The ADMM algorithm for distributed averaging: Convergence rates and optimal parameter selection

Euhanna Ghadimi, André Teixeira, Michael G. Rabbat, and Mikael Johansson

***Abstract*— We derive the optimal step-size and over-relaxation parameter that minimizes the convergence time of two ADMM-based algorithms for distributed averaging. Our study shows that the convergence times for given step-size and over-relaxation parameters depend on the spectral properties of the normalized Laplacian of the underlying communication graph. Motivated by this, we optimize the edge-weights of the communication graph to improve the convergence speed even further. The performance of the ADMM algorithms with our parameter selection are compared with alternatives from the literature in extensive numerical simulations on random graphs.**

I. Introduction

The distributed averaging problem has attracted a strong recent interest due to its many applications in distributed signal processing, communication, and control. Examples include coordination of multi-agent systems [1]–[3], distributed estimation in wireless sensor networks [4]–[6], and communication networks [7]–[9]. In the distributed averaging problem, each node in a network holds an initial value and is only allowed to communicate with its one-hop neighbors to agree on the network-wide average of these initial values. In the simplest distributed algorithm for average seeking, nodes iteratively update their states by forming convex combinations of their own and their neighbors states [10].

As the communication networks become larger, the performance of traditional averaging algorithms often noticeably degrade [11]. There has been an extensive effort to improve the convergence time by finding the best weights that each node uses to form the convex combinations [12], adding memory to account for the values of the past iterates when computing the next [11], [13], and designing new averaging algorithms based on general large-scale optimization techniques [14]. In this paper, we take the latter approach and explore how the alternating direction method of multipliers (ADMM) [15] can be used to derive fast algorithms for distributed averaging.

The ADMM method has been observed to converge fast in many applications and, for certain classes of problems, it has recently been shown to converge at linear rate [16]–[18]. However, the solution times are sensitive to the choice of the algorithm parameters [19]. We demonstrate that ADMM-based algorithms for distributed averaging, when crafted correctly, outperform alternatives from the literature. Our algorithms are based on casting the distributed averaging problem as a least-squares problem where a number of agents collaborate with neighbors in a graph to minimize a convex quadratic function with a specific sparsity structure over a set of shared and private variables. We derive the corresponding iterations for the ADMM method and show that they converge linearly to the optimum. For given algorithm parameters, we show that the performance of the averaging algorithms are characterized by the magnitude of the eigenvalues of the normalized Laplacian related to the communication graph. Optimal choices of the ADMM algorithm parameters, comprising a constant step-size and over-relaxation parameter, are derived. Furthermore, we optimize the weights that each node assigns to its neighbor in order to improve the convergence times. Finally, the performance of ADMM-based algorithms with our parameter selection rules are compared with the state of the art in extensive numerical simulations on random graphs.

The outline of this paper is as follows. Section II presents suitable formulations for distributed averaging along with the corresponding ADMM iterations. Section III characterizes the performance of the resulting ADMM-based averaging algorithms and derives the optimal algorithm parameters and communication weights. Numerical examples illustrating our results and comparing them to the state-of-the art techniques are presented in Section IV. Finally, a discussion and outlook on future research concludes the paper.

II. Problem Formulation

A set of n agents collaborate to compute the network-wide average of their initial values. The nodes can only exchange information with a subset of the other nodes. We encode this restriction by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose vertices $v \in \mathcal{V}$ correspond to agents. There is an edge $(u, v) \in \mathcal{E}$ if and only if agents u and v can exchange information. If $\mathcal{G}$ is connected, then finding the network-wide average is equivalent to solving the collaborative optimization problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\sum_{i\in\mathcal{V}}(x_i - x_i^0)^2 \\ \text{subject to} \quad & x_i = x_j,\ \forall (i,j) \in \mathcal{E}, \end{aligned} \tag{1}$$

where $x_i^0 \in \mathbb{R}$ is the initial value of node i. Note that there is one decision variable x_i for each node, and at the optimum, each x_i equals the network-wide average of the initial values.

Euhanna Ghadimi, André Teixeira, and Mikael Johansson are with the ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden. Michael G. Rabbat is with Department of Electrical and Computer Engineering, McGill University, Montreal, Quebec, Canada. This work was sponsored in part by the Swedish Foundation for Strategic Research (SSF), the Swedish Research Council (VR). {euhanna,andretei}@kth.se, michael.rabbat@mcgill.ca, mikaelj@kth.se

The formulation above does not specify which information is exchanged between the agents and how the constraints between decision variables are enforced. In what comes next, we consider two alternatives that are commonly used in the literature (c.f., *e.g.* [14]).

A. Enforcing agreement with edge-variables

In the first method, for each edge $(j,i) \in \mathcal{E}$, we introduce a shared variable $z_{(j,i)}$ such that $x_j = z_{(j,i)} = x_i$. Rather than the original formulation (1), we consider

$$\begin{aligned} &\underset{x_i, z_{(i,j)}}{\text{minimize}} \sum_{i\in\mathcal{V}} \frac{1}{2} a_i (x_i - b_i x_i^0)^2 \\ &\text{subject to } R_{i,(i,j)} x_i = R_{i,(i,j)} z_{(i,j)},\ \forall i \in \mathcal{V}, \forall (i,j) \in \mathcal{E}, \end{aligned} \tag{2}$$

where $a_i > 0,\ b_i > 0$, and $\mathcal{W}_{i,(i,j)} \triangleq R^2_{i,(i,j)} > 0$ are positive design parameters. These parameters are introduced to increase the degrees of freedom available for tuning our algorithms and must be selected so that the optimal values of (1) and (2) agree. We will return to this issue in Section III. Now, the ADMM iterations for problem (2) read [15]

$$\begin{aligned} x_i^{k+1} &= \frac{a_i b_i x_i^0 + \rho \sum_{(i,j)\in\mathcal{E}} \mathcal{W}_{i,(i,j)} (z^k_{(i,j)} - u^k_{(i,j)})}{a_i + \rho \sum_{(i,j)\in\mathcal{E}} \mathcal{W}_{i,(i,j)}}, \\ \gamma^{k+1}_{(i,j)} &= \alpha x_i^{k+1} + (1-\alpha) z^k_{(i,j)}, \\ z^{k+1}_{(i,j)} &= \frac{\mathcal{W}_{i,(i,j)} (\gamma^{k+1}_{i,j} + u^k_{(i,j)}) + \mathcal{W}_{j,(j,i)} (\gamma^{k+1}_{(j,i)} + u^k_{(j,i)})}{\mathcal{W}_{i,(i,j)} + \mathcal{W}_{j,(j,i)}}, \\ u^{k+1}_{(i,j)} &= u^k_{(i,j)} + \gamma^{k+1}_{(i,j)} - z^{k+1}_{(i,j)}, \end{aligned} \tag{3}$$

where $\rho > 0$ is a constant step-size and $\alpha \in (0,2]$ is a relaxation parameter. Detailed derivations of these iterations can be found in [20]. Note that $\gamma_{(i,j)}$, $\gamma_{(j,i)}$, $z_{(i,j)}$, and $u_{(i,j)}$ are auxiliary variables residing in node i and can be updated using only information from neighbors in $\mathcal{G}$. Moreover, $z^k_{(i,j)} = z^k_{(j,i)}$ in each iteration $k \geq 1$.

To facilitate the performance analysis of distributed averaging algorithms, we rewrite (3) in matrix notation. Assign arbitrary orientations to the edges $(i,j) \in \mathcal{E}$ and define $B \in \mathbb{R}^{|\mathcal{E}|\times n} \triangleq B_I + B_O$ as the unsigned incidence matrix with $[B_I]_{ij} = 1$ ($[B_O]_{ij} = 1$) if node j is the tail (head) of the edge $e_i = (j,k)$ for $k \in \mathcal{N}_j$ and $[B_I]_{ij} = 0$ ($[B_O]_{ij} = 0$) otherwise. Let $R_O, R_I \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ be diagonal matrices defined as follows. Given $[B_O]_{ij} = 1$ ($[B_I]_{ij} = 1$) associated with the edge $e_i = (j,k)$, we have $[R_O]_{ii} = R_{j,(j,k)}$ ($[R_I]_{ii} = R_{j,(j,k)}$). Moreover, we define $\mathcal{W}_O \triangleq R_O^\top R_O$, $\mathcal{W}_I \triangleq R_I^\top R_I$, and $\mathcal{W} \triangleq \mathcal{W}_O + \mathcal{W}_I$. Denoting $E \triangleq [B_O^\top R_O \ B_I^\top R_I]^\top$ and $F \triangleq -[R_O \ R_I]^\top$, the problem (2) reads as

$$\begin{aligned} \underset{x,z}{\text{minimize}} \quad & \frac{1}{2} x^\top Q x - q^\top x \\ \text{subject to} \quad & Ex + Fz = 0, \end{aligned} \tag{4}$$

where $x^\top = [x_1 \dots x_n]$, $z^\top = [z_1 \dots z_{|\mathcal{E}|}]$, $Q \triangleq \text{diag}([a_1 \dots a_n])$, $q^\top = [a_1 b_1 x_1^0 \dots a_n b_n x_n^0]$.

B. Enforcing agreement with node-variables

In the second formulation, we introduce an auxiliary variable z_i for each node in the network, and then require that $x_j = z_i$ for each j such that $(j,i) \in \mathcal{E}$. We then consider the following problem related to (1):

$$\begin{aligned} &\underset{x_i, z_j}{\text{minimize}} \sum_{i\in\mathcal{V}} \frac{1}{2} a_i (x_i - b_i x_i^0)^2 \\ &\text{subject to } R_{i,(i,j)} x_i = R_{i,(i,j)} z_j,\ \forall i \in \mathcal{V}, \forall j \in \{\mathcal{N}_i \cup \{i\}\}, \end{aligned} \tag{5}$$

The ADMM iterations for this formulation read

$$\begin{aligned} x_i^{k+1} &= \frac{a_i b_i x_i^0 + \rho \sum_{(i,j)\in\mathcal{E}} \mathcal{W}_{i,(i,j)} (z^k_j - u^k_{(i,j)})}{a_i + \rho \sum_{(i,j)\in\mathcal{E}} \mathcal{W}_{i,(i,j)}}, \\ \gamma^{k+1}_{(i,j)} &= \alpha x_i^{k+1} + (1-\alpha) z^k_j, \\ z_i^{k+1} &= \frac{\sum_{j\in\mathcal{N}_i\cup\{i\}} \mathcal{W}_{j,(j,i)} (\gamma^{k+1}_{(j,i)} + u^k_{(j,i)})}{\sum_{j\in\mathcal{N}_i\cup\{i\}} \mathcal{W}_{j,(j,i)}}, \\ u^{k+1}_{(i,j)} &= u^k_{(i,j)} + \gamma^{k+1}_{(i,j)} - z^{k+1}_j. \end{aligned} \tag{6}$$

While these iterations only require information exchange among neighbors in $\mathcal{G}$, two rounds of message passing are required in each iteration: the first to exchange the private variables z_j^k for all $j \in \mathcal{N}_i$ to execute the x_i-update and the second to exchange the local variables x_j^{k+1} for all $j \in \mathcal{N}_i$ to conduct the z_i-update (the weights $\mathcal{W}_{j,(j,i)}$ also need to be available for i).

Following a similar approach as in the edge-variable formulation, we rewrite (6) in the matrix form (4). Specifically we define random orientation matrices B_O and B_I similarly to the previous section except that we now augment these matrices to include self-loops (i,i) for all $i \in \mathcal{V}$ and also add them to the edge set $\mathcal{E}$. While the constraint matrix F is defined $F \triangleq -[B_I^\top R_O \ B_O^\top R_I]^\top$ the rest of the variables in (4) remain unchanged.

In the next section, we study the convergnce properties of the ADMM iterations for the two formulations.

III. Analysis of ADMM-based distributed averaging algorithms

Consider the optimization problem (4) with associated variables defined in the previous section. The ADMM-based algorithm to solve this problem takes the form

$$\begin{bmatrix} x^{k+1} \\ y^k \end{bmatrix} = \underbrace{\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & (1-\alpha)I \end{bmatrix}}_{M} \begin{bmatrix} x^k \\ y^{k-1} \end{bmatrix} \tag{7}$$

with $y^k \triangleq E^\top F z^k$, $x^1 = (Q + \rho E^\top E)^{-1} q$, $y^0 = 0$, and

$$\begin{aligned} M_{11} &= \alpha\rho (Q + \rho E^\top E)^{-1} E^\top (2\Pi_{\mathcal{R}(F)} - I) E + I, \\ M_{12} &= \alpha\rho (Q + \rho E^\top E)^{-1}, M_{21} = -\alpha E^\top \Pi_{\mathcal{R}(F)} E, \\ \Pi_{\mathcal{R}(F)} &\triangleq F(F^\top F)^{-1} F^\top. \end{aligned} \tag{8}$$

The convergence behavior of both averaging algorithms is closely related to the spectral properties of the matrix M. In particular, when $\mathcal{G}$ is connected, $\rho > 0$ and $\alpha \in (0,2]$, the general properties of the ADMM method ensure that M has a single eigenvalue equal to 1 whose associated right

eigenvector is the vector of all ones. When $Q = I$, these properties guarantee that (7) converges to the average of the initial values at a linear rate [20].

To optimize the performance of algorithms (3) and (6), we set up the problem so that the M-matrix has a structure that is convenient to analyze:

Assumption 1: The matrix E is constructed in a way such that $E^\top E = \kappa Q$ for some $\kappa > 0$.

There are several ways to satisfy Assumption 1. In this paper, we consider two such techniques:

Lemma 1: Consider the distributed averaging algorithms (3) and (6). Then for given $\kappa > 0$, Assumption 1 holds if and only if we assign local weights $\mathcal{W}_{i,(i,j)}$ for all $i \in \mathcal{V}$ and all $(i,j) \in \mathcal{E}$ such that

$$\sum_{j \in \mathcal{N}_i} \mathcal{W}_{i,(i,j)} = \kappa a_i. \tag{9}$$

Proof: Please refer to [20] for this and all the other proofs. ∎

A simple way to satisfy the conditions in Lemma 1 is to let nodes assign the same weight $\kappa a_i/|\mathcal{N}_i|$ to all its outgoing or incoming edges. We will employ this simple weight selection in Section IV when we compare the performance of the ADMM-based algorithms to other distributed averaging schemes. The next lemma gives an alternative technique for satisfying Assumption 1:

Lemma 2: Consider problems (2) and (5) and let

$$\kappa = \frac{\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \mathcal{W}_{i,(i,j)}}{n}. \tag{10}$$

Then for $a_i = \kappa^{-1}[E^\top E]_{ii} = \kappa^{-1} \sum_{j \in \mathcal{N}_i} \mathcal{W}_{i,(i,j)}$ and $b_i = 1/a_i$, Assumption 1 is satisfied and (2) and (5) converge to the average of the initial values.

The two techniques have different means for satisfying Assumption 1 that are useful in different contexts. Lemma 1 is useful since it allows for a distributed weight selection without altering the overall problem data, while Lemma 2 is centralized in nature and alters the problem data, but will allow for more powerful weight optimization techniques.

Let $|\phi_i|$ with $i = 1, \ldots, 2n$ be the ascending ordered set of the magnitude of eigenvalues of M. Having the largest eigenvalue of M in (7) at 1, i.e., $|\phi_{2n}| = 1$, the convergence behavior of the ADMM-based consensus algorithms is characterized in terms of its second largest eigenvalue in magnitude, i.e., $|\phi_{2n-1}|$ [20]. The smaller $|\phi_{2n-1}|$, the faster the algorithms (3) and (6) converge to the optimality. Next, we find the best algorithm parameters ρ and α to minimize the convergence factor $|\phi_{2n-1}|$.

Theorem 1: Consider the fixed point consensus iterations (7) and let Assumption 1 hold. Let $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$ be the ordered generalized eigenvalues of the matrix pencil $(E^\top(2\Pi_{\mathcal{R}(F)} - I)E, E^\top E)$. Then

C1: If $\lambda_{n-s} > 0$ and $\lambda_{n-1} \geq |\lambda_1|$, the optimal ADMM parameters and the corresponding optimal convergence factor are given as

$$\alpha^\star = 2,\ \rho^\star = \frac{1}{\kappa\sqrt{1-\lambda_{n-1}^2}},\ |\phi_{2n-1}| = \frac{\lambda_{n-1}}{1+\sqrt{1-\lambda_{n-1}^2}}$$

C2: If $|\lambda_1| \geq \lambda_{n-1} > 0$, the parameter choices

$$\alpha^\star = \frac{4}{2-(\lambda_1+\lambda_{n-1})\beta + \sqrt{\lambda_1^2\beta^2 - 2\beta + 1}},$$
$$\rho^\star = \frac{1}{\kappa\sqrt{1-\lambda_{n-1}^2}},$$

with the associated convergence factor

$$|\phi_{2n-1}| = 1 - \frac{\alpha^\star}{2}(1 - \lambda_{n-1}\beta)$$

where $\beta = 1/(1+\sqrt{1-\lambda_{n-1}^2})$ outperform all combinations of $\alpha \in (0, \alpha^\star)$ and $\rho > 0$.

C3: If $0 \geq \lambda_{n-1}$, the parameter selection

$$\rho^\star = \frac{1}{\kappa}, \qquad \alpha^\star = \frac{4}{2-\lambda_1}$$

with associated convergence factor

$$|\phi_{2n-1}| = \frac{-\lambda_1}{2-\lambda_1}$$

outperforms all other choices $\alpha \in (0, \alpha^\star)$ and $\rho > 0$.

Several comments are in order:

(1) While **C1** provides the optimal ADMM parameters, **C2** and **C3** suggest sub-optimal choices. Moreover, by inspecting the values of $\alpha^\star$, it turns out that in all cases the over-relaxed ADMM algorithm (the ones with values of $\alpha > 1$) outperform the standard iterations (with $\alpha = 1$).

(2) The best ADMM-parameters are characterized based on the smallest and second-largest generalized eigenvalues of the matrix pencil $(E^\top(2\Pi_{\mathcal{R}(F)} - I)E, E^\top E)$. In particular, for the edge-variable formulation, we have $E^\top E = B_O^\top \mathcal{W}_O B_O + B_I^\top \mathcal{W}_I B_I$ and $E^\top \Pi_{\mathcal{R}(F)} E = (B_O^\top \mathcal{W}_O + B_I^\top \mathcal{W}_I)\mathcal{W}^{-1}(\mathcal{W}_O B_O + \mathcal{W}_I B_I)$. For simplicity, if we pick symmetric edge-weights $\mathcal{W}_O = \mathcal{W}_I = \mathcal{W}/2$ then the matrix pencil becomes $(B_O^\top \mathcal{W} B_I + B_I^\top \mathcal{W} B_O, B_O^\top \mathcal{W} B_O + B_I^\top \mathcal{W} B_I) \triangleq (\mathcal{A}, \mathcal{D})$, where $\mathcal{A}$ and $\mathcal{D}$ are the weighted adjacency and weighted degree matrices, respectively. The generalized eigenvalues of $(\mathcal{A}, \mathcal{D})$ are highly related to the eigenvalues of the normalized graph Laplacian. In fact, for certain $\mathcal{G}$ there exist analytical expressions characterizing λ_i for this formulation. For example, a path with $n \geq 4$ and a cycle topology with $n \geq 5$ satisfy the condition of **C2**. A complete graph and a complete bipartite graph belong to **C3**. Note that we assume that $\mathcal{W} \succeq 0$ is chosen so that $\mathcal{G}$ is connected and $\mathcal{D} = \kappa Q$.

(3) On the other hand, for the node-variable formulation, we have $E^\top E = B_O^\top \mathcal{W}_O B_O + B_I^\top \mathcal{W}_I B_I$ and

$$E^\top \Pi_{\mathcal{R}(F)} E = (B_O^\top \mathcal{W}_O B_I + B_I^\top \mathcal{W}_I B_O)$$
$$(B_I^\top \mathcal{W}_O B_I + B_O^\top \mathcal{W}_I B_O)^{-1}(B_I^\top \mathcal{W}_O B_O + B_O^\top \mathcal{W}_I B_I).$$

Similarly to the previous case, if we apply symmetric edge-weights $\mathcal{W}_O = \mathcal{W}_I = \mathcal{W}/2$ then $E^\top \Pi_{\mathcal{R}(F)} E = \mathcal{A}\mathcal{D}^{-1}\mathcal{A}$, the matrix pencil for the node-variable formulation becomes $(2\mathcal{A}\mathcal{D}^{-1}\mathcal{A} - \mathcal{D}, \mathcal{D})$.

(4) Considering the convergence factor $|\phi_{2n-1}|$ derived in above cases and the way it relates to λ_1 and λ_{n-1}, we can further decrease its quantity by optimizing the weight matrix

$\mathcal{W}$. Unfortunately, a unique weight optimization problem can not be formulated for all cases and also optimizing the weight for one case may results in changes in λ_1 and λ_{n-1} in a way that it falls into another case. However, good heuristics as the ones presented next can still be applied.

Lemma 3: Let $E = \bar{W}\bar{E}$ and $F = \bar{W}\bar{F}$ with $\bar{W} \triangleq \text{diag}([R_O \;\; R_I]^\top)$ being a diagonal weighting matrix and $\bar{E} \triangleq [B_O^\top \;\; B_I^\top]^\top$ and $\bar{F} \triangleq -[I \; I]^\top$, $\bar{F} \triangleq -[B_I^\top \;\; B_O^\top]^\top$ for edge-variable and node-variable formulations, respectively. Let P be an orthonormal basis spanning the range space of $V^\top$ where $V \triangleq (I - \Pi_{\mathcal{R}(\bar{F})})\bar{E}$ and denote $\mathcal{S}_w$ as the sparsity pattern imposed by $\bar{W}$. Then the weight matrix $W = \bar{W}^2 = \text{diag}([\mathcal{W}_O \;\; \mathcal{W}_I]^\top) \in \mathcal{S}_w$ that minimizes λ_{n-1} the second largest generalized eigenvalue that satisfies Assumption 1 is given by the following quasi-convex optimization problem

$$\begin{aligned} \underset{t,W}{\text{minimize}} \quad & t \\ \text{s.t.} \quad & W \in \mathcal{S}_w,\, W \succ 0,\, \mathbf{1}^\top \bar{E}^\top W \bar{E}\mathbf{1} = \mathbf{1}^\top Q \mathbf{1}, \\ & \begin{bmatrix} (1+t)P^\top \bar{E}^\top W \bar{E} P & P^\top \bar{E}^\top W \bar{F} \\ \bar{F}^\top W \bar{E} P & \frac{1}{2}\bar{F}^\top W \bar{F} \end{bmatrix} \succ 0. \end{aligned} \tag{11}$$

The above lemma is particularly relevant for case **C1** where the optimal convergence factor $|\phi_{2n-1}|$ is obtained by minimizing λ_{n-1}. We notice that under the mapping $X \mapsto 1/2(I+X)$ with $X = (E^\top E)^{-1}E^\top(2\Pi_{\mathcal{R}(F)} - I)E$ we have $\lambda_i \in [0,1]$ without changing the solution to the original problem. In the next section, we show that applying Lemma 3 with the aforementioned transformation and then using the optimal ADMM-parameters for case **C1** in Theorem 1 offers a nice heuristic that outperforms state-of-the-art algorithms for the node-variable formulation.

For the edge-variable formulation and symmetric weights $\mathcal{W}_O = \mathcal{W}_I = \mathcal{W}/2$, we are also able to maximize a lower-bound of the smallest generalized eigenvalue of $(E^\top \Pi_{\mathcal{R}(\bar{F})} E, E^\top E) = (\mathcal{A}, \mathcal{D})$. This is particularly useful if we note that an increased λ_1 in cases **C2** and **C3** leads to a decreased convergence factor. If we further minimize an upper bound on λ_{n-1} using a similar technique as in Lemma 3, we have heuristics for all cases of Theorem 1.

Lemma 4: Consider the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and associated non-negative weights $\mathcal{W} = \{w_{(u,v)}\}$ such that $\mathcal{G}$ is connected. Moreover, let P be the orthonormal basis spanning the null-space of $\mathbf{1}^\top$. The weights $\{w_{(u,v)}\}$ that jointly minimize and maximize the second largest and smallest generalized eigenvalues of $(\mathcal{A}, \mathcal{D})$, respectively, are given via the following quasi-convex problem

$$\begin{aligned} \underset{t,\{w_{(u,v)}\}}{\text{minimize}} \quad & t \\ \text{s.t.} \quad & w_{(u,v)} \geq 0, \;\; \forall u,v \in \mathcal{V}, \\ & \mathcal{A}_{uv} = w_{(u,v)}, \;\; \forall (u,v) \in \mathcal{E}, \\ & \mathcal{A}_{uv} = 0, \;\; \forall (u,v) \notin \mathcal{E}, \\ & \mathcal{D} = \text{diag}(\mathcal{A}\mathbf{1}), \\ & \mathcal{D} - \mathcal{A} + \mathbf{1}\mathbf{1}^\top \succ 0, \\ & P^\top(\mathcal{A} - t\mathcal{D})P \prec 0, \\ & \mathcal{A} + t\mathcal{D} \succ 0. \end{aligned} \tag{12}$$

Having the optimal weights obtained by either of the above weight optimization procedures, one may apply Lemma 2 and then Theorem 1 to derive the optimal ADMM-parameters for the distributed averaging algorithms (3) and (6). At this stage, our weight optimization procedures rely on centralized information and, hence, do not admit an immediate distributed implementation. However, in the next section we still use these weight optimization techniques to compare the best achievable performance of different averaging algorithms.

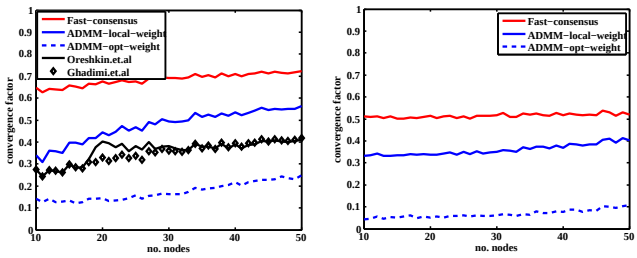
IV. Numerical Experiments

In this section we conduct numerical experiments to evaluate our parameter selection rules and compare the performance of ADMM formulations to the state-of-the-art algorithms for distributed averaging proposed in the literature. In the first experiment, we compare the convergence factors of different methods for the class of Random Geometric Graphs (RGG). In particular, for each simulation instance, n nodes were randomly deployed in the unit square and in order to guarantee the connectivity with high probability, we considered an edge between each pair of nodes if their distance is at most $\sqrt{2\log(n)/n}$ [21].

Fig.1 presents Monte-Carlo simulations of the convergence factors versus the number of nodes $n = [10,\, 50]$. Each data point of the plot is the average of the convergence factors of 50 instances of the randomly generated graphs with the same number of nodes. In the edge-variable scenario, we compare the ADMM iterates to Fast-consensus [14] from the ADMM literature and two recent accelerated consensus schemes: Oreshkin et al. [11] and Ghadimi et al. [22]. These algorithms include a two-tap memory mechanism in which the values of two-last iterates are taken into account in computing the next iterate. We solve the weight optimization problem (12) to find the optimal weights for our ADMM iterates whereas for the alternative algorithms we apply their best known weights. Finally, the ADMM-local-weights algorithm implements the local weights $\mathcal{W}_{v,(v,j)} = 1/\mathcal{N}_v$ and uses the optimal step-size and relaxation parameters derived in Theorem 1.

For the node-variable formulation, we compare the performance of the ADMM method with the local and the optimized weight scheme (11) to the Fast-consensus algorithm. Recall that while each iterate of the edge-variable based algorithms require single message passing within the neighborhood of each node, the node-variable formulation requires two message exchanges between nodes in each iteration. In all scenarios we observe that the ADMM algorithms with our tuning rules outperform the alternatives.

In a second experiment, we compare the performance of different averaging algorithms under fully distributed implementations, where all the algorithm parameters are either computed or estimated in a distributed fashion. For this aim, $n = 200$ nodes were deployed in an RGG topology with initial values $x^0(i) = i/n$. The ADMM (edge-variable) algorithm with local weights (9) is compared to the traditional linear averaging algorithm [12], Fast-consensus and Oreshkin et al., all with Metropolis-Hastings (MH) weight



(a) RGG edge variable (b) RGG node variable

Fig. 1. Performance comparison of the proposed ADMM algorithms with the state-of-the-art algorithms.

matrices. While our weight matrix is constructed locally with each node only knowing its own degree, in MH weights each node needs to know its neighbors' degrees as well. We restricted our method and Oreshkin et al. to the case **C1** (Oreshkin et al. also required a similar condition) by using the aforementioned mapping $X \mapsto 1/2(I + X)$ that can be computed locally. As a result, to compute the algorithm parameters for all methods, one has to compute the second largest eigenvalue of the corresponding weight matrices. The parameter can be obtained by a distributed power method scheme presented in [11]. In particular, nodes iterate 50 consensus rounds to compute the estimated parameter. We neglected this initialization cost to conduct the experiment. Fig. 2 compares the MSE decay rate of different algorithms

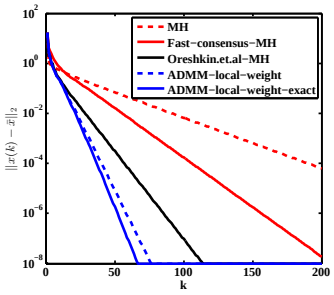


Fig. 2. MSE versus iteration number for $n = 200$ nodes in RGG topology.

versus the number of iterations. The ADMM algorithm with the exact knowledge of the second largest eigenvalue is also included as a reference. The figure indicates that our design rules outperform the alternatives.

V. Conclusions and Future Work

We address the optimal parameter selection of the ADMM method for distributed averaging. Two formulations that yield ADMM iterations which can be executed in a distributed manner were considered. For each formulation, we derived the step-size and relaxation parameters that minimize the convergence factor of the iterates. Under mild assumptions on the communication graph, analytical expressions for the optimal parameters were derived and related to the spectral properties of the communication graph. Supposing the optimal constant parameters were chosen, the convergence factor was further minimized by optimizing the edge weights. Numerical examples confirmed significant performance improvements over the state-of-the-art techniques. As a future work, we plan to extend the results to account for directed communication graphs.

References

[1] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, Sept 2004.

[2] M. Cao, A. Morse, and B. D. O. Anderson, "Agreeing asynchronously," *Automatic Control, IEEE Transactions on*, vol. 53, no. 8, pp. 1826–1838, Sept 2008.

[3] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-raphson consensus for distributed convex optimization," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, Dec 2011.

[4] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. IEEE, 2005, pp. 63–70.

[5] S. Kar and J. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *Signal Processing, IEEE Transactions on*, vol. 57, no. 1, pp. 355–369, Jan 2009.

[6] I. Schizas, G. Giannakis, S. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc wsns with noisy links-part ii: Distributed estimation and smoothing of random signals," *Signal Processing, IEEE Transactions on*, vol. 56, no. 4, pp. 1650–1666, April 2008.

[7] S. Patterson, B. Bamieh, and A. El Abbadi, "Convergence rates of distributed average consensus with stochastic link failures," *Automatic Control, IEEE Transactions on*, vol. 55, no. 4, pp. 880–892, 2010.

[8] S.-J. Kim, E. Dall'Anese, and G. Giannakis, "Cooperative spectrum sensing for cognitive radios using kriged kalman filtering," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 1, pp. 24–36, Feb 2011.

[9] D. Thanou, E. Kokiopoulou, Y. Pu, and P. Frossard, "Distributed average consensus with quantization refinement," *Signal Processing, IEEE Transactions on*, vol. 61, no. 1, pp. 194–205, Jan 2013.

[10] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1984.

[11] B. Oreshkin, M. Coates, and M. Rabbat, "Optimization and analysis of distributed averaging with short node memory," *Signal Processing, IEEE Transactions on*, vol. 58 Issue: 5, pp. 2850 –2865, 2010.

[12] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53 Issue: 1, pp. 65–78, 2004.

[13] B. Johansson, "On distributed optimization in networked systems," Ph.D. dissertation, KTH, Stockholm, Sweden, 2008.

[14] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," *Signal Processing, IEEE Transactions on*, vol. 59, pp. 5523 –5537, 2011.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3 Issue: 1, pp. 1–122, 2011.

[16] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *ArXiv e-prints*, 2012.

[17] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Rice University CAAM Technical Report, Tech. Rep., 2012.

[18] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Explicit convergence rate of a distributed alternating direction method of multipliers," *ArXiv e-prints*, 2013.

[19] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," *ArXiv e-prints*, 2013.

[20] A. Teixeira, E. Ghadimi, I. Shames, H. Sandberg, and M. Johansson, "Optimal scaling of the admm algorithm for distributed quadratic programming," *ArXiv e-prints*, 2014.

[21] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, 2000.

[22] E. Ghadimi, I. Shames, and M. Johansson, "Multi-step gradient methods for networked optimization," *Signal Processing, IEEE Transactions on*, vol. 61, no. 21, pp. 5417–5429, 2013.