

On identifying sparse representations of consensus networks[★]

Neil Dhingra^{*} Fu Lin^{*} Makan Fardad^{**}
Mihailo R. Jovanović^{*}

^{*} *Department of Electrical and Computer Engineering, University of
Minnesota, Minneapolis, MN 55455, (e-mails: dhin0008@umn.edu,
fu@umn.edu, mihailo@umn.edu).*

^{**} *Department of Electrical Engineering and Computer Science,
Syracuse University, NY 13244, (e-mail: makan@syr.edu)*

Abstract: We consider the problem of identifying optimal sparse graph representations of dense consensus networks. The performance of the sparse representation is characterized by the global performance measure which quantifies the difference between the output of the sparse graph and the output of the original graph. By minimizing the sum of this performance measure and a sparsity-promoting penalty function, the alternating direction method of multipliers identifies sparsity structures that strike a balance between the performance measure and the number of edges in the graph. We then optimize the edge weights of sparse graphs over the identified topologies. Two examples are provided to illustrate the utility of the developed approach.

Keywords: Alternating direction method of multipliers, cardinality minimization, consensus networks, ℓ_1 minimization, reweighted ℓ_1 , sparse graph representations, sparsity-promoting optimal control, structured feedback design.

1. INTRODUCTION

We consider the problem of identifying a sparse representation of a given dense graph such that the performance of consensus algorithms operating on both graphs is close in the sense of variance amplification. Consensus networks have garnered much interest for problems dealing with collective decision-making and collective sensing as in Young et al. (2010), Xiao et al. (2007), and Zelazo and Mesbahi (2011). These networks have applications as varied as modeling animal group dynamics, formation flying of spacecraft, and data fusion in sensor networks. See Ballerini et al. (2008), Sumpter et al. (2008), Mesbahi and Hadaegh (2001), and Xiao et al. (2005).

A sparse representation of a dense graph can identify valuable communication links or facilitate understanding of the underlying dynamics of the original graph. Recent related work in Fardad et al. (2011) and Lin et al. (2012) deals with designing a sparse network to minimize input-output variance amplification or focuses on improving the algebraic connectivity of an existing network by adding edges in Ghosh and Boyd (2006) or removing edges of uniformly weighted graphs in Asensio-Marco and Beferull-Lozano (2010). This paper deals with the problem of removing edges from an existing dense network and preserving its input-output behavior as measured by the \mathcal{H}_2 norm.

Our approach minimizes a quadratic performance measure which quantifies the difference between the outputs of the sparse representation and the original dense graph subject

to an stochastic disturbances. The algorithm consists of two steps. First, an optimal sparse network topology is obtained by augmenting the performance measure with a term that penalizes the number of edges in the network. Then, the optimal edge weights are chosen over the identified topology.

The rest of this paper is structured as follows: A description of consensus networks and our problem formulation is given in Section 2. Our algorithm is presented in Section 3. Section 4 provides examples of the algorithm and compares our method to a truncation scheme.

2. PROBLEM FORMULATION

Given a network \mathcal{G} executing a consensus algorithm subject to disturbances, we consider the design of a sparse network $\hat{\mathcal{G}}$ with the same set of nodes but a different set of edges such that the outputs of both networks are close in the \mathcal{H}_2 sense.

Let \mathcal{G} be an undirected connected network with n nodes represented by the set \mathcal{V} and q edges represented by the set \mathcal{E} , where $l \sim (i, j) \in \mathcal{E}$ means that there is an edge between nodes $i, j \in \mathcal{V}$. We consider the consensus algorithm subject to disturbances

$$\dot{x} = -Lx + d, \quad (1)$$

where x is the stacked states of the nodes, L is the weighted Laplacian matrix whose sparsity structure is determined by the topology of \mathcal{G} , and d is the white stochastic process with zero mean and unit variance. In particular, the ij th entry of L is determined by the edge weight between nodes

[★] Financial support from the National Science Foundation under CAREER Award CMMI-06-44793 and under awards CMMI-09-27720 and CMMI-0927509 is gratefully acknowledged.

i and j . This dynamical system can be viewed as a set of single-integrators

$$\dot{x}_i = u_i + d_i,$$

with the feedback control

$$u = -Lx.$$

As shown in Xiao et al. (2007) and Bamieh et al. (2012), the performance of the consensus network with disturbances is quantified by the steady-state variance of the deviation from the consensus value,

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = (1/n)\mathbf{1}^T x,$$

where $\mathbf{1}$ is the vector of all ones. In Lin et al. (2010), the authors designed the edge weights (i.e., the nonzero entries of L) of \mathcal{G} to minimize the steady-state variance of the performance output that encapsulates both the deviation from average and the control input, i.e.,

$$y = \begin{bmatrix} x - \bar{x}\mathbf{1} \\ u \end{bmatrix}. \quad (2)$$

In this paper, our emphasis lies on the sparse representation of a network \mathcal{G} . Namely, given a network \mathcal{G} with the dynamics (1) and the output (2), it is of interest to identify a different set of edges and then design the edge weights such that the performance output of the new network $\hat{\mathcal{G}}$ is close to that of \mathcal{G} . In particular, it is desired that $\hat{\mathcal{G}}$ is a subgraph of \mathcal{G} with much fewer edges.

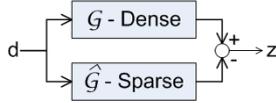


Fig. 1. A parallel connection quantifies the output difference between two systems with a dense graph \mathcal{G} and a sparse graph $\hat{\mathcal{G}}$ subject to stochastic disturbance. The state-space representation of the parallel connection of \mathcal{G} and $\hat{\mathcal{G}}$ shown in Fig. 1 is given by

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} &= \begin{bmatrix} -L & 0 \\ 0 & -\hat{L} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} I \\ I \end{bmatrix} d \\ z &= y - \hat{y} = \begin{bmatrix} H_g & -H_g \\ -L & \hat{L} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}, \end{aligned} \quad (3)$$

where $H_g = I - (1/n)\mathbf{1}\mathbf{1}^T$. Given a Laplacian matrix L , we design both the structure and the entries of another Laplacian matrix \hat{L} such that the steady-state variance of z , denoted by $J(\hat{L})$, is minimized. To induce a sparse structure on \hat{L} and thus minimize the number of edges in $\hat{\mathcal{G}}$, we penalize the number of nonzero entries of \hat{L} , denoted by $\mathbf{card}(\hat{L})$, which leads to the following optimization problem

$$\text{minimize } J(\hat{L}) + \gamma \mathbf{card}(\hat{L}), \quad (4)$$

where the positive parameter γ controls the sparsity level of \hat{L} .

2.1 Change of coordinates to remove the unobservable average mode

It is well-known that the Laplacian matrix has a zero eigenvalue associated with the vector of all ones. Thus, the average mode \bar{x} is not asymptotically stable (see Zelazo

and Mesbahi (2011)) and its steady-state variance is not finite (see Xiao et al. (2007) and Bamieh et al. (2012)). However, it is readily verified that \bar{x} is not observable from the output y . We remove this unobservable mode by a change of coordinates described in Zelazo and Mesbahi (2011) and Lin et al. (2010).

Let edge $l \sim (i, j) \in \mathcal{E}$ be associated with a vector $e_l \in \mathbb{R}^n$ that has 1 and -1 as its i th and j th entries, and 0 elsewhere. The incidence matrix of \mathcal{G} is given by $E = [e_1 \cdots e_q] \in \mathbb{R}^{n \times q}$. Then the weighted Laplacian matrix can be written as

$$L = \sum_{l=1}^q k_l e_l e_l^T = EKE^T$$

where k_l is the edge weight on l and K is the diagonal matrix formed from $k = [k_1 \cdots k_q]$.

Since \mathcal{G} is connected, it has a tree subgraph. Let E_t be the incidence matrix of a tree subgraph of \mathcal{G} . Then it can be shown that the change of coordinates from Zelazo and Mesbahi (2011) and Lin et al. (2010),

$$\begin{aligned} \begin{bmatrix} \psi \\ \bar{x} \end{bmatrix} &= Sx = \begin{bmatrix} E_t^T \\ (1/n)\mathbf{1}^T \end{bmatrix} x \\ S^{-1} &= [E_t(E_t^T E_t)^{-1} \mathbf{1}], \end{aligned} \quad (5)$$

separates the average mode \bar{x} from the differences between nodes across the edges of the tree, i.e., $\psi = E_t^T x$. After removing the unobservable average mode, we have the following minimum realization of the system (1) with the output (2)

$$\begin{aligned} \dot{\psi} &= -E_t^T E_t M K M^T \psi + E_t^T d \\ z &= \begin{bmatrix} E_t(E_t^T E_t)^{-1} \\ -E_t M K M^T \end{bmatrix} \psi, \end{aligned} \quad (6)$$

where $M = (E_t^T E_t)^{-1} E_t^T E$.

2.2 Computation of the steady-state variance

Since \hat{L} is the Laplacian matrix of the graph $\hat{\mathcal{G}}$, it can be written as

$$\hat{L} = \hat{E} \hat{K} \hat{E}^T$$

where \hat{E} is the incidence matrix of a graph with m edges and \hat{K} is the diagonal matrix formed from the edge weights \hat{k} . The graph described by \hat{E} defines the edges available for forming $\hat{\mathcal{G}}$. Taking $\hat{E} = E$ restricts $\hat{\mathcal{G}}$ to subgraphs of \mathcal{G} . Note that setting $\hat{k}_l = 0$ is equivalent to removing edge l from $\hat{\mathcal{G}}$. Thus, the number of nonzero elements of \hat{L} is determined by the number of edges in $\hat{\mathcal{G}}$ (or equivalently $\mathbf{card}(\hat{K})$). It is easy to show that $\mathbf{card}(\hat{L}) = n + 2 \mathbf{card}(\hat{K})$.

After applying the change of coordinates (5) to each subsystem of (3) and removing the unobservable average modes \bar{x} and $\hat{\bar{x}}$, we obtain

$$\begin{aligned} \begin{bmatrix} \dot{\psi} \\ \dot{\hat{\psi}} \end{bmatrix} &= \underbrace{\begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix}}_A \begin{bmatrix} \psi \\ \hat{\psi} \end{bmatrix} + \underbrace{\begin{bmatrix} B_1 \\ B_1 \end{bmatrix}}_B d \\ z &= \underbrace{\begin{bmatrix} C_1 - C_1 \\ C_2 - \hat{C}_2 \end{bmatrix}}_C \begin{bmatrix} \psi \\ \hat{\psi} \end{bmatrix} \end{aligned} \quad (7)$$

where, after defining $\hat{M} = (E_t^T E_t)^{-1} E_t^T \hat{E}$,

$$\begin{aligned} A &= -E_t^T E_t M K M^T & \hat{A} &= -E_t^T E_t \hat{M} \hat{K} \hat{M}^T \\ B_1 &= E_t^T & C_1 &= E_t (E_t^T E_t)^{-1} \\ C_2 &= -E_t M K M^T & \hat{C}_2 &= -E_t \hat{M} \hat{K} \hat{M}^T. \end{aligned}$$

The steady-state variance $J(\hat{K})$ is given by

$$J(\hat{K}) = \text{trace}(\mathcal{C}\mathcal{P}\mathcal{C}^T)$$

where \mathcal{P} is the solution of the Lyapunov equation

$$\mathcal{A}\mathcal{P} + \mathcal{P}\mathcal{A}^T = -\mathcal{B}\mathcal{B}^T. \quad (8)$$

Since \mathcal{P} can be partitioned into 2×2 block matrices

$$\mathcal{P} = \begin{bmatrix} X & Y^T \\ Y & Z \end{bmatrix},$$

equation (8) can be decomposed into one Sylvester equation

$$\hat{A}Y + YA^T = -B_1 B_1^T, \quad (9)$$

and two Lyapunov equations

$$\begin{aligned} AX + XA^T &= -B_1 B_1^T \\ \hat{A}Z + Z\hat{A}^T &= -B_1 B_1^T. \end{aligned} \quad (10)$$

Since $B_1 B_1^T = E_t^T E_t$, we can exploit the structure of A, A^T, \hat{A} and \hat{A}^T as described in Lin et al. (2010) and Zelazo and Mesbahi (2011) to obtain explicit solutions to (10),

$$\begin{aligned} X &= (1/2)(M K M^T)^{-1} \\ Z &= (1/2)(\hat{M} \hat{K} \hat{M}^T)^{-1}. \end{aligned}$$

3. ALGORITHM

The algorithm presented in this paper consists of two steps:

- sparse structure identification,
- optimal weight selection.

The first step identifies an optimal sparsity structure from the solution to a relaxation of (4). The second ‘polishing’ step minimizes the \mathcal{H}_2 norm of (3) over this structure by selecting the optimal edge weights.

Problem (4) is a combinatorial optimization problem whose solution usually requires an intractable combinatorial search. The complexity arises from the presence of the cardinality function; to facilitate tractable computations we relax it with a sparsity-promoting penalty function $g(\hat{K})$ and solve the problem

$$\underset{\hat{K}}{\text{minimize}} \quad J(\hat{K}) + \gamma g(\hat{K}), \quad (11)$$

where $g(\hat{K})$ is determined by the weighted ℓ_1 norm employed by Candès et al. (2008),

$$g(\hat{K}) = \sum_{i=1}^m w_i |\hat{K}_{ii}|. \quad (12)$$

In (12), $w \in \mathbb{R}^m$ is a nonnegative vector of weights. Taking these to be inversely proportional to \hat{K}_{ii} for nonzero \hat{K}_{ii} uncovers the cardinality function. However, this weighting strategy cannot be implemented because it depends on the unknown optimal \hat{K} . We instead employ a reweighted- ℓ_1 strategy from Candès et al. (2008) in which the weights $w_i = (\epsilon + |\hat{K}_{ii}|)^{-1}$ at each iteration are derived from the optimal value found in the previous iteration. The small parameter ϵ ensures that $g(\hat{K})$ is well defined when $\hat{K}_{ii} = 0$.

3.1 Sparse structure identification

For notational convenience we set $\hat{K} = F$ in this section. The solution to (11) identifies an optimal sparsity structure associated with a particular γ . The Alternating Direction Method of Multipliers (ADMM) finds this solution by separating the objective function into its component functions, the \mathcal{H}_2 norm $J(F)$ and the sparsity penalty $g(F)$, and alternating between optimization over each component. This alternation allows us to exploit the differentiability of $J(F)$ and the separable structure of $g(F)$.

We recast the problem (11) in the form that is suitable for the application of ADMM,

$$\begin{aligned} &\text{minimize} && J(F) + \gamma g(G) \\ &\text{subject to} && F - G = 0. \end{aligned} \quad (13)$$

For any feasible F and G , the solution to (13) coincides with the solution to (11). The augmented Lagrangian

$$\begin{aligned} L_\rho(F, G, \Lambda) &= J(F) + \gamma g(G) + \text{trace}(\Lambda^T(F - G)) + \\ &\quad (\rho/2) \|F - G\|_F^2 \end{aligned} \quad (14)$$

incorporates the constraints into a single unconstrained minimization problem. The quadratic term $(\rho/2) \|F - G\|_F^2$ penalizes the violation of the constraint where a larger parameter ρ results in a faster convergence of $\|F - G\|_F^2$ to zero. Minimization of (14) is obtained using the ADMM iterations described in Boyd et al. (2011),

$$\begin{aligned} F^{k+1} &= \underset{F}{\text{argmin}} L_\rho(F, G^k, \Lambda^k) \\ G^{k+1} &= \underset{G}{\text{argmin}} L_\rho(F^{k+1}, G, \Lambda^k) \\ \Lambda^{k+1} &= \Lambda^k + \rho(F^{k+1} - G^{k+1}) \end{aligned} \quad (15)$$

which are performed until the residuals $\|G^k - G^{k-1}\|_F$ and $\|F^k - G^k\|_F$ become smaller than a specified tolerance.

G-Minimization Step Minimizing (14) with respect to G is equivalent to

$$\underset{G}{\text{minimize}} \quad \gamma g(G) + (\rho/2) \|G - V\|_F^2 \quad (16)$$

where $V := (1/\rho)\Lambda + F^{k+1}$. Since the objective function in (16) can be expressed as the sum of the independent terms, $\gamma w_i |G_{ii}| + \frac{\rho}{2} (G_{ii} - V_{ii})^2$, minimization of $g(G)$ yields the soft-thresholding operator described in Boyd et al. (2011) for the elementwise minimization problems

$$G_{ii}^{k+1} = \begin{cases} V_{ii} - \mu w_i, & V_{ii} > \mu w_i \\ 0, & V_{ii} \in [-\mu w_i, \mu w_i] \\ V_{ii} + \mu w_i, & V_{ii} < -\mu w_i \end{cases} \quad (17)$$

where $\mu := \gamma/\rho$.

F-Minimization Step Minimizing (14) with respect to F is equivalent to

$$\underset{F}{\text{minimize}} \quad J(F) + (\rho/2) \|F - U\|_F^2. \quad (18)$$

where $U := G^k - (1/\rho)\Lambda$. The gradient (A.3) and Hessian (A.4), given in the Appendix, are used to solve (18) with descent methods.

We note that, in general, J is a nonconvex function of the unknown matrix F . In view of this, we use descent methods to find a local optimal solution. In particular, we employ the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, which chooses a descent direction based on an approximation of the Hessian. This approach converges to an optimal value in fewer iterations than gradient descent, and is less computationally expensive than Newton's method. We are currently in the process of identifying classes of problems that result in a convex function $J(F)$.

At each iteration r , BFGS approximates the Hessian with the matrix \tilde{H}^r using a rank-two update on \tilde{H}^{r-1} so that \tilde{H}^r incorporates the change in the gradient as a result of the change in the optimization variable. Namely, \tilde{H}^{r-1} is modified to satisfy the secant equation (see Nocedal and Wright (1999)),

$$\nabla J(x^r) = \nabla J(x^{r-1}) + \tilde{H}^r(x^r - x^{r-1}).$$

The Sherman-Morrison formula allows direct computation of $(\tilde{H}^r)^{-1}$ using a rank-two update on $(\tilde{H}^{r-1})^{-1}$.

Computing $(\tilde{H}^r)^{-1}$ requires $O(m^2)$ operations. In comparison, computing the Hessian requires solving $4m$ Sylvester equations of the size $n \times n$; solving each of these requires $O(n^3)$ operations, and inverting the resulting $m \times m$ matrix takes $O(m^3)$ operations. Thus, finding the inverse of the Hessian amounts to operations of order $O(\max\{mn^3, m^3\})$.

For connected undirected graphs, $m \in [n-1, n^2-n]$, so the BFGS update requires between $O(n^2)$ and $O(n^4)$ operations and computing the inverse of the Hessian requires between $O(n^4)$ and $O(n^6)$ operations.

3.2 Polishing - Optimal edge weight selection

The sparsity structure \hat{S} of the edge weights \hat{K} found in the previous step specifies a graph \hat{G} which strikes a balance between the performance of the graph and the number of removed edges. Every zero edge weight $\hat{k}_l = 0$ corresponds a removed edge l . However, the weights \hat{K} minimize the objective function (11), which is not the same as minimizing $J(\hat{K})$ over \hat{G} . To 'polish' the edge weights and to obtain the optimal edge weights for this graph, we solve the problem

$$\begin{aligned} &\text{minimize} && J(\hat{K}) \\ &\text{subject to} && \hat{K} \in \hat{S}. \end{aligned} \quad (19)$$

This can be cast as an unconstrained optimization problem by reforming (7) to reflect the graph described by \hat{S} . We take \hat{E} to be composed of the columns e_l of E which are

associated with nonzero edge weights \hat{k}_l . We then select a tree subgraph \hat{E}_t of \hat{E} ; since \hat{E}_t is also a subgraph of E , we use it to perform the change of coordinates (5) to obtain a system in the form of (7).

This system incorporates the identified topology into its structure and its \mathcal{H}_2 norm quantifies the similarity between the output of the original graph and the output of the graph specified by \hat{S} . Minimization of the \mathcal{H}_2 norm of this system yields the optimal edge weights.

The objective function in (19) is identical to that in (18) without the quadratic penalty. Using the appropriate gradient (A.2), the BFGS method is employed to solve (19) and obtain the optimal edge weights.

4. EXAMPLES

The first example illustrates the performance of our algorithm on a random graph and the tradeoff between performance and sparsity. The second example compares sparsity structures identified by our algorithm with sparsity structures obtained by truncating the edges with the smallest weights.

4.1 Complete graph with random weights

In this example, we consider a complete 25-node, 300-edge graph with random edge weights. Solving (11) with increasing values of γ places increasing importance on the sparsity of the graph. Figure 2 shows the number of edges in the graph \hat{G} as a function of the parameter γ .

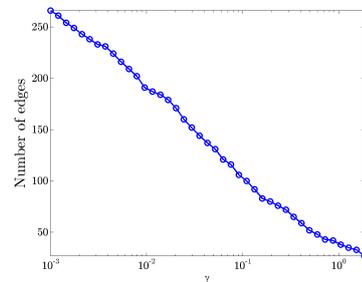


Fig. 2. Number of edges of the sparse graphs at different values of γ for the example of a complete graph with randomly weighted edges.

Figure 3 shows the increasingly sparse topologies identified by the algorithm. The sparser structures correspond to larger values of the parameter γ .

Figure 4 illustrates the tradeoff between sparsity and performance. The performance metric $J(\hat{K})$ is plotted against the numbers of edges removed and illustrates the growing difference between the dense graph and sparser topologies.

4.2 Comparison with truncation

In this example we illustrate the utility of our method by comparing it with a simple truncation algorithm. Truncation of the edges with the smallest weights is an intuitive strategy for identifying a sparsity structure. We consider

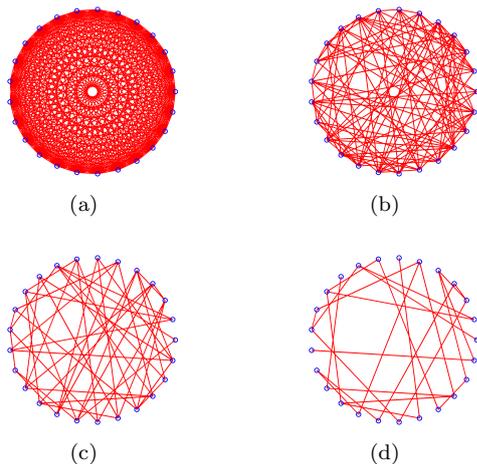


Fig. 3. Identified sparse graphs for the example of a complete graph with randomly weighted edges. (a) 300 edges for $\gamma = 0$ (b) 106 edges for $\gamma = 0.0910$ (c) 48 edges for $\gamma = 0.5964$ (d) 27 edges for $\gamma = 1.8421$.

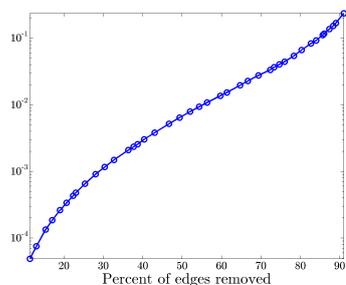


Fig. 4. $J(\hat{K})$ after polishing at different levels of sparsity normalized by the \mathcal{H}_2 norm of \mathcal{G} for the example of a complete graph with randomly weighted edges

a truncation scheme in which the edge with the smallest weight whose removal does not disconnect the graph is iteratively removed until only the desired number of edges remain.

Following an example provided in Motee and Jadbabaie (2008), a graph is formed in the following manner:

- Nodes i are assigned random locations ξ_i .
- Edges are formed between nodes which lie within a certain radius of one another.
- The weight on each edge is a function of the distance between nodes, $k_l = e^{-\|\xi_i - \xi_j\|}$ where edge l connects nodes i and j .

Figure 5 shows such a graph with 40 nodes connected by 425 edges. The truncation and ADMM algorithms were used to identify various sparse topologies with the sparsest topology consisting of 79 edges.

Figure 6 shows the sparsest topology specified by the truncation scheme. Since edges between nodes which are far apart have the smallest weights in this example, they are removed by the truncation algorithm.

Figure 7 shows the sparsest topology specified by the algorithm presented in this paper. The structure achieved here does not show systematic removal of the longest edges

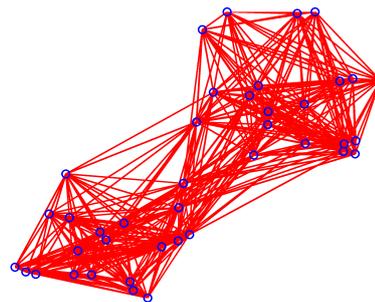


Fig. 5. Original graph with 425 edges.

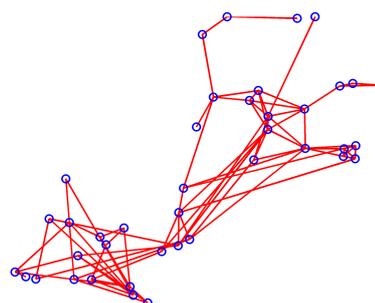


Fig. 6. Sparse graph with 79 edges obtained using truncation.

and qualitatively preserves the ‘skeleton’ of the original graph.

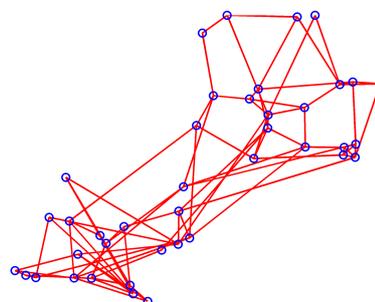


Fig. 7. Sparse graph with 79 edges obtained using ADMM algorithm.

Polishing is performed on the respective topologies at several levels of sparsity. The associated performance metrics $J(\hat{K})$ are plotted in Fig. 8 against the number of edges removed. When the cardinalities of the feedback gain matrices are the same, the sparse graphs identified by our algorithm achieve closer performance to the original graph than those identified by truncation.

Qualitatively, our algorithm preserves the shape of the graph by maintaining the pattern of information exchange. Since by definition the weakest edges connect nodes which are geometrically far, their removal can greatly increase geodesic distances. Our algorithm accounts for this effect because it considers how well the outputs of the dense and the sparse graphs match.

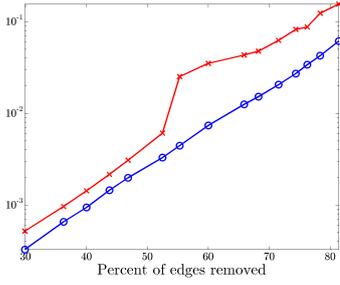


Fig. 8. $J(\hat{K})$ normalized by the \mathcal{H}_2 norm of \mathcal{G} for different sparsity structures identified by truncation (\circ) and by ADMM (\times).

REFERENCES

- Asensio-Marco, C. and Beferull-Lozano, B. (2010). Accelerating consensus gossip algorithms: Sparsifying networks can be good for you. In *Proceedings of the 2010 IEEE International Conference on Communications*, 1–5.
- Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A., Viale, M., and Zdravkovic, V. (2008). Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences*, 105(4), 1232–1237. doi: 10.1073/pnas.0711437105.
- Bamieh, B., Jovanović, M.R., Mitra, P., and Patterson, S. (2012). Coherence in large-scale networks: dimension dependent limitations of local feedback. *IEEE Trans. Automat. Control*. Doi:10.1109/TAC.2012.2202052.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 1–124.
- Candès, E.J., Wakin, M.B., and Boyd, S.P. (2008). Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14, 877–905.
- Fardad, M., Lin, F., and Jovanović, M.R. (2011). Sparsity-promoting optimal control for a class of distributed systems. In *Proceedings of the 2011 American Control Conference*, 2050–2055.
- Ghosh, A. and Boyd, S. (2006). Growing well-connected graphs. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 6605–6611.
- Lin, F., Fardad, M., and Jovanović, M.R. (2010). On the optimal localized feedback design for multi-vehicle systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 5744–5749.
- Lin, F., Fardad, M., and Jovanović, M.R. (2012). Sparse feedback synthesis via the alternating direction method of multipliers. In *Proceedings of the 2012 American Control Conference*, 4765–4770. Montréal, Canada.
- Mesbahi, M. and Hadaegh, F. (2001). Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching. *Journal of Guidance Control and Dynamics*, 24(2), 369–377.
- Motee, N. and Jadbabaie, A. (2008). Optimal control of spatially distributed systems. *IEEE Trans. Automat. Control*, 53(7), 1616–1629.

- Nocedal, J. and Wright, S.J. (1999). *Numerical Optimization*. Springer.
- Sumpter, D., Krause, J., James, R., Couzin, I., and Ward, A. (2008). Consensus decision making by fish. *Current Biology*, 18(22), 1773–1777.
- Xiao, L., Boyd, S., and Kim, S.J. (2007). Distributed average consensus with least-mean-square deviation. *J. Parallel Distrib. Comput.*, 67(1), 33–46.
- Xiao, L., Boyd, S., and Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 63–70.
- Young, G.F., Scardovi, L., and Leonard, N.E. (2010). Robustness of noisy consensus dynamics with directed communication. In *Proceedings of the 2010 American Control Conference*, 6312–6317.
- Zelazo, D. and Mesbahi, M. (2011). Edge agreement: Graph-theoretic performance bounds and passivity analysis. *IEEE Trans. Automat. Control*, 56(3), 544–555.

Appendix A. GRADIENT AND HESSIAN EXPRESSIONS

Computing the order ϵ variations in $J(F)$ as a result of order ϵ variations in F yields

$$\begin{aligned} \nabla J(F) = & \text{diag}(2\hat{M}^T E_t^T E_t L^T Y^T \hat{M} \\ & + \frac{1}{2}\hat{M}^T E_t^T E_t \hat{M} \\ & - 2\hat{M}^T Z(E_t^T E_t)^{-1} Z \hat{M} \\ & + 2\hat{M}^T Y L_e E_t^T E_t \hat{M} \\ & - 2\hat{M}^T E_t^T C_2 Y^T \hat{M}) \end{aligned} \quad (\text{A.1})$$

where L and L_e satisfy the Sylvester equations,

$$\begin{aligned} A^T L + L \hat{A} &= -C_1^T C_1 \\ A^T L_e + L_e \hat{A} &= -C_2^T \hat{C}_2. \end{aligned} \quad (\text{A.2})$$

F is constrained to be diagonal so its gradient is also diagonal. The gradient of (18) is then

$$\nabla_F L_\rho(F, G, \Lambda) = \nabla J(F) + \Lambda + \rho(F - G). \quad (\text{A.3})$$

F and its gradient can be represented by vectors in \mathbb{R}^m , so the Hessian can be represented by a matrix in $\mathbb{R}^{m \times m}$. The Hessian $\nabla^2 J(F)$ quantifies the local curvature of the objective function and the effect of order ϵ variations in F on $\nabla J(F)$. The i th row of $\nabla^2 J(F)$ is the gradient of the i th element of $\nabla J(F)$, which is given by

$$\begin{aligned} e_i^T \nabla^2 J(F) = & \text{diag}(-2\hat{M}^T E_t^T E_t (L_\alpha + L_\delta) Y^T \hat{M} \\ & - 2\hat{M}^T E_t^T E_t L^T L_\beta \hat{M} \\ & + 8\hat{M}^T Z \hat{M} e_i^T \hat{M}^T Z (E_t^T E_t)^{-1} Z \hat{M} \\ & - 2\hat{M}^T L_\beta^T (L_e E_t^T + C_2^T) E_t \hat{M} \\ & - 2\hat{M}^T Y L_\gamma E_t^T E_t \hat{M}) \end{aligned} \quad (\text{A.4})$$

where $L_\alpha, L_\beta, L_\gamma$ and L_δ solve the Sylvester equations

$$\begin{aligned} \hat{A}^T L_\alpha + L_\alpha A + \hat{M} e_i e_i^T \hat{M}^T E_t^T E_t L^T &= 0 \\ A L_\beta + L_\beta \hat{A}^T + Y^T \hat{M} e_i e_i^T \hat{M}^T E_t^T E_t &= 0 \\ A^T L_\gamma + L_\gamma \hat{A} + L_e E_t^T E_t \hat{M} e_i e_i^T \hat{M}^T &= 0 \\ \hat{A}^T L_\delta + L_\delta A + \hat{M} e_i e_i^T \hat{M}^T E_t^T C_2 &= 0. \end{aligned} \quad (\text{A.5})$$