastic, the number of nodes can be asymptotically obtained as

$$\lim_{k \to \infty} \mu_j[k] = \frac{\sum_{\ell=1}^{|\mathcal{V}|} z_\ell[0]}{\sum_{\ell=1}^{|\mathcal{V}|} y_\ell[0]} = \frac{1}{|\mathcal{V}|}, \quad \forall j \in \mathcal{V}. \quad (7)$$

Thus, each node is able to calculate the size of the network.

However, it might be challenging to orchestrate the initial states of all nodes in the network to be zero apart from one node only. Towards this end, it is necessary that we find a way to distributively overcome this problem. Therefore, each node j, before entering the iterative stage, randomly chooses a value between 0 and $w_{\max}$ ($w_{\max}$ can be set to 1 without any loss of generality); this value is used in a *max-consensus* algorithm in order to effectively decide a leader in the graph. Once the leader is determined, the nodes enter a stage where each node $j \in \mathcal{V}$ performs the steps outlined earlier. The problem with the above approach is that, since the number of nodes is not known apriori, the nodes are not in a position to know when the max-consensus algorithm completes; note, however, that in the algorithm below, the leader election — max-consensus — and the iteration in (4) are effectively performed simultaneously. Each node $j \in \mathcal{V}$, does the following:

(i) Initialize $w_j[0] \sim U(0, w_{\max}), y_j[0] := 1, z_j[0] := 1, \phi_j := 1$.
(ii) Calculate the states $y_j[k+1]$ and $z_j[k+1]$ according to iteration (4).
(iii) Update $w_j[k+1]$ with the maximum w_j among the in-neighbors and itself (max-consensus algorithm).
(iv) The first time $w_j[k+1] \neq w_j[k]$, subtract 1 from $z_j[k+1]$.
(v) Calculate the ratio $\mu_j[k] = z_j[k]/y_j[k]$.

Algorithm 1 is described in detail below.

Algorithm 1 Synchronous distributed estimation of the number of nodes $|\mathcal{V}|$ in a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

```
Require: $w_j[0] \sim U(0, w_{\max}), y_j[0] := 1, z_j[0] := 1, \phi_j := 1,\ 1 \leq j \leq n$
Ensure: $\mu_j[k],\ 1 \leq j \leq n$
  for $k = 0, 1, 2, \ldots,$ each node $j$ will do
    RECEIVE: $\{y_i[k], z_i[k], w_i[k] \mid i \in \mathcal{N}_j^-\}$
    $y_j[k+1] := p_{jj} y_j[k] + \sum_{i \in \mathcal{N}_j^-} p_{ji} y_i[k];$
    $z_j[k+1] := p_{jj} z_j[k] + \sum_{i \in \mathcal{N}_j^-} p_{ji} z_i[k];$
    $w_j[k+1] := \max_{i \in \mathcal{N}_j^- \cup \{j\}} (\ldots, w_i[k], \ldots);$
    if $w_j[k+1] \neq w_j[k]$ & $\phi_j = 1$ then
      $z_j[k+1] := z_j[k+1] - 1;$
      $\phi_j := 0;$
    end if
    TRANSMIT: $\{y_j[k+1], z_j[k+1], w_j[k+1]\}$
    $\mu_j[k] := \frac{z_j[k]}{y_j[k]}$
  end for
```

The idea behind the above algorithm is the following. Each node picks a random value ($w_j[0]$) and the max-consensus algorithm ensures that the nodes eventually discover who has selected the maximum value. Initially, each node sets $z_j[0] = 1$ (so that $\sum_\ell z_\ell[0] = n$); however, as soon as a node discovers that it does not have the maximum value, its removes 1 from its z-value, ensuring that after n steps (at which point, all nodes have determined $w_{\max}$), we will have $\sum_j z_j[k] = 1$ for $k \geq n$. Therefore, ratio consensus will converge to

$$\lim_{k \to \infty} \mu_j[k] = \frac{\sum_{\ell=1}^{|\mathcal{V}|} z_\ell[n]}{\sum_{\ell=1}^{|\mathcal{V}|} y_\ell[n]} = \frac{1}{\sum_{\ell=1}^{|\mathcal{V}|} y_\ell[0]} = \frac{1}{|\mathcal{V}|}$$

for all $j \in \mathcal{V}$.

Remark 1: In [25, Theorem 9] the *impossibility result* is stated as follows: there exists no algorithm that is able to compute the number of nodes in an anonymous network, that terminates with the correct result for every finite execution with probability one, and that has a bounded average bit complexity (i.e., the average number of bits used by the algorithm is bounded). The coherency with the *impossibility result* is given by the facts that the algorithm is asymptotic and all the strategies that we propose use max-consensus, which implies that two nodes could choose the same maximum value with some nonzero probability if the number of bits used is bounded.

B. Asynchronous estimation of the number of nodes

The algorithm can be generalised for the case we have asynchronous updates, or equivalently, when we have delays in the system. We assume that the information $\{y_i, z_i, w_i\}$ that arrives to node j from in-neighbor $i \in \mathcal{N}_j^-$ is all transmitted in the same packet, so that delays affect y_i, z_i and w_i in the same manner. The difference from Algorithm 1 is that the iteration is updated by (6). *Algorithm 2* is described in detail below.

Algorithm 2 Asynchronous distributed estimation of the number of nodes $|\mathcal{V}|$ in a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

```
Require: $w_j[0] \sim U(0, w_{max}), y_j[0] := 1, z_j[0] := 1, \phi_j := 1,\ 1 \leq j \leq n$
Ensure: $\mu_j[k],\ 1 \leq j \leq n$
  for $k = 1, 2, \ldots$ each node $j$ will do
    RECEIVE: for $i \in \mathcal{N}_j^-$ and $r = 0, 1, ..., \overline{\tau}$, receive $\{y_i[k-r] I_{k-r,ji}[r], z_i[k-r] I_{k-r,ji}[r], w_i[k-r] I_{k-r,ji}[r]\}$
    $y_j[k+1] := p_{jj} y_j[k] + \sum_{i \in \mathcal{N}_j^-} \sum_{r=0}^{\overline{\tau}} p_{ji} y_i[k-r] I_{k-r,ji}[r];$
    $z_j[k+1] := p_{jj} z_j[k] + \sum_{i \in \mathcal{N}_j^-} \sum_{r=0}^{\overline{\tau}} p_{ji} z_i[k-r] I_{k-r,ji}[r];$
    $w_j[k+1] := \max_{i \in \mathcal{N}_j^- \cup \{j\}, r=0,1,\ldots,\overline{\tau}} (\ldots, w_i[k-r] I_{k-r,ji}[r], \ldots);$
    if $w_j[k+1] \neq w_j[k]$ & $\phi_j = 1$ then
      $z_j[k+1] := z_j[k+1] - 1;$
      $\phi_j := 0;$
    end if
    TRANSMIT: $\{y_j[k+1], z_j[k+1], w_j[k+1]\}$
    $\mu_j[k] := \frac{z_j[k]}{y_j[k]}$
  end for
```

Lemma 3: Consider a network isomorphic to the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Then, under Assumptions (A1) and (A2), Algorithm 2 will lead to $\lim_{k \to \infty} \mu_j[k] = 1/n$, $\forall j \in \mathcal{V}$.

To prove Lemma 3 we first state the following result that is a direct consequence of Lemma 2.

Corollary 1: Consider a network isomorphic to the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Under Assumptions (A1)–(A2), for each $j \in \mathcal{V}$, let

$$y_j[k+1] = p_{jj} y_j[k] + \sum_{i \in \mathcal{N}_j^-} \sum_{r=0}^{\bar{\tau}} p_{ji} y_i[k-r] I_{k-r,ji}[r]$$

$$z_j[k+1] = p_{jj} z_j[k] + \sum_{i \in \mathcal{N}_j^-} \sum_{r=0}^{\bar{\tau}} p_{ji} z_i[k-r] I_{k-r,ji}[r]$$

where all nodes have $y_j[0] = 1$ and $z_j[0] = 0$, except one node, say i, that has initial conditions $y_i[0] = z_i[0] = 1$. Then,

$$\lim_{k \to \infty} \mu_j[k] = \frac{1}{n} \ , \ \ \forall j \in \mathcal{V} \ ,$$

where $\mu_j[k] = \frac{y_j[k]}{z_j[k]}$.

C. Estimation of the number of nodes in a network with varying number of nodes

The number of nodes in a network might change for various reasons (e.g., draining of a battery, removal of a sensor, etc.). Thus, it is of vital importance to be able to estimate the number of nodes in a network when nodes enter or leave the network. In this subsection, we show how our approach can be used to estimate the number of nodes in a network, as nodes join and leave the network, given that the nodes remain aware of the number of their out-neighbors and the graph remains strongly connected.

Consider the case where at time $k = k_s$ a node s joins the network, for the size estimation to work $w_s[k_s] = 0$, $y_s[k_s] = 1$, and $z_s[k_s] = 0$. For the case where a node f fails at time $k = k_f$, we assume that it first transmits $w_f[k_f]$, $y_f[k_f]$, $z_f[k_f]$, d_f^+, and a boolean value $\delta_f[k_f]$. The boolean variable $\delta_f[k_f]$ is equal to one if $w_f[k_f] = w_f[k_f - T_f]$ (zero otherwise), where T_f is the time elapsed since the last time an in-neighbor i of node f, $i \in \mathcal{N}_f^-$, with $\delta_i[k_f - T_f] = 1$ left the network in the past (if no such in-neighbor left the network before, then $T_f = k_f$). For $j \in \mathcal{N}_f^+$ we have

$$\begin{aligned} y_j[k_f] &:= y_j[k_f] + \frac{y_f[k_f] - 1}{d_f^+} \\ z_j[k_f] &:= z_j[k_f] + \frac{z_f[k_f] - \delta_f[k_f]}{d_f^+} \\ w_j[k_f] &:= \begin{cases} w_j[k_f], & \delta_f[k_f] = 0 \\ \sim U(w_f[k_f], w_{max}), & \delta_f[k_f] = 1 \end{cases} \\ z_j[k_f] &:= \begin{cases} z_j[k_f], & \delta_f[k_f] = 0 \\ z_j[k_f] + 1, & \delta_f[k_f] = 1 \text{ and } \delta_j[k_f] = 0 \end{cases} \end{aligned} \tag{8}$$

The directed edges are in principle removed by the nodes initiated, i.e., from the transmitting nodes. This can be done by simply sending an acknowledgement to the receiving node that the edge will be removed, so that both nodes remove the edge (and the node corresponding to this edge). Note that, there is no problem when this is done asynchronously, since the transmitting node does not pass any other information to the receiving node through that edge.

V. Average Degree Estimation and Control

Using the estimates of the number of edges and nodes in the network, each node j can estimate other parameters of the network locally. One such parameter is the *average node degree*. Using this information, each node can control this global parameter of the network through local actions, and consequently the nodes can control the average node degree of the whole graph in a distributed manner.

Consider a time varying network with average node out-degree $\bar{d}[k]$ at time k. It is desired to achieve the average node degree $1 < d^\star \leq n - 1$ through local edge addition and deletion at each node. Note that d^* is taken to be an integer multiple of $\frac{1}{n}$, as other values of d^* would lead to unnecessary oscillations. [However, if this is not the case, we can set below $\gamma[k] = 0$ when $|d^* - \bar{d}[k]| < 1/n$ and avoid oscillations.] Let $\mathcal{N}_j^+[k]$ be the set of out-neighbours of node j at time k, and let $\overline{\mathcal{N}}_j[k] = \mathcal{V} \setminus \mathcal{N}_j^+[k]$. Let $d_j^+[k]$ be the out-degree of node j at time k. At each time step k, node j either adds a new link to node $\bar{l}$, uniformly picked from the nodes in $\overline{\mathcal{N}}_j[k]$, or removes its link to node l uniformly picked from the nodes in $\mathcal{N}_j^+[k]$, or does not do anything. For now, assume that $\bar{d}[k]$ is available through an oracle to each node at time k.

Then, the following equation governs the degree update of node j at each time step k:

$$d_j^+[k+1] = d_j^+[k] + b_j[k]\gamma[k], \tag{9}$$

where $b_j[k] = L[k] + R_j[k]$ is comprised of an integer $L[k] := \lfloor c|d^\star - \bar{d}[k]| \rfloor$ and a Bernoulli variable $R_j[k]$ with probability $\mathbb{P}[k]$:

$$\mathbb{P}[k] = c|d^\star - \bar{d}[k]| - \lfloor c|d^\star - \bar{d}[k]| \rfloor, \tag{10}$$

where $\lfloor \cdot \rfloor$ is the maximum integer that is smaller than its argument, $c \in (0, 1]$ and parameter $\gamma[k]$ is

$$\gamma[k] = \begin{cases} 1, & d^\star - \bar{d}[k] \geq 1/n, \\ 0, & |d^\star - \bar{d}[k]| < 1/n, \\ -1, & d^\star - \bar{d}[k] \leq -1/n. \end{cases} \tag{11}$$

Lemma 4: Under (9)-(11), the expected value of $\bar{d}[k]$ goes to the desired value $d^\star$ as $k \to \infty$.

Proof: First, noting that $\bar{d}[k] = \frac{\sum_{\ell=1}^{n} d_\ell^+[k]}{n}$, (9) becomes

$$\bar{d}[k+1] = \bar{d}[k] + \frac{\sum_{\ell=1}^{n} b_\ell[k]\gamma[k]}{n} \tag{12}$$

Taking the expected value of $\bar{d}[k+1]$ in (12), we obtain

$$\begin{aligned} \mathbb{E}(\bar{d}[k+1]) &= \mathbb{E}(\bar{d}[k]) + \frac{\sum_{\ell=1}^{n} \mathbb{E}(b_\ell[k])\gamma[k]}{n} \\ &= \mathbb{E}(\bar{d}[k]) + (L[k] + \mathbb{P}[k])\gamma[k] \\ &= \mathbb{E}(\bar{d}[k]) + c|d^\star - \bar{d}[k]|\gamma[k], \end{aligned}$$

where $\mathbb{E}(\cdot)$ is the expected value of its argument. Since, $\mathbb{E}(\mathbb{E}(\cdot)) = \mathbb{E}(\cdot)$, when $|d^\star - \bar{d}[k]| > 1/n$, we obtain

$$\mathbb{E}(\bar{d}[k+1]) = (1 - c)\mathbb{E}(\bar{d}[k]) + cd^\star.$$

Hence, $\mathbb{E}(\bar{d}[k])$ converges asymptotically to $d^\star$. ■

Next, we consider the case where nodes do not have access to the global measurements on the average degree of the network. Thus, for node j to be able to achieve the objective, it needs to be able to estimate (in a distributed manner) the value of $\bar{d}[k]$, which we denote by $\hat{d}_j[k]$, using their local measurements. This can be achieved in two ways:
(1) each node estimates the number of nodes and edges in the network and then calculates the average degree.
(2) Each agent j just sets the initial conditions $y_j[0] = d_j^+$ (or $y_j[0] = d_j^-$) and $z_j[0] = 1$, and then uses ratio consensus to (at least asymptotially) obtain the average node degree.

In the rest of this paper, we focus on the second method as it requires less computations and only relies on one ratio consensus algorithm to converge. Additionally, It does not require any coordination among the nodes to determine which $z_j[0]$ should be equal to zero and which one should be equal to one. We propose Algorithm 3 to control and estimate the average node degree distributedly.

Algorithm 3 Simultaneous Average Degree Estimation and Control.

```
Require: $y_j[0] := d_j^+, z_j[0] := 1$, $1 \leq j \leq n$
Ensure: $\hat{d}_j[k] = \bar{d}[k] = d^\star$, $1 \leq j \leq n$, $k \to \infty$
  for $k = 1, 2, \ldots$ each node $j$ will do
    $y_j[k+1] := p_{jj} y_j[k] + \sum_{i \in \mathcal{N}_j^-} \sum_{r=0}^{\bar{\tau}} p_{ji} y_i[k-r] I_{k-r,ji}[r]$;
    $z_j[k+1] := p_{jj} z_j[k] + \sum_{i \in \mathcal{N}_j^-} \sum_{r=0}^{\bar{\tau}} p_{ji} z_i[k-r] I_{k-r,ji}[r]$;
    $\hat{d}_j[k] := \frac{y_j[k]}{z_j[k]}$
    $\mathbb{P}_j[k] = \big(c|d^\star - \hat{d}_j[k]| - \lfloor c|d^\star - \hat{d}_j[k]| \rfloor\big)$
    $L_j[k] := \lfloor c|d^\star - \hat{d}_j[k]| \rfloor$
    $b_j[k] := L_j[k] + \mathbb{P}_j[k]$
    $\gamma_j[k] = \begin{cases} 1, & d^\star - \hat{d}_j[k] \geq 1/n \\ 0, & |d^\star - \hat{d}_j[k]| < 1/n \\ -1, & d^\star - \hat{d}_j[k] \leq -1/n \end{cases}$
    $d_j^+[k+1] = d_j^+[k] + b_j[k]\gamma_j[k]$
    $y_j[k+1] := y_j[k+1] + b_j[k]\gamma_j[k]$
  end for
```

Remark 2: Average node degree can also be accurately estimated using classical consensus algorithms (e.g., [22], [26]), but as in the average consensus algorithms a doubly stochastic matrix (or a sequence of doubly stochastic matrices in the case of time-varying graphs) is required. On the contrary, ratio consensus relies on two iterations using a column stochastic matrix (or a sequence of column stochastic matrices in the case of time-varying graphs), leading to the exact average node degree both synchronously and asynchronously.

Lemma 5: Under Algorithm 3, the expected value of $\bar{d}[k]$ goes to the desired value $d^\star$ as $k \to \infty$.

Proof: For simplicity we will show that the algorithm converges when it operates synchronously. It can be similarly proved for the asynchronous operation. From Algorithm 3 $y_j[k+1]$ can be expressed as

$$y_j[k+1] = p_{jj} y_j[k] + \sum_{i \in \mathcal{N}_j^-} p_{ji} y_i[k] + b_j[k]\gamma_j[k].$$

By taking the expected value of $\mathbb{E}(y_j[k+1])$, this is equivalent to

$$\begin{aligned}\mathbb{E}(y_j[k+1]) =& p_{jj}\mathbb{E}(y_j[k]) \\ &+ \sum_{i \in \mathcal{N}_j^-} p_{ji}\mathbb{E}(y_i[k]) + c\big(d^\star - \mathbb{E}(\hat{d}_j[k])\big).\end{aligned}$$

Substituting $\hat{d}_j[k] = \frac{y_j[k]}{z_j[k]}$, we get

$$\begin{aligned}z_j[k+1]\mathbb{E}(\hat{d}_j[k+1]) &= p_{jj} z_j[k]\mathbb{E}(\hat{d}_j[k]) \\ &+ \sum_{i \in \mathcal{N}_j^-} p_{ji} z_i[k]\mathbb{E}(\hat{d}_i[k]) + c\big(d^\star - \mathbb{E}(\hat{d}_j[k])\big).\end{aligned}$$

Note that the iteration of $z_j[k]$ is deterministic and it is not affected by the estimate for the average degree estimate.Thus, in a strongly connected graph z will converge asymptotically to $z^\star$, whatever happens with the addition or removal of links. Hence, without loss of generality we can assume that the iteration of $z_j[k]$ has already converged. Then,

$$\begin{aligned}\mathbb{E}(\hat{d}_j[k+1]) &= p_{jj}\mathbb{E}(\hat{d}_j[k]) \\ &+ \sum_{i \in \mathcal{N}_j^-} p_{ji}\frac{z_i^\star}{z_j^\star}\mathbb{E}(\hat{d}_i[k]) + \frac{c}{z_j^\star}\big(d^\star - \mathbb{E}(\hat{d}_j[k])\big).\end{aligned}$$

In matrix form for all nodes j in the network,

$$\mathbb{E}(\hat{d}[k+1]) = (Z^{-1}PZ - cZ^{-1})\mathbb{E}(\hat{d}[k]) + cd^\star \bar{z}, \qquad (13)$$

where $\mathbb{E}(\hat{d}[k]) := \big(\mathbb{E}(\hat{d}_1[k]) \ \ldots \ \mathbb{E}(\hat{d}_n[k])\big)^T$, P is a nonnegative stochastic matrix, I is the identity matrix with appropriate dimensions, $Z := \text{diag}_j(z_j^\star)$, and $\bar{z} := \big(z_1^{-1} \ \ldots \ z_n^{-1}\big)^T$. Given that the spectral radius of P, $\rho(P)$, is 1, then $\rho(Z^{-1}PZ) = 1$ and $\mathbb{E}(\bar{d}[k])$ converges to

$$\mathbb{E}(\hat{d}^\star) = (I + cZ^{-1} - Z^{-1}PZ)^{-1} cd^\star \bar{z},$$

where $\mathbb{E}(\hat{d}^\star) := \big(\mathbb{E}(\hat{d}_1^\star) \ \ldots \ \mathbb{E}(\hat{d}_n^\star)\big)^T$. If we pre-multiply Equation (13) with Z when it has converged, we get

$$Z\mathbb{E}(\hat{d}^\star) = (PZ - cI)\mathbb{E}(\hat{d}^\star) + cd^\star \mathbb{1}. \qquad (14)$$

Now, pre-multiplying (14) by $\mathbb{1}^T$ we obtain

$$z^T\mathbb{E}(\hat{d}^\star) = (z^T - c\mathbb{1}^T)\mathbb{E}(\hat{d}^\star) + cd^\star n.$$

After manipulation, this gives $\dfrac{\mathbb{1}^T\mathbb{E}(\hat{d}^\star)}{n} = d^\star$. ■

VI. Numerical Examples

The outcomes of the application of Algorithms 1 and 2 to a network with time varying interconnections are depicted in Figs. 1 and 2, respectively. In the case of asynchronous communications, the maximum delay is assumed to be 5.

In the next scenario we consider the case where nodes have access to global information regarding the average degree of the network, and apply (9) to control this value and achieve

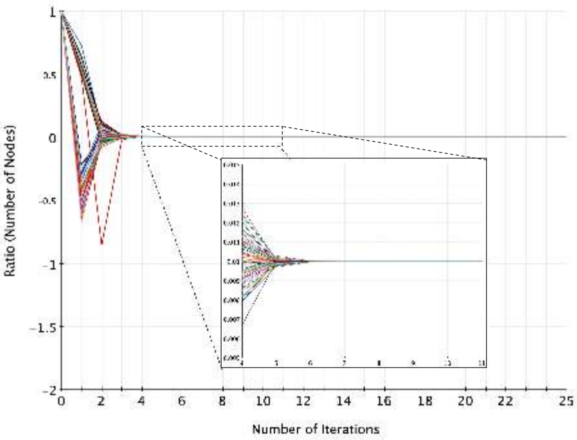

Fig. 1. Estimation of the number of nodes using Algorithm 1 in a network with 100 nodes. All node ratios converge to 0.01 within 10 iterations, and hence, each node can determine that the number of nodes is $1/0.01 = 100$.

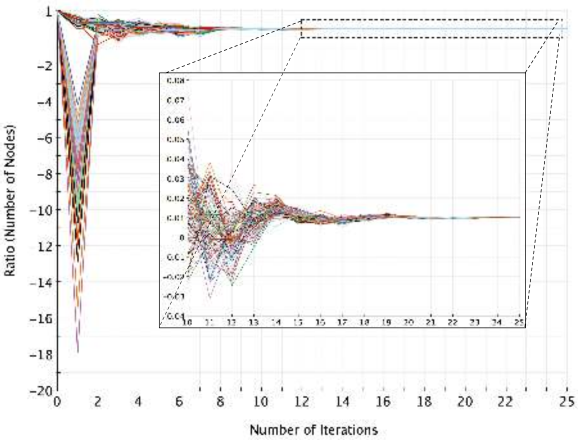

Fig. 2. Estimation of the number of nodes using Algorithm 2 in a network with 100 nodes in the presence of bounded delays (maximum delay is 5).

a desired average degree. The network is assumed to be a random network with 50 nodes and the desired average degree is 43.24. The average degree of the network and the degree of each of the nodes are depicted in Figs. 3 and 4, respectively.

In the last scenario, we assume that the agents do not have access to any global information regarding the average degree of the network and want to achieve the desired average node degree via Algorithm 3. It is assumed that the network is randomly generated initially and the nodes are fixed. Fig. 5 depicts the network average node degree, Fig. 6 represents the degree of each node when Algorithm 3 is applied, and Fig. 7 shows the estimate of the network average degree calculated at each node.

VII. Conclusions

In this paper, we propose synchronous and asynchronous distributed algorithms to accurately estimate the number of nodes and the average node degree of a given distributed system. Furthermore, a distributed algorithm is proposed with which each node can introduce/terminate links, to eventually reach a target average degree for the network. Controlling the node degree can be beneficial in many different practical scenarios; for example, in controlling renewable energy resources in a power network, in decreasing the rate of propagation of a pathogen in the network, and others. As demonstrated in the numerical examples, the algorithms are accurate and extremely efficient.

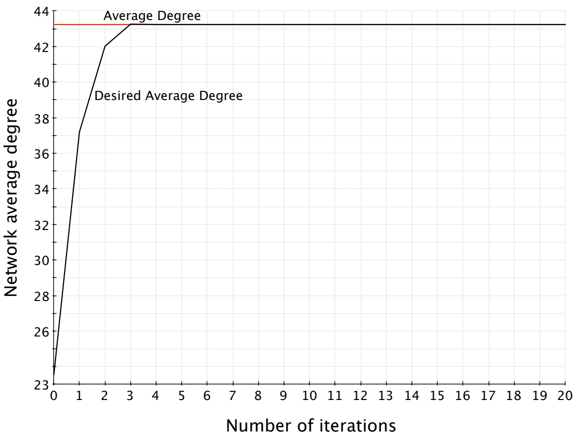

Fig. 3. Network average degree under (9)-(11) with global average degree information in a network with 50 nodes in the presence of bounded delays (maximum delay is assumed to be 5).

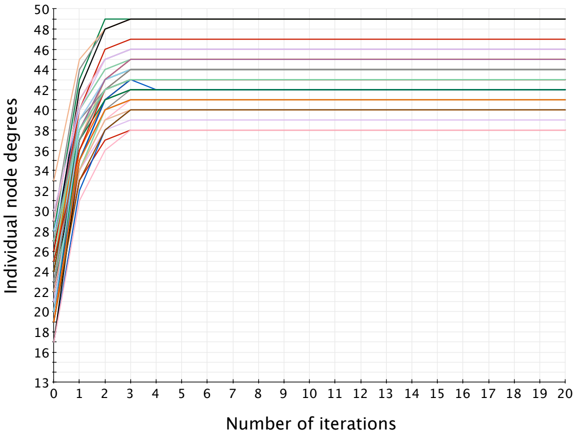

Fig. 4. Nodes degrees under (9)-(11) with global average degree information in a network with 50 nodes in the presence of bounded delays (maximum delay is assumed to be 5).

References

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[2] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.

[3] F. Kuhn, N. Lynch, and R. Oshman, "Distributed computation in dynamic networks," in *Proceedings of the 42nd ACM Symposium on Theory of Computing*, ser. STOC '10. New York, NY, USA: ACM, 2010, pp. 513–522.

[4] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed parametric and nonparametric regression with on-line performance bounds computation," *Automatica*, vol. 48, no. 10, pp. 2468–2481, 2012.

[5] J.-S. Li and X.-D. Zhang, "On the Laplacian eigenvalues of a graph," *Linear Algebra and its Applications*, vol. 285, no. 13, pp. 305–307, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379598101490

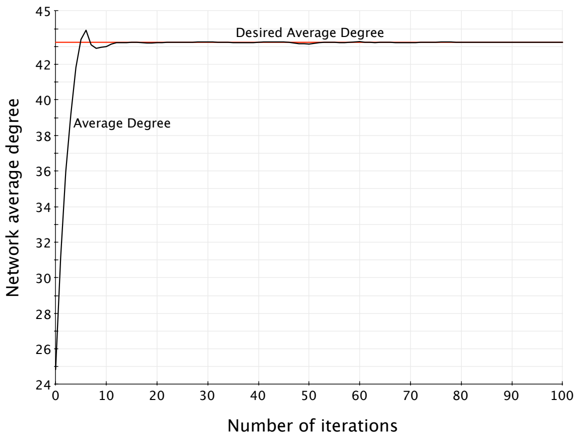

Fig. 5. Network average degree under Algorithm 3 for $c = 1/3$ in a network with 50 nodes in the presence of bounded delays when global average degree information is not available (maximum delay is assumed to be 5).

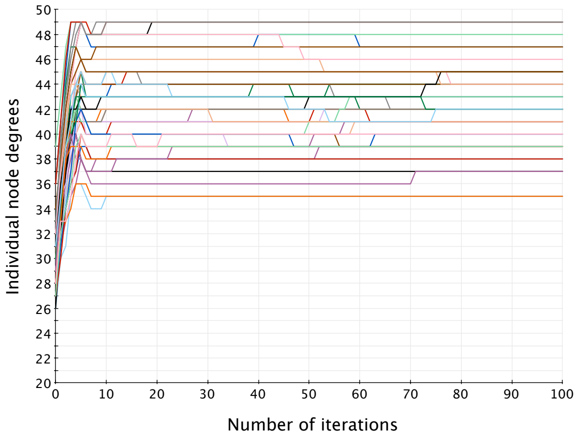

Fig. 6. Nodes degrees under Algorithm 3 for $c = 1/3$ in a network with 50 nodes in the presence of bounded delays (maximum delay is assumed to be 5) when global average degree information is not available.

[6] J. Cardoso, C. Baquero, and P. Almeida, "Probabilistic estimation of network size and diameter," in *Proceedings of Fourth Latin-American Symposium on Dependable Computing (LADC'09)*, 2009, pp. 33–40.

[7] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed statistical estimation of the number of nodes in sensor setworks," in *Proceedings of IEEE Conference on Decision and Control*, Dec. 2010, pp. 1498–1503.

[8] F. Garin, D. Varagnolo, K. Johansson *et al.*, "Distributed estimation of diameter, radius and eccentricities in anonymous networks," in *Proceedings of 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (Necsys' 12)*, 2012.

[9] C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks: algorithms and evaluation," *Performance Evaluation*, vol. 63, no. 3, pp. 241–263, Mar. 2006.

[10] L. Massouliè, E. L. Merrer, A.-M. Kermarrec, and A. Ganesh, "Peer counting and sampling in overlay networks: random walk methods," in *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing*, 2006.

[11] S. Petrovic and P. Brown, "A new statistical approach to estimate global file populations in the eDonkey P2P file sharing system," in *Proceedings of 21st International Teletraffic Congress*, Sep. 2009.

[12] S.-L. Peng, S.-S. Li, X.-K. Liao, Y.-X. Peng, and N. Xiao, "Estimation of a population size in large-scale wireless sensor networks," *Journal of Computer Science and Technology*, vol. 24, no. 5, pp. 987–997, Sep. 2009.

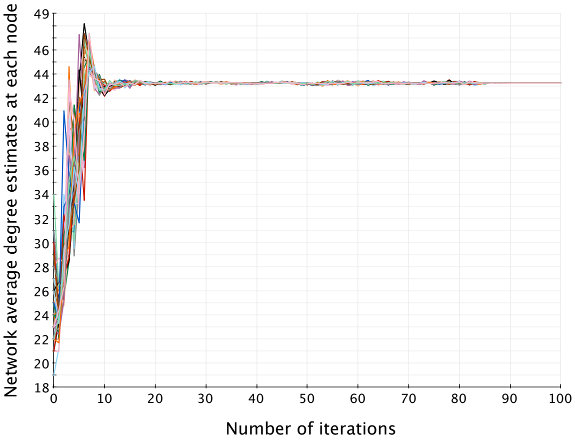

Fig. 7. Average degree estimates at each node under Algorithm 3 for $c = 1/3$ in a network with 50 nodes in the presence of bounded delays (maximum delay is assumed to be 5) when global average degree information is not available.

[13] C. Tang, W. Wang, X. Wu, and B. Wang, "Effects of average degree on cooperation in networked evolutionary game," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 53, no. 3, pp. 411–415, 2006.

[14] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, pp. 1–41, Mar. 2007.

[15] C. Borgs, J. Chayes, A. Ganesh, and A. Saberi, "How to distribute antidote to control epidemics," *Random Structures & Algorithms*, vol. 37, no. 2, pp. 204–222, 2010.

[16] B. Hartnell, "Firefighter! an application of domination," in *Presentation at the 24^{th} Conference on Numerical Mathematics and Computing, University of Manitoba*, 1995.

[17] P. Prałat, "Sparse graphs are not flammable," *Arxiv preprint arXiv:1201.0723*, 2012.

[18] T. Kim, D. Tipper, and P. Krishnamurthy, "Connectivity and critical point behavior in mobile ad hoc and sensor networks," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, 2009, pp. 153–158.

[19] B. C. M. van Wijk, C. J. Stam, and A. Daffertshofer, "Comparing brain networks of different size and connectivity density using graph theory," *PLoS ONE*, vol. 5, no. 10, p. e13701, 10 2010. [Online]. Available: http://dx.doi.org/10.1371\%2Fjournal.pone.0013701

[20] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in 9^{th} *IEEE International Conference on Data Mining (ICDM)*, 2009, pp. 169–178.

[21] O. Goldreich and D. Ron, "On estimating the average degree of a graph," *Electronic Colloquim on Computational Complexity (ECCC)*, 2004.

[22] L. Xiao, S. Boyd, and S. Lall, "Distributed average consensus with time-varying metropolis weights," in *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 63–70.

[23] A. D. Domínguez-García and C. N. Hadjicostis, "Coordination and control of distributed energy resources for provision of ancillary services," in *Proceedings of First IEEE International Conference on Smart Grid Communications*, 2010, pp. 537–542.

[24] C. N. Hadjicostis and T. Charalambous, "Asynchronous coordination of distributed energy resources for the provisioning of ancillary services," in *Proceedings of 49^{th} Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, September 2011, pp. 1500–1507.

[25] I. Cidon and Y. Shavitt, "Message terminating algorithms for anonymous rings of unknown size," *Information Processing Letters*, vol. 54, no. 2, pp. 111–119, Apr. 1995.

[26] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.