

# Algorithms for Leader Selection in Large Dynamical Networks: Noise-corrupted Leaders

Fu Lin, Makan Fardad, and Mihailo R. Jovanović

**Abstract**—We examine the leader selection problem in multi-agent dynamical networks where leaders, in addition to relative information from their neighbors, also have access to their own states. We are interested in selecting an *a priori* specified number of agents as leaders in order to minimize the total variance of the stochastically forced network. Combinatorial nature of this optimal control problem makes computation of the global minimum difficult. We propose a convex relaxation to obtain a lower bound on the global optimal value, and use simple but efficient greedy algorithms to obtain an upper bound. Furthermore, we employ the alternating direction method of multipliers to search for a local minimum. Two examples are provided to illustrate the effectiveness of the developed methods.

**Index Terms**—Alternating direction method of multipliers, consensus, convex optimization/relaxation, greedy algorithm, leader selection, performance bounds, variance amplification.

## I. INTRODUCTION

The consensus of the multi-agent networks is of fundamental interest in the context of distributed control [1]–[4]. It is desired that agents reach agreement on quantities such as heading direction, velocity, and inter-agent spacing in a decentralized fashion. Furthermore, the agents must maintain the agreements in the presence of uncertainty such as exogenous noises. In view of this, the robustness of consensus in networks subject to stochastic disturbances has recently received considerable attention [5]–[7].

In this paper, we consider the networks where all agents use relative information exchange with their neighbors while some agents, *leaders*, also have access to their own states. This setting is particularly relevant to vehicular formations where all vehicles are equipped with ranging devices that allow them to measure relative distances with respect to their neighbors while the leaders additionally have GPS devices.

We are interested in selecting a number of agents as leaders in order to minimize the overall variance in a network subject to stochastic disturbances. This leader selection problem has a convex objective function, but constraints are of combinatorial nature, implying the difficulty in computing the global minimum for large networks. We focus on computing lower and upper bounds on the global optimal value; in particular, we solve a convex relaxation of the combinatorial problem to obtain a lower bound, and develop efficient greedy algorithms to obtain an upper bound. Furthermore, we employ the alternating direction method of multipliers to search for a local minimum of the leader selection problem. This approach is capable of handling the nonconvex

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.

M. Fardad is with the Department of Electrical Engineering and Computer Science, Syracuse University, NY 13244. F. Lin and M. R. Jovanović are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455. E-mails: makan@syr.edu, fu@umn.edu, mihailo@umn.edu.

Boolean constraints *explicitly* by *simple projection*. The main contribution of the paper is the development of these efficient algorithms that facilitate study of the achievable performance in large dynamical networks.

In [8], a closely related leader selection problem has been formulated under the assumption that the leaders are following the desired trajectory at all times and are unaffected by input disturbances. In the companion paper [9], it is shown that the noise-free leader selection problem formulated in [8] can be relaxed to a semidefinite program that can be solved efficiently. This is achieved by linearizing the objective function and by relaxing combinatorial Boolean constraints. Furthermore, an interpretation of the leader selection problems is provided in terms of electrical networks using Kron reduction theory [9].

The paper is organized as follows. In Section II, we formulate the leader selection problem and discuss the connection of our formulation with the one considered in [8] and [9]. In Section III, we obtain a lower bound on the global optimal value of this combinatorial optimal control problem by solving a convex relaxation. In Section IV, we obtain an upper bound on the optimal value using greedy algorithms. In Section V, we employ the alternating direction method of multipliers to search for a local minimum of the leader selection problem. Two examples are provided to demonstrate the effectiveness of the developed methods in Section VI. We summarize our results in Section VII.

## II. LEADER SELECTION PROBLEM

We consider a connected, undirected network of  $n$  single-integrator nodes

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

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

$$u_i = - \sum_{j \in \mathcal{N}_i} (x_i - x_j),$$

where  $\mathcal{N}_i$  is the set of neighbors of node  $i$ ; a node is a *leader* if, in addition to the relative information exchange with its neighbors, it also has access to its own state,

$$u_i = - \sum_{j \in \mathcal{N}_i} (x_i - x_j) - \alpha_i x_i,$$

where  $\alpha_i$  is a positive number.

The state-space representation of the network is given by

$$\dot{x} = -(L + D_\alpha H)x + w,$$

where  $L = L^T$  is the Laplacian matrix of the graph,  $\alpha = [\alpha_1 \cdots \alpha_n]^T$ ,  $h = [h_1 \cdots h_n]^T$ ,  $D_\alpha = \text{diag}\{\alpha\}$ , and

$H = \text{diag}\{h\}$ . Here, the Boolean-valued  $h_i$  for  $i = 1, \dots, n$  indicates that node  $i$  is a leader if  $h_i = 1$ , and that node  $i$  is a follower if  $h_i = 0$ .

Following [6], [8], we consider the steady-state variance of the network

$$V_{ss} = \text{trace} \left( \lim_{t \rightarrow \infty} \mathcal{E}\{x(t)x^T(t)\} \right),$$

which quantifies variance amplification of the network subject to stochastic disturbances. For connected undirected graphs, the steady-state covariance matrix can be obtained explicitly from the corresponding Lyapunov equation,

$$\lim_{t \rightarrow \infty} \mathcal{E}\{x(t)x^T(t)\} = (1/2)(L + D_\alpha H)^{-1},$$

thereby yielding,

$$V_{ss} = (1/2) \text{trace} \left( (L + D_\alpha H)^{-1} \right).$$

In the leader selection problem considered in [8], it is assumed that the leaders are maintaining their desired trajectories at all times and are unaffected by input disturbances. In the coordinates of deviation from the desired trajectory, this assumption implies that the state of every leader is identically equal to zero. Therefore, the rows and columns corresponding to the leaders can be eliminated from the state equations, keeping only the dynamics of the follower nodes [8]

$$\dot{x}_f = -L_h x_f + w_f.$$

Here,  $x_f$  is a vector containing all states of followers,  $w_f$  is the vector of disturbances affecting them, and  $L_h$  is the  $(n-k) \times (n-k)$  submatrix of  $L$  obtained by removing the rows and columns of  $L$  that correspond to the leaders. The steady-state variance of the followers is given by

$$(1/2) \text{trace} (L_h^{-1}).$$

As shown in Appendix A,

$$\text{trace} (L_h^{-1}) = \lim_{\alpha_i \rightarrow \infty} \text{trace} \left( (L + D_\alpha H)^{-1} \right), \quad (1)$$

i.e.,  $\text{trace} (L_h^{-1})$  is equivalent to  $V_{ss}$  when the state feedback gains  $\{\alpha_i\}$  of the leaders all grow to infinity.

We are interested in identifying a subset of nodes that are effective in reducing the variance of the network. Thus, we consider the problem of choosing  $k$  leaders with  $1 \leq k < n$  to minimize the steady-state variance

$$\begin{aligned} & \text{minimize} && J(h) = \text{trace} \left( (L + D_\alpha H)^{-1} \right) \\ & \text{subject to} && h_i \in \{0, 1\}, \quad i = 1, \dots, n, \\ & && \sum_{i=1}^n h_i = k. \end{aligned} \quad (\text{LS})$$

The objective function  $J(h)$  in (LS) is convex for positive definite matrix  $L + D_\alpha H$  and the summation constraint in (LS) is linear. The difficulty of this problem comes from the Boolean constraints  $h_i \in \{0, 1\}$ , which are not convex. For large problems, it is not computationally feasible to perform exhaustive search, since the  $n$ -choose- $k$  combinations grow exponentially fast as the problem size increases.

### III. CONVEX RELAXATION

In this section, we consider a convex relaxation of the leader selection problem (LS). By solving the relaxed problem, we obtain a lower bound on the optimal value  $J_{\text{opt}}$

of (LS). We show that the convex relaxation can be formulated as a semi-definite program (SDP) and thus can be solved by available SDP solvers. Furthermore, we develop an efficient customized interior point method for large problems.

A convex relaxation of the leader selection problem is obtained by enlarging the Boolean constraint sets  $h_i \in \{0, 1\}$  to the convex sets  $h_i \in [0, 1]$ ,

$$\begin{aligned} & \text{minimize} && J(h) = \text{trace} \left( (L + D_\alpha H)^{-1} \right) \\ & \text{subject to} && 0 \leq h_i \leq 1, \quad i = 1, \dots, n \\ & && \sum_{i=1}^n h_i = k. \end{aligned} \quad (\text{CR})$$

Let  $h^*$  be a global minimum of the convex optimization problem (CR). Since we have enlarged the constraint sets, the optimal value  $J(h^*)$  of the relaxed problem provides a lower bound on the optimal value  $J_{\text{opt}}$  of (LS). Note that  $h^*$  may not provide a selection of  $k$  leaders since it may turn out to be non-Boolean-valued.

Note that (CR) can be formulated as a semi-definite program using the Schur complement [10, Appendix 5.5]

$$\begin{aligned} & \text{minimize} && \text{trace}(X) \\ & \text{subject to} && \begin{bmatrix} X & I \\ I & L + D_\alpha H \end{bmatrix} \geq 0 \\ & && 0 \leq h_i \leq 1, \quad i = 1, \dots, n \\ & && \sum_{i=1}^n h_i = k, \end{aligned} \quad (\text{SDP})$$

and thus can be solved efficiently, for small and medium size problems, using standard SDP solvers.

#### A. Customized interior point method

We next develop a customized interior point method to deal with large problems. We consider the following approximation of (CR)

$$\begin{aligned} & \text{minimize} && q(h) = \gamma \text{trace} \left( (L + D_\alpha H)^{-1} \right) + \\ & && \sum_{i=1}^n \left( -\log(h_i) - \log(1 - h_i) \right) \quad (2) \\ & \text{subject to} && \mathbf{1}^T h = k, \end{aligned}$$

where  $\gamma$  is a positive scalar and  $\mathbf{1}$  is the vector of all 1's. The log-barrier function  $-\log(h_i)$  increases to infinity as  $h_i$  approaches zero, while  $-\log(1 - h_i)$  increases to infinity as  $h_i$  approaches 1. Augmenting the objective function with these log-barrier functions provides an approximation to the convex relaxation (CR) with inequality constraints. The quality of this approximation is determined by the positive scalar  $\gamma$ ; the solution of the approximate problem (2) converges to the solution of the convex relaxation (CR) as the parameter  $\gamma$  increases to infinity [10, Section 11.2]. Thus, we solve a sequence of the approximate problems (2) by gradually increasing  $\gamma$ , and by starting each minimization using the solution of the problem for the previous value of  $\gamma$ .

For fixed  $\gamma$ , we employ Newton's method to solve the approximate problem (2). Using standard procedure [10, Section 10.2], we determine the Newton direction

$$h_{\text{nt}} = -(\nabla^2 q)^{-1} \nabla q - \mu (\nabla^2 q)^{-1} \mathbf{1},$$

where

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

Here, we provide the expressions for gradient and Hessian of  $q$  and omit derivation for brevity

$$\begin{aligned} (\nabla q)_i &= -\gamma\alpha_i((L + D_\alpha H)^{-2})_{ii} - h_i^{-1} - (h_i - 1)^{-1}, \\ \nabla^2 q &= 2\gamma(D_\alpha(L + D_\alpha H)^{-2}D_\alpha) \circ (L + D_\alpha H)^{-1} + \\ &\quad \text{diag}\{h_i^{-2} + (1 - h_i)^{-2}\}, \end{aligned}$$

where  $(\nabla q)_i$  denotes the  $i$ th entry of the vector  $\nabla q$ ,  $(\cdot)_{ii}$  denotes the  $ii$ th entry of a given matrix, and  $\circ$  denotes the entry-wise multiplication of matrices.

We next examine complexity of computing the Newton direction. The major cost of forming Hessian  $\nabla^2 q$  is to compute  $(L + D_\alpha H)^{-2}$ , which takes  $(7/3)n^3$  operations to compute  $(L + D_\alpha H)^{-1}$  and  $n^3$  operations to form  $(L + D_\alpha H)^{-2}$ . Computing the Newton direction  $h_{\text{nt}}$  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 the Cholesky factorization. Thus, computation of each Newton step requires  $(11/3)n^3$  operations. In comparison, solving (SDP) using available SDP solvers takes  $O(n^6)$  operations.

#### IV. GREEDY ALGORITHMS

With the lower bound obtained from solving the convex relaxation (CR) in Section III, we next consider two greedy algorithms to obtain an upper bound on the optimal value  $J_{\text{opt}}$ . The first algorithm selects one leader at a time by assigning the node that provides the largest performance improvement as the leader. The second algorithm attempts to improve a given choice of  $k$  leaders by checking possible swaps between leaders and followers. Similar greedy algorithms have been used in [8], [11]. However, in both cases, we show that substantial improvement in algorithmic complexity can be achieved by exploiting structure of the low-rank modifications to the Laplacian.

##### A. One-leader-at-a-time algorithm

We select one leader at a time by assigning the node that results in the largest performance improvement as the leader. To select the first leader, we compute

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

for  $i = 1, \dots, n$ , and assign the node, say  $v_1$ , that achieves the minimum value of  $\{J_1^i\}$ . If two or more nodes provide the largest performance improvement, we select one of these nodes as a leader. After choosing  $s$  leaders,  $v_1, \dots, v_s$ , we compute

$$J_{s+1}^i = \text{trace}((L_s + \alpha_i e_i e_i^T)^{-1}),$$

for  $i \notin \{v_1, \dots, v_s\}$ , and select node  $v_{s+1}$  that yields the minimum value of  $\{J_{s+1}^i\}$ . Here,

$$L_s = L + \alpha_{v_1} e_{v_1} e_{v_1}^T + \dots + \alpha_{v_s} e_{v_s} e_{v_s}^T.$$

This procedure is repeated until all  $k$  leaders are selected.

Without exploiting structure, the above procedure requires  $O(kn^4)$  operations. On the other hand, the rank-1 update

formula obtained from matrix inversion lemma

$$(L_s + \alpha_i e_i e_i^T)^{-1} = L_s^{-1} - \frac{L_s^{-1} \alpha_i e_i e_i^T L_s^{-1}}{1 + \alpha_i e_i^T L_s^{-1} e_i},$$

yields

$$J_{s+1}^i = \text{trace}(L_s^{-1}) - \frac{\alpha_i \|(L_s^{-1})_i\|_2^2}{1 + \alpha_i (L_s^{-1})_{ii}}, \quad (3)$$

where  $(L_s^{-1})_i$  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 [12],

$$L_1^{-1} = L^\dagger - (L^\dagger e_i) \mathbf{1}^T - \mathbf{1} (L^\dagger e_i)^T + ((1/\alpha_i) + e_i^T L^\dagger e_i) \mathbf{1} \mathbf{1}^T,$$

and thus,

$$J_1^i = \text{trace}(L^\dagger) + n((1/\alpha_i) + e_i^T L^\dagger e_i), \quad (4)$$

where  $L^\dagger$  denotes the pseudo-inverse of  $L$  (e.g., see [13])

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

Therefore, matrix inverse can be computed using  $O(n^2)$  operations and  $J_{s+1}^i$  can be evaluated using  $O(n)$  operations. Overall,  $k$  rank-1 updates,  $nk/2$  objective function evaluations, and one full matrix inverse require  $O(kn^2 + n^3)$  operations as opposed to  $O(kn^4)$  operations without exploiting structure.

---

#### Algorithm 1 One-leader-at-a-time algorithm

---

- 1: **for**  $i = 1$  to  $n$  **do**
  - 2:   compute  $J_1^i$  using (4);
  - 3: **end for**
  - 4: choose node  $v_1$  that yields the smallest value of  $\{J_1^i\}$  as a leader;
  - 5: **for**  $s = 1$  to  $k - 1$  **do**
  - 6:   compute  $J_{s+1}^i$  using (3) for  $i \notin \{v_1, \dots, v_s\}$ ;
  - 7:   choose node  $v_{s+1}$  that yields the smallest value of  $\{J_{s+1}^i\}$  as a leader.
  - 8: **end for**
- 

##### B. Swap algorithm

Given a selection of  $k$  leaders, we consider swaps between the leaders and the followers to further improve performance. Specifically, we swap one of the  $k$  leaders with one of the  $n - k$  followers, and check if such a swap would provide a decrease in the objective function  $J$ . If no decrease occurs for all possible  $k(n - k)$  swaps, we stop the algorithm. If a swap provides performance improvement, we update the selection of the leaders and restart checking the  $k(n - k)$  swaps for the updated leader selection. The number of swaps between the leaders and the followers can be quite large. Thus, we terminate the algorithm if it reaches a maximum number  $N_{\text{swap}}$  of leader updates. If  $N_{\text{swap}}$  is chosen to be proportional to the number of nodes  $n$ , then we show below that the worst case computational complexity takes  $O(n^3)$  operations, the same as for the convex relaxation problem (CR).

A swap between a leader (node  $i$ ) and a follower (node  $j$ ) leads to a rank-2 modification of the Laplacian,

$$\bar{L} = L - \alpha_i e_i e_i^T + \alpha_j e_j e_j^T,$$

where  $\bar{L} = L + D_\alpha H$ , with  $H = \text{diag}\{h\}$  for a given leader selection  $h$ . Specifically, we have

$$\begin{aligned} & (\bar{L} - \alpha_i e_i e_i^T + \alpha_j e_j e_j^T)^{-1} = \\ & \bar{L}^{-1} - \bar{L}^{-1} \bar{E}_{ij} (I_2 + E_{ij}^T \bar{L}^{-1} \bar{E}_{ij})^{-1} E_{ij}^T \bar{L}^{-1}, \end{aligned} \quad (5)$$

where  $E_{ij} = [e_i \ e_j]$ ,  $\bar{E}_{ij} = [-\alpha_i e_i \ \alpha_j e_j]$ , and  $I_2$  is a  $2 \times 2$  identity matrix. Thus, the objective function for the leader selection with node  $i$  replaced by node  $j$  is given by

$$J_{ij} = \text{trace}(\bar{L}^{-1} - (I_2 + E_{ij}^T \bar{L}^{-1} \bar{E}_{ij})^{-1} E_{ij}^T \bar{L}^{-2} \bar{E}_{ij}). \quad (6)$$

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

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

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 (5) requires  $O(n^2)$  operations.

Since the total number of leader updates can be large, we set a maximum number of the updates to  $N_{\text{swap}}$ . If we let  $N_{\text{swap}}$  grow as  $O(n)$ , then the worst case computational complexity of the swap algorithm takes  $O(n^3)$  operations (the same as solving the relaxed problem (CR)). It is worth mentioning that in practice the number of leader updates is much less than  $O(n)$ . Similar observation is made in [11] where the swap algorithm is applied to a related sensor selection problem.

---

### Algorithm 2 Swap algorithm

---

- 1: Given a leader selection  $h$ , let  $s_l = \{s_l^1, \dots, s_l^k\}$  and  $s_f = \{s_f^1, \dots, s_f^{n-k}\}$  denote the index sets of leaders and followers, respectively; set the number of swaps  $m = 0$ ;
  - 2: **for**  $i = s_l^1$  to  $s_l^k$  **do**
  - 3:   **for**  $j = s_f^1$  to  $s_f^{n-k}$  **do**
  - 4:     compute  $J_{ij}$  using (6);
  - 5:     **if**  $J_{ij} < J(h)$  **then**
  - 6:       swap  $i$  from  $s_l$  with  $j$  from  $s_f$ ;
  - 7:       set  $J(h) = J_{ij}$ ;
  - 8:       increase  $m$  by 1;
  - 9:     **if**  $m = N_{\text{swap}}$  **then**
  - 10:       stop;
  - 11:     **else**
  - 12:       go to step 2.
  - 13:     **end if**
  - 14:   **end for**
  - 15: **end for**
  - 16: **end for**
- 

## V. ALTERNATING DIRECTION METHOD OF MULTIPLIERS

Since previously introduced greedy algorithms may not yield the optimal leader selection, we next use the alternating direction method of multipliers (ADMM) to search for a local minimum of (LS). The ADMM has been studied extensively since it was introduced in the 1970s [14]–[17]; see [18] for a recent survey. A closely related *split Bregman* approach has

recently attracted considerable attention in the image processing community due to its superior convergence properties in several  $\ell_1$  norm related minimization problems [19], [20].

Owing to its alternating approach, the ADMM is capable of handling the nonconvex Boolean constraints in (LS) explicitly by simple projection. This feature of the ADMM facilitates a straightforward procedure to search for a local minimum of (LS). Let us denote the objective function and the indicator function of the constraint sets in (LS) by

$$\begin{aligned} f(h) &= \text{trace}((L + D_\alpha H)^{-1}), \\ g(h) &= \begin{cases} 0 & \text{if } h_i \in \{0, 1\} \text{ and } \mathbf{1}^T h = k \\ +\infty & \text{otherwise.} \end{cases} \end{aligned}$$

Hence, (LS) can be formulated as an *unconstrained* problem

$$\text{minimize } f(h) + g(h), \quad (7)$$

which can be put into the following form suitable for the ADMM algorithm,

$$\begin{aligned} & \text{minimize } f(h) + g(z) \\ & \text{subject to } h - z = 0, \end{aligned} \quad (8)$$

where  $z$  is the auxiliary variable. Note that both the unconstrained problem (7) and the constrained problem (8) are equivalent to (LS).

We form the augmented Lagrangian associated with (8)

$$\mathcal{L}_\rho(h, z, \lambda) = f(h) + g(z) + \lambda^T (h - z) + (\rho/2) \|h - z\|_2^2,$$

where  $\lambda \in \mathbb{R}^n$  is the dual variable and  $\rho > 0$  is the quadratic penalty weight. Starting from some initial conditions  $z^0$  and  $\lambda^0$ , for  $\kappa = 0, 1, \dots$ , the ADMM algorithm updates

$$h^{\kappa+1} := \arg \min_h \mathcal{L}_\rho(h, z^\kappa, \lambda^\kappa), \quad (9a)$$

$$z^{\kappa+1} := \arg \min_z \mathcal{L}_\rho(h^{\kappa+1}, z, \lambda^\kappa), \quad (9b)$$

$$\lambda^{\kappa+1} := \lambda^\kappa + \rho(h^{\kappa+1} - z^{\kappa+1}), \quad (9c)$$

until

$$\|h^{\kappa+1} - z^{\kappa+1}\|_2 \leq \epsilon \text{ and } \rho \|z^{\kappa+1} - z^\kappa\|_2 \leq \epsilon.$$

An alternating way of updating  $h$  and  $z$  in (9) motivates the name of the method. Splitting the optimization variables into two copies and updating them in an alternating fashion yields the minimization problems (9a) and (9b) that are easy to solve. The minimization of  $\mathcal{L}_\rho(h, z^\kappa, \lambda^\kappa)$  in (9a) with respect to  $h$  can be formulated as an SDP

$$\begin{aligned} & \text{minimize } \text{trace}(X) + (\lambda^\kappa)^T (h - z^\kappa) + (\rho/2) \|h - z^\kappa\|_2^2 \\ & \text{subject to } \begin{bmatrix} X & I \\ I & L + D_\alpha H \end{bmatrix} \geq 0, \end{aligned}$$

which can be solved efficiently for small and medium size problems. Even though the indicator function  $g(z)$  in  $\mathcal{L}_\rho$  is non-convex, we show in Appendix B that the minimizer of  $\mathcal{L}_\rho(h^{\kappa+1}, z, \lambda^\kappa)$  in (9b) with respect to  $z$  is given by

$$z_i = \begin{cases} 1 & \text{if } \bar{z}_i \geq [\bar{z}]_k \\ 0 & \text{if } \bar{z}_i < [\bar{z}]_k \end{cases}$$

where

$$\bar{z} = h^{\kappa+1} + (1/\rho) \lambda^\kappa,$$

and  $[\bar{z}]_k$  denotes the  $k$ th largest entry of  $\bar{z}$ . Thus, the minimizer of (9b) can be determined *explicitly* using *simple*

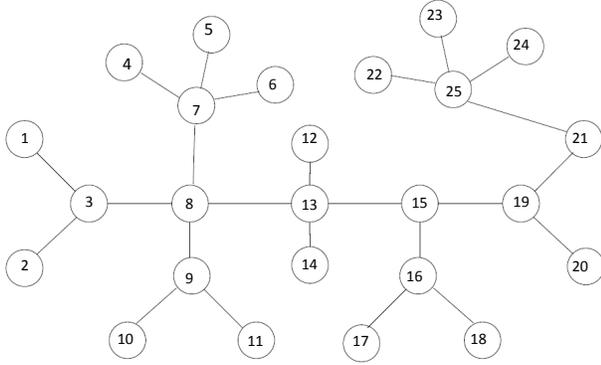


Fig. 1: A network of 25 nodes.

*projection.* For large problems, we employ Newton’s method to solve (9a); the gradient and Hessian of  $\mathcal{L}_\rho$  with respect to  $h$  are given by

$$(\nabla \mathcal{L}_\rho)_i = -\alpha_i((L + D_\alpha H)^{-2})_{ii} + \lambda_i + \rho(h_i - z_i),$$

$$\nabla^2 \mathcal{L}_\rho = 2(D_\alpha(L + D_\alpha H)^{-2}D_\alpha) \circ (L + D_\alpha H)^{-1} + \rho I.$$

## VI. EXAMPLES

In this section, we consider two examples to illustrate the performance of the developed methods. In both examples we set  $\alpha_i$  to be the degree of node  $i$ , that is, the number of neighbors of node  $i$ . We set the penalty weight in the ADMM algorithm to  $\rho = 10^3$ . Note that the convergence of the ADMM depends on the value of  $\rho$ ; in particular, the convergence may not take place if  $\rho$  is not large enough [18].

### A. An example from [8]

We first consider an example from [8]; see Fig. 1. For this small example, the global minima can be determined by exhaustive search. It turns out that the one-leader-at-a-time algorithm followed by the swap algorithm actually finds the global minima. On the other hand, the ADMM provides the global minima for the cases  $k = 4$  and  $k = 5$ ; see Table I. Note that both the greedy algorithms and the ADMM tend to select nodes with large degrees (i.e., with large number of neighbors) as the leaders. It is also worth mentioning that the global optimal leader selections match the results reported in [8].

### B. Regular lattice

We next consider the leader selection problem for a  $10 \times 10$  regular lattice. Figure 2a shows the lower bounds obtained from solving the convex relaxation and the upper bounds obtained from the one-leader-at-a-time algorithm followed by the swap algorithm; for this example, the number of swap updates is between 2 and 30. Figure 2b shows the gap between these bounds. Note that, as the number of leaders increases, the gap between the lower and the upper bounds decreases. Using ADMM, we obtain the upper bounds close to those obtained from the greedy algorithms; see Fig. 3. The leader selections obtained using ADMM for the case with  $k = 4, 20, 40$  are plotted in Fig. 4.

TABLE I: Computational results for the example shown in Fig. 1. Lower bounds  $J_{\text{lb}}$  are obtained from solving the convex relaxation (CR); upper bounds  $J_{\text{ub}}$  for greedy algorithms – the one-leader-at-a-time algorithm followed by the swap algorithm – are actually tight, i.e.,  $J_{\text{ub}} = J_{\text{opt}}$ ; upper bounds  $J_{\text{ub}}$  for ADMM are tight when  $k = 4$  and  $k = 5$ .

$k$	greedy algorithms			ADMM	
	$J_{\text{lb}}$	$J_{\text{ub}}$	leaders	$J_{\text{ub}}$	leaders
1	38.4	72.3	13	118.3	25
2	30.3	43.4	8, 25	47.9	7, 25
3	26.7	35.2	8, 16, 25	36.7	7, 16, 25
4	24.3	30.0	3, 7, 16, 25	30.0	3, 7, 16, 25
5	22.4	25.8	3, 7, 9, 16, 25	25.8	3, 7, 9, 16, 25

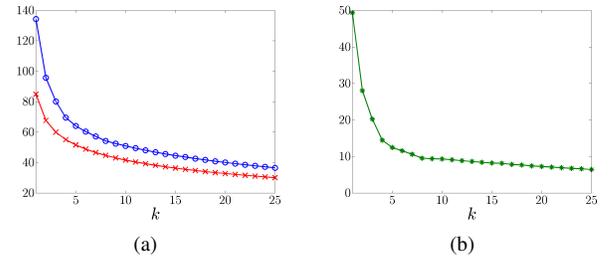


Fig. 2: Regular lattice: (a) the lower bounds ( $\times$ ) obtained from solving the convex relaxation and the upper bounds ( $\circ$ ) obtained from the one-leader-at-a-time algorithm followed by the swap algorithm; (b) the gap between these bounds ( $*$ ).

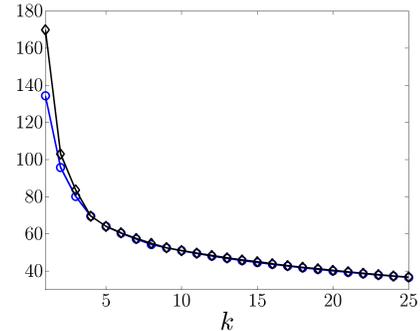


Fig. 3: Regular lattice: the upper bounds obtained from one-leader-at-a-time algorithm followed by the swap algorithm ( $\circ$ ) and the upper bounds obtained from ADMM ( $\diamond$ ).

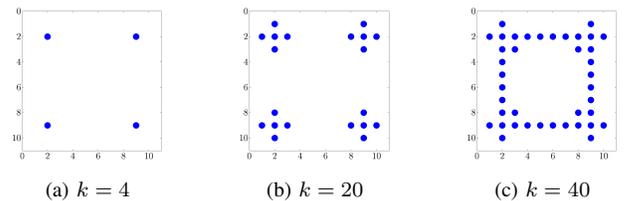


Fig. 4: The leader selections obtained using ADMM for a  $10 \times 10$  regular lattice; leaders are signified by  $\bullet$ .

## VII. CONCLUDING REMARKS

In this paper, we consider the leader selection problem in large dynamical networks. We focus on computing the lower and upper bounds on the global optimal value of this combinatorial optimal control problem. In particular, we obtain a lower bound by solving a convex relaxation and we obtain an upper bound using greedy algorithms. Furthermore, we employ the alternating direction method of multipliers to search for a local minimum. We provide two examples to illustrate the effectiveness of our approaches. As aforementioned, the main contribution of the paper is the development of these efficient algorithms that allow us to estimate the achievable performance in large networks. We intend to apply these tools for leader selection problem in different types of networks, including small-world social networks and random graphs, in our future work.

## APPENDIX

### A. Derivation of (1)

We perform appropriate permutations such that the matrix  $L + D_\alpha H$  can be partitioned into

$$L + D_\alpha H = \begin{bmatrix} L_h & L_b \\ L_b^T & L_l + D_{\alpha l} \end{bmatrix}.$$

Inverting the  $2 \times 2$  block matrix and taking the trace yields

$$\text{trace}(L_h^{-1}) + \text{trace}(L_h^{-1} L_b S_\alpha^{-1} L_b^T L_h^{-1} + S_\alpha^{-1}),$$

where

$$S_\alpha = D_{\alpha l} + L_l - L_b^T L_h^{-1} L_b$$

is the Schur complement of  $L_h$ . Therefore,  $S_\alpha^{-1}$  vanishes as the state feedback gains  $\{\alpha_i\}$  of the leaders all grow to infinity, and (1) follows.

### B. Analytical solution of (9b)

Note that the minimization problem of  $\mathcal{L}_\rho(h^{\kappa+1}, z, \lambda^\kappa)$  with respect to  $z$  is equivalent to

$$\begin{aligned} & \text{minimize} && -(\lambda^\kappa)^T z + (\rho/2) \|z - h^{\kappa+1}\|_2^2 \\ & \text{subject to} && z_i \in \{0, 1\}, \quad \mathbf{1}^T z = k. \end{aligned} \quad (10)$$

The objective function can be rewritten as

$$\frac{\rho}{2} \sum_{i=1}^n ((z_i - \bar{z}_i)^2 + h_i^{\kappa+1} - \bar{z}_i^2), \quad (11)$$

where

$$\bar{z}_i = h_i^{\kappa+1} + (1/\rho)\lambda_i^\kappa.$$

Projecting the minimizer of (11)

$$z_i = \bar{z}_i, \quad i = 1, \dots, n.$$

onto the constraint set  $z_i \in \{0, 1\}$  and  $\mathbf{1}^T z = k$  yields

$$z_i = \begin{cases} 1 & \text{if } \bar{z}_i \geq [\bar{z}]_k \\ 0 & \text{if } \bar{z}_i < [\bar{z}]_k \end{cases} \quad (12)$$

where  $[\bar{z}]_k$  is the  $k$ th largest entry of  $\bar{z}$ .

To see (12), let us consider  $z^1$  satisfying  $\mathbf{1}^T z^1 = k$  and  $z_i^1 \in \{0, 1\}$  but  $z^1$  is not the projection determined by (12). Thus, there exists at least one entry of  $z^1$ , e.g., the  $l$ th entry, such that  $z_l^1 = 1$  with the corresponding  $\bar{z}_l < [\bar{z}]_k$  and at

least one entry, e.g., the  $j$ th entry, such that  $z_j^1 = 0$  with the corresponding  $\bar{z}_j \geq [\bar{z}]_k$ . Let

$$\delta_{lj} = (z_l^1 - \bar{z}_l)^2 + (z_j^1 - \bar{z}_j)^2 = (1 - \bar{z}_l)^2 + \bar{z}_j^2,$$

and let

$$\delta_{jl} = \bar{z}_l^2 + (1 - \bar{z}_j)^2.$$

Since

$$\delta_{lj} - \delta_{jl} = 2(\bar{z}_j - \bar{z}_l) > 0,$$

we conclude that the objective function (11) decreases if we choose  $z_l^1 = 0$  and  $z_j^1 = 1$  instead of  $z_l^1 = 1$  and  $z_j^1 = 0$ . In other words, we can reduce the objective function (11) by exchanging the values of two entries  $z_l^1 = 1$  (with  $\bar{z}_l < [\bar{z}]_k$ ) and  $z_j^1 = 0$  (with  $\bar{z}_j \geq [\bar{z}]_k$ ) until (12) is satisfied. Therefore, the projection (12) provides the solution of problem (10).

## REFERENCES

- [1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [2] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [3] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Automat. Control*, vol. 50, no. 2, pp. 169–182, 2005.
- [4] M. Mesbahi and M. Egerstedt, *Graph-theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
- [5] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.
- [6] B. Bamieh, M. R. Jovanović, P. Mitra, and S. Patterson, "Coherence in large-scale networks: dimension dependent limitations of local feedback," *IEEE Trans. Automat. Control*, 2010, provisionally accepted.
- [7] D. Zelazo and M. Mesbahi, "Edge agreement: Graph-theoretic performance bounds and passivity analysis," *IEEE Trans. Automat. Control*, vol. 56, no. 3, pp. 544–555, 2011.
- [8] S. Patterson and B. Bamieh, "Leader selection for optimal network coherence," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 2692–2697.
- [9] M. Fardad, F. Lin, and M. R. Jovanović, "Algorithms for leader selection in large dynamical networks: Noise-free leaders," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, to appear.
- [10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [11] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, 2009.
- [12] C. D. Meyer, "Generalized inversion of modified matrices," *SIAM Journal of Applied Mathematics*, vol. 24, no. 3, pp. 315–323, 1973.
- [13] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM Review*, vol. 50, no. 1, pp. 37–66, 2008.
- [14] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers and Mathematics with Applications*, vol. 2, pp. 17–40, 1976.
- [15] R. T. Rockafellar, "Augmented lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, vol. 1, pp. 97–116, 1976.
- [16] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.
- [17] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems," *Computational Optimization and Applications*, vol. 1, pp. 93–111, 1992.
- [18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [19] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for  $\ell_1$ -minimization with applications to compressed sensing," *SIAM Journal on Imaging Sciences*, vol. 1, no. 1, pp. 143–168, 2008.
- [20] T. Goldstein and S. Osher, "The split Bregman method for  $\ell_1$  regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.