Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Faster Linear Iterations for Distributed Averaging ⋆

Björn Johansson and Mikael Johansson *

* *School of Electrical Engineering, Royal Institute of Technology (KTH), 100 44 Stockholm (email: firstname.lastname@ee.kth.se).*

Abstract: Distributed averaging problems are a subclass of distributed consensus problems, which have received substantial attention from several research communities. Although many of the proposed algorithms are linear iterations, they vary both in structure and state dimension. In this paper, we investigate the performance benefits of adding extra states to distributed averaging iterations. We establish conditions for convergence and discuss possible ways of optimizing the convergence rates. By numerical examples, it is shown that the performance can be significantly increased by adding extra states. Finally, we provide necessary and sufficient conditions for convergence of a more general version of distributed averaging iterations.

1. INTRODUCTION

In this paper, we discuss distributed averaging, which is a special class of distributed consensus. Distributed averaging has found diverse applications in areas such as multi-robot coordination, distributed synchronization and resource allocation in communication networks. A large body of literature exists, see the recent survey Olfati-Saber et al. (2007) for a good starting point.

Since the distributed consensus problem has received a large attention from several diverse research communities (from physics, mathematical programming to control and communications), it is natural that several different algorithms have been proposed. Many of these are linear iterations where each agent maintains one internal consensus state. Other algorithms are nonlinear, or have a larger state dimension than the basic iteration. One example of this is when the consensus is used for distributed quadratic programming where nodes maintain states corresponding to primal and dual (Lagrange multiplier) iterates. The iterations are then still linear, but of higher state dimension than the basic distributed averaging algorithm.

The objective of this paper is to try to understand the potential benefits of linear averaging iterations with extended state dimension. We establish conditions for convergence and explore how one could optimize the parameters to obtain faster convergence. Somewhat surprisingly, we show that it is possible to make existing distributed averaging algorithms substantially faster with a simple extension that do not change the communication topology.

We consider systems of the following type

$$\begin{cases} x(k+1) = Ax(k) + Bu \\ y(k+1) = Cx(k+1), \end{cases} \tag{1}$$

where $x(k)$ is a vector of the nodes' internal states; $y(k)$ is a vector of the nodes' outputs; and u is a vector with constant input. The asymptotic output should be the average value of the starting values of the nodes. Furthermore, the matrices A, B, and C respect the communication topology of the network, so that the nodes need to rely on only peer-to-peer communication to execute the iterations.

⋆ This work was supported in part by the European Commission.

1.1 Assumptions and Notation

The network is composed of N nodes, and it is described by a graph with vertices $\mathcal{V} = \{1, ..., N\}$ and edges $\mathcal{E} \subseteq \{(i,j) \,|\, i,j \in \mathcal{V}\}$. The network is assumed to be strongly connected and the links are assumed to be bidirectional, i.e., $(i,j) \in \mathcal{E}$ implies $(j,i) \in \mathcal{E}$. The system matrices should respect the communication topology and each node should only need state information from its neighbors. To this end, we define the set $\mathcal{S} = \{S \in \mathbb{R}^{N\times N} | S_{ij} = 0 \,\forall\, (i,j) \notin \mathcal{E}\}$, and all matrices in this set respect the communication topology. Each node i has an initial value, z_i, and the asymptotic output for each node should be the mean value of all nodes' starting values, $\lim_{k\to\infty} y_i(k) = \sum_{j=1}^{N} z_j/N$. The symbol 1_N denotes the $N \times 1$ vector with all entries equal to one, and the symbol I_N denotes the $N \times N$ identity matrix. Finally, we denote the Kronecker product with $\otimes$.

We can use different notions of convergence rates to quantify how fast the different algorithms approaches the desired output or fixed point. One choice is the worst case per step convergence factor,

$$\sup_{x\neq 0} \frac{||Ax||_2}{||x||_2} = ||A||_2,$$

which is the spectral norm. However, the most common choice is the worst case geometric average convergence factor, for which we have the following (Varga, 1962, Theorem 3.2)

$$\lim_{k\to\infty} \left(\sup_{x\neq 0} \frac{||A^k x||_2}{||x||_2} \right)^{1/k} = \rho(A),$$

where $\rho(A)$ is the spectral radius of A, i.e., the eigenvalue of A with the largest magnitude.

1.2 Related Work

Xiao and Boyd (2004) consider the iterations

$$x(k+1) = Ax(k) \tag{2}$$

and provide necessary and sufficient conditions on the matrix A for the iterations to converge to the average of the initial values. The average is reached if A fulfills

$$\lim_{k\to\infty} A^k = \frac{1}{N} 1_N 1_N^\intercal.$$

For the case of a symmetric A, they also provide a convex optimization problem formulation to find an A that minimizes the spectral radius. However, it is possible to do better using a slightly modified algorithm: if we allow scaling of the output, then it is possible to get the spectral radius arbitrarily close to zero (Olshevsky and Tsitsiklis, 2006). The drawback is that the resulting algorithm will suffer from numerical difficulties. Olshevsky and Tsitsiklis (2006) also provide simple and numerically stable algorithms that have good worst case performance. In addition, it is possible to boost the convergence by using a shift-register in the stochastic version of (2), as noted by Cao et al. (2006).

It is also possible to reach similar iterations by a completely different approach. Namely, if we view the average value as the optimal solution to a least-squares problem,

$$\begin{aligned} \underset{\{x_i\}_{i=1}^N}{\text{minimize}} \ & \sum_{i=1}^{N} \frac{1}{2}(x_i - x_i(0))^2 \\ & x_i = x_j, \, \forall (i,j) \in \mathcal{E}, \end{aligned} \tag{3}$$

then a distributed optimization algorithm solving this problem will also be a distributed averaging algorithm. Several optimization algorithms can be used, e.g., dual relaxation combined with a subgradient method (Rabbat et al., 2005) or the alternating direction multiplier method (Schizas et al., 2007). With a carefully posed optimization problem, both of the methods result in iterations of the same form as in (1).

In addition, these alternative formulations can be used to devise iterations with bolstered resilience to communication noise, as shown in Schizas et al. (2007). This extension can be useful, since the iterations (2) is sensitive to communication noise due to its eigenvalue at 1 for A. However, the algorithm presented in Schizas et al. (2007) has the problem that internal values in the nodes blow up, since the desired fixed point is encoded in $x(k+1) - x(k)$.

In this paper we look at convergence conditions and possible optimization schemes for the augmented versions of (2). We also look at convergence conditions for iterations of the more general form (1).

1.3 Outline

In Section 2, the general shift-register case for systems without input is considered. Then, in Section 3, we discuss how the system matrices can be found using different optimization schemes. Next, in Section 4, we investigate the possible benefits of these schemes using numerical examples. In Section 5, we provide necessary and sufficient conditions for convergence for the more general case of (1) with nonzero input. Finally, the paper is concluded with a discussion in Section 6.

2. GENERAL SHIFT-REGISTER CASE

Shift-registers can be used to speed up convergence in the stochastic version of (2) (Cao et al., 2006). As we will see later in this section, shift-registers can be used to speed up convergence in the deterministic case as well. In fact, shift-registers are known to speed up several types of other iterative methods as well (Young, 1972).

For the consensus iteration case, shift-registers result in iterations of the type

$$\begin{cases} x(k+1) = \begin{pmatrix} \beta A_{11} & (1-\beta) I_N \\ I_N & 0 \end{pmatrix} x(k), \, x(0) = \begin{pmatrix} I_N \\ I_N \end{pmatrix} z \\ y(k+1) = \begin{pmatrix} I_N & 0 \end{pmatrix} x(k+1), \end{cases}$$

where β is a scalar constant, A_{11} is a matrix constant (a matrix coming from (2) can be used). The limit (if it exists) has the structure

$$\lim_{k\to\infty} \begin{pmatrix} \beta A_{11} & (1-\beta) I_N \\ I_N & 0 \end{pmatrix}^k = \begin{pmatrix} \alpha\Delta & (1-\alpha)\Delta \\ \alpha\Delta & (1-\alpha)\Delta \end{pmatrix},$$

where α is a function of β.

The generalized version of this iteration, with M copies of the initial state, is the following

$$\begin{cases} x(k+1) = Ax(k), \, x(0) = 1_M \otimes z \\ y(k+1) = \begin{pmatrix} I_N & 1_{M-1}^\intercal \otimes 0 \end{pmatrix} x(k+1), \end{cases} \tag{4}$$

where $A \in \mathbb{R}^{MN \times MN}$. To describe this generalized version in terms of (1), we have that $B = 1_M \otimes 0$ and $C = \begin{pmatrix} I_N & 1_{M-1}^\intercal \otimes 0_N \end{pmatrix}$. Furthermore, in order for the asymptotic output to reach the desired average, A needs to satisfy the limit

$$\lim_{k\to\infty} A^k = \frac{1}{N} \begin{pmatrix} 1_N \\ \vdots \\ 1_N \end{pmatrix} \begin{pmatrix} \alpha_1 1_N^\intercal & \dots & \alpha_M 1_N^\intercal \end{pmatrix}, \, \sum_{i=1}^{M} \alpha_i = 1, \tag{5}$$

since then

$$\lim_{k\to\infty} y(k) = \frac{1}{N} \sum_{i=1}^{M} \alpha_i 1_N 1_N^\intercal z.$$

We have the following theorem (similar to Theorem 2 in Xiao and Boyd (2004)).

Theorem 1. The iteration (4) satisfies (5) if and only if A and α fulfill the following conditions.

a)

$$Af = f, \, f = \frac{1}{N} \begin{pmatrix} 1_N \\ \vdots \\ 1_N \end{pmatrix}. \tag{6}$$

b)

$$g^\intercal(\alpha) A = g^\intercal(\alpha), \, g(\alpha) = \begin{pmatrix} \alpha_1 1_N \\ \vdots \\ \alpha_M 1_N \end{pmatrix}, \, \sum_{i=1}^{M} \alpha_i = 1. \tag{7}$$

c)

$$\rho\left(A - f g^\intercal(\alpha)\right) < 1, \tag{8}$$

where $\rho(\cdot)$ denotes the spectral radius.

Proof. We start with showing sufficiency of the conditions. If conditions a) and b) are satisfied, then we have that

$$(A - fg^\intercal(\alpha))^t = (A - Afg^\intercal(\alpha))^t = A^t(I - fg^\intercal(\alpha))^t$$
$$= A^t(I - fg^\intercal(\alpha)) = A^t - fg^\intercal(\alpha),$$

where we used $Af = f$ in the first equality, and the third equality is based on the fact that $(I - fg^\intercal(\alpha))(I - fg^\intercal(\alpha)) = I - 2fg^\intercal(\alpha) + fg^\intercal(\alpha)fg^\intercal(\alpha) = I - 2fg^\intercal(\alpha) + fg^\intercal(\alpha)(1/N)\sum_{i=1}^{M} N\alpha_i = I - fg^\intercal(\alpha)$. Now condition c) implies that $\lim_{k\to\infty} A^t - fg^\intercal(\alpha) = \lim_{k\to\infty}(A - fg^\intercal(\alpha))^t = 0$, and sufficiency is established. We continue with necessity. The limit $\lim_{k\to\infty} A^t = fg^\intercal(\alpha)$ exists if and only if (Meyer and Plemmons, 1977)

$$A = T\begin{pmatrix} I_\kappa & 0 \\ 0 & Z \end{pmatrix} T^{-1}, \tag{9}$$

where T is an invertible matrix, Z is a matrix with $\rho(Z) < 1$, and I_κ is the κ-dimensional identity matrix. Since

$$T\begin{pmatrix} I_\kappa & 0 \\ 0 & 0 \end{pmatrix} T^{-1} = fg^\intercal(\alpha)$$

and $\operatorname{rank} fg^\intercal(\alpha) = 1$, we know that $\operatorname{rank} I_\kappa = 1$ and $\kappa = 1$. The limit and (9) also imply that $fg^\intercal(\alpha)A = Afg^\intercal(\alpha) = fg^\intercal(\alpha)$, thus f is a right eigenvector and g is a left eigenvector, both with eigenvalue 1. Finally, we also have that

$$\rho(A - fg^\intercal(\alpha)) = \rho\left(T\begin{pmatrix} 0 & 0 \\ 0 & Z \end{pmatrix} T^{-1}\right) < 1.$$

We conclude that the conditions are both necessary and sufficient. ■

3. MINIMIZING THE SPECTRAL RADIUS

As mentioned in Section 1.1 and Theorem 1, the spectral radius is crucial for the convergence rate. Thus, it is of interest to find an A with minimal spectral radius. Optimization of the spectral radius is very hard in general, but in the case of a Hermitian A, the spectral norm and the spectral radius coincide; the spectral norm is much easier to work with.

3.1 Scaling

Inspired by Xiao and Boyd (2004), we will now search for a transformation that allow us to use Hermitian matrices in the optimization problem. We desire that A fulfills the convergence conditions given in Theorem 1. Let $A \in \mathcal{S}$ be given by the following similarity transformation of a Hermitian matrix $\tilde{A}$,

$$A = T\tilde{A}T^{-1}.$$

Condition a) in Theorem 1 becomes

$$Af = f \Leftrightarrow \tilde{A}T^{-1}f = T^{-1}f.$$

Condition b) (with fixed α) in Theorem 1 becomes

$$g^\intercal(\alpha)A = g^\intercal(\alpha) \Leftrightarrow g^\intercal(\alpha)T\tilde{A} = g^\intercal(\alpha)T.$$

Since $\tilde{A}$ is assumed to be Hermitian, a left-eigenvector, $g^\intercal(\alpha)T$, will imply the following right-eigenvector

$$(g^\intercal(\alpha)T\tilde{A})^* = \tilde{A}^*T^*\bar{g} = \tilde{A}T^*\bar{g}.$$

The two eigenvectors should be equal in order for the scaling to be useful,

$$T^*\bar{g} = T^{-1}f \Leftrightarrow TT^*\bar{g} = f.$$

If such a T exists, we can use the following convex optimization problem to find the fastest converging A of the form $A = T\tilde{A}T^{-1}$,

$$\begin{aligned} \underset{\tilde{A},A}{\text{minimize}} \quad & ||\tilde{A} - T^{-1}fg^\intercal(\alpha)T||_2 \\ \text{such that} \quad & \tilde{A}^* = \tilde{A} \\ & \tilde{A}T^{-1}f = T^{-1}f \\ & AT = T\tilde{A} \\ & A \in \mathcal{S}. \end{aligned}$$

The drawback is that it is not known which the best transformation T is, and that the fastest converging A may not be on the form $A = T\tilde{A}T^{-1}$.

3.2 Bilinear Matrix Inequalities

Instead of trying to find a scaling that convert the problem back to the symmetric case, we can directly look for an unsymmetric matrix (for a fixed α in condition b)) that satisfies Theorem 1 and the sparsity constraints. Define the error, $\hat{x}(k)$, as $\hat{x}(k) = x(k) - x^\star$, where $x^\star$ is the fixed point of the linear iteration. The error should decay to zero as fast as possible. Consider the following inequality

$$||\hat{x}(k+1)||_Q^2 - ||\hat{x}(k)||_Q^2 \leq -\psi||\hat{x}(k)||_Q^2,$$

which, using the Schur complement and a congruence transformation, is equivalent with the following positive semidefinite expressions

$$\begin{cases} \begin{pmatrix} (1-\psi)Q^{-1} & Q^{-1}(A - fg^\intercal(\alpha))^\intercal \\ (A - fg^\intercal(\alpha))Q^{-1} & Q^{-1} \end{pmatrix} \geq 0 \\ Q^{-1} > 0. \end{cases} \tag{10}$$

If ψ, Q^{-1}, and A are decision variables, then (10) is a so called bilinear matrix inequality (BMI). We can then use the following optimization problem

$$\begin{aligned} \underset{A,\psi,Q^{-1},\Upsilon}{\text{minimize}} \quad & -\psi \\ \text{such that} \quad & \Upsilon = A - fg^\intercal(\alpha) \\ & 0 \leq \begin{pmatrix} (1-\psi)Q^{-1} & Q^{-1}\Upsilon^\intercal \\ \Upsilon Q^{-1} & Q^{-1} \end{pmatrix} \\ & 0 < Q^{-1} \\ & f = Af \\ & g^\intercal(\alpha) = g^\intercal(\alpha)A \\ & 0 \leq \beta \leq 1 \\ & A \in \mathcal{S}, \end{aligned}$$

to find an unsymmetric A that minimizes $||\hat{x}(k+1)||_Q^2$, and thereby also pushing down the spectral radius of $(A - fg^\intercal(\alpha))$, while still satisfying Theorem 1 and the sparsity constraints. The drawback with this approach is that the optimization problem is nonconvex.

4. NUMERICAL EXAMPLES

To evaluate the performance of the different linear iterations that achieve distributed averaging, we consider the simple network in Figure 1. To avoid numerical difficulties and maintain computational tractability, the network consists of only 7 nodes. We test four methods of finding the system matrices, using the software packages YALMIP (Löfberg, 2004), SDPT3 (Tutuncu et al., 2003), and PENBMI (Kocvara and Stingl, 2003).

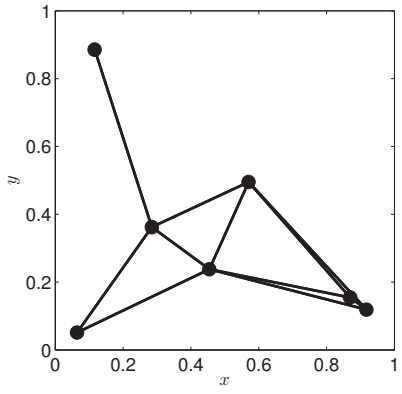

Fig. 1. Topology of the random 7 node network.

4.1 Symmetric Matrix Algorithm. Dimension $N \times N$.

We use the method described in Xiao and Boyd (2004) in conjunction with YALMIP and SDPT3 to find the symmetric A that minimizes the spectral norm (which is the same as the the spectral radius in this case).

4.2 Unsymmetric Matrix Algorithm. Dimension $N \times N$.

The method outlined in Section 3.2 is used in conjunction with YALMIP and PENBMI to find an unsymmetric matrix that is a local minimum of the matrix norm $m(A) = ||A - 1_N 1_N^\intercal/N||_Q^2$, and since we are optimizing over Q as well, this norm is close to the spectral radius.

4.3 Shift-Register Algorithm. Dimension $2N \times 2N$.

Consider the shift-register example (with a fixed matrix A_{11} and M copies of the state) once again. The structure of A is

$$A = \begin{pmatrix} \beta_1 A_{11} & \dots & \beta_{M-1} I & \beta_M I \\ I & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & I & 0 \end{pmatrix}.$$

With this structure and $\sum_{i=1}^{M} \beta_i = 1$, condition a) in Theorem 1 is satisfied. To satisfy condition b) in Theorem 1, α and β need to satisfy

$$\begin{aligned} \alpha_M &= \alpha_1 \beta_M, \\ \alpha_{M-1} &= \alpha_1 \beta_{M-1} + \alpha_M, \\ \alpha_{M-2} &= \alpha_1 \beta_{M-2} + \alpha_{M-1}, \\ &\vdots \\ \alpha_1 &= \alpha_1 \beta_1 + \alpha_2, \end{aligned}$$

in addition with $\sum_{i=1}^{M} \alpha_i = 1$. Finally, to satisfy condition c) in Theorem 1, α, β, and A need to satisfy

$$\rho\left(A(\beta) - f g^\intercal(\alpha)\right) < 1.$$

We use the above conditions with $M = 2$ (this means we have one shift-register and one scalar weight to chose, namely β) and use the A matrix from Section 4.1 as our A_{11} matrix. A search for the optimal β in the interval $[-2, 2]$ is then performed. Since we only have one scalar variable to chose, the interval search is rather computationally cheap. The interval search gave $\beta = 1.1717$.

4.4 General Shift-Register Algorithm. Dimension $2N \times 2N$.

Here we use the f and g vectors from Section 4.3 ($M = 2$ and $\beta = 1.1717$) and search for a general unsymmetric A using the method in Section 3.2, with

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},\ A_{11}, A_{12}, A_{21}, A_{22} \in \mathcal{S}.$$

It is possible to do a search over β in this case as well, but it is not really tractable, since the optimization with BMI is rather time consuming.

4.5 Results

For the communication topology shown in Figure 1, the performance of the four resulting algorithms are shown in Figure 2. We use the following performance metrics: for the $N \times N$ matrices

$$\text{performance}_{N\times N}(k) = \left\| A^k - 1_N 1_N^\intercal/N \right\|_2, \tag{11}$$

and for the $2N \times 2N$ matrices

$$\text{performance}_{2N\times 2N}(k) = \left\| (I_N\ 0)\, A^k \begin{pmatrix} I_N \\ I_N \end{pmatrix} - 1_N 1_N^\intercal/N \right\|_2. \tag{12}$$

The general shift-register algorithm has significantly better performance than the other algorithms. The unsymmetric matrix algorithm is second best, and the shift-register algorithms is quite close. The symmetric matrix algorithm has the worst performance in this example. Thus, this example suggests that much can be gained from using the general shift-register algorithm. However, since this optimization problem is bilinear, it is computationally intractable for topologies with more than ten nodes. The same is valid for the unsymmetric matrix algorithm.

The most computationally viable way to increase the convergence speed is thus the shift-register algorithm from Section 4.3. In Figure 3, we show Monte Carlo simulations for the shift-register algorithm and the optimal symmetric algorithm. For each number of nodes, 1000 connected random networks were generated, and the mean of the performance (using (11) and (12)) for $k = 5, 25, 125$ is shown. The simulations indicate that the shift-register algorithm is better if high accuracy is desired, while the optimal symmetric algorithm has better initial convergence speed.

Thus, a performance boost can be achieved by using the simple shift-register algorithm, with virtually no extra computational burden on the nodes nor on the off-line algorithm that computes the system matrices.

5. GENERAL CASE

We now look at necessary and sufficient conditions for the most general case of (1) to converge to the average of the starting values, while respecting the communication constraints between the nodes.

We consider the following system

$$\begin{cases} x(k+1) = Ax(k) + Bz \\ y(k+1) = Cx(k+1) \\ \quad x(0) = Ez, \end{cases} \tag{13}$$

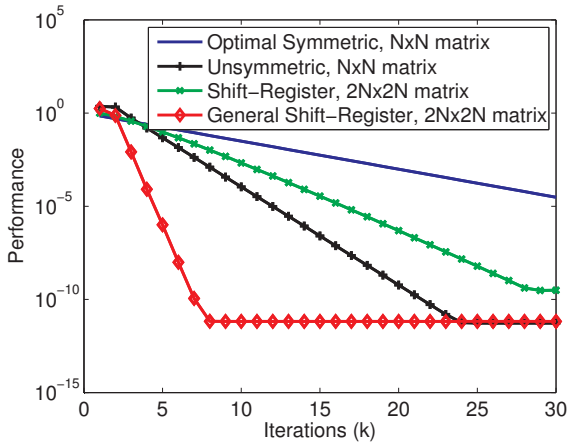

Fig. 2. Performance of the four algorithms for a 7 node network.

with $x(k) \in \mathbb{R}^{NM}$, $y(k) \in \mathbb{R}^{N}$,

$$A = \begin{pmatrix} A_{11} & \dots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \dots & A_{MM} \end{pmatrix}, \; A_{ij} \in \mathcal{S}, \tag{14}$$

$$B = \begin{pmatrix} B_1 \\ \vdots \\ B_M \end{pmatrix}, \; B_i \in \mathbb{R}^{N \times N} \text{ and diagonal}, \tag{15}$$

$$C = (C_1 \; \dots \; C_M), \; C_i \in \mathbb{R}^{N \times N} \text{ and diagonal}, \tag{16}$$

$$E = \begin{pmatrix} E_1 \\ \vdots \\ E_M \end{pmatrix}, \; E_i \in \mathbb{R}^{N \times N} \text{ and diagonal}. \tag{17}$$

We have the following theorem.

Theorem 2. Consider the system defined by (13)-(17). The limit $\lim_{k\to\infty} A^k$ exists, the sequence of states, $\{x(k)\}$, is bounded, and the corresponding system output satisfies $\lim_{k\to\infty} y(k) = 1_N 1_N^\intercal z/N$ if and only if the matrices A, B, C, and E satisfy the following conditions:

a) There exist two matrices, T and Z, and an integer, κ, such that

$$A = T \begin{pmatrix} I_\kappa & 0 \\ 0 & Z \end{pmatrix} T^{-1}, \; \rho(Z) < 1, \; 0 \le \kappa \le NM. \tag{18}$$

We divide the rest of the conditions into $B = 0$ and $B \neq 0$. For $B = 0$, the last condition is:

b) The matrix T from a) satisfies

$$CT \begin{pmatrix} I_\kappa & 0 \\ 0 & 0 \end{pmatrix} T^{-1} E = 1_N 1_N^\intercal / N. \tag{19}$$

For $B \neq 0$, the last conditions are:

c) If κ from a) is greater than zero, then the κ eigenvectors of A with eigenvalue 1 are orthogonal to Bz.

d) There exists a vector, $\bar{x}$, such that

$$\bar{x} = A\bar{x} + Bz \text{ and } C\bar{x} = 1_N 1_N^\intercal z/N. \tag{20}$$

e) The nullspace of $(A - I_{NM})$ is a subset of the nullspace of C.

Proof. The limit $\lim_{k\to\infty} A^k$ exists if and only if (Meyer and Plemmons, 1977) condition a) is satisfied. We divide the rest of the proof into two parts, one for $B = 0$ and one for $B \neq 0$.

When $B = 0$, existence of $\lim_{k\to\infty} A^k$ implies that the sequence $\{x(k)\}$ is bounded. The limit of the output is then

$$\lim_{k\to\infty} CTA^kT^{-1}E = CT \begin{pmatrix} I_\kappa & 0 \\ 0 & 0 \end{pmatrix} T^{-1}E.$$

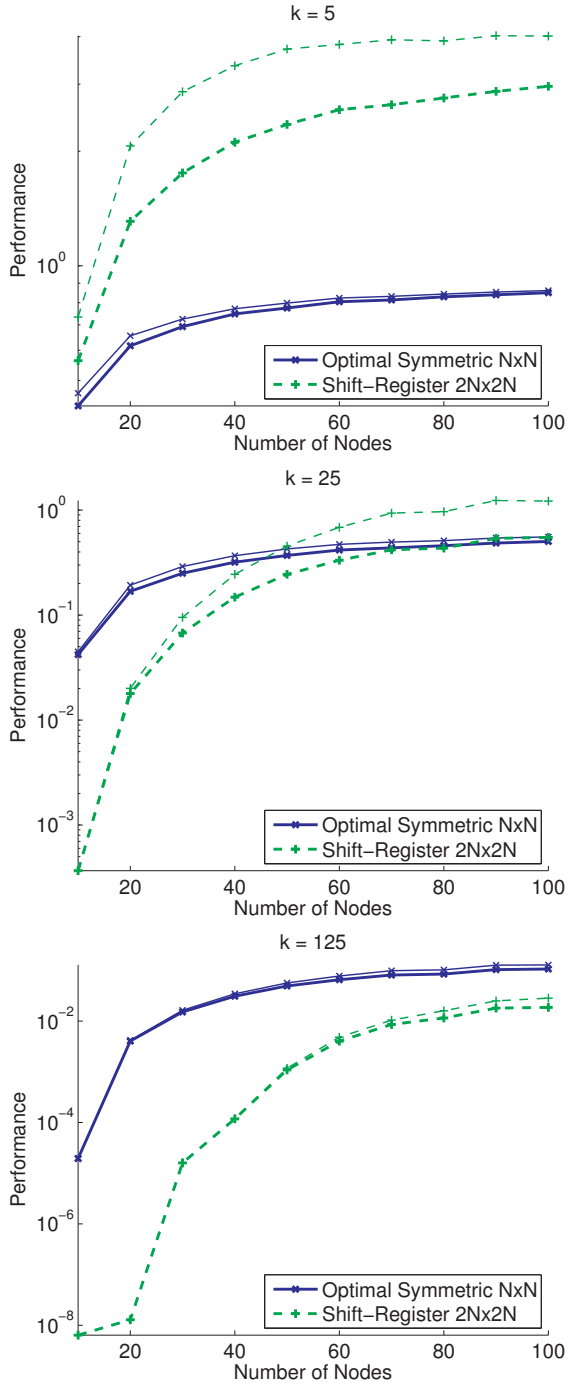

Fig. 3. Average performance of the Symmetric Matrix Algorithm and the Shift-Register Algorithm. For each number of nodes, 1000 random networks were generated and averaged over. The thick line denotes the average and the thin line denotes the sum of the average and the computed variance.

Hence, the desired output can only be reached if and only if condition b) is satisfied.

When $B \neq 0$, we can write the state as

$$x(k) = A^k Ez + \sum_{j=0}^{k-1} A^k Bz = A^k Ez + \sum_{j=0}^{k-1} T \begin{pmatrix} I_\kappa & 0 \\ 0 & Z \end{pmatrix}^k T^{-1} Bz,$$

with

$$T^{-1} = (t_1 \ \dots \ t_N)^\intercal \text{ and } T = \begin{pmatrix} \tilde{t}_1 & \dots & \tilde{t}_N \end{pmatrix}.$$

When $\kappa = 0$, then

$$\lim_{k\to\infty} x(k) = \lim_{k\to\infty} A^k Ez + (I_{NM} - A)^{-1} Bz,$$

which implies that the sequence of states is bounded. When $\kappa > 0$, by using the unique representation of Bz in the basis $\{\tilde{t}_1, \dots, \tilde{t}_N\}$, $Bz = \sum_{i=1}^N \gamma_i \tilde{t}_i$, we can write

$$x(k) - A^k Ez = \sum_{j=0}^{k-1} T \left(\begin{pmatrix} I_\kappa & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & Z^k \end{pmatrix} \right) T^{-1} \sum_{i=1}^N \gamma_i \tilde{t}_i$$

$$= \sum_{j=0}^{k-1} \left(T \begin{pmatrix} I_\kappa & 0 \\ 0 & 0 \end{pmatrix} T^{-1} \sum_{i=1}^{\kappa} \gamma_i \tilde{t}_i + T \begin{pmatrix} 0 & 0 \\ 0 & Z^k \end{pmatrix} T^{-1} \sum_{i=\kappa+1}^N \gamma_i \tilde{t}_i \right)$$

$$= \sum_{j=0}^{k-1} \sum_{i=1}^{\kappa} \gamma_i \tilde{t}_i + \sum_{j=0}^{k-1} T \begin{pmatrix} 0 & 0 \\ 0 & Z^k \end{pmatrix} T^{-1} \sum_{i=\kappa+1}^N \gamma_i \tilde{t}_i,$$

since $t_i^\intercal \tilde{t}_j = 0$ if $i \neq j$ and $t_i^\intercal \tilde{t}_i = 1$. Hence, we have

$$\lim_{k\to\infty} x(k) = T \begin{pmatrix} I_\kappa & 0 \\ 0 & 0 \end{pmatrix} T^{-1} Ez + \lim_{k\to\infty} \sum_{j=0}^{k-1} \sum_{i=1}^{\kappa} \gamma_i \tilde{t}_i +$$

$$+ \left(I_{NM} - T \begin{pmatrix} 0 & 0 \\ 0 & Z \end{pmatrix} T^{-1} \right)^{-1} \sum_{i=\kappa+1}^N \gamma_i \tilde{t}_i,$$

where $\lim_{k\to\infty} \sum_{j=0}^{k-1} \sum_{i=1}^{\kappa} \gamma_i \tilde{t}_i$ exists and the sequence of states is bounded if and only if $\gamma_i = 0$ for all $i = 1, ..., \kappa$, which is equivalent with condition c). Also note that $\lim_{k\to\infty} x(k)$ is a fixed point to (13).

Condition d) is equivalent with that a fixed point with the desired properties exists.

Finally, condition e) is equivalent with that if there are multiple fixed points, all of them give the desired output: assume that there are two fixed points $\bar{x}$ and $\bar{x} + x'$, then the vector x' will be in the nullspace of $(A - I_{NM})$. Now condition e) gives that x' is in the nullspace of C, which implies that $C\bar{x} = C(\bar{x} + x')$. ■

Remark 3. The conditions in Theorem 2 are difficult to use in an optimization problem. For example, condition e) can easily be shown to be nonconvex (in its present form).

Remark 4. It is possible to find matrices fulfilling the conditions in Theorem 2. One choice is the matrices from the algorithms in Section 4. Another choice is the matrices from a dual relaxation solution algorithm to (3).

6. CONCLUSIONS

Motivated by the many different algorithms that have been proposed for distributed averaging, we have investigated the convergence of a more general class of linear averaging iterations. First, we considered iterations that allow nodes to maintain several internal states. We provided necessary and sufficient conditions for convergence, and discussed some possible optimization schemes to improve the convergence rate. It turns out that with a single shift-register it is easy to search for the optimal weights (a one-parameter search), giving a computationally inexpensive way to find a weight matrix A that in some cases converges faster than the original algorithm in Xiao and Boyd (2004). If we search for a nonsymmetric weight matrix, numerical examples indicate that performance can be vastly improved. However, this optimization problem is computationally intractable for larger network topologies. Finally, we looked at necessary and sufficient conditions for the convergence to consensus of a larger class of algorithms with input. These conditions are not amenable for optimization in their present form, but we are looking into this in our future work.

7. ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their detailed comments as well as Alberto Speranzon and Dag Lindbo for helpful discussions.

REFERENCES

M. Cao, D.A. Spielman, and E.M. Yeh. Accelerated gossip algorithms for distributed computation. In *Allerton Conference*, 2006.

M. Kocvara and M. Stingl. PENNON - a code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 18:317–333, 2003.

J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://control.ee.ethz.ch/ joloef/yalmip.php`.

C. D. Meyer and R. J. Plemmons. Convergent powers of a matrix with applications to iterative methods for singular linear systems. *SIAM Journal on Numerical Analysis*, 14:699–705, 1977.

R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

A. Olshevsky and J. Tsitsiklis. Convergence rates in distributed consensus and averaging. In *IEEE CDC*, 2006.

M. Rabbat, R. Nowak, and J. Bucklew. Generalized consensus computation in networked systems with erasure links. In *IEEE SPAWC*, 2005.

I. Schizas, A. Ribeiro, and G. Giannakis. Consensus in ad hoc WSNs with noisy links - part I: Distributed estimation of deterministic signals. *IEEE Transactions on Signal Processing*, 2007. Submitted.

R.H Tutuncu, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming Ser. B*, 95:189–217, 2003.

R. Varga. *Matrix Iterative Analysis*. Prentice-Hall, 1962.

L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

D. M. Young. Second-degree iterative methods for the solution of large linear systems. *Journal of Approximation Theory*, 5:137–148, 1972.