DECENTRALIZED ESTIMATION AND CONTROL OF ALGEBRAIC CONNECTIVITY OF RANDOM AD-HOC NETWORKS

Paolo Di Lorenzo and Sergio Barbarossa

Sapienza University of Rome, DIET, Via Eudossiana 18, 00184 Rome, Italy.
E-mail: dilorenzo,sergio@infocom.uniroma1.it

ABSTRACT

In this paper, we propose a decentralized algorithm for the estimation and control of connectivity of random ad hoc networks. First, we introduce a novel stochastic power iteration method that allows each node to estimate and track the expected algebraic connectivity of a random graph. The proposed method is then used to adapt the power transmitted by each node in order to drive the network connectivity toward a desired value. Numerical results illustrate the main features of the algorithm and its robustness to fluctuations of the network graph due to the presence of random link failures.

***Index Terms*—** Random graph, stochastic power iteration method, topology control.

1. INTRODUCTION AND RELATED WORK

Ad hoc networks are composed of a collective of nodes that can exchange data among each other through wireless links, and where each node has a processing unit to perform local computations. Usually, the topology of this kind of networks is created in an ad hoc way, e.g., based on a nearest neighbor criterion to allow for low-power communication. It is widely recognized that the performance of many distributed algorithms, which operate over wireless ad-hoc networks, highly depends on the network topology [1]-[2]-[3]. For example, highly connected networks generally have significantly faster convergence thanks to a more efficient in-network information diffusion.

Spectral graph theory [4] has been demonstrated to be a very powerful tool for topology inference. The eigenvalues and/or eigenvectors of the Laplacian matrix of the graph have been exploited, e.g., to estimate the connectivity of the network [5]-[10], to find densely connected clusters of nodes [11]-[12], and to search for potential links that would greatly improve the connectivity if they would be established [13]. In all these previous works, it was argued and demonstrated that the most useful eigenvector for graph partitioning is the one corresponding to the second-smallest eigenvalue of the Laplacian matrix. This eigenvalue is referred to as the *algebraic connectivity* and its eigenvector is often referred to as the *Fiedler vector* [5]. It is then of interest to find novel and efficient algorithms for the computation of these connectivity parameters.

The problem of distributed estimation of the eigenvalues of the Laplacian matrix has been considered in several previous works, e.g., [6]-[10]. In [6] a distributed algorithm is proposed to find the n eigenvectors corresponding to the n largest eigenvalues of the Laplacian matrix or the (weighted) adjacency matrix, based on power iteration and random walk techniques. The work in [7] evaluates the eigenstructure of the Laplacian matrix by letting the nodes oscillate at the eigenfrequencies corresponding to the network topology. An efficient and distributed algorithm that computes the Fiedler vector and the algebraic connectivity, with application to topology inference in ad hoc networks, has been also proposed in [8]. In [9], the authors propose a distributed algorithm that allows each node to estimate and track the algebraic connectivity of the graph in mobile wireless sensor networks. Then, based on this estimator, a decentralized gradient controller for each agent helps maintain global connectivity during motion. Finally, reference [10] proposes a distributed algorithm to estimate the algebraic connectivity of a graph, thus applying this method to an event-triggered consensus scenario, where the most recent estimate of the algebraic connectivity is used for adapting the behavior of the average consensus algorithm.

All these previous works assumed ideal communications among the network nodes. However, in a realistic scenario, the wireless channel is affected by random fading and additive noise, which induce errors in the received packets. In such a case, the receiving node could request the retransmission of the erroneous packets, but this would imply random delays in the communication among the nodes and it would be complicated to implement over a totally decentralized system. It is then of interest to analyze networks where the erroneous packets are simply dropped, without requiring a retransmission. The effect of random graphs on distributed algorithms has been thoroughly studied in a series of works, mainly focused on the convergence of consensus algorithms, e.g., [18]-[20]. In this work, we propose a distributed algorithm, which we refer to as the stochastic power iteration algorithm, whose aim is to estimate the algebraic connectivity of the expected Laplacian matrix of a random graph, while assuming random impairments in the exchange of data among neighbor nodes. The proposed method is then used to control the expected connectivity of the network by adapting the power transmitted by each node, in order to drive the network connectivity toward a desired value.

2. NETWORK MODEL

We consider a network composed of N nodes interacting according to a communication topology. The interaction among the nodes is modeled as an undirected graph $G = (V, E)$, where $V = 1, 2, ..., N$ denotes the set of nodes and $E \subseteq V \times V$ is the edge set. The structure of the graph is described by a symmetric $N \times N$ adjacency matrix $\boldsymbol{A} := \{a_{ij}\}$, whose entries a_{ij} are either positive or zero, depending on wether there is a link between nodes i and j or not, i.e., if the distance between nodes i and j is less than a coverage radius, which is dictated by nodes' transmit power and the channel between them. The set of neighbors of a node i is $\mathcal{N}_i$, defined as $\mathcal{N}_i = \{j \in V : a_{ij} > 0\}$. Node i communicates with node j if j is a neighbor of i (or $a_{ij} > 0$). Denoting by $d_{ii} = \sum_{j=1}^{M} a_{ij}$

This work has been supported by TROPIC Project, Nr. 318784.

the degree of node i, the degree matrix $\boldsymbol{D}$ is a diagonal matrix with entries d_{ii} that are the row sums of the adjacency matrix $\boldsymbol{A}$. The graph Laplacian $\boldsymbol{L}$ is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. The spectral properties of $\boldsymbol{L}$ have been shown to be critical in many multiagent applications, such as formation control [14], consensus seeking [15] and direction alignment [17]. We denote by $\lambda_i(\boldsymbol{L})$, $i = 1, \ldots, M$, the eigenvalues of $\boldsymbol{L}$, ordered in increasing sense. The $N \times N$ matrix $\boldsymbol{L}$ always has, by construction, a null eigenvalue $\lambda_1(\boldsymbol{L}) = 0$, with associated eigenvector $\boldsymbol{1}_N$ composed of all ones. For a connected graph, the nullspace of $\boldsymbol{L}$ has dimension 1 and it is spanned by the vector $\boldsymbol{1}$. The quantity $\lambda_2(\boldsymbol{L})$ is known as the *algebraic connectivity* of the graph. This eigenvalue is greater than 0 if and only if G is a connected graph. The magnitude of this value reflects how well connected the overall graph is, and has been used in analysing the robustness and synchronizability of networks [14]-[17].

Random link failures : In a realistic communication scenario, the packets exchanged among network nodes may be received with errors, because of collisions, channel fading or noise. The retransmission of erroneous packets can be incorporated into the system, but packet retransmission introduces a nontrivial additional complexity in decentralized implementations and, more importantly, it also introduces an unknown delay and delay jitter. It is then of interest to examine simple protocols where erroneous packets are simply dropped. We take into account random packet dropping by modeling the coefficient a_{ij} describing the network topology as statistically independent random variables. Then, the Laplacian of the graph varies with time as a sequence of i.i.d. matrices $\{\boldsymbol{L}[k]\}$, which can be written, without any loss of generality, as

$$\boldsymbol{L}[k] = \bar{\boldsymbol{L}} + \tilde{\boldsymbol{L}}[k] \tag{1}$$

where $\bar{\boldsymbol{L}}$ denotes the mean matrix and $\tilde{\boldsymbol{L}}[k]$ are i.i.d. perturbations around the mean. We do not make any assumptions about the link failure model. Although the link failures and the Laplacians are independent over time, during the same iteration, the link failures can still be spatially correlated. It is important to remark that we do not require the random instantiations $G[k]$ of the graph be connected for all k. We only require the graph to be connected on average. This condition is captured by requiring $\lambda_2(\bar{\boldsymbol{L}}) > 0$.

3. DECENTRALIZED ESTIMATION OF EXPECTED CONNECTIVITY

In this section, we propose a novel algorithm aimed at estimating the connectivity of a random graph G by computing the second eigenvalue of the expected Laplacian matrix $\bar{\boldsymbol{L}}$. Since in our setting, the network graph is random due to the presence of link failures, in the following, we introduce a stochastic power iteration method that is able to handle the randomness introduced by the graph fluctuation.

Let us consider the matrix $\boldsymbol{W}[k]$ given at time k by:

$$\boldsymbol{W}[k] = \boldsymbol{I} - \varepsilon \boldsymbol{L}[k] = \bar{\boldsymbol{W}} + \tilde{\boldsymbol{W}}[k] \tag{2}$$

where $\bar{\boldsymbol{W}} = \boldsymbol{I} - \varepsilon\bar{\boldsymbol{L}}$ is the mean matrix, $\tilde{\boldsymbol{W}}[k] = -\varepsilon\tilde{\boldsymbol{L}}[k]$ are i.i.d. fluctuations around the mean, and $0 < \varepsilon < 2/\lambda_N(\boldsymbol{L})$. The matrix $\boldsymbol{W}[k]$ in (2) was already used as the iteration matrix of consensus algorithms over random graphs, see e.g. [19]-[20]. The eigenvalues of the expected Laplacian matrix $\bar{\boldsymbol{L}}$ are directly related to those of the expected consensus matrix $\bar{\boldsymbol{W}}$ in (2) through the relation

$$\lambda_i(\bar{\boldsymbol{L}}) = (1 - \lambda_{N+1-i}(\bar{\boldsymbol{W}}))/\varepsilon \tag{3}$$

and, in particular, the expected algebraic connectivity is given by $\lambda_2(\bar{\boldsymbol{L}}) = (1 - \lambda_{N-1}(\bar{\boldsymbol{W}}))/\varepsilon$. We consider the following assumption on the stochastic matrices describing the network graph:

Assumption A.1 : Every instance of the random matrix $\boldsymbol{W}[k]$ in (2) is doubly stochastic, i.e.,

$$\boldsymbol{1}^T\boldsymbol{W}[k] = \boldsymbol{1}^T, \quad \text{and} \quad \boldsymbol{W}[k]\boldsymbol{1} = \boldsymbol{1}, \quad \text{for all } k. \tag{4}$$

Under Assumption A.1, the eigenvector $\boldsymbol{v}_N[k]$ associated to the largest eigenvalue of the matrix $\boldsymbol{W}[k]$ is equal to $\boldsymbol{1}/\sqrt{N}$, for all k. Exploiting Assumption A.1, we deflate the original matrix $\boldsymbol{W}[k]$ at time k, obtaining the matrix $\boldsymbol{B}[k]$ given by:

$$\begin{aligned}\boldsymbol{B}[k] &= \boldsymbol{W}[k] - \boldsymbol{v}_N[k]\boldsymbol{v}_N[k]^T = \boldsymbol{W}[k] - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T \\ &= \bar{\boldsymbol{W}} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T + \tilde{\boldsymbol{W}}[k] = \bar{\boldsymbol{B}} + \tilde{\boldsymbol{B}}[k]\end{aligned} \tag{5}$$

where $\bar{\boldsymbol{B}} = \bar{\boldsymbol{W}} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$ and $\tilde{\boldsymbol{B}}[k] = \tilde{\boldsymbol{W}}[k] = -\varepsilon\tilde{\boldsymbol{L}}[k]$. In this way, the maximum eigenvalue of the deflated matrix $\bar{\boldsymbol{B}}$ coincides with the second largest eigenvalue of $\bar{\boldsymbol{W}}$.
The main steps of the algorithm are listed in the following.

Stochastic power iteration method

Initialize $\boldsymbol{x}[0]$, $y[0]$, and $z[0]$ randomly. Then, perform the following steps for $k \geq 0$:

1. Build the deflated matrix $\boldsymbol{B}[k] = \boldsymbol{W}[k] - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$;
2. Evaluate the estimate $y[k+1]$ of $\lambda_{N-1}(\bar{\boldsymbol{W}})$ at time $k+1$ as:
$$\bar{y}[k] = \frac{\boldsymbol{x}^T[k]\boldsymbol{B}[k]\boldsymbol{x}[k]}{\boldsymbol{x}^T[k]\boldsymbol{x}[k]} \tag{6}$$
$$y[k+1] = y[k] + \alpha[k]\left(\bar{y}[k] - y[k]\right) \tag{7}$$
where $\alpha[k]$ is an iteration dependent step-size satisfying (10);
3. Perform the following power iteration
$$\boldsymbol{x}[k+1] = \frac{\boldsymbol{B}[k]\boldsymbol{x}[k]}{\|\boldsymbol{B}[k]\boldsymbol{x}[k]\|}; \tag{8}$$
4. Compute the estimate $z[k+1]$ of $\lambda_2(\bar{\boldsymbol{L}})$ at time k as:
$$z[k+1] = (1 - y[k+1])/\varepsilon; \tag{9}$$
5. Go to step 1 and repeat until convergence.

The stochastic power iteration method in (6)-(8) computes an estimate for the largest eigenvalue of the expected matrix $\bar{\boldsymbol{B}}$, which is directly related to the second eigenvalue $\lambda_2(\bar{\boldsymbol{L}})$ of the expected Laplacian through (9). To obtain convergence of the stochastic power iteration method, we also consider the following assumption:

Assumption A.2 : (*Persistence*) The step-size sequence $\alpha[k]$ in (7) satisfies the conditions:

$$\alpha[k] > 0, \quad \sum_{k=0}^{\infty}\alpha[k] = \infty, \quad \sum_{k=0}^{\infty}\alpha^2[k] < \infty. \tag{10}$$

Conditions (10) are standard in stochastic approximation and adaptive signal processing [21]-[22]; the effect of the step-size in (10) is to drive to zero the variance of the additive disturbance due to the presence of link failures. Then, the convergence of the iterative procedure is determined only by the expected graph of the network. We are now able to state the main theorem on the convergence of the stochastic power iteration method.

Theorem : Let $z[k]$ the sequence generated in (9) by the stochastic power iteration algorithm. If $\lambda_2(\bar{\boldsymbol{L}}) > 0$, under Assumptions A.1 and A.2, we have

$$\lim_{k\to\infty} z[k] = \lambda_2(\bar{\boldsymbol{L}}), \quad \text{almost surely (w.p.1).} \tag{11}$$

Proof. The proof is omitted due to lack of space. A detailed convergence analysis can be found in [23]. ■

Decentralized implementation : The stochastic power iteration method has been described up to now in a centralized fashion. In the following, we propose a method based on average consensus [15]-[16] to decentralize the computation. The two operations that the nodes have to parallelize are (6) and (8), whereas all the other computations can be done locally. Let $\boldsymbol{b}[k] = \boldsymbol{B}[k]\boldsymbol{x}[k]$. The i-th component of the vector $\boldsymbol{b}[k]$ can be evaluated locally. Indeed, exploiting the structure of the deflated matrix in (5), we have

$$b_i[k] = x_i[k] + \varepsilon \sum_{j=1}^{N} a_{ij}[k](x_j[k] - x_i[k]) - m[k] \tag{12}$$

where $m[k] = \dfrac{1}{N}\boldsymbol{1}^T\boldsymbol{x}[k]$ is a global parameter. Indeed, the value $m[k]$ is given by the average of the values $x_i[k]$ stored locally at each node, which can be computed in a decentralized fashion using an average consensus step [15]-[16].

The next step is to evaluate in a distributed fashion the ratio in (6). In particular, we notice that expression (6) can be rewritten as

$$\frac{\boldsymbol{x}^T[k]\boldsymbol{B}[k]\boldsymbol{x}[k]}{\boldsymbol{x}^T[k]\boldsymbol{x}[k]} = \frac{\frac{1}{N}\sum_{i=1}^{N} x_i[k]b_i[k]}{\frac{1}{N}\sum_{i=1}^{N} x_i^2[k]}, \tag{13}$$

where both numerator and denominator are written as inner products. This notation is convenient because it enables us to compute these expressions through a weighted average consensus step [15]-[16], which evaluates in a distributed manner the ratio in (13). Thus, at this stage, each node is able to compute (6) and (7) locally.

To complete the series of operations of the stochastic power iteration algorithm, we need still to evaluate (8) in a distributed fashion. Expression (8) can be locally evaluated by node i, $\forall i$, as:

$$x_i[k+1] = \frac{b_i[k]}{\|\boldsymbol{b}[k]\|}. \tag{14}$$

Since the numerator has been already computed through (12), we need only to compute the denominator of (14). In particular, we consider the evaluation of $\|\boldsymbol{b}[k]\|_N = \sqrt{\sum_{i=1}^{N} b_i^2[k]/N}$, which is a scaled version of $\|\boldsymbol{b}[k]\|$, and can be computed in a distributed fashion by taking the square root of the output of an average consensus step [15]-[16]. Each node then computes $\hat{x}_i[k+1] = \dfrac{b_i[k]}{\|\boldsymbol{b}[k]\|_N} = \sqrt{N}x_i[k]$, which is a scaled version of the true value $x_i[k]$ that the algorithm should compute in (14). However, even in the presence of such update, the method still works correctly because, at time $k+1$, the step in (6) is a Rayleigh ratio, whose result is not affected by the scaling $\sqrt{N}$, thus leading to the correct update of the algorithm.

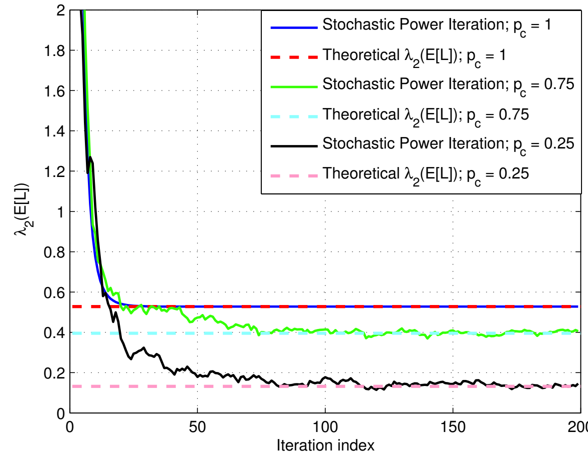

Fig. 1: Estimate of $\lambda_2(\bar{\boldsymbol{L}})$ versus iteration index.

Numerical example - Estimation of $\lambda_2(\bar{\boldsymbol{L}})$: The aim of this example is to test the estimation capabilities of the stochastic power iteration method. We consider a connected network composed of 20 nodes, where the communication among nodes is impaired by random link failures so that each link is on with a probability p_c, i.e., the probability that a packet is exchanged correctly over a communication link. In Fig. 1, we report the behavior of the estimate of $\lambda_2(\bar{\boldsymbol{L}})$ in (9) versus the iteration index, considering different values of the probability p_c. The ideal case, which corresponds to a probability $p_c = 1$, is reported as a benchmark. The theoretical values of $\lambda_2(\bar{\boldsymbol{L}})$ are also reported as dashed lines. The step-size sequence is chosen so that $\alpha[k] = \alpha_0/(k^\tau)$, where $\alpha_0 = 1$ and $\tau = 0.55$, in order to satisfy (10). As we can notice from Fig. 1, the algorithm converges to the theoretical values of $\lambda_2(\bar{\boldsymbol{L}})$ of the corresponding expected graph. As expected, reducing the probability to establish a link, the algorithm needs more time to reach the final convergence value.

4. CONTROL OF EXPECTED CONNECTIVITY

The estimation of $\lambda_2(\bar{\boldsymbol{L}})$ carried out by the stochastic power iteration method can be used to adapt the power transmitted by each node, in order to drive the network connectivity toward a desired value. This can be obtained through a power control step, where each node update its transmission power as

$$p_i[k+1] = p_i[k] + \beta\left(\lambda^* - z[k+1]\right) \tag{15}$$

for $k \geq 0$, where λ^* is a positive constant used to enforce a desired connectivity value, β is a positive step-size, and $z[k+1]$ is the estimate of $\lambda_2(\bar{\boldsymbol{L}})$, at time $k+1$, carried out by the stochastic power iteration method in (9). Then, at each time $k+1$, the covering radius of each node is updated according to

$$r_i[k+1] = \sqrt[\gamma]{\frac{p_i[k+1]}{P_{th}}} \tag{16}$$

where γ is the path-loss exponent, which can assume values from 2 to 6 according to the considered propagation environment, and P_{th} is the minimum received power needed to establish a communication link among two nodes. The updates in (15)-(16) can be inserted as fifth and sixth steps of the stochastic power method in (6)-(9),

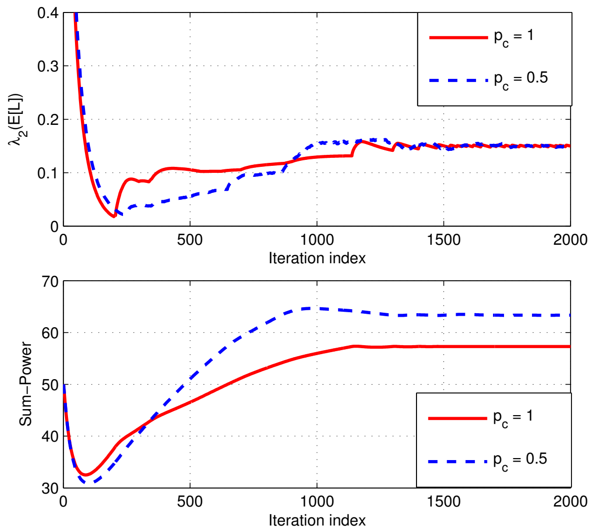

Fig. 2: (Top) Behavior of $\lambda_2(\bar{\boldsymbol{L}})$ versus iteration index. (Bottom) Temporal behavior of the sum of powers transmitted by nodes.

thus leading to a dynamic change of the network topology toward a desired connectivity value.

Numerical example - Control of the expected connectivity : In this example we combine the stochastic power iteration step in (6)-(9) with the power control step in (15)-(16), thus illustrating the capability of the resulting strategy to control the expected connectivity of the network graph. As a starting point, we consider a network composed of 50 nodes deployed over a geographic area of 2500 m^2 according to the topology (1) shown in Fig. 3. The initial power transmitted by each node is $p_i[0] = 1$ Watt, whereas the threshold power needed to establish a communication link among two nodes is $P_{th} = 0.01$ Watt. We assume a free-space path loss as a propagation environment, i.e. $\gamma = 2$. The network topology (1) in Fig. 3 has a value of algebraic connectivity $\lambda_2(\boldsymbol{L}) = 0.105$. Our goal is to use the proposed algorithm to drive the expected connectivity of the graph (1) in Fig. 3 toward a desired value λ^* equal to 0.15, considering two different values of probability to establish a link, e.g., $p_c = 1$, and $p_c = 0.5$. In the top part of Fig. 2, we report the behavior of the estimate of $\lambda_2(\bar{\boldsymbol{L}})$ in (9) versus the iteration index, whereas, in the bottom part, we illustrate the temporal behavior of the sum of the powers transmitted by the network nodes. The continuous curves illustrate the case $p_c = 1$, whereas the dashed curves the case $p_c = 0.5$. The step size $\alpha[k]$ has been chosen as before, and $\beta = 0.1$. As we can notice from Fig. 2 (top), the value of the expected algebraic connectivity of the graph converges close to the desired value λ^* for both values of p_c. The correspondent network topologies are shown in Fig. 3, (2) and (3), respectively for $p_c = 1$ and $p_c = 0.5$. As we can notice from Fig. 3, (2) and (3), the network graph obtained in the case of a lower probability p_c is much more connected. This happens because, reducing the probability to establish a communication link with respect to the ideal case, the algorithm will increase the number of links of the resulting network in order to reach the target value of expected connectivity. Consequently, each node will transmit more power to enlarge its own subset of neighbors. This behavior can be noticed from Fig. 2 (bottom), where we can see how the sum of the powers transmitted by

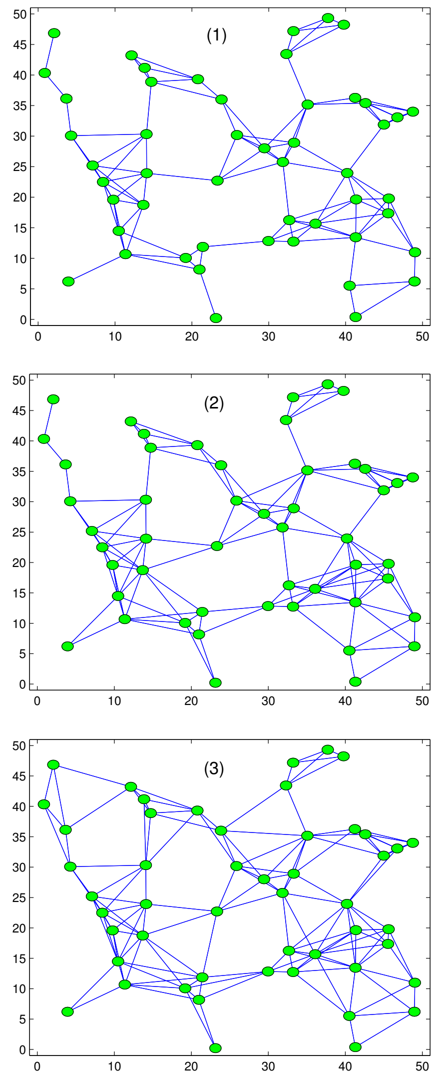

Fig. 3: Examples of network graphs.

the network nodes converge to a fixed value, which depends on the probability to establish a communication link.

5. CONCLUSIONS

In this paper we have proposed a decentralized stochastic power iteration algorithm aimed at estimating the algebraic connectivity of an ad hoc network in case the communications among secondary users are affected by random link failures. The algorithm converges almost surely to the second smaller eigenvalue of the expected Laplacian of the graph. The estimation is then robust against the graph randomness, whose effect is only to slow down the convergence process. Finally, the proposed method is coupled with a power control mechanism that drives the network connectivity toward a desired value. Numerical simulations show the main features of the algorithm in the presence of random link failures.

6. REFERENCES

[1] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," Systems and Control Letters, vol. 53, no. 1, pp. 65-78, 2004.

[2] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, pp. 1035-1048, March 2010.

[3] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Performance analysis of the consensus-based distributed LMS algorithm," *EURASIP Journ. on Adv. in Signal Processing*, vol. 57, no. 6, pp. 2365-2382, June 2009.

[4] F. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[5] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 98, pp. 298–305, 1973.

[6] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. ACM symposium on Theory of computing*, New York, NY, USA, 2004, pp. 561-568.

[7] M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu, "Decentralized laplacian eigenvalues estimation for networked multi-agent systems," in *Proc. IEEE Conf. on Decision and Control*, dec. 2009, pp. 2717-2722.

[8] Alexander Bertrand and Marc Moonen, "Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks," *Internal report KU Leuven ESAT-SCD*, 2012.

[9] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, R. Sukthankar, "Decentralized Estimation and Control of Graph Connectivity in Mobile Sensor Networks," *Proc. American Control Conference*, pp. 2678-2683, 11-13 June 2008, Evanston, IL.

[10] R. Aragues, G. Shi, D. V. Dimarogonas, C. Sagues, and K. H. Johansson, "Distributed algebraic connectivity estimation for adaptive event-triggered consensus," in Proc. American Control Conference, Fairmont Queen Elizabeth, Montreal, Canada, June 2012, pp. 32-37.

[11] M. Bojan, "Laplace eigenvalues of graphs - a survey," Discrete Mathematics, vol. 109, no. 13, pp. 171-183, 1992.

[12] L. Hagen and A. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074-1085, sep 1992.

[13] A. Ghosh and S. Boyd, "Growing well-connected graphs," in *IEEE Conference on Decision and Control*, dec. 2006, pp. 6605-6611.

[14] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol 49, no. 9, pp. 1465–1476, Sept. 2004.

[15] R. Olfati-Saber, J. A. Fax, R. M. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proc. of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[16] S. Barbarossa and G. Scutari, "Bio-inspired sensor network design: distributed decision through self-synchronization," *IEEE Signal Processing Magazine*, Vol. 24, no. 3, pp. 26-35, May 2007.

[17] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.

[18] S. Chatterjee and E. Seneta, "Towards consensus: Some convergence theorems on repeated averaging," *Journal of Applied Probability*, vol. 14, no. 1, pp. 89–97, Mar. 1977.

[19] A. Tahbaz Salehi and A. Jadbabaie, "On consensus over random networks," *Proc. of 44th Allerton Conference*, UIUC, Illinois, USA, Sept. 27–29, 2006.

[20] A. Tahbaz Salehi and A. Jadbabaie, "Consensus over ergodic stationary graph processes," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, Jan. 2010.

[21] A. H. Sayed, *Adaptive Filters*, Wiley, NJ, 2008.

[22] M. Nevelson and R. Hasminskii, "Stochastic approximation and recursive estimation," Providence, Rhode Island: American Mathematical Society, 1973.

[23] P. Di Lorenzo and Sergio Barbarossa, " Decentralized estimation and control of expected algebraic connectivity of random graphs," to be submitted to *IEEE Trans. on Signal Processing*.