

# Algorithms for Leader Selection in Stochastically Forced Consensus Networks

Fu Lin, *Member, IEEE*, Makan Fardad, *Member, IEEE*, and Mihailo R. Jovanović, *Senior Member, IEEE*

**Abstract**—We are interested in assigning a pre-specified number of nodes as leaders in order to minimize the mean-square deviation from consensus in stochastically forced networks. This problem arises in several applications including control of vehicular formations and localization in sensor networks. For networks with leaders subject to noise, we show that the Boolean constraints (which indicate whether a node is a leader) are the only source of nonconvexity. By relaxing these constraints to their convex hull we obtain a lower bound on the global optimal value. We also use a simple but efficient greedy algorithm to identify leaders and to compute an upper bound. For networks with leaders that perfectly follow their desired trajectories, we identify an additional source of nonconvexity in the form of a rank constraint. Removal of the rank constraint and relaxation of the Boolean constraints yields a semidefinite program for which we develop a customized algorithm well-suited for large networks. Several examples ranging from regular lattices to random graphs are provided to illustrate the effectiveness of the developed algorithms.

**Index Terms**—Alternating direction method of multipliers (ADMMs), consensus networks, convex optimization, convex relaxations, greedy algorithm, leader selection, performance bounds, semidefinite programming (SDP), sensor selection, variance amplification.

## I. INTRODUCTION

REACHING consensus in a decentralized fashion is an important problem in network science [1]. This problem is often encountered in social networks where a group of individuals is trying to agree on a certain issue [2], [3]. A related load balancing problem has been studied extensively in computer science with the objective of distributing evenly computational load over a network of processors [4], [5]. Recently, consensus problems have received considerable attention in the context of distributed control [6], [7]. For example, in cooperative control of vehicular formations, it is desired to use local interactions between vehicles in order to reach agreement on quantities such as heading angle, velocity, and inter-vehicular spacing.

Manuscript received October 11, 2012; revised May 29, 2013 and October 28, 2013; accepted February 21, 2014. Date of publication March 28, 2014; date of current version June 19, 2014. This work was supported by the National Science Foundation under CAREER Award CMMI-06-44793 and under awards CMMI-09-27720 and CMMI-0927509. Recommended by Associate Editor D. Bauso.

F. Lin is with the Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 USA (e-mail: fulin@mcs.anl.gov).

M. Fardad is with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244 USA (e-mail: makan@syr.edu).

M. R. Jovanović is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: mihailo@umn.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2014.2314223

Since vehicles have to maintain agreement in the presence of uncertainty, it is important to study the robustness of consensus. Several authors have recently used the steady-state variance of the deviation from consensus to characterize fundamental performance limitations in stochastically forced networks [8]–[14].

In this paper, we consider undirected consensus networks with two groups of nodes. Ordinary nodes, the so-called *followers*, form their action using relative information exchange with their neighbors; special nodes, the so-called *leaders*, also have access to their own states. This setting may arise in the control of vehicular formations and in distributed localization in sensor networks. In vehicular formations, all vehicles are equipped with ranging devices (that provide information about relative distances with respect to their neighbors), and the leaders additionally have GPS devices (that provide information with respect to a global frame of reference).

We are interested in assigning a pre-specified number of nodes as leaders in order to minimize the mean-square deviation from consensus. For undirected networks in which all nodes are subject to stochastic disturbances, we show that the Boolean constraints (which indicates whether a node is either a leader or it is not) are the only source of nonconvexity. The combinatorial nature of these constraints makes the determination of the global minimum challenging. Instead, we focus on computing lower and upper bounds on the global optimal value. A convex relaxation of Boolean constraints is used to obtain a lower bound, and a greedy algorithm is used to obtain an upper bound and to identify leaders. We show that the convex relaxation can be formulated as a semidefinite program (SDP) which can be solved efficiently for small networks. We also develop an efficient customized interior point method that is well-suited for large-scale problems. Furthermore, we improve the performance of one-leader-at-a-time (greedy) approach using a procedure that checks for possible swaps between leaders and followers. In both steps, algorithmic complexity is significantly reduced by exploiting the structure of low-rank modifications to Laplacian matrices. The computational efficiency of our algorithms makes them well-suited for establishing achievable performance bounds for leader selection problem in large stochastically forced networks.

Following [15]–[18], we also examine consensus networks in which leaders follow desired trajectories at all times. For consensus networks with at least one leader, adding leaders always improves performance [15]. In view of this, a greedy algorithm that selects one leader at a time by assigning the node that leads to the largest performance improvement as a leader was proposed in [15]. Furthermore, it was proved in [17] that the mean-square deviation from consensus is a supermodular

function of the set of noise-free leaders. Thus, the supermodular optimization framework in conjunction with the greedy algorithm can be used to provide selection of leaders that is within a provable bound from globally optimal solution [17].

In contrast to [15]–[18], we use convex optimization to quantify performance bounds for the noise-free leader selection problem. While we show that the leader selection is additionally complicated by the presence of a nonconvex rank constraint, we obtain an SDP relaxation by dropping the rank constraint and by relaxing the aforementioned Boolean constraints. Furthermore, we exploit the separable structure of the resulting constraint set and develop an efficient algorithm based on the alternating direction method of multipliers (ADMM). As in the noise-corrupted problem, we use a greedy approach followed by a swap procedure to compute an upper bound and to select leaders. In both steps, we exploit the properties of low-rank modifications to Laplacian matrices to reduce computational complexity. We have implemented the developed algorithms in Matlab and made them publicly available at [www.ece.umn.edu/~mihailo/software/leaders](http://www.ece.umn.edu/~mihailo/software/leaders).

Several recent efforts have focused on characterizing graph-theoretic conditions for controllability of networks in which a pre-specified number of leaders act as control inputs [19]–[24]. In contrast, our objective is to identify leaders that are most effective in minimizing the deviation from consensus in the presence of disturbances. Alternative performance indices for the selection of leaders have been also recently examined in [23], [25], and [26]. Other related work on augmenting topologies of networks to improve their algebraic connectivity includes [27], [28].

After our original submission, the leader selection problem has been studied in [29]–[32]. In [29], an information centrality measure was employed to provide an ordering of nodes. Leveraging results from [29], an information-centrality-based formulation was provided in [30] for the optimal selection of both noise-corrupted and noise-free leaders. In [31], upper and lower bounds on the convergence rate to the consensus value were derived using the maximum distance from leaders to followers. In [32], the joint problem of choosing a set of controlling agents and of designing the corresponding control inputs has been studied.

We finally comment on the necessity of considering two different problem formulations. The noise-free leader selection problem, aimed at identifying influential nodes in undirected networks, was originally formulated in [15] and consequently studied in [16]–[18], and [33]. To the best of our knowledge, the noise-corrupted leader selection problem first appeared in a preliminary version of this work [34]. The noise-corrupted formulation is introduced for two reasons: First, it is well-suited for applications where a certain number of nodes are to be equipped with additional capabilities (e.g., the GPS devices) in order to improve the network's performance; for example, this setup may be encountered in vehicular formation and sensor localization problems. And, second, in contrast to the noise-free formulation, the Boolean constraints are the only source of nonconvexity in the noise-corrupted problem; consequently, a convex relaxation in this case is readily obtained by enlarging Boolean constraints to their convex hull. Even though these for-

mulations have close connections in a certain limit, the differences between them are significant enough to warrant separate treatment. As we show in Sections III and IV, the structure of the corresponding optimization problems necessitates separate convex relaxations and the development of different customized optimization algorithms. The noise-free and noise-corrupted setups are thus of independent interest from the application, problem formulation, and algorithmic points of view.

The paper is organized as follows. In Section II, we formulate the problem and establish connections between the leader selection and the sensor selection problems. In Section III, we develop efficient algorithms to compute lower and upper bounds on the global optimal value for the noise-corrupted leader selection problem. In Section IV, we provide an SDP relaxation of the noise-free formulation and employ the ADMM algorithm to deal with large-scale problems. We conclude the paper with a summary of our contributions in Section V.

## II. PROBLEM FORMULATION

In this section, we formulate the noise-corrupted and noise-free leader selection problems in consensus networks and make connections to sensor selection in distributed localization problems. Furthermore, we establish an equivalence between the two problem formulations when all leaders use arbitrarily large feedback gains on their own states.

### A. Leader Selection in Consensus Networks

We consider networks in which each node updates a scalar state  $\psi_i$

$$\dot{\psi}_i = u_i + w_i, \quad i = 1, \dots, n$$

where  $u_i$  is the control input and  $w_i$  is the white stochastic disturbance with zero-mean and unit-variance. A node is a *follower* if it uses *only* relative information exchange with its neighbors to form its control action

$$u_i = -\sum_{j \in \mathcal{N}_i} (\psi_i - \psi_j).$$

A node is a *leader* if, in addition to relative information exchange with its neighbors, it also has access to its own state

$$u_i = -\sum_{j \in \mathcal{N}_i} (\psi_i - \psi_j) - \kappa_i \psi_i.$$

Here,  $\kappa_i$  is a positive number and  $\mathcal{N}_i$  is the set of all nodes that node  $i$  communicates with.

The control objective is to strategically deploy leaders in order to reduce the variance amplification in stochastically forced consensus networks. The communication network is modeled by a connected, undirected graph; thus, the graph Laplacian  $L$  is a symmetric positive semidefinite matrix with a single eigenvalue at zero and the corresponding eigenvector  $\mathbf{1}$  of all ones [1]. A state-space representation of the leader-follower consensus network is therefore given by

$$\dot{\psi} = -(L + D_\kappa D_x)\psi + w \quad (1)$$

where  $\mathcal{E}(w(t)w^T(\tau)) = I\delta(t - \tau)$ ,  $\mathcal{E}(\cdot)$  is the expectation operator, and

$$D_\kappa := \text{diag}(\kappa), \quad D_x := \text{diag}(x)$$

are diagonal matrices formed from the vectors  $\kappa = [\kappa_1 \cdots \kappa_n]^T$  and  $x = [x_1 \cdots x_n]^T$ . Here,  $x$  is a Boolean-valued vector with its  $i$ th entry  $x_i \in \{0, 1\}$ , indicating that node  $i$  is a leader if  $x_i = 1$  and that node  $i$  is a follower if  $x_i = 0$ . In connected networks with at least one leader,  $L + D_\kappa D_x$  is a positive definite matrix [21]. The steady-state covariance matrix of  $\psi$

$$\Sigma := \lim_{t \rightarrow \infty} \mathcal{E}(\psi(t)\psi^T(t))$$

can thus be determined from the Lyapunov equation

$$(L + D_\kappa D_x)\Sigma + \Sigma(L + D_\kappa D_x) = I$$

whose unique solution is given by

$$\Sigma = \frac{1}{2}(L + D_\kappa D_x)^{-1}.$$

Following [10], [13], we use the total steady-state variance

$$\text{trace}(\Sigma) = \frac{1}{2}\text{trace}((L + D_\kappa D_x)^{-1}) \quad (2)$$

to quantify performance of stochastically forced consensus networks.

We are interested in identifying  $N_l$  leaders that are most effective in reducing the steady-state variance (2). For an *a priori* specified number of leaders  $N_l < n$ , the leader selection problem can thus be formulated as

$$\begin{aligned} & \underset{x}{\text{minimize}} && J(x) = \text{trace}((L + D_\kappa D_x)^{-1}) \\ & \text{subject to} && x_i \in \{0, 1\}, \quad i = 1, \dots, n \\ & && \mathbb{1}^T x = N_l. \end{aligned} \quad (\text{LS1})$$

In (LS1), the number of leaders  $N_l$  as well as the matrices  $L$  and  $D_\kappa$  are the problem data, and the vector  $x$  is the optimization variable. As we show in Section III, for a positive definite matrix  $L + D_\kappa D_x$ , the objective function  $J$  in (LS1) is a convex function of  $x$ . The challenging aspect of (LS1) comes from the nonconvex Boolean constraints  $x_i \in \{0, 1\}$ ; in general, finding the solution to (LS1) requires an intractable combinatorial search.

Since the leaders are subject to stochastic disturbances, we refer to (LS1) as the *noise-corrupted* leader selection problem. We also consider the selection of *noise-free* leaders which follow their desired trajectories at all times [15]. Equivalently, in coordinates that determine deviation from the desired trajectory, the state of every leader is identically equal to zero, and the network dynamics are thereby governed by the dynamics of the following:

$$\dot{\psi}_f = -L_f \psi_f + w_f.$$

Here,  $L_f$  is obtained from  $L$  by eliminating all rows and columns associated with the leaders. Thus, the problem of

selecting leaders that minimize the steady-state variance of  $\psi_f$  amounts to

$$\begin{aligned} & \underset{x}{\text{minimize}} && J_f(x) = \text{trace}(L_f^{-1}) \\ & \text{subject to} && x_i \in \{0, 1\}, \quad i = 1, \dots, n \\ & && \mathbb{1}^T x = N_l. \end{aligned} \quad (\text{LS2})$$

As in (LS1), the Boolean constraints  $x_i \in \{0, 1\}$  are nonconvex. Furthermore, as we demonstrate in Section IV-B, the objective function  $J_f$  in (LS2) is a nonconvex function of  $x$ .

We note that the noise-free leader selection problem (LS2) cannot be uncovered from the noise-corrupted leader selection problem (LS1) by setting the variance of disturbances (that act on noise-corrupted leaders) to zero. Even when leaders are not directly subject to disturbances, their interactions with followers would prevent them from perfectly following their desired trajectories. In what follows, we establish the equivalence between the noise-corrupted and noise-free leader selection problems (LS1) and (LS2) in the situations when all noise-corrupted leaders use arbitrarily large feedback gains on their own states. Specifically, for white in time stochastic disturbance  $w$  with unit variance, the variance of noise-corrupted leaders in (1) decreases to zero as feedback gains on their states increase to infinity; see Appendix I-A.

## B. Connections to the Sensor Selection Problem

The problem of estimating a vector  $\psi \in \mathbb{R}^n$  from  $m$  relative measurements that are corrupted by additive white noise

$$y_{ij} = \psi_i - \psi_j + w_{ij}$$

arises in distributed localization in sensor networks. We consider the simplest scenario in which all  $\psi_i$ 's are scalar-valued, with  $\psi_i$  denoting the position of sensor  $i$ ; see [8], [9] for vector-valued localization problems. Let  $\mathcal{I}_r$  denote the index set of the  $m$  pairs of distinct nodes between which the relative measurements are taken and let  $e_{ij}$  belong to  $\mathbb{R}^n$  with 1 and  $-1$  at its  $i$ th and  $j$ th elements, respectively, and zero everywhere else. Then

$$y_{ij} = e_{ij}^T \psi + w_{ij}, \quad (i, j) \in \mathcal{I}_r$$

or, equivalently in the matrix form

$$y_r = E_r^T \psi + w_r \quad (3)$$

where  $y_r$  is the vector of relative measurements and  $E_r \in \mathbb{R}^{n \times m}$  is the matrix whose columns are determined by  $e_{ij}$  for  $(i, j) \in \mathcal{I}_r$ . Since  $\psi + a\mathbb{1}$  for any scalar  $a$  results in the same  $y_r$ , use of relative measurements provides estimate of the position vector  $\psi$  only up to an additive constant. This can be also verified by noting that  $E_r^T \mathbb{1} = 0$ .

Suppose that  $N_l$  sensors can be equipped with GPS devices that allow them to measure their absolute positions

$$y_a = E_a^T \psi + E_a^T w_a$$

where  $E_a \in \mathbb{R}^{n \times N_l}$  is the matrix whose columns are determined by  $e_i$ , the  $i$ th unit vector in  $\mathbb{R}^n$ , for  $i \in \mathcal{I}_a$ , the index set of absolute measurements. Then the vector of all measurements is given by

$$\begin{bmatrix} y_r \\ y_a \end{bmatrix} = \begin{bmatrix} E_r^T \\ E_a^T \end{bmatrix} \psi + \begin{bmatrix} I & 0 \\ 0 & E_a^T \end{bmatrix} \begin{bmatrix} w_r \\ w_a \end{bmatrix} \quad (4)$$

where  $w_r$  and  $w_a$  are zero-mean white stochastic disturbances with

$$\mathcal{E}(w_r w_r^T) = W_r, \quad \mathcal{E}(w_a w_a^T) = W_a, \quad \mathcal{E}(w_r w_a^T) = 0.$$

In Appendix I-B, we show that the problem of choosing  $N_l$  absolute position measurements among  $n$  sensors to minimize the variance of the estimation error is equivalent to the noise-corrupted leader selection problem (LS1). Furthermore, when the positions of  $N_l$  sensors are known *a priori* we show that the problem of assigning  $N_l$  sensors to minimize the variance of the estimation error amounts to solving the noise-free leader selection problem (LS2).

### III. LOWER AND UPPER BOUNDS ON GLOBAL PERFORMANCE: NOISE-CORRUPTED LEADERS

In this section, we show that the objective function  $J$  in the noise-corrupted leader selection problem (LS1) is convex. Convexity of  $J$  is utilized to develop efficient algorithms for computation of lower and upper bounds on the global optimal value (LS1). A lower bound results from convex relaxation of Boolean constraints in (LS1) which yields an SDP that can be solved efficiently using a customized interior point method. On the other hand, an upper bound is obtained using a greedy algorithm that selects one leader at a time. Since the greedy algorithm introduces low-rank modifications to Laplacian matrices, we exploit this feature in conjunction with the matrix inversion lemma to gain computational efficiency. Finally, we provide two examples to illustrate performance of the developed approach.

#### A. Convex Relaxation to Obtain a Lower Bound

Since the objective function  $J$  in (LS1) is the composition of a convex function  $\text{trace}(\bar{L}^{-1})$  of a positive definite matrix  $\bar{L} \succ 0$  with an affine function  $\bar{L} := L + D_\kappa D_x$ , it follows that  $J$  is a convex function of  $x$ . By enlarging the Boolean constraint set  $x_i \in \{0, 1\}$  to its convex hull  $x_i \in [0, 1]$ , we obtain the following convex relaxation of (LS1):

$$\begin{aligned} & \underset{x}{\text{minimize}} && J(x) = \text{trace}((L + D_\kappa D_x)^{-1}) \\ & \text{subject to} && \mathbf{1}^T x = N_l, \quad 0 \leq x_i \leq 1, \quad i = 1, \dots, n. \end{aligned} \quad (\text{CR1})$$

Since we have enlarged the constraint set, the solution  $x^*$  of the relaxed problem (CR1) provides a lower bound on  $J_{\text{opt}}$ . However,  $x^*$  may not provide a selection of  $N_l$  leaders, as it may not be Boolean-valued. If  $x^*$  is Boolean-valued, then it is the global solution of (LS1).

The Schur complement can be used to formulate the optimization problem (CR1) as an SDP

$$\begin{aligned} & \underset{X, x}{\text{minimize}} && \text{trace}(X) \\ & \text{subject to} && \begin{bmatrix} X & I \\ I & L + D_\kappa D_x \end{bmatrix} \succeq 0 \\ & && \mathbf{1}^T x = N_l, \quad 0 \leq x_i \leq 1, \quad i = 1, \dots, n. \end{aligned}$$

For small networks (e.g.,  $n \leq 30$ ), this problem can be solved efficiently using standard SDP solvers. For large networks, we develop a customized interior point method in Appendix I-C.

#### B. Greedy Algorithm to Obtain an Upper Bound

With the lower bound on the optimal value  $J_{\text{opt}}$  resulting from the convex relaxation (CR1), we next use a greedy algorithm to compute an upper bound on  $J_{\text{opt}}$ . This algorithm selects one leader at a time by assigning the node that provides the largest performance improvement as the leader. Once this is done, an attempt to improve a selection of  $N_l$  leaders is made by checking possible swaps between the leaders and the followers. In both steps, we show that substantial improvement in algorithmic complexity can be achieved by exploiting structure of the low-rank modifications to Laplacian matrices.

1) *One-Leader-at-a-Time Algorithm*: As the name suggests, we select one leader at a time by assigning the node that results in the largest performance improvement as the leader. For  $i = 1, \dots, n$ , we compute

$$J_1^i = \text{trace}((L + \kappa_i e_i e_i^T)^{-1})$$

and assign the node, say  $v_1$ , that achieves the minimum value of  $\{J_1^i\}$  as the first leader. If two or more nodes provide the optimal performance, we select one of these nodes as a leader. After choosing  $s$  leaders,  $v_1, \dots, v_s$ , we compute

$$\begin{aligned} J_{s+1}^i &= \text{trace}((L_s + \kappa_i e_i e_i^T)^{-1}) \\ L_s &:= L + \kappa_{v_1} e_{v_1} e_{v_1}^T + \dots + \kappa_{v_s} e_{v_s} e_{v_s}^T \end{aligned}$$

for  $i \notin \{v_1, \dots, v_s\}$ , and select node  $v_{s+1}$  that yields the minimum value of  $\{J_{s+1}^i\}$  as the  $(s+1)$ th leader. This procedure is repeated until all  $N_l$  leaders are selected.

Without exploiting structure, the above procedure requires  $O(n^4 N_l)$  operations. On the other hand, the rank-1 update formula resulting from the matrix inversion lemma

$$(L_s + \kappa_i e_i e_i^T)^{-1} = L_s^{-1} - \frac{L_s^{-1} \kappa_i e_i e_i^T L_s^{-1}}{1 + \kappa_i e_i^T L_s^{-1} e_i} \quad (5)$$

yields

$$J_{s+1}^i = \text{trace}(L_s^{-1}) - \frac{\kappa_i \|(L_s^{-1})_i\|_2^2}{1 + \kappa_i (L_s^{-1})_{ii}}$$

Here,  $(L_s^{-1})_i$  is the  $i$ th column of  $L_s^{-1}$  and  $(L_s^{-1})_{ii}$  is the  $ii$ th entry of  $L_s^{-1}$ . To initiate the algorithm, we use the generalized rank-1 update [35]

$$L_1^{-1} = L^\dagger - (L^\dagger e_i) \mathbf{1}^T - \mathbf{1} (L^\dagger e_i)^T + \left( \left( \frac{1}{\kappa_i} \right) + e_i^T L^\dagger e_i \right) \mathbf{1} \mathbf{1}^T$$

which thereby yields

$$J_1^i = \text{trace}(L^\dagger) + n \left( \left( \frac{1}{\kappa_i} \right) + e_i^T L^\dagger e_i \right)$$

where  $L^\dagger$  denotes the pseudo-inverse of  $L$  (e.g., see [36])

$$L^\dagger = (L + \mathbf{1}\mathbf{1}^T/n)^{-1} - \mathbf{1}\mathbf{1}^T/n.$$

Therefore, once  $L_s^{-1}$  is determined, the matrix inverse on the left-hand-side of (5) can be computed using  $O(n^2)$  operations and  $J_{s+1}^i$  can be evaluated using  $O(n)$  operations. Overall,  $N_l$  rank-1 updates,  $nN_l/2$  objective function evaluations, and one full matrix inverse (for computing  $L_s^{-1}$ ) require  $O(n^2N_l + n^3)$  operations as opposed to  $O(n^4N_l)$  operations without exploiting the low-rank structure. In large-scale networks, further computational advantage may be gained by exploiting structure of the underlying Laplacian matrices; e.g., see [37].

2) *Swap Algorithm*: After leaders are selected using the one-leader-at-a-time algorithm, we swap one of the  $N_l$  leaders with one of the  $n - N_l$  followers, and check if such a swap leads to a decrease in  $J$ . If no decrease occurs for all  $(n - N_l)N_l$  swaps, the algorithm terminates; if a decrease in  $J$  occurs, we update the set of leaders and then check again the possible  $(n - N_l)N_l$  swaps for the new leader selection. A similar swap procedure has been used as an effective means for improving performance of combinatorial algorithms encountered in graph partitioning [38], sensor selection [39], and community detection problems [40].

Since a swap between a leader  $i$  and a follower  $j$  leads to a rank-2 modification (6) to the matrix  $\bar{L} := L + D_\kappa D_x$ , we can exploit this low-rank structure to gain computational efficiency. Using the matrix inversion lemma, we have

$$\begin{aligned} & (\bar{L} - \kappa_i e_i e_i^T + \kappa_j e_j e_j^T)^{-1} \\ &= \bar{L}^{-1} - \bar{L}^{-1} \bar{E}_{ij} (I_2 + E_{ij}^T \bar{L}^{-1} \bar{E}_{ij})^{-1} E_{ij}^T \bar{L}^{-1} \end{aligned} \quad (6)$$

where  $E_{ij} = [e_i \ e_j]$ ,  $\bar{E}_{ij} = [-\kappa_i e_i \ \kappa_j e_j]$ , and  $I_2$  is the  $2 \times 2$  identity matrix. Thus, the objective function after the swap between leader  $i$  and follower  $j$  is given by

$$J_{ij} = J - \text{trace} \left( (I_2 + E_{ij}^T \bar{L}^{-1} \bar{E}_{ij})^{-1} E_{ij}^T \bar{L}^{-2} \bar{E}_{ij} \right). \quad (7)$$

Here, we do not need to form the full matrix  $\bar{L}^{-2}$ , since

$$E_{ij}^T \bar{L}^{-2} \bar{E}_{ij} = \begin{bmatrix} -\kappa_i (\bar{L}^{-2})_{ii} & \kappa_j (\bar{L}^{-2})_{ij} \\ -\kappa_i (\bar{L}^{-2})_{ji} & \kappa_j (\bar{L}^{-2})_{jj} \end{bmatrix}$$

and the  $ij$ th entry of  $\bar{L}^{-2}$  can be computed by multiplying the  $i$ th row of  $\bar{L}^{-1}$  with the  $j$ th column of  $\bar{L}^{-1}$ . Thus, evaluation of  $J_{ij}$  takes  $O(n)$  operations and computation of the matrix inverse in (6) requires  $O(n^2)$  operations. Since the total number of swaps for large-scale networks can be large, we follow [39] and limit the maximum number of swaps with a linear function of the number of nodes  $n$ .

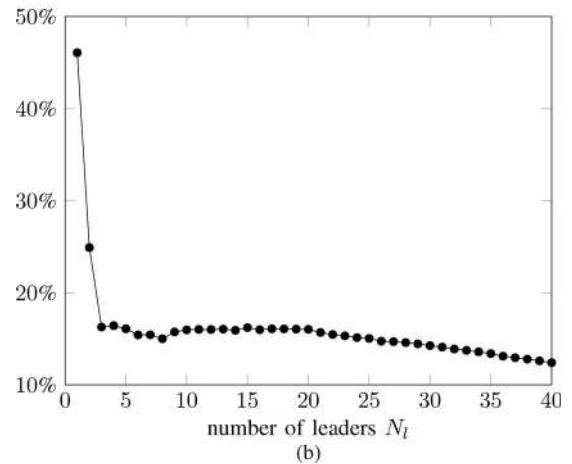
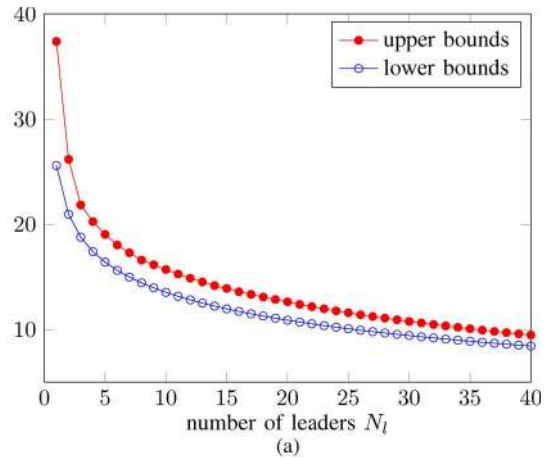


Fig. 1. Bounds on the global optimal value for noise-corrupted leader selection (LS1) for the random network example. (a) Upper and lower bounds,  $J_{up}$  and  $J_{low}$ , resulting from convex relaxation (CR1) and greedy algorithm, respectively. (b) Performance gap between upper and lower bounds, in fraction of lower bounds,  $100(J_{up} - J_{low})/J_{low}$ .

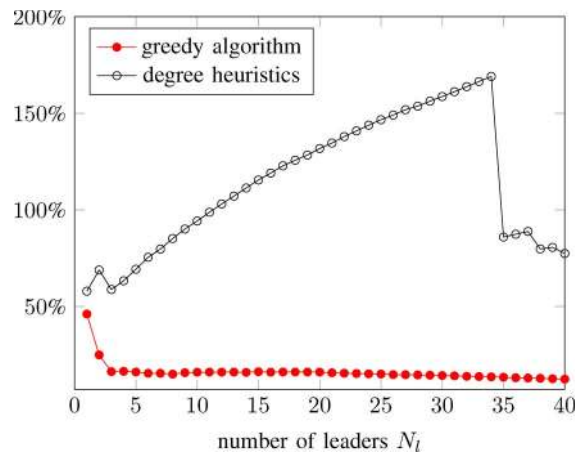


Fig. 2. The random network example: performance obtained using the greedy algorithm and the degree heuristics in fraction of lower bounds resulting from convex relaxation (CR1).

### C. Example

We next provide two examples to illustrate performance of developed methods.

1) *A Random Network Example*: We consider the selection of noise-corrupted leaders in a network with 100 randomly

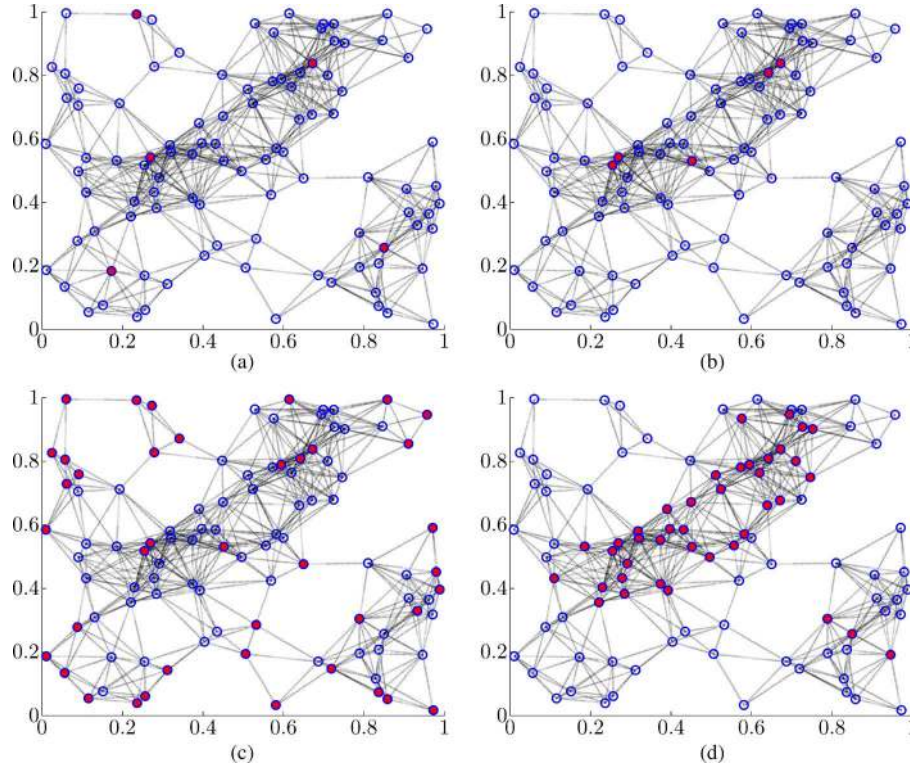


Fig. 3. Selection of leaders (●) for the random network example using greedy algorithm in (a) and (c) and using degree heuristics in (b) and (d). (a) Greedy algorithm:  $N_l = 5$ ,  $J = 19.0$ ; (b) Degree heuristics:  $N_l = 5$ ,  $J = 27.8$ ; (c) Greedy algorithm:  $N_l = 40$ ,  $J = 9.5$ ; (d) Degree heuristics:  $N_l = 40$ ,  $J = 15.0$ .

distributed nodes in a unit square. A pair of nodes communicate with each other if their distance is not greater than 0.2 units. This scenario may arise in sensor networks with prescribed omnidirectional (i.e., disk shape) sensing range [1], [41].

Fig. 1(a) shows lower bounds resulting from the convex relaxation (CR1) and upper bounds resulting from the greedy algorithm (i.e., the one-leader-at-a-time algorithm followed by the swap algorithm). Fig. 1(b) shows performance gap between upper and lower bounds, in fraction of lower bounds,  $100(J_{\text{up}} - J_{\text{low}})/J_{\text{low}}$ . For  $N_l = 1, \dots, 40$ , the number of swap updates ranges between 1 and 26 and the average number of swaps is 8.

As shown in Fig. 2, the greedy algorithm significantly outperforms the degree-heuristics-based-selection. To gain some insight into the selection of leaders, we compare results obtained using the greedy method with the degree heuristics. As shown in Fig. 3(b), the degree heuristics chooses nodes that turn out to be in the proximity of each other. In contrast, the greedy method selects leaders that, in addition to having large degrees, are far from each other; see Fig. 3(a). As a result, the selected leaders can influence more followers and thus more effectively improve the performance of the network.

The contrast between degree heuristics and greedy algorithms becomes even more dramatic for large number of leaders. As shown in Fig. 3(c) and 3(d), the leader sets obtained using the greedy algorithm and degree heuristics are almost complements of each other. While the degree heuristics clusters the leaders around the center of the network, the greedy algorithm distributes the leaders around the boundary of the network.

2) *A 2-D Lattice*: We next consider the noise-corrupted leader selection problem (LS1) for a 2-D regular lattice with 81 nodes. Fig. 4(a) shows lower bounds resulting from convex relaxation (CR1) and upper bounds resulting from the greedy algorithm. Fig. 4(b) shows the performance gap normalized by lower bounds,  $100(J_{\text{up}} - J_{\text{low}})/J_{\text{low}}$ . For  $N_l = 1, \dots, 40$ , the number of swap updates ranges between 1 and 19 and the average number of swaps is 10.

Fig. 5 shows selection of leaders resulting from the greedy algorithm for different choices of  $N_l$ . For  $N_l = 1$ , the center node (5, 5) provides the optimal selection of a single leader. As  $N_l$  increases, nodes away from the center node are selected; for example, for  $N_l = 2$ , nodes  $\{(3, 3), (7, 7)\}$  are selected and for  $N_l = 3$ , nodes  $\{(2, 6), (6, 2), (8, 8)\}$  are selected. Selection of nodes farther away from the center becomes more significant for  $N_l = 4$  and  $N_l = 8$ .

As shown in Fig. 5, the selection of leaders exhibits symmetry with respect to the center of the lattice. In particular, when  $N_l$  is large, almost uniform spacing between the leaders is observed; see Fig. 5(f) for  $N_l = 31$ . This is in contrast to the random network example where boundary nodes were selected as leaders; see Fig. 3(c).

#### IV. LOWER AND UPPER BOUNDS ON GLOBAL PERFORMANCE: NOISE-FREE LEADERS

We now turn our attention to the noise-free leader selection problem (LS2). An explicit expression for the objective function  $J_f$  that we develop in (LS2) allows us to identify the source of nonconvexity and to suggest a convex relaxation.

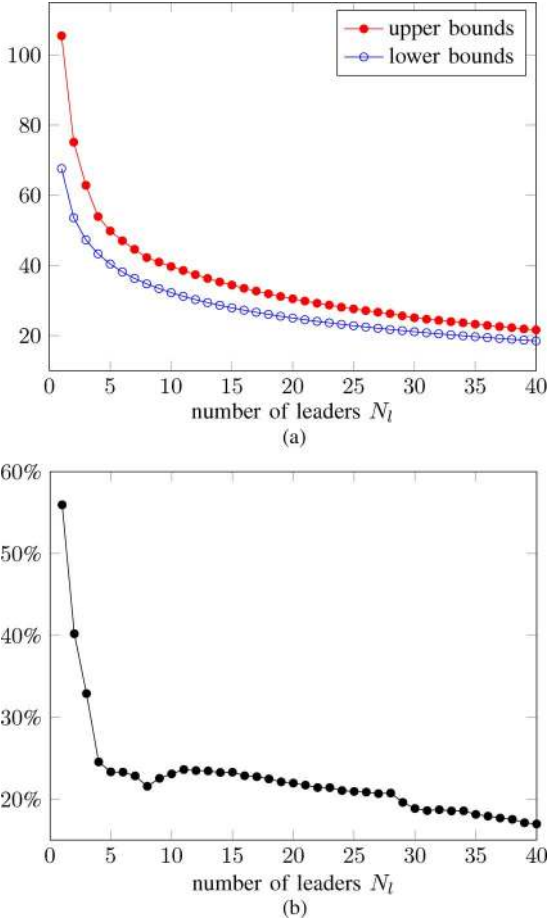


Fig. 4. Bounds on the global optimal value for noise-corrupted leader selection (LS1) for a 2-D lattice. (a) Upper and lower bounds,  $J_{\text{up}}$  and  $J_{\text{low}}$ , resulting from convex relaxation (CR1) and greedy algorithm, respectively. (b) Performance gap between upper and lower bounds, in fraction of lower bounds,  $100(J_{\text{up}} - J_{\text{low}})/J_{\text{low}}$ .

The resulting convex relaxation, which comes in the form of a semidefinite program, is used to obtain a lower bound on the global optimal value of (LS2). In order to increase computational efficiency, we employ the alternating direction method of multipliers to decompose the relaxed problem into a sequence of subproblems that can be solved efficiently. We also use the greedy algorithm to compute an upper bound and to identify noise-free leaders. As in the noise-corrupted leader selection problem, we take advantage of low-rank modifications to Laplacian matrices to reduce computational complexity. An example from sensor networks is provided to illustrate performance of the developed approach.

#### A. An Explicit Expression for the Objective Function $J_f$ in (LS2)

Since the objective function  $J_f$  in (LS2) is not expressed explicitly in terms of the optimization variable  $x$ , it is difficult to examine its basic properties (including convexity). In Proposition 1, we provide an alternative expression for  $J_f$  that allows us to establish the lack of convexity and to suggest a convex relaxation of  $J_f$ .

*Proposition 1:* For networks with at least one leader, the objective function  $J_f$  in the noise-free leader selection problem (LS2) can be written as

$$J_f = \text{trace} \left( (L \circ ((\mathbf{1} - x)(\mathbf{1} - x)^T) + \text{diag}(x))^{-1} \right) - \mathbf{1}^T x \quad (8)$$

where  $\circ$  denotes the elementwise multiplication of matrices.

*Proof:* Let the graph Laplacian  $L$  be partitioned into  $2 \times 2$  block matrices which, respectively, correspond to the set of leaders and the set of followers

$$L = \begin{bmatrix} L_l & L_0 \\ L_0^T & L_f \end{bmatrix}. \quad (9)$$

Furthermore, let the Boolean-valued vector  $x$  be partitioned conformably

$$x := \begin{bmatrix} \mathbf{1}_{N_l}^T & 0_{N_f}^T \end{bmatrix}^T \quad (10)$$

where  $\mathbf{1}_{N_l}$  is an  $N_l$ -vector with all ones,  $0_{N_f}$  is an  $N_f$ -vector with all zeros, and

$$N_f := n - N_l$$

is the number of followers. The elementwise multiplication of matrices can be used to set the rows and columns of  $L$  that correspond to leaders to zero

$$\begin{aligned} L \circ ((\mathbf{1} - x)(\mathbf{1} - x)^T) &= \begin{bmatrix} L_l & L_0 \\ L_0^T & L_f \end{bmatrix} \circ \begin{bmatrix} 0_{N_l \times N_l} & 0_{N_l \times N_f} \\ 0_{N_f \times N_l} & \mathbf{1}_{N_f \times N_f} \end{bmatrix} \\ &= \begin{bmatrix} 0_{N_l \times N_l} & 0_{N_l \times N_f} \\ 0_{N_f \times N_l} & L_f \end{bmatrix}. \end{aligned}$$

Using this expression and the definition of the vector  $x$  in (10) we obtain

$$(L \circ ((\mathbf{1} - x)(\mathbf{1} - x)^T) + \text{diag}(x))^{-1} = \begin{bmatrix} I_{N_l \times N_l} & 0_{N_l \times N_f} \\ 0_{N_f \times N_l} & L_f^{-1} \end{bmatrix}. \quad (11)$$

Finally, taking trace of (11) and subtracting  $\mathbf{1}^T x = N_l$  yields the desired result (8).

Thus, the noise-free leader selection problem (LS2) can be formulated as

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & J_f(x) = \text{trace} \left( (L \circ ((\mathbf{1} - x)(\mathbf{1} - x)^T) \right. \\ & \left. + \text{diag}(x))^{-1} \right) - N_l \\ \text{subject to} \quad & x_i \in \{0, 1\}, \quad i = 1, \dots, n \\ & \mathbf{1}^T x = N_l \end{aligned} \quad (\text{LS2}') \quad (12)$$

where the constraint  $\mathbf{1}^T x = N_l$  is used to obtain the expression for the objective function  $J_f$  in (LS2'). An example can be provided to demonstrate the lack of convexity of  $J_f(x)$ . In fact, it turns out that  $J_f$  is not convex even if all  $x_i$ 's are restricted to the interval  $[0, 1]$ . In Section IV-B, we introduce a change of variables to show that the lack of convexity of  $J_f$  can be equivalently recast as a rank constraint.

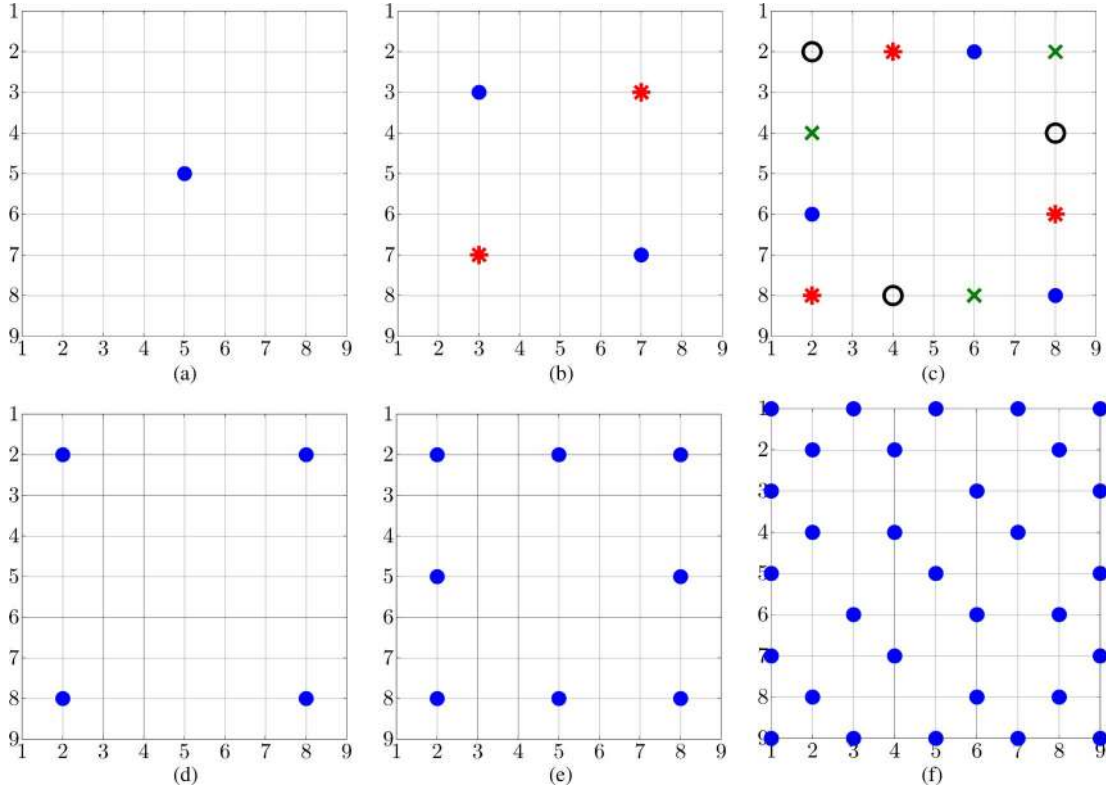


Fig. 5. Selections of noise-corrupted leaders (●) obtained using the one-at-a-time algorithm followed by the swap algorithm for a 2-D lattice. (b) The two selections of two leaders denoted by (●) and (\*) provide the same objective function  $J$ . (c) The four selections of three leaders denoted by (●), (\*), (×), and (○) provide the same  $J$ . (a)  $N_l = 1$ ,  $J = 105.5$ ; (b)  $N_l = 2$ ,  $J = 75.2$ ; (c)  $N_l = 3$ ,  $J = 62.9$ ; (d)  $N_l = 4$ ,  $J = 53.9$ ; (e)  $N_l = 8$ ,  $J = 42.3$ ; (f)  $N_l = 31$ ,  $J = 24.7$ .

**B. Reformulation and Convex Relaxation of (LS2')**

By introducing a new variable  $y := \mathbb{1} - x$ , we can rewrite (LS2') as

$$\begin{aligned} & \underset{Y,y}{\text{minimize}} \quad J_f(Y,y) = \text{trace} \left( (L \circ Y + \text{diag}(\mathbb{1} - y))^{-1} \right) - N_l \\ & \text{subject to} \quad Y = yy^T \\ & \quad \quad \quad y_i \in \{0, 1\}, \quad i = 1, \dots, n \\ & \quad \quad \quad \mathbb{1}^T y = N_f. \end{aligned}$$

Since  $Y := yy^T$ , it follows that  $Y$  is a Boolean-valued matrix with  $\mathbb{1}^T Y \mathbb{1} = N_f^2$ . Expressing these implicit constraints as

$$Y_{ij} \in \{0, 1\}, \quad i, j = 1, \dots, n, \quad \mathbb{1}^T Y \mathbb{1} = N_f^2$$

leads to the following equivalent formulation:

$$\begin{aligned} & \underset{Y,y}{\text{minimize}} \quad J_f(Y,y) = \text{trace} \left( (L \circ Y + \text{diag}(\mathbb{1} - y))^{-1} \right) - N_l \\ & \text{subject to} \quad Y = yy^T \\ & \quad \quad \quad y_i \in \{0, 1\}, \quad i = 1, \dots, n \\ & \quad \quad \quad Y_{ij} \in \{0, 1\}, \quad i, j = 1, \dots, n \\ & \quad \quad \quad \mathbb{1}^T y = N_f \\ & \quad \quad \quad \mathbb{1}^T Y \mathbb{1} = N_f^2. \end{aligned}$$

Since

$$Y = yy^T \iff \{Y \succeq 0, \text{rank}(Y) = 1\}$$

we drop the nonconvex rank constraint and enlarge the Boolean set  $\{0, 1\}$  to its convex hull  $[0, 1]$  to obtain the following convex relaxation of the leader selection problem (LS2)

$$\begin{aligned} & \underset{Y,y}{\text{minimize}} \quad J_f(Y,y) = \text{trace} \left( (L \circ Y + \text{diag}(\mathbb{1} - y))^{-1} \right) - N_l \\ & \text{subject to} \quad y_i \in [0, 1], \quad i = 1, \dots, n \\ & \quad \quad \quad Y_{ij} \in [0, 1], \quad i, j = 1, \dots, n \\ & \quad \quad \quad \mathbb{1}^T y = N_f \\ & \quad \quad \quad \mathbb{1}^T Y \mathbb{1} = N_f^2 \\ & \quad \quad \quad Y \succeq 0. \end{aligned} \tag{CR2}$$

The objective function in (CR2) is convex because it is a composition of a convex function  $\text{trace}(W^{-1})$  of a positive definite matrix  $W$  with an affine function  $W := L \circ Y + \text{diag}(\mathbb{1} - y)$  of  $Y$  and  $y$ . The constraint set for  $y$  is convex because it is the simplex set defined as

$$\mathcal{C}_1 := \{y | y_i \in [0, 1], \quad i = 1, \dots, n, \quad \mathbb{1}^T y = N_f\}. \tag{C1}$$

The constraint set for  $Y$  is also convex because it is the intersection of the simplex set

$$\mathcal{C}_2 := \{Y | Y_{ij} \in [0, 1], \quad i, j = 1, \dots, n, \quad \mathbb{1}^T Y \mathbb{1} = N_f^2\} \tag{C2}$$

and the positive semidefinite cone

$$\mathcal{C}_3 := \{Y | Y \succeq 0\}. \tag{C3}$$



Following a similar procedure to that in Section III-A, we use Schur complement to cast (CR2) as an SDP. Furthermore, since the constraints (C1)–(C3) are *decoupled* over  $y$  and  $Y$ , we exploit this *separable* structure in Section IV-C and develop an efficient algorithm to solve (CR2).

### C. Solving the Convex Relaxation (CR2) Using ADMM

For small networks (e.g.,  $n \leq 30$ ), the convex relaxation (CR2) can be solved using general-purpose SDP solvers, with computational complexity of order  $n^6$ . We next exploit the separable structure of the constraint set (C1)–(C3) and develop an alternative approach that is well-suited for large problems. In our approach, we use the alternating direction method of multipliers (ADMM) to decompose (CR2) into a sequence of subproblems which can be solved with computational complexity of order  $n^3$ .

Let  $\phi_1(y)$  be the indicator function of the simplex set in (C1)

$$\phi_1(y) := \begin{cases} 0, & y \in \mathcal{C}_1 \\ \infty, & y \notin \mathcal{C}_1. \end{cases}$$

Similarly, let  $\phi_2(Y)$  and  $\phi_3(Y)$  be the indicator functions of the simplex set (C2) and the positive semidefinite cone (C3), respectively. Then the convex relaxation (CR2) can be expressed as a sum of convex functions

$$\underset{Y,y}{\text{minimize}} \quad J_f(Y,y) + \phi_1(y) + \phi_2(Y) + \phi_3(Y).$$

We now introduce additional variables  $\{Z, z\}$  and rewrite (CR2) as

$$\begin{aligned} & \underset{Y,y;Z,z}{\text{minimize}} && f(Y,y) + g(Z,z) \\ & \text{subject to} && Z = Y, \quad z = y \end{aligned} \quad (12)$$

where

$$\begin{aligned} f(Y,y) &:= J_f(Y,y) + \phi_3(Y) \\ g(Z,z) &:= \phi_1(z) + \phi_2(Z). \end{aligned}$$

In (12),  $f$  and  $g$  are two independent functions over two different sets of variables  $\{Y, y\}$  and  $\{Z, z\}$ , respectively. As we describe below, this separable feature of the objective function in (12) in conjunction with the separability of the constraint set (C1)–(C3) is amenable to the application of the ADMM algorithm.

We form the augmented Lagrangian associated with (12)

$$\begin{aligned} \mathcal{L}_\rho(Y,y;Z,z;\Lambda,\lambda) &= f(Y,y) + g(Z,z) + \langle \Lambda, Y - Z \rangle \\ &+ \lambda^T (y - z) + \frac{\rho}{2} \|Y - Z\|_F^2 + \frac{\rho}{2} \|y - z\|_2^2 \end{aligned}$$

where  $\Lambda$  and  $\lambda$  are Lagrange multipliers,  $\rho$  is a positive scalar,  $\langle \cdot, \cdot \rangle$  is the inner product of two matrices,  $\langle M_1, M_2 \rangle := \text{trace}(M_1^T M_2)$ , and  $\|\cdot\|_F$  is the Frobenius norm. To find the

solution of (12), the ADMM algorithm uses a sequence of iterations

$$(Y,y)^{k+1} := \arg \min_{Y,y} \mathcal{L}_\rho(Y,y;Z^k,z^k;\Lambda^k,\lambda^k) \quad (13a)$$

$$(Z,z)^{k+1} := \arg \min_{Z,z} \mathcal{L}_\rho(Y^{k+1},y^{k+1};Z,z;\Lambda^k,\lambda^k) \quad (13b)$$

$$\Lambda^{k+1} := \Lambda^k + \rho(Y^{k+1} - Z^{k+1}) \quad (13c)$$

$$\lambda^{k+1} := \lambda^k + \rho(y^{k+1} - z^{k+1}) \quad (13d)$$

until the primal and dual residuals are sufficiently small [42, Section 3.3]

$$\|Y^{k+1} - Z^{k+1}\|_F + \|y^{k+1} - z^{k+1}\|_2 \leq \epsilon$$

$$\|Z^{k+1} - Z^k\|_F + \|z^{k+1} - z^k\|_2 \leq \epsilon.$$

The convergence of ADMM for convex problems is guaranteed under fairly mild conditions [42, Section 3.2]. Furthermore, for a fixed value of parameter  $\rho$ , a linear convergence rate of ADMM has been established in [43]. In practice, the convergence rate of ADMM can be improved by appropriately updating  $\rho$  to balance the primal and dual residuals; see [42, Section 3.4.1].

In what follows, we show that the  $(Y,y)$ -minimization step (13a) amounts to the minimization of a smooth convex function over the positive semidefinite cone  $\mathcal{C}_3$ . We use a gradient projection method to solve this problem. On the other hand, the  $(Z,z)$ -minimization step (13b) amounts to projections on simplex sets  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , both of which can be computed efficiently.

1)  $(Y,y)$ -Minimization Step: Using completion of squares, we express the  $(Y,y)$ -minimization problem (13a) as

$$\begin{aligned} & \underset{Y,y}{\text{minimize}} && h(Y,y) = \text{trace} \left( (L \circ Y + \text{diag}(\mathbb{1} - y))^{-1} \right) \\ & && + \frac{\rho}{2} \|Y - U^k\|_F^2 + \frac{\rho}{2} \|y - u^k\|_2^2 \\ & \text{subject to} && Y \succeq 0 \end{aligned} \quad (14)$$

where  $U^k := Z^k - (1/\rho)\Lambda^k$  and  $u^k := z^k - (1/\rho)\lambda^k$ . A gradient projection method is used to minimize the smooth convex function  $h$  in (14) over the positive semidefinite cone  $Y \succeq 0$ . This iterative descent scheme guarantees feasibility in each iteration [44, Section 2.3] by updating  $Y$  as follows:

$$Y^{r+1} = Y^r + s^r (\bar{Y}^r - Y^r). \quad (15)$$

Here, the scalar  $s^r$  is the stepsize of the  $r$ th gradient projection iteration and

$$\bar{Y}^r := [Y^r - \nabla_Y h]^+ \quad (16)$$

is the projection of the matrix  $Y^r - \nabla_Y h$  on the positive semidefinite cone  $\mathcal{C}_3$ . This projection can be obtained from an eigenvalue decomposition by replacing the negative eigenvalues with zero. On the other hand, since no constraints are imposed on  $y$ , it is updated using standard gradient descent

$$y^{r+1} = y^r - s^r \nabla_y h$$

where the stepsize  $s^r$  is the same as in (15) and it is obtained, e.g., using the Armijo rule [44, Section 2.3]. Here, we provide expressions for the gradient direction

$$\begin{aligned}\nabla_Y h &= -(L \circ Y + \text{diag}(\mathbb{1} - y))^{-2} \circ L + \rho(Y - U^k) \\ \nabla_y h &= \text{diag} \left( (L \circ Y + \text{diag}(\mathbb{1} - y))^{-2} \right) + \rho(y - u^k)\end{aligned}\quad (17)$$

and note that the KKT conditions for (14) are given by

$$Y \succeq 0, \quad \nabla_Y h \succeq 0, \quad \langle Y, \nabla_Y h \rangle = 0, \quad \nabla_y h = 0.$$

Thus, the gradient projection method terminates when  $(Y^r, y^r)$  satisfies

$$\begin{aligned}Y^r &\succeq 0, \quad \nabla_Y h(Y^r) \succeq 0 \\ \langle Y^r, \nabla_Y h(Y^r) \rangle &\leq \epsilon, \quad \|\nabla_y h(y^r)\|_2 \leq \epsilon.\end{aligned}$$

Finally, we note that each iteration of the gradient projection method takes  $O(n^3)$  operations. This is because the projection (16) on the positive semidefinite cone requires an eigenvalue decomposition and the gradient direction (17) requires computation of a matrix inverse.

2)  $(Z, z)$ -Minimization Step: We now turn to the  $(Z, z)$ -minimization problem (13b), which can be expressed as

$$\begin{aligned}\underset{Z, z}{\text{minimize}} \quad & \frac{\rho}{2} \|z - v^k\|_2^2 + \frac{\rho}{2} \|Z - V^k\|_F^2 \\ \text{subject to} \quad & z \in \mathcal{C}_1, \quad Z \in \mathcal{C}_2\end{aligned}\quad (18)$$

where  $V^k := Y^{k+1} + (1/\rho)\Lambda^k$  and  $v^k := y^{k+1} + (1/\rho)\lambda^k$ . The separable structure of (18) can be used to decompose it into two independent problems

$$\underset{z \in \mathcal{C}_1}{\text{minimize}} \quad \frac{\rho}{2} \|z - v^k\|_2^2 \quad (19a)$$

$$\underset{Z \in \mathcal{C}_2}{\text{minimize}} \quad \frac{\rho}{2} \|Z - V^k\|_2^2 \quad (19b)$$

whose solutions are determined by projections of  $v^k$  and  $V^k$  on convex sets  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , respectively.

In what follows, we focus on the projection on  $\mathcal{C}_1$ ; the projection on  $\mathcal{C}_2$  can be obtained in a similar fashion. For  $N_f = 1$ ,  $\mathcal{C}_1$  becomes a *probability simplex*,

$$\mathcal{C}_1 = \{z | z_i \in [0, 1], \quad i = 1, \dots, n, \quad \mathbb{1}^T z = 1\}$$

and customized algorithms for projection on probability simplex can be used; e.g., see [45] and [46, Section 6.2.5]. Since for  $N_f \geq 2$  these algorithms are not applicable, we view the simplex  $\mathcal{C}_1$  as the intersection of the hyperplane  $\{z | \mathbb{1}^T z = N_f\}$  and the unit box  $\{z | 0 \leq z \leq \mathbb{1}\}$  and employ an ADMM-based alternating projection method in conjunction with simple analytical expressions developed in [46, Section 6.2]; see Appendix I-D for details.

#### D. Greedy Algorithm to Obtain an Upper Bound

Having determined a lower bound on the global optimal value of (LS2) by solving the convex relaxation (CR2), we next quantify the performance gap and provide a computationally

attractive way for selecting leaders. As in the noise-corrupted case, we use the one-leader-at-a-time algorithm followed by the swap algorithm to compute an upper bound. Rank-2 modifications to the resulting Laplacian matrices allow us to compute the inverse of  $L_f$  using  $O(n^2)$  operations.

Let  $[L]_i$  be the principal submatrix of  $L$  obtained by deleting its  $i$ th row and column. To select the first leader, we compute

$$J_1^i = \text{trace}([L]_i^{-1}), \quad i = 1, \dots, n$$

and assign the node, say  $v_1$ , that achieves the minimum value of  $\{J_1^i\}$ . After choosing  $s$  noise-free leaders  $\nu = \{v_1, \dots, v_s\}$ , we compute

$$J_{s+1}^i = \text{trace}([L]_{\nu \cup i}^{-1}), \quad i \notin \nu$$

and choose node  $v_{s+1}$  that achieves the minimum value of  $\{J_{s+1}^i\}$ . We repeat this procedure until all  $N_l$  leaders are selected.

For  $N_l \ll n$ , the one-at-a-time greedy algorithm that ignores the low-rank structure requires  $O(n^4 N_l)$  operations. We next exploit the low-rank structure to reduce complexity to  $O(n^3 N_l)$  operations. The key observation is that the difference between two *consecutive* principal submatrices  $[L]_i$  and  $[L]_{i+1}$  leads to a rank-2 matrix. To see this, let us partition the Laplacian matrix as

$$L = \begin{bmatrix} L_1 & c_i & c_{i+1} & L_0 \\ c_i^T & a_i & d_i & b_i^T \\ c_{i+1}^T & d_i & a_{i+1} & b_{i+1}^T \\ L_0^T & b_i & b_{i+1} & L_2 \end{bmatrix} \quad \begin{array}{l} \leftarrow i\text{th row} \\ \leftarrow (i+1)\text{th row} \end{array}$$

where the  $i$ th column of  $L$  consists of  $\{c_i, a_i, d_i, b_i\}$  and the  $(i+1)$ th column consists of  $\{c_{i+1}, d_i, a_{i+1}, b_{i+1}\}$ . Deleting the  $i$ th row and column and deleting the  $(i+1)$ th row and column respectively yields

$$[L]_i = \begin{bmatrix} L_1 & c_{i+1} & L_0 \\ c_{i+1}^T & a_{i+1} & b_{i+1}^T \\ L_0^T & b_{i+1} & L_2 \end{bmatrix}, \quad [L]_{i+1} = \begin{bmatrix} L_1 & c_i & L_0 \\ c_i^T & a_i & b_i^T \\ L_0^T & b_i & L_2 \end{bmatrix}. \quad (20)$$

Thus, the difference between two consecutive principal submatrices of  $L$  can be written as

$$[L]_{i+1} - [L]_i = e_i \xi_i^T + \xi_i e_i^T$$

where  $e_i$  is the  $i$ th unit vector and  $\xi_i^T := [c_i^T - c_{i+1}^T \ (1/2)(a_i - a_{i+1}) \ b_i^T - b_{i+1}^T]$ . Hence, once  $[L]_i^{-1}$  is determined, computing  $[L]_{i+1}^{-1}$  via matrix inversion lemma takes  $O(n^2)$  operations; cf. (6). The selection of the first leader requires one matrix inverse and  $n-1$  times rank-2 updates, resulting in  $O(n^3)$  operations. For  $N_l \ll n$ , the total cost of the greedy algorithm is thus reduced to  $O(n^3 N_l)$  operations.

As in Section III-B2, after selecting  $N_l$  leaders using the one-leader-at-a-time algorithm we employ the swap algorithm to further improve performance. Similar to the noise-corrupted case, a swap between a noise-free leader and a follower leads to a rank-2 modification to the reduced Laplacian  $L_f$ . Thus, after a swap, the evaluation of the objective function  $J_f$  can be carried out with  $O(n^2)$  operations. If  $L$  is partitioned as in (9), a swap between leader  $i$  and follower  $N_l + j$  amounts to

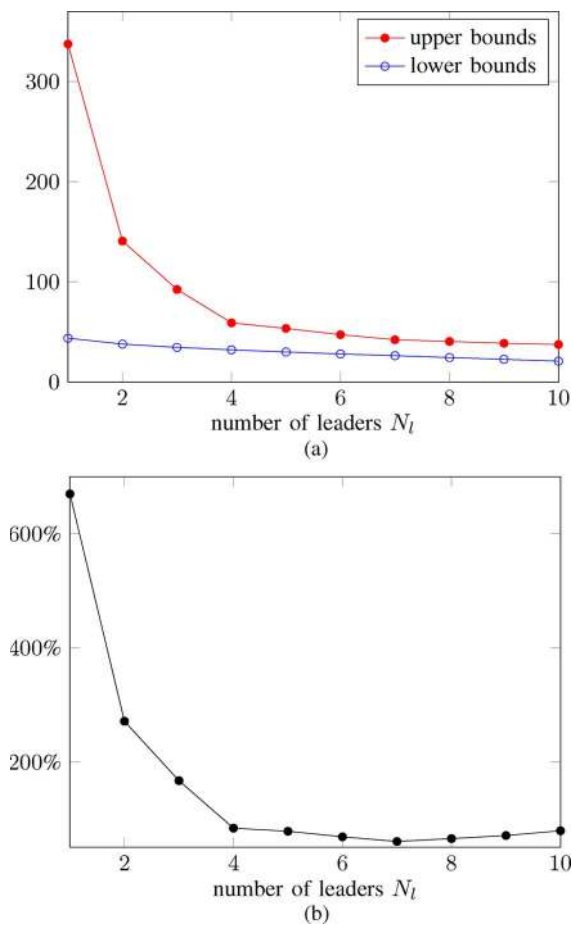


Fig. 6. Bounds on the global optimal value of the noise-free leader selection problem (LS2) for the network shown in Fig. 7. (a) Upper and lower bounds,  $J_{up}$  and  $J_{low}$ , resulting from convex relaxation (CR2) and greedy algorithm, respectively. (b) Performance gap between upper and lower bounds, in fraction of lower bounds,  $100(J_{up} - J_{low})/J_{low}$ .

replacing (i) the  $j$ th row of  $L_f$  with the  $i$ th row of  $L_0$ ; and (ii) the  $i$ th column of  $L_f$  with the  $i$ th column of  $L_0^T$ . Thus, a swap introduces a rank-2 modification to  $L_f$ .

E. An Example

We consider a network with 200 randomly distributed nodes in a C-shaped region within a unit square; see Fig. 7. A pair of nodes communicates with each other if their distance is not greater than 0.1 units. This example was used in [41] as a benchmark for testing algorithms for the sensor localization problem. Lower and upper bounds on the global optimal value of the noise-free leader selection problem (LS2) are computed using approaches developed in this section. The lower and upper bounds along with the normalized performance gap are shown in Fig. 6. For  $N_l = 1, \dots, 10$ , the number of the swap updates ranges from 1 to 16 and the average number of swaps is 8.

The greedy algorithm selects leaders that have large degrees and that are geographically far from each other; see Fig. 7. Similar leader selection strategies have been observed in the noise-corrupted case of Section III-C. For the C-shaped network, we note that the noise-free and noise-corrupted formulations lead to almost identical selection of leaders.

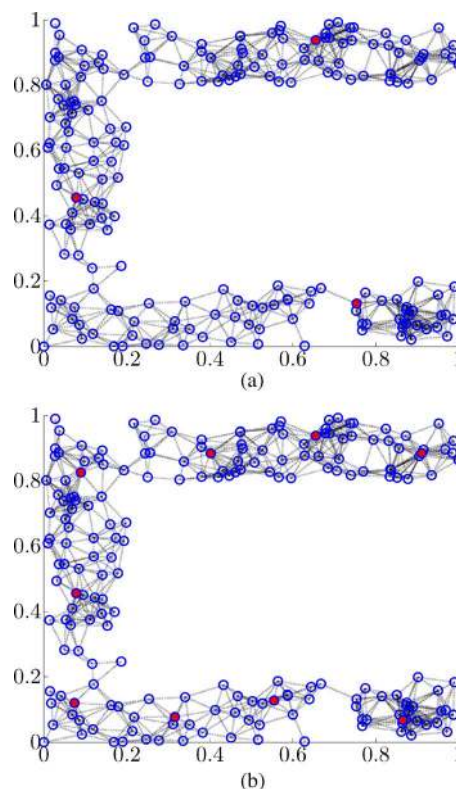


Fig. 7. Selection of noise-free leaders (●) using the greedy algorithm for the C-shaped network. (a)  $N_l = 3$ ,  $J_f = 92.5$ ; (b)  $N_l = 9$ ,  $J_f = 41.2$ .

V. CONCLUSION

The main contribution of this paper is the development of efficient algorithms for the selection of leaders in large stochastically forced consensus networks. For both noise-corrupted and noise-free formulations, we focus on computing lower and upper bounds on the global optimal value. Lower bounds are obtained by solving convex relaxations and upper bounds result from simple but efficient greedy algorithms.

Even though the convex relaxations can be cast as semidefinite programs and solved using general-purpose SDP solvers, we take advantage of the problem structure (such as separability of constraint sets) and develop customized algorithms for large-scale networks. We also improve the computational efficiency of greedy algorithms by exploiting the properties of low-rank modifications to Laplacian matrices. Several examples ranging from regular lattices to random networks are provided to illustrate the effectiveness of the developed algorithms.

We are currently applying the developed tools for leader selection in different types of networks, including small-world and social networks [47], [48]. Furthermore, the flexibility of our framework makes it well-suited for quantifying performance bounds and selecting leaders in problem formulations with alternative objective functions [23], [25]. An open question of theoretical interest is whether leaders can be selected based on the solutions of the convex relaxations (CR1) and (CR2). Since our computations suggest that the solution  $Y^* \succeq 0$  to (CR2) has a small number of dominant eigenvalues, it is of interest to quantify the level of conservatism of the lower bounds that result from these low-rank solutions and to

investigate scenarios under which (CR2) yields a rank-1 solution. The use of randomized algorithms [49], [50] may provide a viable approach to addressing the former question.

## APPENDIX I

### A. Connection Between Noise-Free and Noise-Corrupted Formulations

Partitioning  $\psi$  into the state of the leader nodes  $\psi_l$  and the state of the follower nodes  $\psi_f$  brings system (1) to the following form<sup>1</sup>

$$\begin{bmatrix} \dot{\psi}_l \\ \dot{\psi}_f \end{bmatrix} = - \begin{bmatrix} L_l + D_{\kappa_l} & L_0 \\ L_0^T & L_f \end{bmatrix} \begin{bmatrix} \psi_l \\ \psi_f \end{bmatrix} + \begin{bmatrix} w_l \\ w_f \end{bmatrix}. \quad (21)$$

Here,  $D_{\kappa_l} := \text{diag}(\kappa_l)$  with  $\kappa_l \in \mathbb{R}^{N_l}$  being the vector of feedback gains associated with the leaders. Taking the trace of the inverse of the  $2 \times 2$  block matrix in (21) yields

$$J = \text{trace} \left( L_f^{-1} + L_f^{-1} L_0^T S_{\kappa_l}^{-1} L_0 L_f^{-1} + S_{\kappa_l}^{-1} \right)$$

where

$$S_{\kappa_l} := L_l + D_{\kappa_l} - L_0 L_f^{-1} L_0^T$$

is the Schur complement of  $L_f$ . Since  $S_{\kappa_l}^{-1}$  vanishes as each component of the vector  $\kappa_l$  goes to infinity, the variance of the network in this case is determined by the variance of the followers,  $J_f = \text{trace}(L_f^{-1})$ . Here,  $L_f$  denotes the reduced Laplacian matrix obtained by removing all rows and columns that correspond to the leaders from  $L$ .

### B. Equivalence Between Leader Selection and Sensor Selection Problems

We next show that the problem of choosing  $N_l$  absolute position measurements among  $n$  sensors to minimize the variance of the estimation error in Section II-B is equivalent to the noise-corrupted leader selection problem (LS1).

Given the measurement vector  $y$  in (4), the linear minimum variance unbiased estimate of  $\psi$  is determined by [51, Chapter 4.4]

$$\hat{\psi} = \left( E_r W_r^{-1} E_r^T + E_a (E_a^T W_a E_a)^{-1} E_a^T \right)^{-1} \\ \times \left( E_r W_r^{-1} y_r + E_a (E_a^T W_a E_a)^{-1} y_a \right)$$

with the covariance of the estimation error

$$\Sigma = \mathcal{E} \left( (\psi - \hat{\psi})(\psi - \hat{\psi})^T \right) \\ = \left( E_r W_r^{-1} E_r^T + E_a (E_a^T W_a E_a)^{-1} E_a^T \right)^{-1}.$$

Furthermore, let us assume that  $W_r = I$  and  $W_a = D_{\kappa}^{-1}$ . The choice of  $W_a$  indicates that a larger value of  $\kappa_i$  corresponds to

a more accurate absolute measurement of sensor  $i$ . Then

$$(E_a^T W_a E_a)^{-1} = (E_a^T D_{\kappa}^{-1} E_a)^{-1} = E_a^T D_{\kappa} E_a$$

and thus

$$\Sigma = (E_r E_r^T + E_a E_a^T D_{\kappa} E_a E_a^T)^{-1}.$$

Since  $E_a E_a^T$  is a diagonal matrix with its  $i$ th diagonal element being 1 for  $i \in \mathcal{I}_a$  and  $E_r E_r^T$  is the Laplacian matrix of the relative measurement graph, it follows that:

$$D_x = E_a E_a^T, \quad L = E_r E_r^T, \\ \Sigma = (L + D_x D_{\kappa} D_x)^{-1} = (L + D_{\kappa} D_x)^{-1}$$

where  $D_x D_{\kappa} D_x = D_{\kappa} D_x$  because  $D_x$  and  $D_{\kappa}$  commute and  $D_x D_x = D_x$ . Therefore, we have established the equivalence between the noise-corrupted leader selection problem (LS1) and the problem of choosing  $N_l$  sensors with absolute position measurements such that the variance of the estimation error is minimized.

To formulate an estimation problem that is equivalent to the noise-free leader selection problem (LS2), we follow [8] and assume that the positions of  $N_l$  sensors are known *a priori*. Let  $\psi_l$  denote the positions of these *reference sensors* and let  $\psi_f$  denote the positions of the other sensors. We can thus write the relative measurement (3) as

$$y_r = E_r^T \psi + w_r = E_l^T \psi_l + E_f^T \psi_f + w_r$$

and the linear minimum variance unbiased estimate of  $\psi_f$  is given by

$$\hat{\psi}_f = (E_f E_f^T)^{-1} E_f W_r^{-1} (y_r - E_l^T \psi_l)$$

with covariance of the estimation error  $\Sigma_f = (E_f E_f^T)^{-1}$ . Identifying  $E_f E_f^T$  with  $L_f$  in the Laplacian matrix

$$L = E_r E_r^T = \begin{bmatrix} E_l E_l^T & E_l E_f^T \\ E_f E_l^T & E_f E_f^T \end{bmatrix} = \begin{bmatrix} L_l & L_0 \\ L_0^T & L_f \end{bmatrix}$$

establishes equivalence between problem (LS2) and the problem of assigning  $N_l$  sensors with known reference positions to minimize the variance of the estimation error of sensor network.

### C. Customized Interior Point Method for (CR1)

We begin by augmenting the objective function in (CR1) with log-barrier functions associated with the inequality constraints on  $x_i$

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & q(x) = \tau \text{trace} \left( (L + D_{\kappa} D_x)^{-1} \right) \\ & + \sum_{i=1}^n (-\log(x_i) - \log(1 - x_i)) \\ \text{subject to} \quad & \mathbf{1}^T x = N_l. \end{aligned} \quad (22)$$

As the positive scalar  $\tau$  increases to infinity, the solution of the approximate problem (22) converges to the solution of the convex relaxation (CR1) [52, Section 11.2]. We solve a

<sup>1</sup>Since the partition is performed with respect to the indices of the 0 and 1 elements of  $x$ , the matrix  $D_x$  does not show in (21).

sequence of problems (22) by gradually increasing  $\tau$ , and by starting each minimization using the solution from the previous value of  $\tau$ . We use Newton's method to solve (22) for a fixed  $\tau$ , and the Newton direction is given by

$$x_{\text{nt}} = -(\nabla^2 q)^{-1} \nabla q - \delta (\nabla^2 q)^{-1} \mathbf{1}$$

$$\delta := -\frac{\mathbf{1}^T (\nabla^2 q)^{-1} \nabla q}{\mathbf{1}^T (\nabla^2 q)^{-1} \mathbf{1}}.$$

Here, the expressions for the  $i$ th entry of the gradient direction  $\nabla q$  and for the Hessian matrix are given by

$$(\nabla q)_i = -\tau \kappa_i \left( (L + D_\kappa D_x)^{-2} \right)_{ii} - x_i^{-1} - (x_i - 1)^{-1}$$

$$\nabla^2 q = 2\tau \left( D_\kappa (L + D_\kappa D_x)^{-2} D_\kappa \right) \circ (L + D_\kappa D_x)^{-1}$$

$$+ \text{diag} \left( x_i^{-2} + (1 - x_i)^{-2} \right).$$

We next examine complexity of computing the Newton direction  $x_{\text{nt}}$ . The major cost of computing  $\nabla^2 q$  is to form  $(L + D_\kappa D_x)^{-2}$ , which takes  $(7/3)n^3$  operations to form  $(L + D_\kappa D_x)^{-1}$  and  $n^3$  operations to form  $(L + D_\kappa D_x)^{-2}$ . Computing  $x_{\text{nt}}$  requires solving two linear equations

$$(\nabla^2 q)y = -\nabla q, \quad (\nabla^2 q)z = -\mathbf{1}$$

which takes  $(1/3)n^3$  operations using Cholesky factorization. Thus, the computation of each Newton step requires  $(7/3 + 1 + 1/3)n^3 = (11/3)n^3$  operations. Finally, it is worth noting that better-conditioned linear systems for computing the Newton direction in the interior point method can be obtained using the approach suggested in [53, Chapter 19].

#### D. Solving (19a) Using ADMM

Since the solution of (19a) does not depend on the value of  $\rho$ , and since the constraint set is the intersection of the hyperplane and the unit box, we can express (19a) as

$$\begin{aligned} & \underset{z, w}{\text{minimize}} && \frac{1}{2} \|z - v^k\|_2^2 + \phi_4(z) + \phi_5(w) \\ & \text{subject to} && z - w = 0. \end{aligned} \quad (23)$$

Here,  $\phi_4$  and  $\phi_5$  are the indicator functions of the hyperplane  $\{z | \mathbf{1}^T z = N_f\}$  and the box  $\{w | 0 \leq w \leq \mathbf{1}\}$ , respectively. The augmented Lagrangian associated with (23) is given by

$$\begin{aligned} \mathcal{L}_\varrho(z, w, \lambda) = & \frac{1}{2} \|z - v^k\|_2^2 + \phi_4(z) + \phi_5(w) \\ & + \langle \lambda, z - w \rangle + \frac{\varrho}{2} \|z - w\|_2^2 \end{aligned}$$

and the ADMM algorithm uses the sequence of iterations

$$z^{s+1} := \arg \min_z \left( \frac{1}{2} \|z - v^k\|_2^2 + \phi_4(z) + \frac{\varrho}{2} \|z - (w^s - \lambda^s / \varrho)\|_2^2 \right) \quad (24a)$$

$$w^{s+1} := \arg \min_w \left( \phi_5(w) + \frac{\varrho}{2} \|w - (z^{s+1} + \lambda^s / \varrho)\|_2^2 \right) \quad (24b)$$

$$\lambda^{s+1} := \lambda^s + \varrho(z^{s+1} - w^{s+1}) \quad (24c)$$

until  $\|z^{s+1} - w^{s+1}\|_2 \leq \epsilon$  and  $\|w^{s+1} - w^s\|_2 \leq \epsilon$ . By solving the KKT conditions for (24a), we obtain an analytical solution

$$z^{s+1} = \frac{(\varrho w^s - \lambda^s + v^k - \eta \mathbf{1})}{(\varrho + 1)} \quad (25)$$

where the scalar  $\eta$  is given by

$$\eta = \frac{(\mathbf{1}^T (\varrho w^s - \lambda^s + v^k) - (\varrho + 1) N_f)}{n}.$$

On the other hand, the solution to (24b) is determined by the projection of  $\mu = z^{s+1} + \lambda^s / \varrho$  on the box  $\{w | 0 \leq w \leq \mathbf{1}\}$

$$w_i^{s+1} = \begin{cases} 1, & \mu_i > 1 \\ \mu_i, & 0 \leq \mu_i \leq 1 \\ 0, & \mu_i < 0. \end{cases} \quad (26)$$

Both the solution (25) and the projection (26) take  $O(n)$  operations.

#### REFERENCES

- [1] M. Mesbahi and M. Egerstedt, *Graph-theoretic Methods in Multiagent Networks*. Princeton, NJ: Princeton Univ. Press, 2010.
- [2] M. H. DeGroot, "Reaching a consensus," *J. Amer. Statist. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.
- [3] B. Golub and M. Jackson, "Naive learning social networks and the wisdom of crowds," *American Economic J.: Microecon.*, vol. 2, no. 1, pp. 112–149, 2010.
- [4] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *J. Parallel Distrib. Comput.*, vol. 7, no. 2, pp. 279–301, 1989.
- [5] J. E. Boillat, "Load balancing and Poisson equation in a graph," *Concurrency: Practice and Experience*, vol. 2, no. 4, pp. 289–313, 1990.
- [6] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [7] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *Automatica*, vol. 44, no. 3, pp. 671–684, 2007.
- [8] P. Barooah and J. P. Hespanha, "Estimation on graphs from relative measurements: Distributed algorithms and fundamental limits," *IEEE Control Syst. Mag.*, vol. 27, no. 4, pp. 57–74, 2007.
- [9] P. Barooah and J. P. Hespanha, "Estimation from relative measurements: Electrical analogy and large graphs," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2181–2193, 2008.
- [10] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, 2007.
- [11] G. F. Young, L. Scardovi, and N. E. Leonard, "Robustness of noisy consensus dynamics with directed communication," in *Proc. Amer. Control Conf.*, 2010, pp. 6312–6317.
- [12] D. Zelazo and M. Mesbahi, "Edge agreement: Graph-theoretic performance bounds and passivity analysis," *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 544–555, 2011.
- [13] B. Bamieh, M. R. Jovanović, P. Mitra, and S. Patterson, "Coherence in large-scale networks: Dimension dependent limitations of local feedback," *IEEE Trans. Autom. Control*, vol. 57, no. 9, pp. 2235–2249, Sep. 2012.
- [14] F. Lin, M. Fardad, and M. R. Jovanović, "Optimal control of vehicular formations with nearest neighbor interactions," *IEEE Trans. Autom. Control*, vol. 57, no. 9, pp. 2203–2218, Sep. 2012.
- [15] S. Patterson and B. Bamieh, "Leader selection for optimal network coherence," in *Proc. 49th IEEE Conf. Decision Control*, 2010, pp. 2692–2697.
- [16] A. Clark and R. Poovendran, "A submodular optimization framework for leader selection in linear multi-agent systems," in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf.*, 2011, pp. 3614–3621.
- [17] A. Clark, L. Bushnell, and R. Poovendran, "A supermodular optimization framework for leader selection under link noise in linear multi-agent systems," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 283–296, 2014.
- [18] H. Kawashima and M. Egerstedt, "Leader selection via the manipulability of leader-follower networks," in *Proc. Amer. Control Conf.*, 2012, pp. 6053–6058.

- [19] H. G. Tanner, "On the controllability of nearest neighbor interconnections," in *Proc. 43rd IEEE Conf. Decision Control*, 2004, pp. 2467–2472.
- [20] B. Liu, T. Chu, L. Wang, and G. Xie, "Controllability of a leader-follower dynamic network with switching topology," *IEEE Trans. Autom. Control*, vol. 53, no. 4, pp. 1009–1013, 2008.
- [21] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph theoretic perspective," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 162–186, 2009.
- [22] Z. Jia, Z. Wang, H. Lin, and Z. Wang, "Interconnection topologies for multi-agent coordination under leader-follower framework," *Automatica*, vol. 45, no. 12, pp. 2857–2863, 2009.
- [23] A. Clark, L. Bushnell, and R. Poovendran, "On leader selection for performance and controllability in multi-agent systems," in *Proc. 51st IEEE Conf. Decision Control*, 2012, pp. 86–93.
- [24] A. Y. Yazicioglu, W. Abbas, and M. Egerstedt, "A tight lower bound on the controllability of networks with multiple leaders," in *Proc. 51st IEEE Conf. Decision Control*, 2012, pp. 1978–1983.
- [25] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, "Leader selection in multi-agent systems for smooth convergence via fast mixing," in *Proc. 51st IEEE Conf. Decision Control*, 2012, pp. 818–824.
- [26] A. Clark, L. Bushnell, and R. Poovendran, "Joint leader and weight selection for fast convergence in multi-agent systems," in *Proc. Amer. Control Conf.*, 2013, pp. 3814–3820.
- [27] A. Ghosh and S. Boyd, "Growing well-connected graphs," in *Proc. 45th IEEE Conf. Decision Control*, 2006, pp. 6605–6611.
- [28] D. Zelazo, S. Schuler, and F. Allgöwer, "Performance and design of cycles in consensus networks," *Syst. Control Lett.*, vol. 62, no. 1, pp. 85–96, 2013.
- [29] I. Poulakakis, L. Scardovi, and N. E. Leonard, Node Classification in Networks of Stochastic Evidence Accumulators, 2012, *arXiv:1210.4235*.
- [30] K. Fitch and N. E. Leonard, "Information centrality and optimal leader selection in noisy networks," in *Proc. 52nd IEEE Conf. Decision Control*, Florence, Italy, 2013, pp. 7510–7515.
- [31] G. Shi, K. C. Sou, H. Sandberg, and K. H. Johansson, "A graph-theoretic approach on optimizing informed-node selection in multi-agent tracking control," *Physica D: Nonlin. Phenom.*, vol. 267, pp. 104–111, 2014.
- [32] F. Pasqualetti, S. Zampieri, and F. Bullo, Controllability Metrics and Algorithms for Complex Networks, 2013, *arXiv:1308.1201*.
- [33] M. Fardad, F. Lin, and M. R. Jovanović, "Algorithms for leader selection in large dynamical networks: Noise-free leaders," in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf.*, Orlando, FL, 2011, pp. 7188–7193.
- [34] F. Lin, M. Fardad, and M. R. Jovanović, "Algorithms for leader selection in large dynamical networks: Noise-corrupted leaders," in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf.*, Orlando, FL, 2011, pp. 2932–2937.
- [35] C. D. Meyer, "Generalized inversion of modified matrices," *SIAM J. Appl. Math.*, vol. 24, no. 3, pp. 315–323, 1973.
- [36] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM Rev.*, vol. 50, no. 1, pp. 37–66, 2008.
- [37] D. A. Spielman, "Algorithms, graph theory, linear equations in Laplacian matrices," *Proc. Int. Congress Math.*, vol. IV, pp. 2698–2722, 2010.
- [38] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Syst. Tech. J.*, vol. 49, pp. 291–307, 1970.
- [39] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, 2009.
- [40] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, p. 036104, 2006.
- [41] S. Srirangarajan, A. H. Tewfik, and Z.-Q. Luo, "Distributed sensor network localization using SOCP relaxation," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4886–4895, 2008.
- [42] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [43] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Mathematical Programming*, 2013, *arXiv:1208.3922*.
- [44] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Boston, MA: Athena Scientific, 1999.
- [45] Y. Chen and X. Ye, Projection Onto a Simplex, 2011, *arXiv:1101.6081*.
- [46] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2013.
- [47] M. Fardad, X. Zhang, F. Lin, and M. R. Jovanović, "On the optimal dissemination of information in social networks," in *Proc. 51th IEEE Conf. Decision Control*, Maui, HI, 2012, pp. 2539–2544.
- [48] M. Fardad, F. Lin, X. Zhang, and M. R. Jovanović, "On new characterizations of social influence in social networks," in *Proc. Amer. Control Conf.*, Washington, DC, 2013, pp. 4784–4789.
- [49] M. Kisialiou, X. Luo, and Z.-Q. Luo, "Efficient implementation of quasi-maximum-likelihood detection based on semidefinite relaxation," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4811–4822, 2009.
- [50] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, 2010.
- [51] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1968.
- [52] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [53] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer, 2006.



**Fu Lin** (S'06–M'13) received the B.S. degree in instrument science and engineering from Shanghai Jiaotong University, Shanghai, China, in 2005, and the Ph.D. degree in electrical and computer engineering from University of Minnesota, Minneapolis, in 2012.

He was a Postdoctoral Research Associate at the University of Minnesota, Minneapolis, before joining the mathematics and computer science division of Argonne National Laboratory in 2013. His primary research interests are in the analysis and design

of optimal distributed controllers using tools from convex optimization, compressive sensing, and graph theory.



**Makan Fardad** (M'08) received the B.S. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, the M.S. degree in electrical engineering from the Iran University of Science and Technology, Tehran, and the Ph.D. degree in mechanical engineering from the University of California, Santa Barbara, in 2006.

He was a Postdoctoral Associate at the University of Minnesota, Minneapolis, before joining the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY, in 2008.

His research interests are in modeling, analysis, and optimization of distributed and large-scale interconnected systems.



**Mihailo R. Jovanović** (S'00–M'05–SM'13) received the Dipl. Ing. and M.S. degrees from the University of Belgrade, Serbia, in 1995 and 1998, respectively, and the Ph.D. degree from the University of California, Santa Barbara, in 2004.

Before joining the University of Minnesota, Minneapolis, he was a Visiting Researcher with the Department of Mechanics, the Royal Institute of Technology, Stockholm, Sweden, from September to December 2004. Currently, he is an Associate Professor of Electrical and Computer Engineering

at the University of Minnesota, Minneapolis, where he also serves as the Director of Graduate Studies in the interdisciplinary Ph.D. program in Control Science and Dynamical Systems. He has held visiting positions with Stanford University and the Institute for Mathematics and its Applications. His expertise is in modeling, dynamics, and control of large-scale and distributed systems and his current research focuses on sparsity-promoting optimal control, dynamics and control of fluid flows, and fundamental performance limitations in the design of large dynamic networks.

Dr. Jovanović received a CAREER Award from the National Science Foundation in 2007, an Early Career Award from the University of Minnesota Initiative for Renewable Energy and the Environment in 2010, a Resident Fellowship within the Institute on the Environment at the University of Minnesota in 2012, and the George S. Axelby Outstanding Paper Award from the IEEE Control Systems Society in 2013. He is a member of APS and SIAM. He currently serves as an Associate Editor of the *SIAM Journal on Control and Optimization* and has served as an Associate Editor of the IEEE Control Systems Society Conference Editorial Board from July 2006 until December 2010.