Weight Design for Consensus Algorithms in Static and Random Topologies: Finite Time Horizon

Dusan Jakovetić

Abstract—We address the problem of the weights design for consensus algorithms under random network topology. We differ two different cases: 1) high estimate precision is required, and there is no firm restriction on the number of available iterations; 2) there is only a small, limited budget of iterations available. For the first case, we show that minimizing the mean squared error contraction factor is a convex optimization problem and we globally solve it by the subgradient algorithm.

For the case of limited budget of iterations, we consider both static and random topologies. For static topology, we define the objective as to minimize the sum of $k$ largest eigenvalues of the consensus error contraction matrix. We show that the proposed optimization problem is convex which enables finding globally optimal solution. Further, it has been discovered that there is a trade off for the choice of the parameter $k = 1, ..., N$: larger $k$ yields faster convergence in the transient regime and slower long run convergence. Thus, the parameter $k$ should be tuned according to a specific application and requirements. All results are extended to the case of random topology, and the validity of the proposed weight choice is verified through simulations.

I. INTRODUCTION

Consensus algorithm is a distributed averaging algorithm for networked multi agent systems. It appears as a canonical problem in many applications, e.g., distributed data fusion [1], [2], distributed optimization algorithms [3], flocking [4], etc.

In wireless sensor networks (WSNs), consensus algorithm operates under random network topology. Packet losses or drops may occur randomly. Also, a subset of sensors is usually kept in the sleep mode in order to save power. In our model, we assume that the network links are random switches, i.e., the link may be online or off line at a time step with probabilities $P_{ij}$ and $1 - P_{ij}$, respectively. Consensus under random or time varying topologies has been extensively studied in the literature [5], [6], [7], [8], [9], [10].

It has been observed in the literature that the convergence rate of the consensus algorithm depends significantly on the choice of the weights [11]. Fast convergence of the algorithm is of great importance, since fast convergence implies reducing the amount of communication and thus saving power. In this paper we address the problem of finding the weights that maximize the consensus algorithm convergence speed. The paper consists of the two parts: the first part considers the consensus algorithm under random topology when there is no firm restriction on the budget of the available iterations (communication cycles). More detailed treatment of this problem can be found in [12].

In the second part of the paper, we extend the mentioned work to the case when only a small budget of iterations is available. We refer to this scenario as finite time horizon case.

We propose several objective functions for finding the weights. Further, we show that the proposed functions are convex, and thus the optimal weights can be found in efficient way. Simulation examples demonstrate that the resulting weights lead to faster convergence than the choices previously proposed in the literature. Also, we show that among the cost function choices that we propose, there exists a trade off in the following sense: better performance in the transient regime leads to the worse performance in the asymptotic regime and vice versa. Thus, depending on the requirements, one can choose the appropriate cost function to achieve the desired performance.

Paper organization. In the next paragraph, notation used throughout the paper is introduced. Section II describes the problem model. In section III we address the problem of weight optimization for the case of large budget of iterations. In section IV we consider the case when only a small budget of iterations is available, for static topologies. Section V studies the same scenario for random topologies. Finally, section VI concludes the paper.

Notation. Constant matrices are denoted by capital letters (e.g., $X$) and random matrices are denoted by calligraphic letters (e.g., $X$). Columns and entries of matrix $X$ are denoted by $X_i$ and $X_{ij}$, respectively. Quantities $X \otimes Y$, $X \odot Y$, $X \oplus Y$ denote the Kronecker product, the Hadamard product and the direct sum of the matrices $X$ and $Y$, respectively. The matrix $\text{diag} (X)$ is diagonal matrix with the diagonal equal to the direct sum of the matrices $X$ and $Y$, respectively. The matrix $\text{diag} (X)$ is diagonal matrix with the diagonal equal to the diagonal of $X$. Vectors (random and deterministic) are denoted with lower case letters (e.g., $x$). Inequality $X \succeq Y$ means that the matrix $X - Y$ is positive semidefinite and $X \succeq Y$ or $x < y$ is understood entry wise. Quantities $\lambda_{\text{max}} (X)$ and $r(X)$ denote the maximal eigenvalue and the spectral radius of $X$, respectively. The identity matrix is denoted by $I$. Matrix $\text{diag} (x_1, ..., x_N)$ denotes the diagonal $N \times N$ matrix with the $i$-th diagonal entry equal to scalar $x_i$. The $N$-dimensional column vector of ones is denoted with 1. Symbol $J = \frac{1}{N} 1 1^T$. The $i$-th canonical unit vector, i.e., the $i$-th column of $J$, is denoted by $e_i$. Symbol $|S|$ denotes the cardinality of a set $S$.

II. PRELIMINARIES AND ASSUMPTIONS

A. Network connectivity model

The maximal realizable network, i.e., the network that contains all realizable links, is modeled by the supergraph. The supergraph $G$ is a graph $(V, E)$ where $V$ is the set of sensors ($|V| = N$) and $E$ is the set of edges or realizable communication channels. We refer to $E$ as the superset ($|E| = M$). Supergraph $G$ is assumed to be connected and...
the quantities \( \Omega_i(k) \) are completely described by \( A(k) \).

The weights rule for the consensus algorithm (2) becomes:

\[
W_{ij}(k) = \begin{cases} 
W_{ij} & \text{if } j \in \Omega_i(k) \\
1 - \sum_{j \in \Omega_i(k)} W_{ij}(k) & \text{if } i = j \\
0 & \text{otherwise.}
\end{cases}
\]

In view of (4) we rewrite (2) in compact form:

\[
x(k + 1) = W(k)x(k) \\
W(k) = I - \text{diag}(W.A(k)) + W \odot A(k)
\]

We calculate closed form expressions for the expected value of \( W \) and the expected value of the square of \( W \) because they will play an important role in the convergence rate of the consensus algorithm.

**Lemma 1** Denote \( E \left[ W \right] = \overline{W} \). Then:

\[
\begin{align*}
E \left[ W^2 \right] &= \overline{W}^2 + R_C \\
\frac{1}{2} R_C &= \text{diag} \left\{ ((11^T) \odot P) (W \odot W) \right\} \\
&\quad - (11^T) \odot P \odot W \odot W
\end{align*}
\]

Derivation of the expressions in Lemma 1 can be found in [12].

### III. Weight optimization: large budget of iterations

In this section, we consider the case when a large number of iterations of the consensus algorithm is available. This setup is appropriately suited for the case when the primal goal is to achieve high precision (small error), and the communication constraints are not firm.

We introduce the optimization problem in steps. In III-A we give the expression for the mean squared convergence rate \( \rho(W) \) that we want to minimize. In III-B we give the conditions for m.s.s. and a.s. convergence that define the constraint set for \( W \). The final problem introduced in III-C is unconstrained.

#### A. Objective function: Mean square convergence rate

Define the consensus error vector \( e(k) \), the error covariance matrix \( \Sigma(k) \), and the mean squared error \( E \left[ e(k)^T e(k) \right] \) as

\[
e(k) = x(k) - x_{\text{avg}} \]  
\[
\Sigma(k) = E \left[ e(k)^T e(k) \right] 
\]

The next Lemma states that the mean squared error decays at the rate \( \rho(W) \) given by

\[
\rho(W) = \lambda_{\text{max}} \left( E[|W|^2] - J \right) 
\]
Lemma 2 (m.s.s convergence rate) Consider the consensus algorithm given by eqn. (5). Then:
\[ \text{tr} (\Sigma(k)) \leq (\rho(W))^k \text{tr} (\Sigma(0)) \]  
(7)

The proof can be found in [12], and similar result for the case of gossip algorithm can be found in [13].

B. Constraint set: Convergence conditions for consensus algorithm

From Lemma 2, it can be seen that $\rho(W) < 1$ is a sufficient condition for the m.s.s. convergence of consensus (5). We state this as a theorem.

Theorem 3 (m.s.s. convergence) Assume $\rho(W) < 1$. Then the sequence $x(k)$ generated by the consensus (5) converges to $x_{avg}$ in the mean squared sense.

It turns out that $\rho(W) < 1$ is also a sufficient condition for almost sure convergence. The proof can be derived involving the theory of stationary ergodic random matrix sequences (for the proof see [12]), in particular, by the Fuerstenberg-Kersten theorem [14] and the Theorem 5 from the reference [10].

Theorem 4 (a.s. convergence) Assume $\rho(W) < 1$. Then the sequence $x(k)$ generated by the consensus algorithm (5) converges to $x_{avg}$ almost surely.

The set
\[ S_{\text{conv}} = \{ W \in S_W : \rho(W) < 1 \} \]
(8)
defines the set of matrices $W$ that are of interest.

C. Weight optimization problem formulation

Now we state the weight optimization problem that we solve. The weights that maximize the mean square convergence rate are the solution of

\[ \begin{array}{l}
\text{minimize} & \rho(W) \\
\text{subject to} & W \in S_W 
\end{array} \]  
(9)

$S_W$ is defined in eqn. (3) and $\rho(W)$ is given by (6). We note that, although the set of interest is $S_{\text{conv}}$, we can search over all the matrices $W \in S_W$, resulting with the unconstrained problem (9).

Since there is no any other constraint on $W$ except the sparsity structure of the supergraph, some entries of $W$ may be negative. Moreover, we observed in simulation examples that it happens that some of the entries for the optimal $W$ are negative. In [8] it has been shown that under the assumption of the connected supergraph, and all the realizations of $W$ being stochastic with positive diagonal, consensus algorithm converges a.s. However, the requirement of all realizations of $W$ being stochastic for the consensus rule (4) implies a conservative restriction to the weight matrices $W$. This restriction might preclude us from finding the truly optimal $W$. We ask the reader also to refer to the Section 4, Concluding remarks, in the reference [7].

It is straightforward to check that, in the case of static topology, the optimization problem (9) and the optimization problem (14) in reference [11] are equivalent.

D. Weight optimization problem: solution

It can be shown that the function $\rho(W)$ is convex, which implies that the optimization problem (9) is convex, unconstrained. We solve numerically (9) by the subgradient algorithm.

Lemma 5 (Convexity of $\rho(W)$) Function $\rho : S_W \rightarrow \mathbb{R}^+$ is convex.

Proof can be found in [12].

E. Simulation example

We illustrate the performance of the proposed solution with a simulation example. We consider $N = 120$ nodes. Average degree in the supergraph is 20. The network is modeled as a geometric random graph. Nodes are randomly placed on a unit square, and the edge superset $E$ is defined in the following way: $\{i,j\} \in E$, if $\Delta_{ij} < r$, where $r = 0.3$ and $\Delta_{ij}$ denotes the Euclidean distance between nodes $i$ and $j$. For any edge in the superset $E$, we model the link formation probability $P_{ij}$ in the following way: $P_{ij} = 1 - c(\Delta_{ij}/r)^2$, where $c = 0.5$, for this example.

We find the opweights by solving (9). We refer to our weight matrix solution as PBW (probability based weights). We compare PBW to the following different choices: 1) Metropolis weights, proposed in [11]; 2) SGBW (supergraph-based weights), proposed in [10]. We estimate the expected value of the error vector norm $E(||e(k)||)$ versus the iteration number $k$ by Monte Carlo simulations as

\[ E(||e(k)||) = \frac{1}{S} \sum_{s=1}^{S} ||e(k,s)||. \]
(10)

Quantity $e(k,s)$ is the error vector at iteration $k$ for the sample path $s$. The number of sample paths is $S = 100$. Obtained results are depicted in Figure 1.

PBW outperform MW and SGBW for the high precision requirement. For example, if we want that the estimate goes below 1% of the initial error, we need 37 iterations for PBW, 60 iterations for SGBW and 75 for MW.

However, we remark that in the first several iterations, MW perform better than PBW. This fact suggests that if we are limited only to the first several iterations, we should try to look for an alternative weight choice. In this paper we propose several alternative choices for the cost function and illustrate with simulations that the improvement can be achieved, compared to $\rho(W)$. Detailed treatment of the described problem is given in the following sections.

IV. Static topology: Finite time horizon case

Now we consider the weight optimization for the consensus algorithm when only a finite budget of algorithm iterations is available. Weight optimization for the case of static topology
is studied in [11]. Proposed weights are the solution of the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad r(W - J) \\
\text{subject to} & \quad W_{ij} = 0, \quad \{i, j\} \notin E
\end{align*}
\]  
(11)

This choice for the objective function is motivated by the two facts:

1) Spectral radius of the matrix \(W - J\) is a time asymptotic convergence rate, i.e.,

\[
r(W - J) = \sup_{x(0) \neq 0} \lim_{k \to \infty} \left( \frac{\|e(k)\|}{\|e(0)\|} \right)^{1/k}
\]  
(12)

2) Spectral radius of the matrix \(W - J\) is the worst case per step convergence rate, i.e.,

\[
r(W - J) = \sup_{x(0) \neq x_{avg}} \left( \frac{\|e(k + 1)\|}{\|e(k)\|} \right)
\]  
(13)

However, one can argue that:

1) For the case of the finite budget of the algorithm iterations, we are not in the time asymptotic regime, but rather in the transient regime.

2) We might be interested in the average performance, and not the worst case performance. Also, the worst case occurs unlikely.

Having in mind the simulation results in Figure 1, it is tempting to try to find the alternative objective function for the case of the finite budget of iterations.

We remark that in the case of the static topology, weight matrix is deterministic. It is straightforward to show that the recurrence relation for the consensus error is given by:

\[
e(k + 1) = (W - J)e(k)
\]  
(14)

Further, decompose the matrix \(W - J\) by the eigenvalue decomposition

\[
W - J = QAQ^T
\]

\[
\Lambda = \text{diag}(\lambda_i), \quad i = 1, ..., N
\]

Introduce \(\zeta(k) = Q^T e(k)\).

Then we can write the following equation:

\[
\zeta(k + 1) = \sum_{i=1}^{N} \lambda_i^2 \zeta_i(0)
\]  
(15)

Remark that if the consensus algorithm converges, then the matrix \(W - J\) has all the eigenvalues with the modulus less than one. It is clear that for very large \(k\), only the largest eigenvalues play important role in the expression (15), other terms being negligible. On the other hand, for very small \(k\), not only the largest but all the eigenvalues are of interest.

Having in mind this consideration, we propose to minimize the sum of the squares of the eigenvalues of \(W - J\), i.e., we propose the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \text{tr} ((W - J)^2) \\
\text{subject to} & \quad W_{ij} = 0, \quad \{i, j\} \notin E
\end{align*}
\]  
(16)

Further, we may reason as follows. For the \(k\) being very large, only the largest eigenvalue is of interest; for the \(k\) being very small, all the eigenvalues should be taken into account. For some "middle" range of the algorithm iterations, it is thus reasonable to try to minimize \(k\) largest eigenvalues of \((W - J)^2\), \(1 < k < N\). This leads to the following weight optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{k} \lambda_i^2 (W - J) \\
\text{subject to} & \quad W_{ij} = 0, \quad \{i, j\} \notin E
\end{align*}
\]  
(17)

We will show that the optimization problems (16) and (17) are both convex (first being a special case of the second one for \(k = 1\)), and can be efficiently solved globally.

Lemma 6 Function \(\psi(W) = \sum_{i=1}^{k} \lambda_i^2 (W - J)\) is convex for any \(k \in \{1, ..., N\}\).

Proof: It is well known [2] that the sum of the \(k\) largest eigenvalues of the matrix \((W - J)^2\) can be expressed as:

\[
\max_{Q^T Q = I_k} \quad Q^T (W - J)^2 Q
\]  
(18)

Denote the \(i\)-th column of \(Q\) by \(q_i\), and the \(i\)-th column of \(W\) and \(J\) by \(W_i\) and \(J_i\), respectively. We next develop (19) in the following way:

\[
Q^T (W - J)^2 Q = \sum_{i=1}^{k} q_i^T (W - J)^2 q_i
\]

\[
= \sum_{i=1}^{k} \text{tr} ((W - J)q_i q_i^T (W - J))
\]

\[
= \sum_{i=1}^{k} \sum_{j=1}^{N} ((W_i - J_i)q_i q_j)^2 = \theta(W, Q)
\]

We remark that for the fixed \(Q\), function \(\theta(W, Q)\) is a convex
quadratic function. Therefore, the function \( \psi(W) \) is also convex as the maximum of convex functions. This completes the proof.

A. Simulation example

We consider the geometric random graph with \( N = 30 \) sensors. Each sensor has 3 neighbors on average. The graph is connected. To evaluate the performance of different choices for the cost function, we run Monte Carlo simulations. We are interested in the average performance, rather than the worst case performance. We generate 100 sample paths, each time for different, randomly generated initial state \( x(0) \). Then we average the relative mean squared error at each time step over the sample paths, i.e., we compute:

\[
\epsilon(k) = \frac{1}{S} \sum_{s=1}^{S} e(k,s) e^T(k,s) e(0,s) \tag{20}
\]

We compare several different choices of weights: 1) Metropolis weights, proposed in [11]; 2) SGBW (supergraph-based weights), proposed in [10]; 3) Simple choice of equal weights, i.e. off-diagonal weights are chosen to be 1\%/N for all available links; 4) weights that are the solution of (11); 5) weights that are the solution of (16); 6) weights that are the solution of (17) for case \( k=5 \). As we can see in the Figure 2, weight choices (11), (16) and (17) perform better than the Metropolis weights, and of course much better than the equal weights choice. Also, we note that the novel proposed choices (16),(17), and (11) exhibit the tradeoff in the following sense: (16) has the best transient regime, while (11) has the best steady state, or asymptotic regime. Practically, this means that one should choose (11) to achieve high precision (of order 1\% of error) and one should choose (16) if only a very small number of iterations is available. Solution that corresponds to the case \( 1 < k < N \) represents a tradeoff, meaning that it has both satisfactory transient, but also steady state regime.

In Figure 3, distributions of the eigenvalues for different choices of weight matrices are plotted. One can note a clear trade off between the maximal-modulus eigenvalue(i.e., spectral radius) and the number(cardinality) of large eigenvalues. For example, for the case of \( k = 1 \) (minimizing \( \lambda_{\text{max}} \)), resulting spectral radius of \( W - J \) is equal 0.98. For the case of \( k = N \) (minimizing of trace) spectral radius is equal to 0.99. On the other hand, many eigenvalues for the case \( k = N \) are concentrated around zero; for the case \( k = 1 \), many eigenvalues are close in modulus to the maximal one.

![Fig. 2. Estimated mean squared error versus iteration number k for the static network with N=30 sensors](image)

![Fig. 3. Distributions of eigenvalues of matrix W-J for the different choices of objective function. Upper left: k = 1, optimization problem (17); upper right, k = N, optimization problem (17); lower right, k = 5, optimization problem (17); Metropolis weights](image)

V. RANDOM TOPOLOGY:FINITE TIME HORIZON CASE

We extend the results from the previous section to the case of the random network topology. Generalizing optimization problem (17) to the case of the random topology, we propose to solve the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{k} \lambda_i \left( \mathbb{E} [W^2 - J] \right) \\
\text{subject to} & \quad W_{ij} = 0, \quad \{i,j\} \notin E
\end{align*} \tag{21}
\]

The case \( k = 1 \) boils down to the optimization problem (9), and the case \( k = N \) leads to the minimization of the trace of the matrix \( \mathbb{E} [W^2 - J] \). Denote \( \gamma(A) = \sum_{i=1}^{k} \lambda_i(A), A \in S^N \). It can be shown that the function \( \gamma(A) \) is convex. We will use this fact to show that the function \( \psi(W) \) is convex, where \( \psi(W) \) is given by

\[
\psi(W) = \sum_{i=1}^{k} \lambda_i \left( \mathbb{E} [W^2 - J] \right) \tag{22}
\]

**Fact.** Function \( \psi(\cdot) \) given by 22 is convex.

Proof: Choose and fix arbitrary \( X, Y \in S_W \). We restrict our attention to the matrices \( W \) of the form

\[
W = X + t Y, \quad t \in \mathbb{R}. \tag{23}
\]
Recall the expression for $W$ given by (5). For the matrix $W$ given by (23) we have for $W = W(t)$

$$
W(t) = I - \text{diag}((X + tY) A) + (X + tY) \odot A \tag{24}
$$

$$
\mathcal{X} = X \odot A + I - \text{diag}(X A) \tag{25}
$$

$$
\mathcal{Y} = Y \odot A - \text{diag}(X A) \tag{26}
$$

Introduce the auxiliary function $\phi : \mathbb{R} \to \mathbb{R}_+$,

$$
\phi(t) = \gamma \left( E \left[ W(t)^2 \right] - J \right)
$$

To prove that $\rho(W)$ is convex, it suffices to prove that the function $\phi$ is convex. Introduce $Z(t)$ and compute successively

$$
Z(t) = W(t)^2 - J
$$

$$
= (\mathcal{X} + t\mathcal{Y})^2 - J
$$

$$
= t^2 \mathcal{Y}^2 + t (\mathcal{X} \mathcal{Y} + \mathcal{Y} \mathcal{X}) + \mathcal{X}^2 - J
$$

$$
= t^2 Z_2 + t Z_1 + Z_0
$$

The random matrices $Z_2$, $Z_1$ and $Z_0$ do not depend on $t$. Also, $Z_2$ is semidefinite positive. The function $\phi(t)$ can now be expressed as

$$
\phi(t) = \gamma \left( E \left[ Z(t) \right] \right)
$$

We will now derive that

$$
Z ((1 - \alpha)t + \alpha u) \leq (1 - \alpha) Z(t) + \alpha Z(u), \quad \forall \alpha \in [0, 1], \forall t, u \in \mathbb{R} \tag{27}
$$

Since $\psi(t) = t^2$ is convex, the following inequality holds:

$$
((1 - \alpha)t + \alpha u)^2 \leq (1 - \alpha)t^2 + \alpha u^2, \quad \alpha \in [0, 1] \tag{28}
$$

Since the matrix $Z_2$ is positive semidefinite, eqn. (28) implies that:

$$
\left( ((1 - \alpha)t + \alpha u)^2 \right) Z_2 \leq (1 - \alpha) t^2 Z_2 + \alpha u^2 Z_2, \quad \alpha \in [0, 1]
$$

After adding to both sides $((1 - \alpha)t + \alpha u) Z_1 + Z_0$, we get eqn. (28). Taking the expectation to both sides of (28), we get:

$$
E \left[ Z ((1 - \alpha)t + \alpha u) \right] \leq E \left[ (1 - \alpha) Z(t) + \alpha Z(u) \right]
$$

$$
= (1 - \alpha) E \left[ Z(t) \right] + \alpha E \left[ Z(u) \right]
$$

$$
\alpha \in [0, 1]
$$

Now, we have that:

$$
\phi ((1 - \alpha)t + \alpha u) = \gamma \left( E \left[ Z ((1 - \alpha)t + \alpha u) \right] \right)
$$

$$
\leq \gamma \left( (1 - \alpha) E \left[ Z(t) \right] + \alpha E \left[ Z(u) \right] \right)
$$

$$
= (1 - \alpha) \phi(t) + \alpha \phi(u), \quad \alpha \in [0, 1]
$$

The last inequality holds since $\gamma(\cdot)$ is convex. This implies $\phi(t)$ is convex and hence $\rho(W)$ is convex.

A. Simulation example

Now we consider the case of $N = 80$ sensors. The network is modeled as a geometric disc graph. Average degree in the network is 21. Link formation probabilities are generated in the similar way as for the case considered in section III-E.

![Fig. 4. Estimated $E(||e(k)||)$ versus iteration number $k$ for the random network with N=80 nodes](image)

In Figure 5 we compare the proposed solution to the Metropolis weights and equal weights. It can be seen that both for choices $k = 1$ and $k = N$, the solution of (21) has better performance than the Metropolis weights and equal weights.

VI. Conclusion

In this paper, we studied the consensus algorithm with random topology. We addressed the problem of finding the optimal weights $W$ that yield the best average consensus performance, when only a limited number of iterations is available. We formulated a class of optimization problems for finding the optimal weights and showed that these optimization problems are convex. Optimization consists of minimizing the sum of $k$ largest eigenvalues of the matrix that describes the error decaying, where $k = 1, ..., N$. For any choice of $k$, optimization procedure provides the solutions comparable to or better than the weight choices previously proposed in the
literature. If, however, we want to compare our two solutions that correspond to two different $k$’s, we face a trade off: larger $k$ yields faster convergence in the transient regime and slower convergence in the long run regime. Thus, the parameter $k$ should be tuned according to a specific application and requirements.

REFERENCES