Distributed Weight Selection in Consensus Protocols by Schatten Norm Minimization

Mahmoud El Chamie, Giovanni Neglia, and Konstantin Avrachenkov*

***Abstract*—This paper studies the weight optimization problem for average consensus protocols by reformulating it as a Schatten norm minimization with parameter p. We show that as p approaches infinity, the optimal solution of the Schatten norm induced problem recovers the optimal solution of the original problem. Moreover, by tuning the parameter p in our proposed minimization, we can simply trade-off the quality of the solution (i.e., the speed of convergence) for communication/computation requirements (in terms of number of messages exchanged and volume of data processed). We then propose a distributed algorithm to solve the Schatten norm minimization and we show that it outperforms the other distributed weight selection methods.**

I. Introduction

A network is formed of nodes (or agents) and communication links that allow these nodes to share information and resources. We consider each node i in the network to store in a local memory a scalar $x_i(0) \in \mathbb{R}$, called the estimate. Average consensus protocols are algorithms that can allow nodes to reach consensus on the average of all initial estimates ($\frac{\sum_i x_i(0)}{n}$) relying only on neighbor to neighbor communication. Consensus algorithms are used in many applications and distributed control problems for different and various systems [1], [2], and [3]. For a complete overview of consensus protocols we refer the reader to [4], [5], [6], [7] and the references therein.

An iterative algorithm for achieving the average consensus is the following: at iteration $k+1$, node i updates its state value x_i as follows:

$$x_i(k+1) = w_{ii}x_i(k) + \sum_{j \in N_i} w_{ij}x_j(k), \qquad (1)$$

where N_i is the set neighbors of node i, w_{ij} is the weight selected by node i for the value sent by its neighbor j and w_{ii} is the weight selected by node i for its own value. We can put the weights in an n by n matrix W where n is the number of nodes in the network. A necessary and sufficient condition for system (1) to converge starting from any initial condition [8] is to select W such that $W\mathbf{1} = \mathbf{1}$, $\mathbf{1}^T W = \mathbf{1}^T$, and $\mu(W) < 1$ where $\mathbf{1}$ is the vector of all ones and $\mu(W)$ is the second largest eigenvalue of W in module.

Xiao and Boyd in [8] formulated a general Fastest Distributed Linear Averaging (FDLA) problem as a non-convex optimization to find W that guarantees the fastest convergence. They showed that if the solution space is restricted to symmetric weight matrices, then the symmetric FDLA problem can be formulated as a Semi-Definite Program (SDP) that can be solved by a centralized unit using interior point methods. Kim *et al.* in [9] approximate the general FDLA using the qth-order spectral norm minimization (q-SNM). For a symmetric weight matrix, solving the q-SNM is equivalent to solving the symmetric FDLA problem. Their algorithm is centralized and has a higher complexity than the SDP. Solving the problem of optimal weight selection in a distributed way is still an open problem.

In this paper we study distributed techniques to optimally select the weights of average consensus protocols. We address the problem of selecting the weights in a given network in order to have a fast speed of convergence for these protocols. We approximate the problem of optimal weight selection by the minimization of the Schatten p-norm of a matrix with some constraints related to the connectivity of the underlying network. We then provide a totally distributed gradient method to solve the Schatten norm optimization problem. By tuning the parameter p in our proposed minimization, we can simply trade-off the quality of the solution (i.e., the speed of convergence) for communication/computation requirements (in terms of number of messages exchanged and volume of data processed). The theoretical contribution of this paper is as follows: we formulate a new optimization problem (the Schatten norm minimization) for weight selection problem of average consensus, then we show that i) the formulated problem is an approximation of the optimal weight selection problem, and ii) it can be implemented in a totally decentralized fashion. Simulation results on random graphs and on real networks show that our approach provides very good performance already for values of p that only need limited information exchange.

The paper is organized as follows: In section II we introduce the notation used across the paper. In section III we propose Schatten p-norm minimization as an approximation of the original problem and in section IV we show how its solution can be computed in a distributed way and evaluate its computation and communication costs. Section V compares the performance of our algorithm and that of other known weight selection algorithms on different graph topologies. Section VI summarizes the paper.

II. Notation

The network is considered as an undirected graph $G = (V, E)$ where V is the set of nodes ($V = \{1, \ldots, n\}$), E is

*The authors are with INRIA Sophia Antipolis - Méditerranée, 2004 Route des Lucioles, B.P. 93, 06902 Sophia Antipolis, France, Emails: {mahmoud.el_chamie, giovanni.neglia, k.avrachenkov}@inria.fr

the set of edges, $(i,j) \in E$ if nodes i and j are neighbors and can communicate ($E = \{1,\dots,m\}$, i.e., we label the edges from 1 to m). If link (i,j) has label l, we write $l \sim (i,j)$. We denote by $\mathbf{w}$ the vector of dimensions $m \times 1$, whose l-th element w_l is the weight associated to link l, then if $l \sim (i,j)$ it holds $w_l = w_{ij} = w_{ji}$. A is the adjacency matrix of graph G, i.e., $a_{ij} = 1$ if $(i,j) \in E$ and $a_{ij} = 0$ otherwise. $\mathcal{C}_G$ is the set of all real $n \times n$ matrices M corresponding to graph G, i.e., $m_{ij} = 0$ if $(i,j) \notin E$. D is a diagonal matrix where d_{ii} (or simply d_i) is the degree of node i in the graph G. Q is the $n \times m$ incidence matrix of the graph, such that for each $l \sim (i,j) \in E$, $Q_{il} = +1$ and $Q_{jl} = -1$ and the rest of the elements of the matrix are null. L is the laplacian matrix of the graph, so $L = D - A$. It can also be seen that $L = QQ^T$. The $n \times n$ identity matrix is denoted by I_n. Given that W is real and symmetric, it has real eigenvalues (and then they can be ordered). We denote by λ_i the i-th largest eigenvalue of W, and by μ the largest eigenvalue in module non considering λ_1, i.e., $\mu = \max\{\lambda_2, -\lambda_n\}$. Let σ_i be the i-th largest singular value of a matrix, $\mathrm{Tr}(X)$ be the trace of the matrix X and $\rho(X)$ be its spectral radius. $||X||_{\sigma p}$ denotes the Schatten p-norm of matrix X, i.e., $||X||_{\sigma p} = (\sum_i \sigma_i^p)^{1/p}$. Finally we use the symbol $\frac{\mathrm{d}}{\mathrm{d}X} f(X)$, where f is a differentiable scalar-valued function $f(X)$ with matrix argument $X \in \mathbb{R}^{m \times n}$, to denote the $n \times m$ matrix whose (i,j) entry is $\frac{\partial f(X)}{\partial x_{ji}}$.

III. Schatten Norm Minimization

The problem of finding the weight matrix that guarantees the fastest convergence can be formalized as follows (see [8]):

$$\begin{array}{ll} \underset{W}{\text{Argmin}} & \mu(W) \\ \text{subject to} & W = W^T,\ W\mathbf{1} = \mathbf{1},\ W \in \mathcal{C}_G, \end{array} \tag{2}$$

where the last constraint on the matrix W is derived from the assumption that nodes can only communicate with their neighbors and then necessarily $w_{ij} = 0$ if $(i,j) \notin E$. Problem (2) is called in [8] the "symmetric FDLA problem".

The above minimization problem is a convex one and the function $\mu(W)$ is non-smooth convex function. It is convex since when W is a symmetric matrix, we have $\mu(W) = \rho(W - J) = ||W - J||_2$ where $J = \frac{1}{n}\mathbf{1}\mathbf{1}^T$, so $\mu(W)$ is a composition between an affine function and the convex matrix L-2 norm function [10]. The function $\mu(W) = \rho(W - J)$ is non-smooth since the spectral radius of a matrix is not differentiable at points where the eigenvalues coalesce [11]. The process of minimization itself in (2) tends to make them coalesce at the solution.

We change the original minimization problem in (2) by considering a different cost function that is a monotonic function of the Schatten Norm. The minimization problem we propose is the following one:

$$\begin{array}{ll} \underset{W}{\text{Argmin}} & f(W) = ||W||_{\sigma p}^p \\ \text{subject to} & W = W^T,\ W\mathbf{1} = \mathbf{1},\ W \in \mathcal{C}_G, \end{array} \tag{3}$$

where p is an even positive integer. The following result establishes that (3) is a smooth convex optimization problem and also it provides an alternative expression of the cost function in terms of the trace of W^p. For this reason we refer to our problem also as *Trace Minimization* (TM).

Proposition 1. *$f(W) = ||W||_{\sigma p}^p = Tr(W^p)$ is a scalar-valued smooth convex function on its feasible domain when p is an even positive integer.*

Proof. We have $\mathrm{Tr}(W^p) = \sum_{i=1}^n \lambda_i^p$. Since W is symmetric, its non-zero singular values are the absolute values of its non-zero eigenvalues [12]. If p is even, $\sum_{i=1}^n \lambda_i^p = \sum_{i=1}^n \sigma_i^p$. Therefore, $\mathrm{Tr}(W^p) = ||W||_{\sigma p}^p$.

The Schatten norm $||W||_{\sigma p}$ is a nonnegative convex function. Hence f is convex because it is the composition of a non-decreasing convex function—the function x^p where x is non-negative—and a convex function [10].

The function is also differentiable and we have

$$\frac{\mathrm{d}}{\mathrm{d}W}\mathrm{Tr}(W^p) = pW^{p-1}, \tag{4}$$

(see [13, p. 411]). □

We now illustrate the relation between (3) and the optimization (2). The following lemmas will prepare the result:

Lemma 1. *For any symmetric weight matrix W whose rows (and columns) sum to 1 and with eigevalues $\lambda_1(W) \geq \lambda_2(W) \geq \cdots \geq \lambda_n(W)$, there exist two integers $K_1 \in \{1,2,\dots n-1\}, K_2 \in \{0,1,2,\dots n-1\}$ and a positive constant $\alpha < 1$ such that for any positive even integer p we have:*

$$1 + \tau(W)^p K_1 \leq Tr(W^p) \leq 1 + \tau(W)^p(K_1 + K_2\alpha^p), \tag{5}$$

where

$$\tau(W) = \begin{cases} \rho(W) = \max\{\lambda_1(W), -\lambda_n(W)\} & \text{if } \rho(W) > 1, \\ \mu(W) = \max\{\lambda_2(W), -\lambda_n(W)\} & \text{if } \rho(W) \leq 1. \end{cases} \tag{6}$$

Proof. Due to space limits the proof is presented in [14]. □

We now show that the the Schatten-p optimization problem (3) gives a stable matrix.

Lemma 2. *Let us denote by $W_{(p)}$ the solution of the minimization problem* (3)*. If the graph of the network is connected then $\tau\left(W_{(p)}\right) < 1$ for p sufficiently large.*

Proof. Due to space limits the proof is presented in [14]. □

We are now ready to state our main results in the following propositions:

Proposition 2. *If the underlying graph is connected, then the solution of the Schatten Norm minimization problem* (3) *satisfies the consensus protocol convergence conditions for p sufficiently large. Moreover as p approaches ∞, this minimization problem is equivalent to the minimization problem* (2) *(i.e., to minimize the second largest eigenvalue $\mu(W)$).*

Proof. Due to space limits the proof is presented in [14]. □

Proposition 3. *The Schatten Norm minimization* (3) *is an approximation for the original problem* (2) *with a guaranteed error bound,*

$$|\mu(W_{(SDP)}) - \mu(W_{(p)})| \leq \mu(W_{(SDP)}) \times \epsilon(p),$$

where $\epsilon(p) = (n-1)^{1/p} - 1$ *and where* $W_{(SDP)}$ *and* $W_{(p)}$ *are the solutions of* (2) *and* (3) *respectively.*

Proof: Due to space limits the proof is presented in [14]. ∎

Remark: Comparing the results of Schatten Norm minimization (3) with the original problem (2), we observe that on some graphs the solution of problem (3) already for $p = 2$ gives the optimal solution of the original problem (2); this is for example the case for complete graphs.[1] However, on some other graphs, it may give a weight matrix that does not guarantee the convergence of the consensus protocol because the second largest eigenvalue is larger than or equal to 1 (the other convergence conditions are intrinsically satisfied). Nevertheless, a suitable projection of the matrix on the set of stable matrices can be done in distributed way (see our technical report [14] for more details).

IV. A Distributed Algorithm for Schatten Norm minimization

Given that problem (3) is smooth and convex, it can be solved by interior point methods which would be a centralized solution. In this section we are going to show a distributed gradient-type algorithm to solve problem (3). We call an algorithm distributed when each node only needs to retrieve information from a limited neighborhood (possibly larger than N_i) in order to calculate the weights on its incident links. The constraint $W = W^T$ in the optimization requires any two neighbors i and j to choose the same weight on their common link $l \sim (i,j)$ i.e., $w_{ij} = w_{ji} = w_l$. The last condition $W\mathbf{1} = \mathbf{1}$ means that at every node i the sum of all weights on its incident links plus its self-weight w_{ii} must be equal to one. This condition is satisfied if nodes choose first weights on links, and then adapt consequently their self-weights w_{ii}. Moreover these two constraints lead to the possibility to write W as follows: $W = I - Q \times \mathrm{diag}(\mathbf{w}) \times Q^T$, where $\mathbf{w} \in \mathbb{R}^m$ is the vector of all the weight links w_l, $l = 1...m$. It follows that Schatten Norm minimization (3) is equivalent to the following unconstrained problem:

$$\text{minimize } h(\mathbf{w}) = Tr\left((I - Q \times \mathrm{diag}(\mathbf{w}) \times Q^T)^p\right). \quad (7)$$

We will give a distributed algorithm to solve the Schatten Norm minimization (3) by applying gradient techniques to problem (7). Since the cost function to optimize is smooth and convex as we proved in Proposition 1, if the gradient technique converges to a stationary point, then it converges to the global optimum. The gradient method uses the simple iteration:

$$w_l^{(k+1)} = w_l^{(k)} - \gamma^{(k)} g_l^{(k)} \ \forall l = 1...m \ ,$$

where $\gamma^{(k)}$ is the stepsize at iteration k and $g_l^{(k)}$ is the l-th component of the gradient $\mathbf{g}^{(k)}$ of the function $h(\mathbf{w})$. At every iteration k, starting with a feasible solution for link weights, $w_l^{(k)}$, we calculate the gradient $g_l^{(k)}$ for every link, and then we obtain a new weight value $w_l^{(k+1)}$. There are different conditions on the function $h(.)$ and on the stepsize sequence that can guarantee convergence. A distributed computational model for optimizing a sum of non-smooth convex functions is proposed in [15], [16] and its convergence is proved for bounded (sub)gradients for different network dynamics. For a similar objective function, the authors in [17] study the convergence of a projected (sub)-gradient method with constant stepsize. For unbounded gradients, the algorithm in [18, Section 5.3.2, p. 140] guarantees global convergence but requires a centralized calculation of the stepsize sequence. Because the objective function in (7) has unbounded gradient, our distributed implementation combines ideas from unbounded gradients methods and the projecting methods using theorems from [19].

In particular, we will add a further constraint to (7), looking for a solution in a compact set X, and we will consider the following projected gradient method:

$$\mathbf{w}^{(k+1)} = P_X\left(\mathbf{w}^{(k)} - \gamma^{(k)}\mathbf{g}^{(k)}\right),$$

where $P_X()$ is the projection on the set X. We can show that by a particular choice of X and $\gamma^{(k)}$ the method converges to the solution of the original problem. Moreover, all the calculations can be performed in a distributed way on the basis of local knowledge. In particular, we will show that:

- nodes incident to l are able to calculate $g_l^{(k)}$ using only information they can retrieve from their (possibly extended) neighborhood;
- the stepsize sequence $\gamma^{(k)}$ is determined a priori and then nodes do not need to evaluate the function h or any other global quantity to calculate it;
- the projection on set X can be performed component-wise, and locally at each node;
- the global convergence of the projected gradient method is guaranteed.

We will start by g_l and show that it only depends on information local to nodes i and j incident to the link $l \sim (i,j)$, then we will discuss the choice of the stepsize $\gamma^{(k)}$ and of the projection set X.

A. Locally Computed Gradient

Consider the link $l \sim (i,j)$, since $w_l = w_{ij} = w_{ji}$ and $w_{ii} = 1 - \sum_{s \in N_i} w_{is}$, we have:

$$\frac{\mathrm{d}w_{st}}{\mathrm{d}w_l} = \begin{cases} +1 & \text{if } s = i \text{ and } t = j \\ +1 & \text{if } s = j \text{ and } t = i \\ -1 & \text{if } s = i \text{ and } t = i \\ -1 & \text{if } s = j \text{ and } t = j \\ 0 & \text{else.} \end{cases} \quad (8)$$

[1] This can be easily checked. In fact, for any matrix that guarantees convergence of average consensus protocol, it holds $\mu(W) \geq 0$ and $\mathrm{Tr}(W^2) \geq 1$ (because 1 is an eigenvalue of W). The matrix $\hat{W} = 1/n\mathbf{1}\mathbf{1}^T$ (corresponding to each link having the same weight $1/n$) has eigenvalues 1 and 0 with multiplicity 1 and $n-1$ respectively. Then $\mu(\hat{W}) = 0$ and $\mathrm{Tr}(\hat{W}^2) = 1$. It follows that $\hat{W}$ minimizes both the cost function of problem (2) and (3).

The gradient g_l of the function $h(\mathbf{w})$ for $l \sim (i,j)$ can be calculated as follows:

$$\begin{aligned} g_l &= \frac{\mathrm{d}h(\mathbf{w})}{\mathrm{d}w_l} = \frac{\mathrm{d}f(I - Q \times \mathrm{diag}(\mathbf{w}) \times Q^T)}{\mathrm{d}w_l} \\ &= \sum_{s,t} \frac{\partial f}{\partial w_{st}} \frac{\mathrm{d}w_{st}}{\mathrm{d}w_l} \\ &= \frac{\partial f}{\partial w_{ij}} \frac{\mathrm{d}w_{ij}}{\mathrm{d}w_l} + \frac{\partial f}{\partial w_{ji}} \frac{\mathrm{d}w_{ji}}{\mathrm{d}w_l} + \frac{\partial f}{\partial w_{ii}} \frac{\mathrm{d}w_{ii}}{\mathrm{d}w_l} + \frac{\partial f}{\partial w_{jj}} \frac{\mathrm{d}w_{jj}}{\mathrm{d}w_l} \\ &= \frac{\partial f}{\partial w_{ij}} + \frac{\partial f}{\partial w_{ji}} - \frac{\partial f}{\partial w_{ii}} - \frac{\partial f}{\partial w_{jj}} \\ &= p\big((W^{p-1})_{ji} + (W^{p-1})_{ij} - (W^{p-1})_{ii} - (W^{p-1})_{jj}\big). \end{aligned} \tag{9}$$

In the last equality we used equation (4).

It is well know from graph theory that if we consider W to be the adjacency matrix of a weighted graph G, then $(W^s)_{ij}$ is a function of the weights on the edges of the $i-j$ walks (i.e., the walks from i to j) of length exactly s (in particular if the graph is unweighted $(W^s)_{ij}$ is the number of distinct $i-j$ s-walks [20]). Since for a given p the gradient g_l, $l \sim (i,j)$, depends on the $\{ii, jj, ij, ji\}$ terms of the matrix W^{p-1}, g_l can be calculated locally by using only the weights of links and nodes at most $\frac{p}{2}$ hops away from i or j.[2] Practically speaking, at each step, nodes i and j need to contact all the nodes up to $p/2$ hops away in order to retrieve the current values of the weights on the links of these nodes and the values of weights on the nodes themselves. An advantage of our approach is that it provides a trade-off between locality and optimality. In fact, the larger the parameter p, the better the solution of problem (3) approximates the solution of problem (2), but at the same time the larger is the neighborhood from which each node needs to retrieve the information. When $p = 2$, then g_l where $l \sim (i,j)$ only depends on the weights of subgraph induced by the two nodes i and j. For $p = 4$, the gradient g_l depends only on the weights found on the subgraph induced by the set of vertices $N_i \cup N_j$.

B. Choice of Stepsize and Projection set

The global convergence of gradient methods (i.e., for any initial condition) has been proved under a variety of different hypotheses on the function h to minimize and on the step size sequence $\gamma^{(k)}$. In many cases the step size has to be adaptively selected on the basis of the value of the function or of the module of its gradient at the current estimate, but this cannot be done in a distributed way for the function $h(\mathbf{w})$. This leads us to look for convergence results where the step size sequence can be fixed ahead of time. Moreover the usual conditions, like Lipschitzianity or boundness of the gradient, are not satisfied by the function $h(.)$ over all the feasible set. For this reason we add another constraint to our original problem (7) by considering that the solution has to belong to a given convex and compact set X. Before further specifying how we choose the set X, we state our convergence result.

[2]If a link or a node is more than $p/2$ hops away both from node i and node j, then it cannot belong to a $i-j$ walk of length p.

Proposition 4. *Given the following problem*

$$\begin{aligned} &\textit{minimize} && h(\mathbf{w}) = Tr\left((I - Q \times diag(\mathbf{w}) \times Q^T)^p\right), \\ &\textit{subject to} && \mathbf{w} \in X \end{aligned} \tag{10}$$

where $X \subseteq \mathbb{R}^m$ is a convex and compact set, if $\sum_k \gamma^{(k)} = \infty$ and $\sum_k \left(\gamma^{(k)}\right)^2 < \infty$, then the following iterative procedure converges to the minimum of h in X:

$$\mathbf{w}^{(k+1)} = P_X\left(\mathbf{w}^{(k)} - \gamma^{(k)} \mathbf{g}^{(k)}\right), \tag{11}$$

where $P_X(.)$ is the projection operator on the set X and $\mathbf{g}^{(k)}$ is the gradient of h evaluated in $\mathbf{w}^{(k)}$.

Proof. The function h is continuous on a compact set X, so it has a point of minimum. Moreover also the gradient $\mathbf{g}$ is continuous and then bounded on X. The result then follows from Proposition 8.2.6 in [19, pp. 480]. □

For example, $\gamma^{(k)} = a/(b+k)$ where $a > 0$ and $b \geq 0$ satisfies the step size condition in Proposition 4.

While the convergence is guaranteed for any set X convex and compact, we have two other requirements. First, it should be possible to calculate the projection P_X in a distributed way. Second, the set X should contain the solution of the optimization problem (7). About the first issue, we observe that if X is the cartesian product of real intervals, i.e., if $X = [a_1, b_1] \times [a_2, b_2] \times \ldots [a_m, b_m]$, then we have that the l-th component of the projection on X of a vector $\mathbf{y}$ is simply the projection of the l-th component of the vector on the interval $[a_l, b_l]$, i.e.,

$$[P_X(\mathbf{y})]_l = P_{[a_l,b_l]}(y_l) = \begin{cases} a_l & \text{if } y_l < a_l, \\ y_l & \text{if } a_l \leq y_l \leq b_l, \\ b_l & \text{if } b_l < y_l. \end{cases} \tag{12}$$

Then in this case Eq. (11) can be rewritten component-wise as

$$w_l^{(k+1)} = P_{[a_l,b_l]}(w_l^{(k)} - \gamma^{(k)} g_l^{(k)}).$$

We have shown in the previous section that g_l can be calculated in a distributed way, then the iterative procedure can be distributed. About the second issue, we have from the bound of matrix norms (see [13])

$$||W||_\infty \leq \rho(W), \tag{13}$$

where $||W||_\infty = \max_{i,j} |w_{ij}|$. A consequence of inequality (13) is that if we choose $X = [-1, 1]^m$, we include in the feasibility set all the weight matrices with spectral radius at most 1, and in particular the the matrix solution of problem (2) (it satisfies the convergence conditions). The same is true for the solution of problem (7) for p large enough because of Proposition 2. The following proposition summarizes our results.

Proposition 5. *If the underlying graph is connected, then the following distributed algorithm converges to the solution of the Schatten norm minimization problem for p large enough:*

$$w_l^{(k+1)} = P_{[-1,1]}(w_l^{(k)} - \gamma^{(k)} g_l^{(k)}), \quad \forall l = 1, \ldots, m, \tag{14}$$

where $\sum_k \gamma^{(k)} = \infty$ and $\sum_k \left(\gamma^{(k)}\right)^2 < \infty$.

C. Complexity of the Algorithm

Our distributed algorithm for Schatten Norm minimization requires to calculate at every iteration, the stepsize $\gamma^{(k)}$, the gradient $g_l^{(k)}$ for every link, and a projection on the feasible set X. Its complexity is determined by the calculation of link gradient g_l, while the cost of the other operations is negligible. For $p = 2$, $g_l = 2 \times (2W_{ij} - W_{ii} - W_{jj})$, so the computational complexity for nodes is negligible, but the communication complexity is 1 message carrying a single real value (w_{ii}) per link, per node and per iteration. For $p = 4$, $g_l = 4\big((W^3)_{ij}+(W^3)_{ji}-(W^3)_{ii}-(W^3)_{jj}\big)$, and as discussed in the previous section, each of the 4 terms can be calculated only locally from the weights within 2-hops from i or j. The computational complexity for calculating g_l is in the worst case $O(\Delta^3)$ where Δ is the largest degree in the network. In order to calculate g_l locally, the communication complexity would be to send two messages per link per node and per iteration. The first message carries at most Δ values (the weight vector $\mathbf{W}_i$) and the second message carries one real value ($(W^3)_{ii}$). Due to space limit, we refer the reader to the technical report [14] for details on both computational and communication costs.

Remark: If p is larger than twice the diameter of the graph, then each agent requires the knowledge of all the link weights in the network. In this case, it is better to revert to the centralized version of the Schatten norm minimization.

V. Performance Evaluation

In this section we compare the speed of convergence of the average consensus protocol when the weight matrix W is the solution of the Schatten norm minimization, the solution of the FDLA problem, or it is selected according to other distributed approaches (see [21], [8]): max degree weights (MD) ($w_l = \frac{1}{\max_i\{d_i\}+1} \ \forall l = 1,\dots,m$), local degree (metropolis) weights (LD) ($w_l = \frac{1}{\max\{d_i,d_j\}+1}$ $l \sim (i,j) \ \ \forall l = 1,\dots,m$), and optimal constant weights (OC) ($w_l = \frac{2}{\lambda_1(L)+\lambda_{n-1}(L)} \ \ \forall l = 1,\dots,m$).

As we have discussed in section III, the speed of convergence is asymptotically determined by the second largest eigenvalue in absolute value ($\mu(W)$). The simulations are done on random graphs: Erdös-Renyi (ER) graphs and Random Geometric Graphs (RGG), given that they are connected. The random graphs are generated as following :

- For the ER random graphs, we start from n nodes fully connected graph, and then every link is removed from the graph with probability $1 - q$ and is kept with a probability q. We have tested the performance for different probabilities q given that the graph is connected.
- For the RGG random graphs, n nodes are thrown uniformly at random on a unit square area, and any two nodes within a connectivity radius r are connected by a link. We have tested the performance for different connectivity radii given that the graph is connected. It is known that for small connectivity radii the nodes tend to form clusters.

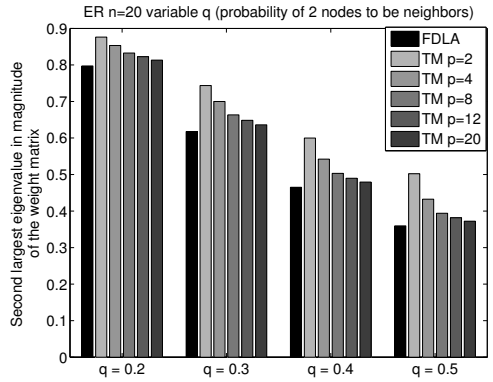

Fig. 1. Performance comparison between the optimal solution of the FDLA problem (labeled FDLA) and the approximated solutions obtained solving the Schatten Norm minimization for different values of p (labeled TM).

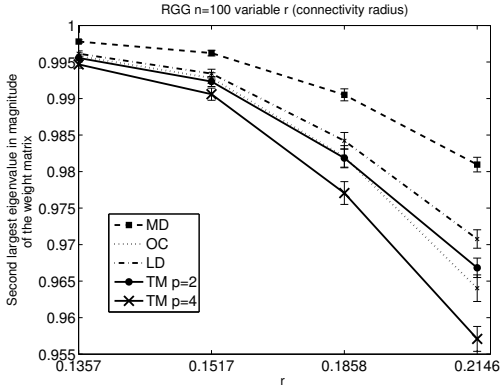

Fig. 2. Performance comparison between Schatten Norm minimization (TM) for $p = 2$ and $p = 4$ with other weight selection algorithms on RGG graphs.

A. Comparison with the optimal solution

We first compare $\mu\left(W_{(p)}\right)$ of the solution $W_{(p)}$ of the Schatten p-norm (or Trace) minimization problem (3) with its minimum value obtained solving the symmetric FDLA problem (2). To this purpose we used the `CVX` solver ([22]). This allows us also to evaluate how well problem (3) approximates problem (2) for finite values of the parameter p. The results in Fig. 1 have been averaged over 100 random graphs with 20 nodes generated according to the Erdos-Renyi (ER) model, where each link is included with probability $q \in \{0.2, 0.3, 0.4, 0.5\}$. We see from the results that as we solve the trace minimization for larger p, the asymptotic convergence speed of our approach converges to the optimal one as proven in Proposition 2.

B. Other distributed approaches: Asymptotic Convergence Rate

We compare now our algorithm for $p = 2$ and $p = 4$ with other distributed weight selection approaches. Fig. 2 shows the results on connected Random Geometric Graphs (RGG) with 100 nodes for different values of the connectivity radius r. We provide 95% confidence intervals by averaging each metric over 100 different samples. We see in Fig. 2

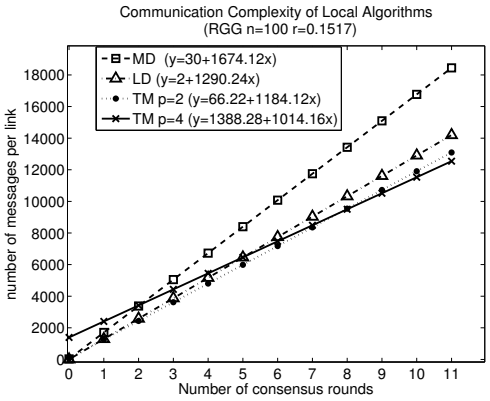

Fig. 3. Communication overhead of local algorithms for a different number of consensus rounds.

that on RGG for $p = 2$ and $p = 4$ the TM algorithm reaches faster convergence than the other known algorithms even when the graph is well connected (large connectivity radius). However, the larger the degrees of nodes, the higher the complexity of our algorithm. Interestingly even performing trace minimization for the smallest value $p = 2$ nodes are able to achieve faster speed of convergence than a centralized solution like the OC algorithm.

C. Communication Complexity of Local Algorithms

We have shown above that the weight matrix with minimum Schatten norm allows nodes to converge faster than the other heuristics, and then to exchange less messages, if a mechanism is implemented to stop consensus when estimates are close enough to the actual average. At the same time, the Schatten norm minimization algorithm may require itself a large number of messages to calculate the weights, while other local weight selection algorithms, like MD or LD, require a negligible communication exchange. In order to have a fair comparison, it is important then to consider on how many "consensus rounds" the additional communication overhead of our algorithm can be amortized.

In this section we perform such comparison on RGG with 100 nodes and connectivity radius 0.1517. The total communication overhead of the local algorithms is plotted in Fig. 3. The figure shows the total number of messages transmitted on a link, considering both those needed initially to calculate the weights and those needed to determine the average with a relative error from consensus precision (10^{-3}) for a different number of consensus rounds. The TM algorithms have high initial communication overhead (due to the slow convergence of the gradient method for weight calculation), but then the more the consensus rounds we have the more the messages are saved in comparison to the simpler methods. Note that the asymptotic results are reflected in the slopes of the lines. As the figure shows, if the network is used for more than 8 consensus rounds then $TM\ p = 4$ is recommended, while $TM\ p = 2$ starts outperforming LD and MD already for 2 consensus rounds.

VI. Conclusion

We have proposed in this paper an approximated solution for the FDLA problem by minimizing the Schatten p-norm of the weight matrix. Our approximated algorithm converges to the solution of the FDLA problem as p approaches infinity, and in comparison to it, has the advantage to be suitable for a distributed implementation. Moreover, simulations on random and real networks show that the algorithm outperforms other common distributed algorithms for weight selection.

References

[1] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2007.

[2] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Decision and Control, 2007 46th IEEE Conference on*, 2007, pp. 5492–5498.

[3] N. Elia and S. Mitter, "Stabilization of linear systems with limited information," *IEEE Transactions on Automatic Control*, vol. 46, no. 9, pp. 1384–1400, 2001.

[4] K. Avrachenkov, M. El Chamie, and G. Neglia, "A local average consensus algorithm for wireless sensor networks," in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, June 2011, pp. 1–6.

[5] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, pp. 1520–1533, September 2004.

[6] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Trans. Autom. Control*, vol. 54, November 2009.

[7] S. Kar and J. M. F. Moura, "Topology for global average consensus," *in 40th Asilomar Conference on Signals, Systems, and Computers*, October 2006.

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

[9] Y. Kim, D.-W. Gu, and I. Postlethwaite, "Spectral radius minimization for optimal average consensus and output feedback stabilization ," *Automatica*, vol. 45, no. 6, pp. 1379 – 1386, 2009.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, March 2004.

[11] M. K. Fan and B. Nekooie, "On minimizing the largest eigenvalue of a symmetric matrix," *Linear Algebra and its Applications*, vol. 214, pp. 225–246, 1995.

[12] C. D. Meyer, Ed., *Matrix analysis and applied linear algebra*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000.

[13] D. Bernstein, *Matrix mathematics: theory, facts, and formulas*. Princeton University Press, 2005.

[14] M. El Chamie, G. Neglia, and K. Avrachenkov, "Distributed Weight Selection in Consensus Protocols by Schatten Norm Minimization," INRIA, INRIA Research Report RR-8078, Oct 2012. [Online]. Available: http://hal.inria.fr/hal-00738249

[15] A. Nedić and A. Ozdaglar, "Distributed Subgradient Methods for Multi-Agent Optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan 2009.

[16] I. Lobel and A. Ozdaglar, "Distributed Subgradient Methods for Convex Optimization Over Random Networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1291–1306, June 2011.

[17] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *47th IEEE Conference on Decision and Control CDC 2008*, Dec 2008, pp. 4185–4190.

[18] B. Polyak, *Introduction to Optimization*. New York: Optimization Software, 1987.

[19] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[20] D. B. West, *Introduction to Graph Theory (2nd Edition)*. Prentice Hall, Aug. 2000.

[21] L. Xiao, S. Boyd, and S. jean Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, pp. 33–46, 2005.

[22] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 1.21," April 2011.