

# Distributed Consensus in Wireless Networks With Probabilistic Broadcast Scheduling

Daniel Pérez Herrera, Zheng Chen and Erik G Larsson

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-192531>

N.B.: When citing this work, cite the original publication.

Pérez Herrera, D., Chen, Z., Larsson, E. G. (2023), Distributed Consensus in Wireless Networks With Probabilistic Broadcast Scheduling, *IEEE Signal Processing Letters*, 30, 41-45.  
<https://doi.org/10.1109/LSP.2023.3240647>

Original publication available at:

<https://doi.org/10.1109/LSP.2023.3240647>

©2023 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

<https://www.ieee.org/>

# Distributed Consensus in Wireless Networks with Probabilistic Broadcast Scheduling

Daniel Pérez Herrera, Zheng Chen and Erik G. Larsson

**Abstract**—We consider distributed average consensus in a wireless network with partial communication to reduce the number of transmissions in every iteration/round. Considering the broadcast nature of wireless channels, we propose a probabilistic approach that schedules a subset of nodes for broadcasting information to their neighbors in every round. We compare several heuristic methods for assigning the node broadcast probabilities under a fixed number of transmissions per round. Furthermore, we introduce a pre-compensation method to correct the bias between the consensus value and the average of the initial values, and suggest possible extensions for our design. Our results are particularly relevant for developing communication-efficient consensus protocols in a wireless environment with limited frequency/time resources.

**Index Terms**—Average consensus, broadcast transmission, scheduling, wireless networks.

## I. INTRODUCTION

Distributed consensus deals with the problem of reaching an agreement among networked agents on a certain quantity of interest that depends on the initial states of all agents. The interaction rule that specifies the information exchange between an agent (node) and its neighbors is known as consensus protocol or algorithm [1]. Traditional studies of consensus problems usually assume a fixed topology with perfect communication between the nodes in the network. For this ideal case, the convergence conditions and convergence speed of discrete-time average consensus are provided in [2] and [3]. As an extension to the main theme, different methods can be used to accelerate convergence, such as extrapolation [4] and stepsize adjustment [5]. For varying topologies, *necessary and sufficient* conditions to reach consensus over random networks are presented in [6], using ergodicity and probabilistic arguments.

Conventional works on consensus algorithms tend to rely on idealized assumptions (e.g., instantaneous, error-free) on the communication links between connected nodes. Consensus algorithms may suffer from imperfect communication between the nodes due to, for example, link failures [7], [8], [9]. To deal with the missing information, two compensation (biased and balanced) methods are presented in [10]. The main difference between these methods is how the weights of the failed links are re-distributed among the successful links. Both algorithms converge almost surely and in the mean-square sense to a value that generally is not equal to the average of the initial values.

In addition to the conventional consensus protocols where every node communicates with all their neighbors in every iteration, gossip algorithms can also be used to achieve consensus with randomized and asynchronous communication [11], [12]. A key feature of these algorithms is the asynchronous time model, where information is exchanged between a subset of nodes. This information is processed by the receiving nodes to compute local update via a linear combination of its own value and the received one. There are different variants of gossip algorithms, such as geographic gossip [13], randomized gossip [14] and broadcast gossip [15]. Particularly, broadcast gossip is more relevant to the implementation of gossip algorithms in wireless networks, where the transmission of one node can reach any other nodes within its communication range.

Unlike the case with broadcast gossip, where the nodes update their states immediately after receiving information from another node, we propose a probabilistic broadcast-based method to schedule a subset of nodes for broadcasting in every communication round. The nodes update their values after receiving information from all the scheduled broadcasting nodes. Since only a subset of nodes is selected for transmission in every round, the number of transmission slots needed in every round is reduced as compared to the full communication case. To compensate for the lack of information with our partial communication approach, we use a biased compensation method similar to the one proposed in [10]. Our objective is to achieve fast convergence with fewer transmission slots while maintaining small deviation from the average of the initial values. We investigate several heuristic methods for assigning the broadcast probabilities of the nodes, combined with a pre-compensation mechanism to eliminate the bias caused by unbalanced information exchange.

## II. SYSTEM MODEL

### A. Graph Model

We consider a wireless network modeled by an undirected and connected graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ , where  $\mathcal{N}$  is the set of nodes and  $\mathcal{E}$  is the set of edges. The total number of nodes is  $N$ . Each edge  $\{i, j\} \in \mathcal{E}$  indicates the existence of a link between the pair of nodes  $i$  and  $j$ . The set of neighbors of node  $i$  is  $\mathcal{N}_i = \{j | \{i, j\} \in \mathcal{E}\}$ .<sup>1</sup>

The connectivity of the graph can be represented by its adjacency matrix  $\mathbf{A}$ , with elements  $a_{ij} = 1$  if  $\{i, j\} \in \mathcal{E}$ ,

The authors are with the Department of Electrical Engineering, Linköping University, Linköping, SE-58183 Sweden. E-mail: {daniel.perez.herrera, zheng.chen, erik.g.larsson}@liu.se. This work was supported by Zenith, ELLIIT, and the KAW foundation.

<sup>1</sup>In a wireless network, the notion of connectivity is not strictly defined since every node can hear the transmission of other nodes in the network. We consider a simplified model where a link between two nodes does not exist when the distance is larger than some predefined threshold.

and  $a_{ij} = 0$  otherwise. Another matrix representation of the graph is the Laplacian matrix, defined as  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , where  $\mathbf{D}$  is a diagonal matrix whose  $i$ -th entry is  $d_i = |\mathcal{N}_i|$ .

### B. Consensus Algorithm

We consider the problem of distributed average consensus, where the goal is for all nodes in the network to reach consensus on the arithmetic mean of their initial values, by only communicating with their neighbors. To accomplish this, we use the following distributed linear iteration

$$\mathbf{x}(t+1) = \mathbf{W}\mathbf{x}(t), \quad (1)$$

where  $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_N(t)]^T$  contains the values of every node in iteration  $t \geq 0$ .  $\mathbf{W}$  is the mixing matrix, defined as  $\mathbf{W} = \mathbf{I} - \epsilon\mathbf{L}$ , where  $\mathbf{I}$  is the identity matrix and  $\epsilon \in \mathbb{R}$  is the step size. Let  $\Delta(\mathcal{G})$  be the maximum degree of the graph  $\mathcal{G}$ . When  $0 < \epsilon < 1/\Delta(\mathcal{G})$ , the following are necessary and sufficient conditions for  $\mathbf{x}(t)$  to converge to the average of the initial values [2]:

- 1)  $\mathbf{W}\mathbf{u} = \mathbf{u}$ ,
- 2)  $\mathbf{u}^T\mathbf{W} = \mathbf{u}^T$ ,
- 3)  $\rho(\mathbf{W} - \mathbf{u}\mathbf{u}^T/N) < 1$ ,

where  $\rho(\cdot)$  denotes the spectral radius of a matrix and  $\mathbf{u}$  the all 1's column vector. Note that since  $\mathbf{W}$  is symmetric, conditions 1 and 2 are equivalent. Condition 3 is also related to the convergence speed, since minimizing the second largest eigenvalue of  $\mathbf{W}$  can speed up convergence of distributed averaging algorithms [2].

### III. PARTIAL COMMUNICATION WITH PROBABILISTIC BROADCAST SCHEDULING

Distributed consensus algorithms rely on an iterative process where in each iteration, every node must communicate with all its neighbors before the updating step in (1). The communication design of consensus algorithms in wireless networks is non-trivial due to the ‘‘many-to-many’’ communication topology. If all nodes transmit information simultaneously using the same frequency-time resources, then all signals will be garbled. One simple way to avoid interference between concurrent transmissions is to assign orthogonal resources to each transmitting node. For instance, each node occupies one dedicated time slot for broadcasting its information to the neighbors. In this way, it takes  $N$  times slots in total for the entire network to complete information fusion in one consensus iteration with full communication.

We refer to one *communication round* as one iteration interval during which all nodes exchange information with the neighbors and update their state values, and one *transmission slot* as the time interval during which one node broadcasts its value. With full communication over the network, after  $J$  iterations of the consensus algorithm, the total number of required transmission slots is  $NJ$ .

To reduce the communication costs (time slots), we consider a partial communication design where in every round we only select a subset of nodes for broadcasting their information. Let  $\mathbf{p} = [p_1, p_2, \dots, p_N]^T$  be a vector that contains the broadcast probabilities of the nodes with  $\sum_{i=1}^N p_i = K$  and  $K < N$ .

This means that in every communication round,  $K$  nodes are selected on average for broadcasting. Therefore, the average number of transmission slots for completing  $J$  iterations is  $KJ < NJ$ .

Since in every iteration, some nodes are not scheduled for broadcasting their values, we compensate the missing information by using the biased compensation method proposed in [10]. The idea is to use a new mixing matrix  $\overline{\mathbf{W}}(t)$  in every round  $t$ , whose elements are given by

$$\overline{w}_{ij}(t) = \begin{cases} w_{ij}v_j(t), & \text{if } j \neq i. \\ 1 - \sum_{k=1, k \neq i}^N w_{ik}v_k(t), & \text{if } j = i. \end{cases} \quad (2)$$

Here,  $v_j(t) = 1$  if node  $j$  is selected for broadcasting in the  $t$ -th communication round, and  $v_j(t) = 0$  otherwise. With our probabilistic scheduling approach, we have  $p_i = \mathbb{E}[v_i(t)]$ . Though the biased compensation method ensures convergence in mean square, in general the consensus value is not equal to the average of the initial values, since the new mixing matrix is only row stochastic but not necessarily doubly stochastic.

#### A. Heuristic Designs for Broadcast Probability Vector

Intuitively, the broadcast probabilities should be determined in a way that can reflect the ‘‘importance’’ of the nodes in a network. As a first approach, we consider different methods of creating the broadcast probability vector based on commonly used metrics for network connectivity, such as degree, PageRank and betweenness centrality [16].

1) *Degree-based method*: Degree centrality of a node is the number of neighbors of the node: the degree of node  $j$  is  $d_j = |\mathcal{N}_j|$ . We can choose the broadcast probability of node  $i$  as

$$p_i = \min\{1, \gamma d_i\} \quad (3)$$

where the constant  $\gamma$  is chosen such that  $\sum_{j=1}^N p_j = K$ , for a given value of  $K$ .

2) *PageRank-based method*: PageRank was first proposed to rank webpages using the hyperlink network structure of the web. It can be used in any network as a measure of the node importance. The PageRank-based method gives us a probability vector  $\mathbf{p}$  in the same way, but using the PageRank centrality of the nodes instead of their degrees.

3) *Betweenness-based method*: Betweenness centrality of a node is defined as the fraction of shortest paths between pairs of other nodes that pass through it. Let  $b_i$  represent the betweenness centrality of node  $i$ . Since  $b_i$  is zero if node  $i$  has degree one, a small positive value is added to this metric, i.e., the new score  $\tilde{b}_i$  of node  $i$  is:

$$\tilde{b}_i = b_i + \beta, \quad (4)$$

where  $\beta$  is a small positive number. The probability vector  $\mathbf{p}$  is then created in the same way as for the degree-based method, but using this new score instead of the node degree.

## B. Bias Correction

With our probabilistic scheduling of broadcasting nodes in every round, the matrix  $\overline{\mathbf{W}}(t)$  is row stochastic with positive diagonals (as long as  $p_i > 0, \forall i$ ). As a consequence of Theorem 2.1 in [10], the product  $\prod_{t=0}^J \overline{\mathbf{W}}(t)$  converges almost surely, when  $J \rightarrow \infty$ , to a random rank-one matrix,  $\mathbf{u}\boldsymbol{\alpha}^T$ , for some vector  $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ . In general, a bias will be introduced, meaning that the consensus value will not be the average of the initial values. There are several existing bias correction approaches in the literature of distributed consensus, such as the corrective consensus in [17], and the re-scaling method in [18]. The method proposed in [18] is not directly applicable in our system because of the time-varying communication topology. However, it is possible to correct the bias by using a pre-compensation method as follows.

For a given topology  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ , a set of initial values  $\{x_i(0)\}_{i=1}^N$  and a scheduling probability vector  $\mathbf{p}$ :

- 1) Run  $N$  consensus processes with probabilistic scheduling of broadcast transmissions. In the  $i$ -th consensus process, the initial value of every node is zero, except for node  $i$ , whose initial value is one. In this case all nodes will converge to  $\alpha_i$ , since  $\prod_{t=0}^{\infty} \overline{\mathbf{W}}(t)\mathbf{e}_i = \mathbf{u}\boldsymbol{\alpha}^T\mathbf{e}_i = \alpha_i\mathbf{u}$ , where  $\mathbf{e}_i$  is the  $i$ -th column of the  $N \times N$  identity matrix.
- 2) Pre-compensate the initial values: for node  $i$ , the new initial value becomes  $\tilde{x}_i(0) = \frac{x_i(0)}{\alpha_i N}, i = 1, \dots, N$ .
- 3) Run another consensus algorithm with initial node values  $\{\tilde{x}_1(0), \tilde{x}_2(0), \dots, \tilde{x}_N(0)\}$ . This consensus process will converge to  $\frac{1}{N} \sum_{i=1}^N x_i(0)$ .

Importantly, the same seed must be used for the generation of (pseudo-)random numbers in the consensus mechanisms in Steps 1 and 3. This can be achieved, for example, by using a source of shared randomness. Note that the pre-compensation weights computation (Step 1) before the actual consensus process (Step 3) requires extra communication and computation. Specifically, it will consume  $JKN$  transmission slots, where  $J$  is the number of iterations in Step 1. However, this extra cost is incurred only once, since the pre-compensation weights can be reused for any number of future consensus processes, as long as the network topology and the probability vector remain the same.

## C. Possible Extensions

1) *Numerical Optimization of Broadcast Probabilities:* As mentioned in Sec. II-B, for a fixed graph, we can optimize the second largest eigenvalue of the mixing matrix to achieve fast convergence as shown in [2]. However, following our approach with partial communication,  $\overline{\mathbf{W}}(t)$  is a time-varying random matrix whose elements are determined by the scheduling decision in every communication round. It is generally difficult to analyze the eigenvalues of  $\prod_{t=0}^{\infty} \overline{\mathbf{W}}(t)$ . Therefore, we focus on the expectation of  $\overline{\mathbf{W}}(t)$  and optimize the probability vector  $\mathbf{p}$  by minimizing the second largest eigenvalue of  $\mathbb{E}[\overline{\mathbf{W}}(t)]$ . The elements of  $\mathbb{E}[\overline{\mathbf{W}}(t)]$  are:

$$\mathbb{E}[\overline{w}_{ij}(t)] = \begin{cases} w_{ij}p_j, & \text{if } j \neq i. \\ 1 - \sum_{k=1, k \neq i}^N w_{ik}p_k, & \text{if } j = i. \end{cases} \quad (5)$$

The optimization problem can be defined as:

$$\begin{aligned} \min_{\mathbf{p}} \quad & \rho(\mathbb{E}[\overline{\mathbf{W}}(t)] - \mathbf{u}\mathbf{u}^T/N) \\ \text{s.t.} \quad & \sum_{i=1}^N p_i = K \\ & 0 \leq p_i \leq 1, \quad i = 1, \dots, N \end{aligned} \quad (6)$$

for a given value of  $K$ .

To the best of our knowledge, there is no tractable expression that can characterize the relation between the second largest eigenvalue of  $\mathbb{E}[\overline{\mathbf{W}}(t)]$  and the broadcast probability vector  $\mathbf{p}$ . It is possible to apply some derivative-free optimization method, such as Simultaneous Perturbation Stochastic Approximation (SPSA), for optimizing the broadcasting probabilities [19], [20]. The convergence analysis of SPSA has been established in the literature with different assumptions on the objective function. However, due to the unknown structure of the objective function in (6), the convergence to the optimal solution in this case is not guaranteed. Another challenge of using this method for the eigenvalue optimization problem is that some broadcast probabilities can be zero depending on the network topology and the given  $K$ , which is undesirable. In practice, this can be handled by imposing a minimum value of the probabilities.

2) *Alternative Bias Correction Methods:* Another possible bias correction method is to apply the corrective consensus mechanism proposed in [17]. In that mechanism, a new set of auxiliary variables  $\phi_{ij}(t)$  is introduced to represent the amount of change that node  $i$  has made to its state value  $x_i(t)$ , due to the information received from its neighbor  $j$  at iteration  $t$ . Periodically, one corrective iteration takes place, where every node  $i$  transmits  $\phi_{ij}(t)$  to the corresponding neighbor  $j$  and computes  $\Delta_{ij}(t) = \phi_{ij}(t) + \phi_{ji}(t)$ . This new set of variables  $\Delta_{ij}$  represents the bias accumulated in both directions and it is used to correct the values of  $x_i$  and  $\phi_{ij}$ . Note that this bias correction method introduces extra communication overhead. In every corrective iteration,  $\sum_{i=1}^N d_i$  transmission slots are needed for the exchange of  $\phi_{ij}$ . Another issue with this method is that convergence is not guaranteed, even for a complete graph. (The convergence criteria given in [17] assume retransmissions).

## IV. SIMULATION RESULTS

For simulations, we generate one instance of an Erdős-Rényi random graph with  $N = 100$  nodes, that is undirected and connected. The initial values of the nodes are created by using a normal distribution with zero mean and unit variance. Every plot is obtained by averaging 10 realizations with the same graph but different initial values. The mixing matrix  $\mathbf{W} = \mathbf{I} - \epsilon\mathbf{L}$  is computed for  $\epsilon = \frac{1}{\Delta(\mathcal{G})+1}$ .

### A. Broadcast Probability Vector Design

First, we show the performance of our partial communication design with different heuristic choices for the broadcast probabilities. We choose  $K = 80$ , which corresponds to 80% of the nodes scheduled for broadcasting their values in every communication round. The results are presented in Fig. 1. We

observe a clear gain in terms of convergence speed (measured in transmission slots) with our partial communication design, especially with betweenness-based method. However, the converged value is not equal to the average of the initial values, which implies a tradeoff between the convergence speed and the bias in the consensus value.

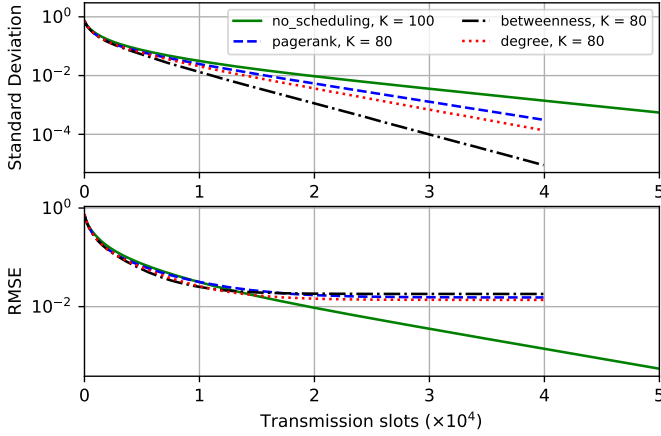


Fig. 1. Comparison of the standard deviation of the nodes values and its distance from true average (RMSE) for different heuristic methods.

### B. Partial Communication with Bias Correction

To deal with the bias introduced by our partial communication design, we implement the pre-compensation mechanism introduced in Sec. III-B. The results are presented in Fig. 2. We can see that by introducing a pre-compensation step before the actual consensus process, we can eliminate the bias in the consensus value while maintaining the advantage of our proposed design in reducing the communication costs.

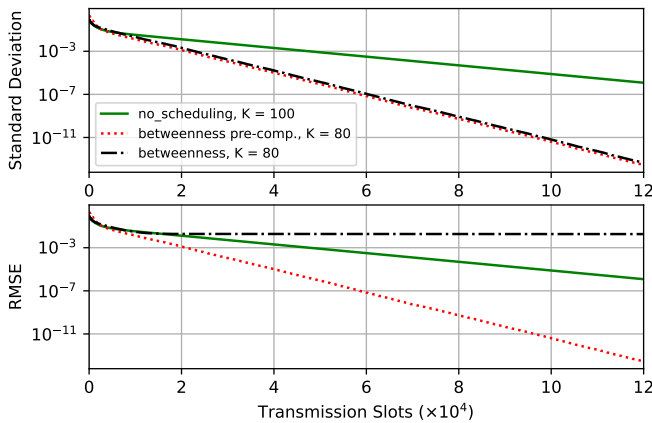


Fig. 2. Performance with the proposed bias correction mechanism, using the betweenness-based probability vector design.

### C. Potential Improvement by Using SPSA

Finally, we show the performance of our partial communication design with the scheduling probabilities obtained with SPSA. In Fig. 3, we compare the SPSA-based method and the

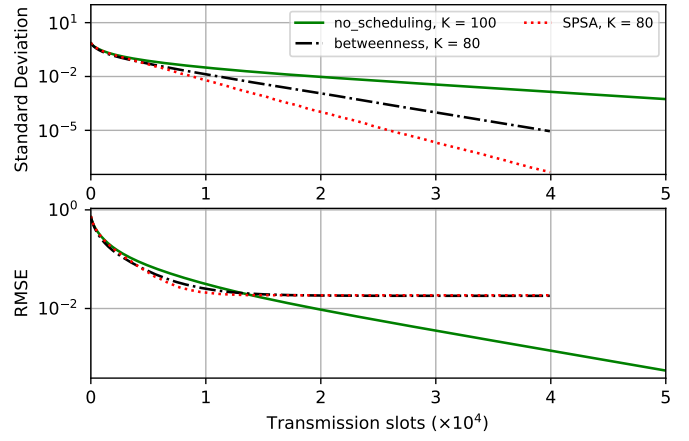


Fig. 3. Comparison of the standard deviation of the nodes values and the RMSE for no scheduling, betweenness-based and SPSA-based methods.

heuristic design of the probability vector using betweenness centrality, as detailed in Sec. IV-A.

We observe that with the SPSA-based method, we can achieve faster convergence as compared to the heuristic design, while keeping similar RMSE. This result is well supported by the fact that reducing the second largest eigenvalue of the expected mixing matrix improves convergence speed.

## V. CONCLUSION

We proposed a partial communication design for distributed average consensus over wireless networks using probabilistic broadcast scheduling. A trade-off between convergence speed in terms of transmission slots and the bias in the consensus value was observed from simulations. Several heuristic methods for assigning the node broadcast probabilities were proposed, as well as a pre-compensation mechanism for bias correction. We concluded that distributed consensus algorithms in wireless networks can benefit from partial communication in achieving consensus with reduced communication costs. As future work, an optimal selection of the number of broadcasting nodes per round could be investigated. The possibility of adaptively changing  $\epsilon$  could also be considered.

## REFERENCES

- [1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
- [2] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.
- [3] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 33–55, 2009.
- [4] E. Kokiopoulou and P. Frossard, "Accelerating distributed consensus using extrapolation," *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 665–668, Oct. 2007.
- [5] C. Mosquera, R. Lopez-Valcarce, and S. K. Jayaweera, "Stepsize sequence design for distributed average consensus," *IEEE Signal Process. Lett.*, vol. 17, no. 2, pp. 169–172, Feb. 2010.
- [6] A. Tahbaz-Salehi and A. Jadbabaie, "A necessary and sufficient condition for consensus over random networks," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 791–795, Mar. 2008.
- [7] S. Patterson, B. Bamieh, and A. El Abbadi, "Convergence rates of distributed average consensus with stochastic link failures," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 880–892, Apr. 2010.

- [8] S. Kar and J. M. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.
- [9] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 634–649, May 2008.
- [10] —, "Average consensus with packet drop communication," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 102–133, 2009.
- [11] A. G. Dimakis, S. Kar, J. M. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.
- [12] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *Proc. IEEE INFOCOM*, 2005, pp. 1653–1664.
- [13] A. D. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: Efficient averaging for sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1205–1216, 2008.
- [14] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
- [15] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2748–2761, 2009.
- [16] V. Latora, V. Nicosia, and G. Russo, *Complex networks: principles, methods and applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [17] Y. Chen, R. Tron, A. Terzis, and R. Vidal, "Corrective consensus with asymmetric wireless links," *Proc. IEEE 50th Conf. Decis. Control*, 2011, pp. 6660–6665.
- [18] C. Xi, V. S. Mai, R. Xin, E. H. Abed, and U. A. Khan, "Linear convergence in optimization over directed graphs with row-stochastic matrices," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3558–3565, Oct. 2018.
- [19] J. C. Spall, "An overview of the simultaneous perturbation method for efficient optimization," *Johns Hopkins APL Techn. Dig.*, vol. 19, no. 4, pp. 482–492, 1998.
- [20] P. Sadegh, "Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation," *Automatica*, vol. 33, pp. 889–892, 1997.