FINITE-TIME DISTRIBUTED CONSENSUS THROUGH GRAPH FILTERS

Aliaksei Sandryhaila, Soummya Kar, and José M. F. Moura

Electrical and Computer Engineering, Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

We propose a new framework for distributed computation of average consensus. The presented framework leads to a systematic design of iterative algorithms that compute the consensus exactly, are guaranteed to converge in finite time, are computationally efficient, and require no online memory. We demonstrate that our approach is applicable to a broad class of networks. For remaining networks, our framework leads to the construction of approximating algorithms for consensus that are also guaranteed to compute in finite time. Our approach is inspired by graph filters introduced by the theoretical framework of signal processing on graphs.

Index Terms— Consensus, distributed average, graph filters, network.

1. INTRODUCTION

We consider a network of N agents that is described by a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Each node v_n represents the nth agent, and nodes v_n and v_m are linked by a *directed* edge $(v_n, v_m) \in \mathcal{E}$ if the agent v_n communicates to the agent v_m. Indices of nodes that communicate to the nth agent form a set $\Omega_n = \{m \mid (v_m, v_n) \in \mathcal{E}\}$ called the *neighborhood* of v_n.

Each agent v_n at time t holds a scalar value $x_n(t) \in \mathbb{C}$. Agents communicate with their neighbors and update their values through distributed linear iterations of the form

$$x_n(t+1) = w_{nn}(t) x_n(t) + \sum_{m \in \Omega_n} w_{nm}(t) x_m(t), \qquad (1)$$

where, in general, the weights $w_{nm}(t) \in \mathbb{C}$ are complex-valued scalars that change with time t.

The *distributed average consensus* problem refers to the situation when all agents seek to compute the average of all initial values

$$y_n = \frac{1}{N} \left(x_1(0) + x_2(0) + \ldots + x_N(0) \right) \qquad (2)$$

in a distributed manner, that is, through iterative communication of the form (1). Key questions that need to be answered for this problem are: a) what conditions should be imposed on the weights $w_{nm}(t)$ to compute the average consensus (2) exactly or approximately, and b) how the corresponding computational algorithm can be constructed.

Distributed computation of the average consensus was originally formulated in [1] and has been extensively studied in numerous works, including [2, 3, 4, 5, 6, 7, 8, 9, 10] and others. Multiple implementations of the distributed average consensus have been proposed in the literature. They differ in preprocessing, online memory, and computational requirements, as well as the nature of convergence (finite-time vs. asymptotic convergence). These metrics lead to various design trade-offs; for instance, it is expected that algorithms with asymptotic convergence (see [4, 6] and references therein) are less memory and computationally intensive than algorithms with finite-time convergence, such as [7, 8].

Contribution. In this paper we present a new framework for distributed computation of the average consensus (2) that avoids the trade-off between convergence and efficiency. Our approach leads to a systematic design of algorithms that, on the one hand, are *exact* and guaranteed to converge in *finite time*, and, on the other hand, are computationally efficient and require no online memory. We identify a family of networks that can compute the average consensus exactly and in finite time, determine a lower bound on the number of iterations required for exact computation, and provide a constructive procedure for the corresponding algorithms. We also study the problem of approximate computation of average consensus in finite time and demonstrate that it can be formulated as a semidefinite program.

The presented approach is inspired by a class of operators called *graph filters* that were introduced by the theoretical framework of discrete signal processing on graphs [11, 12] as generalizations of linear time-invariant filters from regular lattices to arbitrary graphs. It offers a general, principled formulation of the distributed computation of linear functions that can be applied to various problems, not only the distributed average consensus. This contrasts our framework with other existing implementations of distributed average consensus, such as [10], that also consider exact and finite-time computation of distributed consensus, but do not generalize straightforwardly to other transforms.

2. COMPUTATION MODEL

In this section, we introduce specific assumptions on the communication model (1) and formalize our computational framework for agent networks.

We assume that weights $w_{nm}(t)$ change from time t to time $t+1$ proportionally to each other, so that the ratio $w_{nm}(t+1)/w_{nm}(t)$ is constant for all $n \neq m$. We formalize this assumption by expressing the weights in the form

$$\begin{cases} w_{nm}(t) = \beta_t w_{nm}, & n \neq m, \\ w_{nn}(t) = \alpha_t + \beta_t w_{nn}, & \end{cases} \qquad (3)$$

where $w_{nm} \in \mathbb{C}$ stand for $w_{nm}(0)$ and $\alpha_t, \beta_t \in \mathbb{C}$ are arbitrary scalar coefficients. We assume that the weights w_{nm} are known and provided as a part of the network topology, while coefficients α_t and β_t are unknown variables. Unlike many existing algorithms for distributed consensus, we do not require the weights $w_{nm}(t)$ to be non-negative.

This work was supported in part by AFOSR grant FA95501210087 and NSF grant ECCS-1306128.

It is reasonable to assume that $\beta_t \neq 0$ for each t in (3), since otherwise the corresponding weights $w_{nm}(t)$ would be equal to zero for all $m \neq n$. Hence, we can rewrite the iteration step (1) using the assumptions (3) as

$$x_n(t+1) = \beta_t \left(\left(\frac{\alpha_t}{\beta_t} + w_{nn} \right) x_n(t) + \sum_{m \in \Omega_n} w_{nm} x_m(t) \right). \tag{4}$$

We define the vector of values

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & \dots & x_N(t) \end{bmatrix}^T$$

and write (4) in the matrix-vector form

$$\begin{aligned} \mathbf{x}(t+1) &= \beta_t \left(\frac{\alpha_t}{\beta_t} \mathbf{I} + \mathbf{W} \right) \mathbf{x}(t) \\ &= \prod_{i=0}^{t} \beta_i \left(\frac{\alpha_i}{\beta_i} \mathbf{I} + \mathbf{W} \right) \mathbf{x}(0), \end{aligned} \tag{5}$$

where $\mathbf{I}$ is a $N \times N$ identity matrix and

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & \dots & w_{1,N} \\ \vdots & \vdots & \vdots \\ w_{N,1} & \dots & w_{N,N} \end{bmatrix} \tag{6}$$

is and $N \times N$ matrix of weights w_{nm}. We then define coefficients

$$g_i = \alpha_i / \beta_i$$

and

$$f_{t+1} = \beta_0 \beta_1 \cdots \beta_t$$

and write the computation model (5) as

$$\mathbf{x}(t+1) = f_{t+1} \prod_{i=0}^{t} (g_i \, \mathbf{I} + \mathbf{W}) \mathbf{x}(0). \tag{7}$$

We describe computation algorithms using the model (7) rather than (4); that is, in terms of coefficients f_{t+1} and g_i rather than α_t and β_t. Since the weight matrix $\mathbf{W}$ is known, an algorithm is completely specified by the corresponding coefficients f_{t+1} and g_i.

In addition, we assume that the weight matrix (6) is symmetric[1], so that weights in (3) satisfy $w_{nm} = w_{mn}$. In other words, the network is undirected. We also assume that the weights w_{nn} for all $1 \leq n \leq N$ satisfy

$$w_{nn} = - \sum_{m \in \Omega_n} w_{nm}. \tag{8}$$

In this paper, we study the computation of the average consensus (2) using algorithms of the form (7). In particular, let us write the average consensus in the matrix-vector form

$$\mathbf{y} = \mathbf{P} \mathbf{x}^{(0)}. \tag{9}$$

Here,

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_N \end{bmatrix}^T$$

is the vector of averages (2). The matrix

$$\begin{aligned} \mathbf{P} &= \frac{1}{N} \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \dots & 1 \end{bmatrix} \\ &= \frac{1}{N} \mathbf{1} \, \mathbf{1}^T \end{aligned} \tag{10}$$

is the average consensus matrix; the vector

$$\mathbf{1} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T \tag{11}$$

denotes a vector of all ones, . The average consensus (9) can be computed *exactly* and in *finite time* by a network of agents if there exist a non-negative finite integer t and coefficients f_t and $g_0, g_1, \dots, g_{t-1}$ that satisfy

$$\mathbf{P} = f_t \prod_{i=0}^{t-1} (g_i \, \mathbf{I} + \mathbf{W}). \tag{12}$$

Observe that by fixing t, one can attempt to find coefficients f_t and $g_0, g_1, \dots, g_{t-1}$ by casting (12) as a minimization problem. However, it is not obvious how to select the value for t and what is the smallest t that leads to an exact solution. Moreover, depending on $\mathbf{W}$, the search for a solution can also be complicated by slow convergence and numerical instability. In Section 3, we provide a principled approach to solving (12).

Connection with graph filters. The right-hand side of (7) can be written as a matrix polynomial

$$h(\mathbf{W}) = h_0 \, \mathbf{I} + h_1 \mathbf{W} + h_2 \mathbf{W}^2 + \dots + h_t \mathbf{W}^t. \tag{13}$$

It is obtained by evaluating the scalar-coefficient polynomial

$$\begin{aligned} h(z) &= f_t \prod_{i=0}^{t-1} (g_i + z) &(14) \\ &= h_0 + h_1 z + h_2 z^2 + \dots + h_t z^t. &(15) \end{aligned}$$

at $z = \mathbf{W}$. By comparing polynomials (14) and (15), we immediately conclude that the coefficient f_t in (7) and coefficient h_t in (15) are equal:

$$f_t = h_t. \tag{16}$$

The matrix polynomial $h(\mathbf{W})$ in (13) has the same form as *graph filters* introduced by the discrete signal processing on graphs [11, 12]. This framework studies the analysis and processing of signals residing on graphs, and graph filters operate by combining weighted averages of neighborhoods of different radii for each node. Hence, the computation model (12) can be seen as an implementation of the distributed average consensus using an appropriately designed graph filter on the graph G that describes the agent network and has the adjacency matrix (6).

3. EXACT COMPUTATION

In this section, we identify a broad class of networks that can compute the average consensus exactly and in finite time. We also determine the minimal number of iterations required for the exact computation of the consensus and provide a construction procedure for the corresponding algorithm.

Since $\mathbf{W}$ is a symmetric matrix, it has a full set of orthonormal eigenvectors [13]. The eigendecomposition of $\mathbf{W}$ is given by

$$\mathbf{W} = \mathbf{V} \, \mathbf{\Lambda} \, \mathbf{V}^T, \tag{17}$$

where the columns of the eigenvector matrix

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots \mathbf{v}_N \end{bmatrix}$$

are the orthonormal eigenvectors of $\mathbf{W}$, and

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 \, \mathbf{I}_{m_1} & & \\ & \ddots & \\ & & \lambda_K \, \mathbf{I}_{m_K} \end{bmatrix} \tag{18}$$

[1] Assuming that $\mathbf{W}$ is Hermitian leads to practically identical results.

is a diagonal matrix of eigenvalues, where each distinct eigenvalue λ_k for $1 \leq k \leq K$ has algebraic multiplicity m_k; the multiplicities add up to

$$m_1 + m_2 + \ldots + m_K = N.$$

Furthermore, it follows from (8) that matrix $\mathbf{W}$ has an eigenvalue 0, which we assign to be the first eigenvalue $\lambda_1 = 0$. Its corresponding eigenvector is

$$\mathbf{v}_1 = \frac{1}{\sqrt{N}} \mathbf{1}\,. \tag{19}$$

As follows from (17) and (19),

$$\mathbf{V}\,\mathbf{v}_1 = \frac{1}{\sqrt{N}} \mathbf{V}^T \mathbf{1} = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T. \tag{20}$$

The following theorem establishes a family of networks that can compute the average consensus exactly and in finite time. The proof is omitted due to space restrictions.

Theorem 1 The distributed average consensus can be computed exactly and in finite time by *any* network with a symmetric communication matrix $\mathbf{W}$ that satisfies the condition (8) and has a simple eigenvalue $\lambda_1 = 0$ (that is, $m_1 = 1$).

The corresponding polynomial $h(z)$ of the form (15) satisfies the set of conditions

$$\begin{cases} h(0) = h_0 = 1, \\ h(\lambda_k) = 0, & 2 \leq k \leq K. \end{cases} \tag{21}$$

Recall that we do not make any assumptions about the weights w_{nm}. If we assume that they are non-negative, Theorem 1 can be formulated as follows.

Corollary 1 The distributed average consensus can be computed exactly and in finite time by any connected network with a symmetric communication matrix $\mathbf{W}$ that satisfies the condition (8) and has non-negative weights $w_{nm} \geq 0$ for $n \neq m$.

Proof: Observe that matrix $\mathbf{W}$ that satisfies (8) and has positive weights $w_{nm} > 0$ for $n \neq m$ can be seen as a negative Laplacian matrix of the underlying graph. In this case, the multiplicity of the eigenvalue $\lambda_1 = 0$ equals to the number of connected components in the graph [14], that is, $m_1 = 1$, since the graph is connected. Hence, the conditions of Theorem 1 are satisfied, and the average consensus can be computed exactly in finite time by the considered network. □

The next theorem identifies the smallest number of iterations required to compute the average consensus.

Theorem 2 Consider a network of N agents and a polynomial $h(z)$ that satisfies the conditions of Theorem 1. The minimum number of iterations required to compute the distributed average consensus (10) by this network is $K - 1$.

Proof: The system (21) of $K - 1$ linear equations can be written in the matrix-vector form as

$$\begin{bmatrix} 1 & \lambda_1 & \ldots & \lambda_1^t \\ 1 & \lambda_2 & \ldots & \lambda_2^t \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_K & \ldots & \lambda_K^t \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_t \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{22}$$

Since the system matrix of (22) is a Vandermonde matrix, this linear system has an exact solution if $t \geq K - 1$ [13]. Hence, the minimum number of iterations required to compute the distributed average consensus is $K - 1$. □

Notice that Theorems 1 and 2 only specify how to determine coefficients h_i of the polynomial $h(z)$. However, the implementation (12) requires the knowledge of the coefficients g_i, which in turn requires the factorization of the corresponding polynomial $h(z)$. For networks with a large number of agents N this task can be computationally expensive and numerically unstable.

As we demonstrate next, the factorization of the polynomial $h(z)$ actually can be completely avoided for the computation algorithm that used the fewest possible number of iterations $K - 1$.

Theorem 3 The coefficients g_i and f_t required to compute the average consensus operator in exactly $K - 1$ iterations are

$$\begin{cases} g_k = -\lambda_{k+2}, & 0 \leq k \leq K - 2, \\ f_t = \frac{(-1)^{K-1}}{\lambda_2 \lambda_3 \cdots \lambda_K}. \end{cases} \tag{23}$$

Proof: As follows from Theorem 2, the minimal possible degree for polynomial $h(z)$ that satisfies (12) is $t = K - 1$. In this case, it has $t = K - 1$ roots $-g_i$ for $0 \leq i \leq t - 1$.

By direct inspection of conditions (21) we conclude that these roots are precisely $-g_i = \lambda_{i+2}$ for $0 \leq i \leq K - 2$, which leads to the first part of (23).

Combining this result with the fact that $h_0 = 1$, as established by the equality (21) in Theorem 1, we obtain

$$h_t = \frac{(-1)^{K-1}}{\lambda_2 \lambda_3 \cdots \lambda_K}.$$

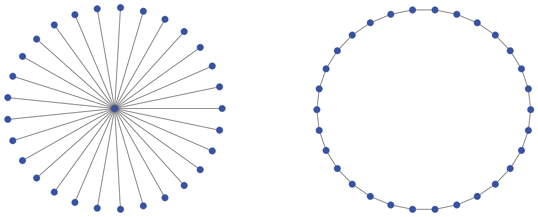
Since $f_t = h_t$ (see (16)), we obtain the second part of (23). □

We would like to point out here that a result similar to Theorem 3 was obtained in [10] for the distributed average consensus computation in exactly $K-1$ iterations in sensor networks with non-negative weights $w_{nm} \geq 0$. In contrast, our approach applies to networks with arbitrary weights. Moreover, Theorems 1 and 2 provide a general solution for the exact implementation of the average consensus in an arbitrary number of iterations $t \geq K - 1$. The solution can be obtained by solving (22) for the desired value of t.

Discussion. Given a network with a known weight matrix $\mathbf{W}$, we can use the solution (23) to quickly determine the number of iterations required to computed the distributed averaging operator (10) and corresponding coefficients.

Recall that the minimal number of iterations required to compute the average consensus in any network cannot be smaller than the diameter of the network. In general, the fastest algorithms that compute the average consensus in the smallest possible number of iterations require careful selection and tuning of parameters $w_{nm}(t)$ in (1) for each time step t. However, despite the restrictions (3), our framework can also produce algorithms that compute the average consensus in the smallest number of iterations.

For example, consider the star network of N agents in Fig. 1(a). Its diameter is two; hence, the fastest algorithm for average consensus would require two iterations. Let us set all non-zero coefficients w_{nm} to 1 and ensure the condition (8) by setting $w_{11} = 1 - N$ and $w_{nn} = -1$ for $2 \leq n \leq N$. The corresponding communication matrix $\mathbf{W}$ has $K = 3$ distinct eigenvalues $\lambda_1 = 0$, $\lambda_2 = -1$, and $\lambda_3 = -N$ with respective multiplicities $m_1 = 1$, $m_2 = N - 2$, and $m_3 = 1$. As follows from Theorem 2, the computation algorithm



(a) Star (b) Circle

Fig. 1. Example graphs representing networks of agents. All edges are undirected.

requires $t = K - 1 = 2$ iterations. It is thus the fastest algorithm. As follows from Theorem 3, the corresponding coefficients are $f_2 = 1/N$, $g_0 = 1$ and $g_1 = N$.

Another example is the circle network of N agents in Fig. 1(b). Its diameter is $N/2$ (we assume that N is even). Again, we set non-zero coefficients w_{nm} to 1 and satisfy the condition (8) by setting $w_{nn} = -2$ for all n. In this case, the eigenvalues of corresponding $\mathbf{W}$ are $\lambda_k = 2\cos(2\pi k/N) - 2$ for $0 \leq k \leq N/2$ with multiplicity $m_k = 1$ for $k \in \{0, N/2\}$ and $m_k = 2$ otherwise. Hence, the distributed average consensus can be computed by this circle network in $N/2$ iterations, which is the fastest possible algorithm.

4. APPROXIMATE COMPUTATION

The average consensus cannot be computed exactly and in finite time by a network that does not satisfy the conditions of Theorem 1. In this case, we can only compute the consensus in finite time approximately.

The closest approximation $h(\mathbf{W})$ of the average consensus operator $\mathbf{P}$ minimizes the output error of the operator. It minimizes the spectral norm of the difference $h(\mathbf{W}) - \mathbf{P}$ and can be found by solving the minimization problem

$$\underset{h(z)}{\text{minimize}} \qquad ||h(\mathbf{W}) - \mathbf{P}||_2 \,. \tag{24}$$

By introducing a slack variable s, (24) can be formulated and solved as a semidefinite program [5]

$$\begin{aligned} &\underset{h(z)}{\text{minimize}} && s \\ &\text{subject to} && \begin{bmatrix} s\,\mathbf{I} & h(\mathbf{W}) - \mathbf{P} \\ h(\mathbf{W}) - \mathbf{P} & s\,\mathbf{I} \end{bmatrix} \succeq 0. \end{aligned}$$

Here, $\succeq$ denotes matrix inequality: the relation $\mathbf{W} \succeq \mathbf{B}$ means that $\mathbf{W} - \mathbf{B}$ is a positive semidefinite matrix.

Discussion. For some networks, the search for the best approximation to average consensus can be simplified even further or even solved exactly. For example, consider a network with a symmetric matrix $\mathbf{W}$ that satisfies condition (8), but its eigenvalue $\lambda_1 = 0$ is not simple. In this case, the objective function in (24) can be modified as

$$\begin{aligned} ||h(\mathbf{W}) - \mathbf{P}||_2 &= \left|\left|\mathbf{V}^T \left(h(\mathbf{W}) - \mathbf{P}\right) \mathbf{V}\right|\right|_2 \\ &= \left|\left|\begin{bmatrix} h(\lambda_1) - 1 & & & \\ & h(\lambda_1) & & \\ & & \ddots & \\ & & & h(\lambda_K) \end{bmatrix}\right|\right|_2 \\ &= \max\left\{(h(\lambda_1) - 1)^2, h(\lambda_1)^2, \ldots, h(\lambda_K)^2\right\}. \end{aligned}$$

Since the values $h(\lambda_k)$ for $2 \leq k \leq K$ can be set arbitrarily small, the minimization problem (24) is equivalent to the problem

$$\underset{h(z)}{\text{minimize}} \qquad \max\left\{(h(\lambda_1) - 1)^2, h(\lambda_1)^2\right\}, \tag{25}$$

which has the exact solution corresponding to $h(\lambda_1) = 1/2$. Hence, there exist infinitely many optimal approximate finite-time algorithms for the computation of the average consensus by a network with symmetric matrix $\mathbf{W}$ that satisfies condition (8) and has a repeating eigenvalue $\lambda_1 = 0$. These algorithms can be found by solving the system of equations

$$\begin{cases} h(0) = h_0 = 1/2, & \\ h(\lambda_k) = \gamma_k, & 2 \leq k \leq K, \end{cases} \tag{26}$$

where γ_k are arbitrary constants that satisfy $-1/2 \leq \gamma_k \leq 1/2$ for $2 \leq k \leq K$. In particular, if we set all $\gamma_k = 0$ in (26), we immediately obtain

$$\begin{cases} g_k = -\lambda_{k+2}, & 0 \leq k \leq K-2, \\ f_t = \frac{(-1)^{K-1}}{2\lambda_2\lambda_3\cdots\lambda_K}. & \end{cases}$$

As an example, consider an undirected network that consists of several components that are not connected to each other. The average consensus cannot be computed in this network. Its optimal approximation is an algorithm in which every agent computes an average within its own component.

Our approach yields this optimal approximation algorithm that computes in finite time. For instance, consider a network consisting of two components with M and $N - M$ agents. For this network, $m_1 \geq 2$. Assuming that $m_1 = 2$, the eigenvalue $\lambda_1 = 0$ has two orthonormal eigenvectors $\mathbf{v}_1$ given by (19) and

$$\mathbf{v}_2 = \sqrt{\frac{N-M}{NM}} \begin{bmatrix} \mathbf{1}_M \\ -\frac{M}{N-M}\mathbf{1}_{N-M} \end{bmatrix},$$

where $\mathbf{1}_n$ denotes a vector of length n containing all ones. In this case, the solution to (26) with all $\gamma_k = 0$ yields a polynomial $h(z)$ of degree $K - 1$ that satisfies

$$\begin{aligned} h(\mathbf{W}) &= \mathbf{V} \begin{bmatrix} \frac{1}{2} & & & & \\ & \frac{1}{2} & & & \\ & & & 0 & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \mathbf{V}^T \\ &= \frac{1}{2} \begin{bmatrix} \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T & \\ & \frac{1}{N-M}\mathbf{1}_{N-M}\mathbf{1}_{N-M}^T \end{bmatrix}. \end{aligned} \tag{27}$$

Comparing (27) with the consensus matrix (10), we observe that this algorithm computes the average consensus in each component in $t = \deg h(z) = K - 1$ iterations.

5. CONCLUSIONS

We have presented a new framework for distributed average consensus computation by agent networks. The proposed framework leads to a systematic design of iterative algorithms that compute the consensus exactly, are guaranteed to converge in finite time, are computationally efficient, and require no online memory. We demonstrated that our approach can be used with a broad class of networks. In addition, we demonstrated that for all other networks the presented framework leads to the construction of approximating algorithms that are also guaranteed to compute in finite time and can be found by solving a semidefinite program.

6. REFERENCES

[1] M. H. DeGroot, “Reaching a consensus,” *J. Amer. Statistic. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.

[2] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, September 1986.

[3] 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, Jun. 2003.

[4] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.

[5] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[6] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[7] S. Sundaram and C. N. Hadjicostis, “Distributed function calculation and consensus using linear iterative strategies,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 650–660, 2008.

[8] S. Pequito, S. Kruzick, S. Kar, J. M. F. Moura, and A. P. Aguiar, “Optimal design of distributed sensor networks for field reconstruction,” in *21st European Signal Processing Conference (EUSIPCO)*, Marrakech, Morocco, Sept. 9 - 11 2013.

[9] A. Olshevsky and J. N. Tsitsiklis, “Convergence rates in distributed consensus and averaging,” in *Proc. IEEE Conf. Decis. Contr.*, 2006, pp. 3387–3392.

[10] A. Y. Kibangou, “Finite-time average consensus based protocol for distributed estimation over AWGN channels,” in *Proc. IEEE Conf. Decis. Contr.*, 2011, pp. 5595–5600.

[11] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs,” *IEEE Trans. Signal Proc.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[12] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Graph filters,” in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, 2013, pp. 6163–6166.

[13] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, 2nd edition, 1985.

[14] S. Kar and J. M. F. Moura, “Sensor networks with random links: Topology design for distributed consensus,” *IEEE Trans. Signal Proc.*, vol. 56, no. 7, pp. 3315–3326, 2008.