Key Nodes Selection in Controlling Complex Networks via Convex Optimization

Jie Ding, Changyun Wen, Fellow, IEEE, Guoqi Li, and Zhenghua Chen

Abstract—Key nodes are the nodes connected with a given number of external source controllers that result in minimal control cost. Finding such a subset of nodes is a challenging task since it is impossible to list and evaluate all possible solutions unless the network is small. In this paper, we approximately solve this problem by proposing three algorithms step by step. By relaxing the Boolean constraints in the original optimization model, a convex problem is obtained. Then inexact alternating direction method of multipliers (IADMMs) is proposed and convergence property is theoretically established. Based on the degree distribution, an extension method named degree-based IADMM (D-IADMM) is proposed such that key nodes are pinpointed. In addition, with the technique of local optimization employed on the results of D-IADMM, we also develop LD-IADMM and the performance is greatly improved. The effectiveness of the proposed algorithms is validated on different networks ranging from Erdős–Rényi networks and scale-free networks to some real-life networks.

Index Terms—Complex networks, key node selection, optimal control.

I. INTRODUCTION

Controlling complex networks is one of the most important and challenging problems in network science and engineering. On the one hand, the ability to control the networked system, for instance, smart grid [1], [2]; sensor networks [3]–[5]; and transportation networks [6], [7], is a guarantee of their reliable and efficient operation. On the other hand, many complex dynamics and behaviors can be analyzed and understood from a control system perspective, ranging from social networks [8]–[10] and economic networks [11], [12] to gene networks [13], [14] and brain networks [15], [16].

According to control theory, a system is controllable, if it can be driven from any initial state to a pregiven state with a suitable input. Due to the couplings between nodes, it is actually difficult to choose such suitable inputs. Recently, many important advances have been made with regards to the controllability problem of complex networks. A prominent work is presented in [17] to identify the minimum number of driver nodes/inputs that ensure the controllability of the network based on the concept of structural controllability [18] by ignoring the weights of coupling between nodes. It is also shown that the minimum number of driver nodes required for complete control mainly depends on the network’s degree distribution. Different from [17], the minimal controllability problem (MCP) of controlling the network with minimum state variables, that is, actuated nodes, is addressed in [19]. Shortly after [17], many extensions about MCP have emerged (see [20]–[23] and references therein). However, there is no consideration of cost in all of the aforementioned works, while their solutions assure controllability or structural controllability of the system. Actually, when the Gramian controllability matrix is nearly singular, the control cost becomes excessively high, making it impossible to fully control the whole network. In addition, it is arguably even more important to evaluate the energy cost in controlling real-life networked systems due to limited and valuable resources. Therefore, researchers start to investigate how to control the network from the energy point of view. Yan et al. [24] solved this problem only by analyzing a single external input which is connected with one node. In [25], the smallest eigenvalue of controllability Gramian is adopted as a metric to derive a lower bound on control cost for a discrete linear system. But no general algorithm is proposed to achieve such a lower bound. Such a problem is also addressed in [26] and [27], where the authors show submodularity of controllability metrics proposed in [28]. However, to find the node set with minimum control cost, a set of nodes that provides controllability of the whole network should be given first. Then, extra nodes are added to the given node set by optimizing the energy metrics. In [29] and [30], an optimization model is built to minimize an energy cost function on a sphere surface determined by trace boundary conditions. It is found that key nodes are related to the network’s degree distribution.

Departing from the aforementioned works, in this paper, we consider the problem of identifying a subset of nodes connected with external inputs under the condition that the control cost is minimized. Specifically, given a number of external
control sources, there will be multiple possible ways to connect them with nodes in the network and each way gives a set of nodes. Among all possible sets, we hope to locate the one which can control the network with minimum control cost, that is, the key node set [30]–[32]. As pointed out in [19] and [33], it is an intractable combinatorial problem and NP-hard. A straightforward idea is to list all possible combinations with the given size and select the one with minimum control cost. However, this is almost impossible for large-scale complex networks due to high-computational cost.

Motivated by [29], we adopt the average energy and solve the problem via convex relaxation. Specifically, to minimize the average control cost, we first formulate the optimization model in Boolean constraints where nodes and external controllers are in one-to-one connections. By relaxing the constraints to their convex hull, a convex problem is obtained. Then, an efficient algorithm called the inexact alternating direction method of multipliers (IADMMs) is developed to solve the problem iteratively. The convergence of IADMM is theoretically analyzed. Through rounding the solution by exploiting degree information, degree-based IADMM (D-IADMM) is proposed such that the key nodes are pinpointed. To further improve the computational efficiency is significantly improved compared with ADMM, yet obtaining convergence property of IADMM becomes more challenging due to inexact solutions. Nevertheless, this difficulty is successfully overcome and such a property is still theoretically established by exploring different techniques, such as involving monotone operator and Bregman distance.

2) To solve the convex problem, we propose an algorithm named IADMMs to avoid solving the subproblem proposed in ADMM exactly. The computational efficiency is significantly improved compared with ADMM, yet obtaining convergence property of IADMM becomes more challenging due to inexact solutions. Nevertheless, this difficulty is successfully overcome and such a property is still theoretically established by exploring different techniques, such as involving monotone operator and Bregman distance.

3) With the obtained solution of IADMM, we need to solve the original nonconvex problem to pinpoint the key node set, which is another challenge encountered in this paper. Instead of simply truncating the solution of IADMM, we take the degree information of the network into account when locating the key nodes. Thus, D-IADMMs are proposed.

The rest of this paper is organized as follows. Section II presents some preliminaries which will be referred to later. In Section III, we formulate the problem of selecting $M$ key nodes in one-to-one connection with external control sources. In Section IV, we relax the model to a convex problem and solve it by IADMM. The convergence of IADMM is also analyzed. Section V deals with the problem of key nodes determination, where D-IADMM and LD-IADMM are proposed. In Section VI, we validate our proposed algorithms through extensive numerical examples. A comparison with PGME, which is proposed in [30] and [31], is presented. Section VII concludes this paper.

II. Preliminaries

As mentioned earlier, the monotone operator and Bregman distance will be exploited to obtain the basic convergence results of IADMM. These preliminaries are now introduced in this section.

A. Subdifferential and Monotone Operator

On Euclidean space $\mathcal{H}$, an operator is a set-valued mapping, that is, $\mathcal{P} : \mathcal{H} \rightarrow 2^{\mathcal{H}}$. Given a convex function $f$, the subdifferential at $x \in \text{dom} f$ is a set-valued operator, $\partial f := \{ p \in \mathcal{H} | f(y) > f(x) + p^T(y-x), \forall y \in \mathcal{H} \}$.

Let $T$ be a set-valued operator with graph $\mathcal{P}$ defined as $\text{gra} \mathcal{P} := \{(x,y) \in \mathcal{H} \times \mathcal{H} | y \in \mathcal{P} x \}$. The operator $\mathcal{P}$ is called monotone if

$$\forall (x,y), (x',y') \in \text{gra} \mathcal{P}, \| x - y, x' - y' \| \geq 0.$$ 

A monotone operator $\mathcal{P}$ is maximal monotone if no monotone operator $\mathcal{P}'$ such that $\text{gra} \mathcal{P} \subseteq \text{gra} \mathcal{P}'$. The best-known example of the maximal monotone operator is the subdifferential $\partial f$ of a closed proper convex function. The resolvent of $\mathcal{P}$ is $R_{\tau \mathcal{P}} = (I + \tau \mathcal{P})^{-1}$. An application of the resolvent operator is the proximity operator which is defined as

$$\text{prox}_{\tau f}(v) = \arg \min_{x \in \mathcal{H}} f(x) + \frac{1}{2\tau} \| x - v \|^2.$$
B. Bregman Distance

Consider a convex function \( f \), the Bregman distance between \( x, x' \in \text{dom} f \) is defined as

\[
D_f(x, x') = f(x) - f(x') - q^T(x - x')
\]

where \( q \in \partial f(x') \).

Bregman distance has the following properties.

1. \( D_f(x, y) \geq 0 \);
2. \( D_f(x, y) = 0 \iff x = y \);
3. \( D_f(x, y) = D_f(y, x) \);
4. \( D_f(x, z) \leq D_f(x, y) + D_f(y, z) \).

C. Rules of Derivatives

Denote \( [ \cdot ]_{ij} \) as the \((i, j)\)th element of the matrix \([ \cdot ]\).

For compatible matrices \(A, B, C\), and \(X\) such that \(X = AB\), then \(X_{ij} = \sum_k A_{ik}B_{kj}\). It can be written as \(X_{ij} = A_{ik} \cdot B_{kj}\) by introducing the \([ \cdot ]\) operator. In what follows, the summation notation is omitted over repeated indices (Einstein notation) for the convenience of representation. Similarly, for compatible matrices \(A, B, C, X\) such that \(X = ABC\), then \(X_{ij} = (ABC)_{ij} = \sum_k \sum_p A_{ik}B_{kp}C_{pj}\) can be written as \(X_{ij} = A_{ik} \cdot B_{kp} \cdot C_{pj}\). Define that \(\delta_{ij} = 1\) if \(i = j; 0\) otherwise. Also denote \(X_{ij}\) as the \(ij\)th element of a matrix \(X\).

Lemma 1 [35]: If the elements of \(X\) are independent, we have the following.

1. \( \partial X_{ij} / \partial X_{ij} = \delta_{ij} \cdot \delta_{ij} \);
2. \( \partial X_{ij}^T / \partial X_{ij} = \delta_{ij} \cdot \delta_{ij} \);
3. \( \Delta X_{ij} = X_{ij}^T \);
4. \( \delta_{ij} \cdot \delta_{ij} = \delta_{ij} \).

Lemma 2 (1-Chain Rule [36]): Suppose that \(U \in \mathbb{R}^{N \times M}\) is a function of matrix \(X\), that is, \(U = g(X)\), the derivative of the function \(h(U) = h(g(X))\) with respect to \(X\) is given by the chain rule as follows:

\[
\frac{\partial h(U)}{\partial X_{ij}} = \sum_{m,n} \frac{\partial h(U)}{\partial U_{mn}} \frac{\partial U_{mn}}{\partial X_{ij}} = \text{tr}\left(\left[\frac{\partial h(U)}{\partial U}\right]^T \left[\frac{\partial U}{\partial X_{ij}}\right]\right) = \frac{\partial h(U)}{\partial U} \cdot \frac{\partial U_{mn}}{\partial X_{ij}}.
\]

Thus, the \(ij\)th element of matrix \([\partial h(g(X))]/[\partial X]\), which is the derivative of \(h(g(X))\) with respect to matrix \(X\), is obtained by

\[
\frac{\partial h(g(X))}{\partial X} = \frac{\partial h(g(X))}{\partial X_{ij}}.
\]

III. Problem Formulation

Given a directed graph \(G(V, E)\), where \(V = \{v_1, v_2, \ldots, v_N\}\) denotes a set of vertices or nodes and \(E\) is a set of edges connecting pairs of vertices. Significant studies have mainly been focused on directed networks with \(N\)-dimensional linear dynamics as follows [8], [17]:

\[
x(t) = Ax(t) + Bu(t), \quad x(0) = x_0
\]

where \(x(t) = [x_1(t), \ldots, x_N(t)]^T\) is the state vector of \(N\) nodes at time \(t\) with the initial state being \(x_0\), \(u(t) = [u_1(t), \ldots, u_M(t)]^T\) is a time-dependent input vector of external signals with \(M (M \leq N)\) being the number of control sources where the same \(u_i(t)\) may drive multiple nodes. The element \(A_{ij}\) of the \(N \times N\) network weighted adjacency matrix describes the connection/interaction strength/weight from node \(j\) to node \(i\) for directed networks (between node \(i\) and node \(j\) for undirected networks), and the element \(B_{im}\) of the \(N \times M\) input matrix \(B\) is nonzero if an external control source \(m\) is connected to node \(i\) and zero otherwise. As mentioned in Section I, we consider how to identify \(M\) key nodes in one-to-one connection with external control sources, such that the states of system (1) can be driven to the origin at \(t = t_f\), that is, \(x(t_f) = 0\), \(0, \ldots, 0\)^T, subject to the condition that a given control cost is minimized. To achieve this objective, the following assumptions are made.

Assumption 1: The number of external control sources is sufficient such that \((A, B)\) is controllable at an initial time.

Assumption 2: Suppose that \(BB^T = \text{diag}(b)\), where \(b\) is an 
\(N\)-dimensional vector, that is, \(BB^T\) is an \(N \times N\) square diagonal matrix with all elements of vector \(b\) on its main diagonal. Under this assumption, \(b_i = 1\) if node \(i\) is connected with external control source; \(b_i = 0\) otherwise.

Remark 1: With the condition that \((A, B)\) is controllable, \(M\) should be greater than the number of driver nodes, denoted as \(N_D\), which can be determined by the maximum matching algorithm in [17]. On the other hand, as pointed out in [37], there exists a numerical controllability transition problem in controlling complex networks, which means that a small number of key nodes is barely enough to ensure numerical controllability. Such a failure of finding a numerical solution may not be overcome by merely increasing numerical precision. That is to say, we need more input signals to overcome the numerical controllability transition problem. In [37], it further shows that the numerical success rate increases sharply from zero to one as the number of external inputs is increased. The transition point, measured by \(M/N\), is around 0.2. Therefore, we generally set \(M \geq \min[ND, 0.2N]\) to ensure controllability of the networks and avoid the numerical controllability transition problem.

Mathematically, the optimization problem is formulated as

\[
\min E(t_f, B) = \int_0^{t_f} u^T(t, B)u(t, B)dt
\]

s.t. \(x(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad x(t_f) = 0\)

where \(u(t) = [u_1(t, B), \ldots, u_M(t, B)]^T \in \mathbb{R}^{M \times 1}\) means that the input signal to be designed at time \(t\) is dependent on \(B\). In this case, the state vector \(x\) at time \(t\) is also dependent on \(B\) since

\[
x(t, B) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau.
\]

Under Assumption 1 and from [38], the control cost \(\int_0^{t_f} u(t, B)u(t, B)dt\) can be minimized when the input vector is given by

\[
u(t, B) = -B^Te^{AT(\tau)}W_B^{-1}e^{A\tau}x_0
\]

where

\[
W_B = \int_0^{t_f} e^{AT\tau}B^TBe^{A\tau}d\tau.
\]
is the Gramian matrix [37], [38] depending on A when $t_f$ and $B$ are pregiven. At the same time, at time $t_f$, the constraint $x(t_f) = 0$ can be satisfied.

By substituting (6) and (7) into the cost function in (4), we have

$$
\int_0^{t_f} u^T(t, B)u(t, B)dt
= \int_0^{t_f} x_0^T e^{A(t-t_f)} B B^T e^{A(t-t_f)} B^T e^{A(t-t_f)} x_0 dt
= x_0^T e^{A(t-t_f)} B B^T \int_0^{t_f} e^{A(t-t_f)} B^T e^{A(t-t_f)} dt \cdot B^T e^{A(t-t_f)} x_0
= x_0^T e^{A(t-t_f)} B^T e^{A(t-t_f)} x_0
= \text{tr}\left(e^{A(t-t_f)} B^T e^{A(t-t_f)} x_0^T x_0 \right).
$$

Assume that each element of initial state $x_0 = [x_1(0), \ldots, x_N(0)]^T$ is an identically independently distributed variable with zero mean and variance 1. This assumption holds in various applications of science and engineering since the elements of $x_0$ are the initial states of different nodes. Under this assumption, $x_i(0)$ and $x_j(0)$ are two independent random variables with zero mean and unit variance. When summing all realizations of $x_i(0)$ and $x_j(0)$ together, we obtain that $\mathbb{E}[x_i(0)x_j(0)] = \delta_{ij}$. Thus, $\mathbb{E}[x_0x_0^T]$ can be regarded as an identity matrix since

$$
\mathbb{E}[x_0x_0^T] = 
\begin{pmatrix}
\mathbb{E}(x_1(0)x_1(0)) & \cdots & \mathbb{E}(x_1(0)x_N(0)) \\
\mathbb{E}(x_2(0)x_1(0)) & \cdots & \mathbb{E}(x_2(0)x_N(0)) \\
\vdots & \ddots & \vdots \\
\mathbb{E}(x_N(0)x_1(0)) & \cdots & \mathbb{E}(x_N(0)x_N(0))
\end{pmatrix}
= I_N.
$$

Therefore, under Assumption 2, we have

$$
\mathcal{E}(B) = \text{tr}\left(e^{A(t-t_f)} B B^T e^{A(t-t_f)} x_0^T \right)
= \text{tr}\left(e^{A(t-t_f)} B B^T e^{A(t-t_f)} x_0^T \right)
= \text{tr}\left(\int_0^{t_f} e^{A(t-t_f)} B B^T e^{A(t-t_f)} dt \right)^{-1}
= \text{tr}\left(\int_0^{t_f} e^{-A(t-t_f)} B B^T e^{-A(t-t_f)} dt \right)^{-1}.
$$

To identify M key nodes and minimize the control cost, the optimization model is finally given by

$$
\begin{align*}
\arg \min_b \mathcal{E}(b) &= \text{tr}\left(\int_0^{t_f} e^{-A(t-t_f)} B B^T e^{-A(t-t_f)} dt \right)^{-1} \\
\text{s.t.} \quad b^T 1 &= M, \quad b_i \in [0, 1], \quad i = 1, \ldots, N
\end{align*}
$$

where 1 is an all-ones vector.

Remark 2: When the mean of $x_i(0)$ is not equal to 0, the objective function needs to be changed to $\text{tr}\left(\int_0^{t_f} e^{-A(t-t_f)} B B^T e^{-A(t-t_f)} dt \right)^{-1} \mathbb{E}[x_0x_0^T]$ and the proposed algorithms are still applicable to this case.

IV. CONVEX RELAXATION FOR KEY NODES SELECTION

In this section, we relax the optimization problem (10) into a convex model. An efficient and effective algorithm called IADMM is developed. The convergence property of IADMM is further established.

A. Inexact Alternating Direction Multipliers of Method

Note that (10) is a nonconvex problem due to the Boolean constraints $b_i \in \{0, 1\}$, which is too difficult to be solved directly. Now, we relax this constraint to its convex hull 0 $\leq b_i \leq 1$, which yields the following optimization problem:

$$
\begin{align*}
\arg \min_b \mathcal{E}(b) &= \text{tr}\left(\int_0^{t_f} e^{-A(t-t_f)} B B^T e^{-A(t-t_f)} dt \right)^{-1} \\
\text{s.t.} \quad b^T 1 &= M, \quad 0 \leq b_i \leq 1, \quad i = 1, \ldots, N
\end{align*}
$$

Unlike the original problem (10), this problem is proved to be convex, so that it can be solved efficiently. Note that the feasible solution set of the relaxed convex problem (11) contains the feasible set of the original problem (10). Therefore, its optimal value of the objective function $\mathcal{E}(b)$ cannot be larger than that of the original problem [39]. We derive the solution of the original problem based on (11), which will be explored in Section V.

Before solving the problem (11), we first prove (11) is a convex problem.

Lemma 3 [40]: Let $X, Y$ be $\mathbb{R}^{n \times n}$ Hermitian matrices. If $X$ is positive definite, then there is a nonsingular $S \in \mathbb{R}^{n \times n}$ such that $X = S^* S$ and $Y = S A S^*$, in which $\Lambda$ is real diagonal. The inertias of $Y$ and $\Lambda$ are the same, so $\Lambda$ is non-negative diagonal (positive diagonal) if $Y$ is positive semidefinite (positive definite). The main diagonal entries of $\Lambda$ are the eigenvalues of the diagonalizable matrix $X^{-1} Y$.

Proposition 1: The optimization model (11) is strictly convex. That is, $\mathcal{E}(b)$ is strictly convex on 0 $\leq b_i \leq 1 (i = 1, \ldots, N), \quad b^T 1 = M$.

Proof: See Appendix A.

Before solving the problem in (11), we first obtain the derivative of $\mathcal{E}(b)$.

Lemma 4: Let $H_b = \int_0^{t_f} e^{-A(t-t_f)} B B^T e^{-A(t-t_f)} dt$. The gradient of $\mathcal{E}(b)$ in (10) with regards to $b$ yields that

$$
\frac{\partial \mathcal{E}(b)}{\partial b_i} = -\left(\int_0^{t_f} e^{-A(t-t_f)} H_b^{-2} e^{-A(t-t_f)} dt \right)_{ii}.
$$

Proof: See Appendix B.

As the optimization problem (11) is demonstrated to be strictly convex in Theorem 1 and the gradient of the objective function is given in Lemma 4, now it is time for us to introduce IADMMs to solve the problem in (11).

Define $S_1 := \{-b : 0 \leq b_i \leq 1, \quad b^T 1 \geq \delta\}$, where $\delta$ is a sufficiently small positive number. $S_2 := \{1^T b = M\}, \quad \phi_1(b), \quad \phi_2(b)$ are indicator functions of $S_1$ and $S_2$, respectively, which are given by

$$
\phi_1(b) = \begin{cases} 0 & b \in S_1 \\infty & b \notin S_1 \end{cases}
$$

$$
\phi_2(b) = \begin{cases} 0 & b \in S_2 \\infty & b \notin S_2 \end{cases}
$$
Then problem (11) is equivalent to
\[
\arg\min_b \ E(b) + \phi_1(b) + \phi_2(b).
\] (13)

Now, we introduce an additional variable \(z\) and rewrite (13) as the following optimization problem:
\[
\arg\min_b \ f(b) + g(z)
\] s.t. \(b - z = 0\) (14)
where \(f(b) = E(b) + \phi_1(b)\) and \(g(z) = \phi_2(z)\).

Note that the augmented Lagrangian associated with (14) is
\[
\mathcal{L}_\rho(b, z, \lambda) = f(b) + g(z) + \lambda^T(b - z) + \frac{\rho}{2}\|b - z\|_2^2
\] (15)
where \(\lambda\) is the Lagrangian multiplier and \(\rho > 0\).

To find the optimal solution of (14), we develop IADMM with the help of augmented Lagrangian function (15) through the following alternating iterations:
\[
b^{k+1} = \Pi_{S_1}\left(\frac{b^k - \mu}{\rho}\left(\nabla E(b^k) + \lambda + \rho(b^k - z^k)\right)\right)
\] (16a)
\[
z^{k+1} = \arg\min_z\left(g(z) - \lambda^T z + \frac{\rho}{2}\|b^{k+1} - z\|_2^2\right)
\] (16b)
\[
\lambda^{k+1} = \lambda^k + \rho(b^{k+1} - z^{k+1})
\] (16c)
where \(\mu\) is a step size and \(\Pi_{S_1}\) is the Euclidean projection onto the set \(S_1\). The update of \(\lambda\) is actually the dual ascent step.

**Remark 3:** Note that if ADMM [34] was applied here, the update for \(b^{k+1}\) in (16a) would be replaced by minimizing the subproblem, i.e., \(b^{k+1} = \arg\min_b(\nabla E(b) + \phi_1(b) + \lambda^T b + \rho/2\|b - z^k\|_2^2)\), which is of high-computational complexity. Instead of obtaining the exact minimum point of the subproblem, here we update \(b\) inexactly by a single gradient step in (16a), yet the convergence is still ensured which will be presented in Theorem 1 later. Therefore, we term the proposed algorithm as IADMMs.

For \(z\)-minimization step in (16b), it is equivalent to the following optimization problem:
\[
\arg\min_z -\lambda^T z + \frac{\rho}{2}\|b^{k+1} - z\|_2^2
\] s.t. \(\|b^{k+1} - z\|_2^2 = M\). (17)

The Lagrangian function of (17) is given by
\[
\mathcal{L}_\mu(z) = -\lambda^T z + \frac{\rho}{2}\|b^{k+1} - z\|_2^2 + \kappa(\|b^{k+1} - z\|_2^2 - M).
\]

Based on the Karush–Kuhn–Tucker condition [41], we can obtain that
\[
\begin{align*}
\frac{\partial \mathcal{L}_\mu(z)}{\partial z} = 0, \\
\frac{\partial \mathcal{L}_\mu(z)}{\partial \kappa} = 0. \tag{18}
\end{align*}
\]

By solving the equations in (18), the exact solution for \(z\)-update in each iteration is given as follows:
\[
z^{k+1} = \left(1 - \frac{\|b^k - z^k\|_2^2}{\rho}\right)\left(\frac{b^k + \lambda^k}{\rho}\right) + \frac{M}{\rho}1. \tag{19}
\]

**Algorithm 1 IADMMs**

**Input:** Functions \(f(b)\) and \(g(z)\), parameter \(\rho\).

**Output:** \(b^k\)

1. **Initialization:** Initialize \(b, z, \nu\)
2. **repeat**
3. \[
b^{k+1} = \Pi_{S_1}\left(\frac{b^k - \mu}{\rho}\left(\nabla E(b^k) + \lambda + \rho(b^k - z^k)\right)\right)
\]
\[
z^{k+1} = \left(1 - \frac{1}{\rho}\right)b^{k+1} + \frac{\lambda^k}{\rho} + M\frac{1}{\rho}
\]
\[
\lambda^{k+1} = \lambda^k + \rho(b^{k+1} - z^{k+1})
\]
4. **until** meet certain stopping criterion
5. **return** \(b^* = b\)

The whole iterative process of (16) will be terminated until the primal and dual residuals are sufficiently small as specified by
\[
\frac{\|b^{k+1} - z^{k+1}\|_2}{\rho} < \epsilon_{\text{pri}} \quad \text{(primal residual)}
\]
\[
\frac{\rho\|z^{k+1} - z^k\|_2}{2} < \epsilon_{\text{dual}} \quad \text{(dual residual)} \tag{20}
\]
where \(\epsilon_{\text{pri}}\) and \(\epsilon_{\text{dual}}\) are positive small constants specified by user.

Denote the solution obtained by the proposed algorithm IADMM as \(b^*\). The above iterative process is summarized in Algorithm 1. The relevant convergence analysis of IADMM will be presented later.

**B. Convergence Analysis of IADMM**

In this section, we mainly study the convergence properties of the proposed algorithm IADMM. Before proceeding to the main result, we introduce the following lemmas.

**Lemma 5:** The update of \(b\) in (16a) is equivalent to
\[
\mu\phi_1(b^{k+1}) = b^k - b^{k+1} - \mu(v\nabla E(b^k) + \lambda^k + \rho(b^k - z^k)).
\]

**Proof:** See Appendix C.

Note that \(\nabla E(b)\) is continuously differentiable on the compact set \(S_1\). Therefore, \(\nabla E(b)\) is Lipschitz continuous in the domain \(S_1\), that is,
\[
\|\nabla E(b) - \nabla E(b')\| \leq L\|b - b'\| \quad \forall b, b' \in S_1. \tag{21}
\]

Now, we are ready to introduce the following theorem.

**Theorem 1:** Given \(\mu < 1/(\rho + L)\), the sequence \([b^k, z^k, \lambda^k]\) generated by the proposed IADMM will converge to the saddle point \((b^*, z^*, \lambda^*)\) of \(\mathcal{L}_\rho(b, z, \lambda)\).

**Proof:** Let us first consider the optimality condition of problem (19). Let \(G = (\|b^k - z^k\|_2)\). Then the primal and dual optimality condition satisfied by individual steps in Algorithm 1 can be expressed as follows:
\[
\begin{align*}
\|b^k - z^k\|_2 = M = 0 & \quad \text{(primal feasibility)} \\
\|b^k - z^k\|_2 = 0 & \quad \text{(primal feasibility)} \\
\frac{\partial f(b^*) + \lambda^*}{\partial b} = 0 & \quad \text{(dual feasibility)} \\
\frac{\partial g(b^*) - \lambda^*}{\partial b} = 0 & \quad \text{(dual feasibility)} \tag{22}
\end{align*}
\]
where \(\frac{\partial f(b^*)}{\partial b} = \nabla E(b^*) + \frac{\partial \phi_1(b^*)}{\partial b^*}\).
Apparently, $Gb^* = (M/N)\mathbb{1}$. Then, we substitute (16b) into (16c) in the main paper and obtain that
\[
\lambda^{k+1} = \rho Gb^{k+1} + G\lambda^k - \rho Gb^*
\] (23)

Multiplying $G$ on both sides of (23) and using the fact $G^2 = G$, it follows that:
\[
\begin{cases}
(I - G)\lambda^k = 0 \\
\lambda^{k+1} - \lambda^k = \rho G(b^{k+1} - b^*)
\end{cases}
\] (24)

By Lemma 5, we have
\[
\mu \partial f(b^{k+1}) = \mu \nabla \mathcal{E}(b^{k+1}) + \mu \partial \phi_1(b^{k+1})
\]
\[
= b^k - b^{k+1} + \mu \bigg( \nabla \mathcal{E}(b^{k+1}) - \nabla \mathcal{E}(b^k) \bigg) - \mu \lambda^k - \mu \rho (b^k - z^k).
\] (25)

Knowing that \(\partial f\) is maximal monotone, based on \(\partial f(b^*) + \lambda^* = 0\) from the optimality condition (22) and properties in (24), we have
\[
\mu \partial f(b^{k+1}) - \partial f(b^*) = (b^k - b^{k+1} + \mu \bigg( \nabla \mathcal{E}(b^{k+1}) - \nabla \mathcal{E}(b^k) \bigg) + \mu (\lambda^* - \lambda^k) + \mu (b^{k+1} - b^*)
\]
\[
= b^k - b^{k+1} + \mu \bigg( \nabla \mathcal{E}(b^{k+1}) - \nabla \mathcal{E}(b^k) \bigg) + \mu (\lambda^* - \lambda^k).
\] (26)

For the first term of the above inequality, we have
\[
\bigg\langle b^k - b^{k+1}, b^{k+1} - b^* \bigg\rangle \\
= \frac{1}{2} \bigg( \|b^k - b^*\|^2 - \|b^{k+1} - b^*\|^2 - \|b^{k+1} - b^k\|^2 \bigg)
\] (27)

With the identity of Bregman distance, it can be obtained that
\[
\nabla \mathcal{E}(b^{k+1}) - \nabla \mathcal{E}(b^k), b^{k+1} - b^*
\]
\[
= D_G(b^*, b^{k+1}) - D_G(b^*, b^k) + D_G(b^{k+1}, b^*).
\] (28)

With (24), we have
\[
\bigg\langle \lambda^* - \lambda^k, b^{k+1} - b^* \bigg\rangle \\
= \bigg\langle G\big(\lambda^* - \lambda^k\big), b^{k+1} - b^* \bigg\rangle \\
= \bigg\langle \lambda^* - \lambda^k, G\big(b^{k+1} - b^*\big) \bigg\rangle \\
= \frac{1}{\rho} \bigg\langle \lambda^* - \lambda^k, b^{k+1} - \lambda^k \bigg\rangle \\
= \frac{1}{2\rho} \bigg( \|\lambda^{k+1} - \lambda^k\|^2 + \|\lambda^k - \lambda^*\|^2 - \|\lambda^{k+1} - \lambda^*\|^2 \bigg)
\] (29)

\[
\bigg\langle \lambda^{k+1} - \lambda^k, b^{k+1} - b^* \bigg\rangle \\
= \bigg\langle G\big(\lambda^{k+1} - \lambda^k\big), b^{k+1} - b^* \bigg\rangle \\
= \rho\big\langle G\big(b^* - b^k\big), b^{k+1} - b^* \big\rangle
\]
\[
= \frac{\rho}{2} \big\| b^{k+1} - b^k \big\|^2 + \frac{\rho}{2} \big\| b^k - b^* \big\|^2 - \frac{\rho}{2} \big\| b^{k+1} - b^* \big\|^2
\]
\[
= \frac{\rho}{2} \big\| b^{k+1} - b^k \big\|^2 - \frac{1}{2\rho} \big\| \lambda^{k+1} - \lambda^k \big\|^2 - \frac{1}{2\rho} \big\| \lambda^{k+1} - \lambda^k \big\|^2
\] (30)

where \(\|b^{k+1} - b^k\|_G = \sqrt{\big\langle G(b^{k+1} - b^k), b^{k+1} - b^k \big\rangle}\) and \(\|G\|_G\) is a seminorm.

Now, substituting (27)–(30) into (26) yields
\[
\big\| b^k - b^* \big\|^2 - \big\| b^{k+1} - b^* \big\|^2 - \big\| b^{k+1} - b^k \big\|^2
\]
\[
= -\rho \big\| b^{k+1} - b^k \big\|^2 - \frac{\rho}{2} \big\| b^k - b^* \big\|^2 + 2\mu D_G\big(b^k, b^{k+1}\big)
\]
\[
= -\rho \big\| b^{k+1} - b^k \big\|^2 - \frac{\rho}{2} \big\| b^k - b^* \big\|^2 + 2\mu D_G\big(b^k, b^{k+1}\big).
\] (31)

Let \(V_k = \|b^k - b^*\|^2 + (\mu/\rho)\|\lambda^k - \lambda^*\|^2 - 2\mu D_G\big(b^k, b^*\big)\). As \(\mu < 1/(\rho + L)\), \(\|b^k - b^*\|^2 \geq (2/L)D_G\big(b^*, b^k\big) > (2/L + \rho)D_G\big(b^*, b^k\big) > 2\mu D_G\big(b^*, b^k\big)\) such that \(V_k\) is positive. Then, we have
\[
V_k - V_{k+1} \geq \frac{\mu}{\rho} \|\lambda^k - \lambda^{k-1}\|^2 + \|b^{k+1} - b^k\|^2
\]
\[
- \rho \big\| b^{k+1} - b^k \big\|^2 - \rho \big\| b^{k+1} - b^* \big\|^2
\]
\[
= \frac{\mu}{\rho} \|\lambda^k - \lambda^{k-1}\|^2 + \|b^{k+1} - b^k\|^2 - \rho \|b^{k+1} - b^*\|^2 - 2\mu D_G\big(b^{k+1}, b^*\big).
\] (32)

As the maximum eigenvalue of \(G\) is 1, \(I - \mu G\rho\) is semidefinite positive such that \(\|b^{k+1} - b^k\|^2 - \|b^{k+1} - b^*\|^2 > \mu L\|b^{k+1} - b^*\|^2 > 2\mu D_G\big(b^{k+1}, b^*\big)\). Then the above inequality can be rewritten as
\[
V_k - V_{k+1} \geq \frac{\mu}{\rho} \|\lambda^k - \lambda^{k-1}\|^2 + (1 - \mu - \mu L) \|b^{k+1} - b^k\|^2.
\] (33)

This shows that \(V_k\) is nonincreasing. Because \(V_k \leq V_0\), it follows that \(b^k\) and \(\lambda^k\) are bounded. Summing (33) over \(k\) from 0 to infinity leads to
\[
\sum_{k=0}^{\infty} \frac{\mu}{\rho} \|\lambda^k - \lambda^{k-1}\|^2 + (1 - \mu - \mu L) \|b^{k+1} - b^k\|^2 \leq V_0
\]
which implies \(\lim_{k \to \infty} \lambda^k - \lambda^{k-1} = 0\) and \(\lim_{k \to \infty} b^{k+1} - b^k = 0\). Through (16b), it is easy to obtain \(\lim_{k \to \infty} x^{k+1} - x^k = (G\lim_{k \to \infty} b^{k+1} - b^k) + G/\rho (\lim_{k \to \infty} \lambda^k - \lambda^{k-1}) = 0\).

Thus with (31) and the fact that \(\{V_k\}_{k \geq 0}\) converges, we conclude that when \(\mu < 1/(\rho + L)\), the sequence \(\{b^k, z^k, \lambda^k\}\) uniquely converges to the saddle point \(\{b^*, z^*, \lambda^*\}\) of the augmented Lagrangian \(L_\rho(b, z, \lambda)\).  

\[\]
V. DETERMINATION OF KEY NODES

In this section, we discuss how to select key nodes and construct the solution of the original problem (10) based on the solution of IADMM. By taking the degree information of nodes, D-IADMM is proposed and key nodes are identified. To further improve the performance of D-IADMM, we employ a greedy algorithm and propose LD-IADMM.

A. Key Nodes Selection Based on the Degree Distribution

Note that \( b^* \) is the solution of the problem (11) where \( 0 \leq b^*_i \leq 1 \). This implies that node \( i \) is connected with external controller \( i \) with input weight \( b^*_i \), \( i = 1, \ldots, N \). Summing all these input weights leads to \( b^*_i = M \). Getting back to the original problem (10), we hope to find \( M \) most important nodes to construct the key node set with obtained weight information \( b^*_i \). On the other hand, in [30], it is stated that key nodes are related to networks’ degree distribution trend for directed networks. And in [17], it is pointed out that each stem of the directed network must be driven by an independent external control source to ensure the controllability of the networked system [8]. Therefore, motivated by these results, we propose another algorithm, named D-IADMMs to identify key node set based on the solution of IADMM by exploiting degree information.

Denote the in-degree and out-degree of node \( i \) as \( k^i_{in} \) and \( k^i_{out} \), respectively. Similar to [8], sources are defined as nodes with no edges pointing to them, i.e., \( k^i_{in} = 0 \). Sinks are nodes with edges only pointing toward them. That is, \( k^i_{out} = 0 \) and \( k^i_{in} \neq 0 \). On the one hand, based on the results in [8], we know that the sources are definitely connected with external inputs to ensure the controllability of networks. On the other hand, in [42] and [43], it is concluded that the longest control chains (LLC) dominate the control cost of the network and the extra control inputs should be placed near the middle of the chain so as to minimize the length of the LCC. Therefore, the sinks are almost unlikely to be key nodes except the case where two sinks have the same parent node. Such an exception is illustrated by an example in Fig. 1. Note that in-degree and out-degree of Node 2/Node 3 are 1 and 0, respectively, and they have a common parent node, i.e., Node 1. However, according to [17], there is still a possibility for Node 2 and/or Node 3 to be a key node. We call the pattern illustrated in Fig. 1 as \( \Lambda \)-pattern.

After predetermination of sources as key nodes and pre-elimination of some sinks from key node set based on above descriptions, now we determine the remaining key nodes. Suppose that the number of predetermined key nodes is \( N' \) and the number of eliminated nodes from key node set is \( N_e \). Then we calculate \( b^*_{i}/k^i_{in} \) for the remaining \( N - N' - N_e \) nodes. Finally, nodes with \( M - N_e \) largest values are chosen as key nodes [29]. Let \( \mathcal{K}^* \) be the key node set and \( b^* \) be the solution obtained by D-IADMM. Obviously, if node \( i \) is a key node, \( b^*_i = 1 \) and \( i \in \mathcal{K}^* \). Otherwise, \( b^*_i = 0 \) and \( i \notin \mathcal{K}^* \).

Combining Algorithms 1 and 2, we term the whole process of determining key nodes as D-IADMMs.

### Algorithm 2 Key Node Selection in Directed Networks Based on Nodes’ Degree Distribution (D-IADMM)

**Input:** \( \mathcal{G}(V,E), b^* \) obtained by IADMM

**Output:** \( b^* \), key node set \( \mathcal{K}^* \)

1. **Initialization:** Initialize \( \mathcal{K} = \emptyset \), \( \mathcal{T} = \emptyset \), \( b = 0 \)
2. **Calculation:** Calculate the in-degree \( k^i_{in} \) and out-degree \( k^i_{out} \) of node \( i \) respectively (\( i = 1, 2, \ldots, N \)).
3. for \( i \) from 1 to \( N \) do
   4. if \( k^i_{in} = 0 \) then
      5. \( \mathcal{K} \leftarrow \mathcal{K} \cup \{i\} \)
   6. else if \( k^i_{out} = 0 \) and \( i \notin \Lambda \)-pattern then
      7. \( \mathcal{T} \leftarrow \mathcal{T} \cup \{i\} \)
   8. end if
4. end for
5. while \( |\mathcal{K}| < M \) do
   6. \( i \in \arg \max_{i \in \mathcal{V} \setminus (\mathcal{K} \cup \mathcal{T})} \frac{b^*_i}{k^i_{in}} \)
   7. \( b^*_i = 1 \)
   8. \( \mathcal{K} \leftarrow \mathcal{K} \cup \{i\} \)
6. end while
7. return \( b^* = b \) and \( \mathcal{K}^* = \mathcal{K} \).

B. Local Optimization

Note that the key node set identified by D-IADMM yields a control cost no more than a near-optimal solution. Therefore, we hope to further decrease the control cost. In this section, we employ a greedy algorithm, i.e., LO [39] based on the result of D-IADMM.

Specifically, for a key node set \( \mathcal{K}^* \) obtained from Algorithm 2, we first obtain the set of nodes with zero in-degree in \( \mathcal{K}^* \) and denote it as set \( \mathcal{K}^{in} \). The nodes in \( \mathcal{K}^{in} \) are kept in the optimal key node set since they are sources and definitely connected with input signals [8]. Denote \( \mathcal{K}^* \setminus \mathcal{K}^{in} \) as the relative complement set of \( \mathcal{K}^{in} \) with respect to \( \mathcal{K}^* \), that is, the nodes in \( \mathcal{K}^* \) but not in \( \mathcal{K}^{in} \). If the key node set is not optimal, we will swap each node in \( \mathcal{K}^* \setminus \mathcal{K}^{in} \) with the node in candidate key node set \( \mathcal{V} \setminus \mathcal{K}^* \) which can help to maximize the reduction of the control cost to further decrease the cost. This technique is similar to Fedorov’s exchange algorithm [44] and Wynn’s algorithm [45]. For large-scale networks, as the total number of swaps may be large, the maximum number of swaps will be restricted as a linear function of network size [46]. Denote the key node set obtained by LD-IADMM as \( \mathcal{K}' \) and the solution as \( b' \). Similar to D-IADMM, if node \( i \) is a key node, \( b'_i = 1 \) and \( i \in \mathcal{K}' \). Otherwise, \( b'_i = 0 \) and \( i \notin \mathcal{K}' \). The detailed procedure is given in Algorithm 3.
Algorithm 3 LO With the Solution Obtained by LD-IADMM

**Input:** \( \mathcal{G}(V, E), b^* \) and \( K^* \) obtained by D-IADMM

**Output:** \( b', K' \) obtained by LD-IADMM

1: Initialization: Initialize \( K = \emptyset, b = 0, \) swap_num = 0
2: Calculation: Calculate the set \( K^{\text{in}} \) where nodes have zero in-degree.
3: repeat
4: for each node \( i \in K^* \setminus K^{\text{in}} \) do
5: for each node \( j \in V \setminus K^* \) do
6: \( b_{ij}^{\text{temp}} = b^* - e_i + e_j \) and calculate \( E_{ij}(b_{ij}^{\text{temp}}) \)
7: end for
8: \( j \in \arg\min_j E_{ij}(b_{ij}^{\text{temp}}) \)
9: if \( E_{ij}(b_{ij}^{\text{temp}}) < E(b^*) \) then
10: \( b^* = b_{ij}^{\text{temp}} \)
11: \( K^* \leftarrow K^* \setminus \{i\} \cup \{j\} \)
12: swap_num = swap_num + 1
13: end if
14: end for
15: until meet certain stopping criterion, e.g., no change of swap_num, meeting maximum swap number
16: return \( b' = b^* \) and \( K' = K^* \).

TABLE I

<table>
<thead>
<tr>
<th>( N )</th>
<th>( M )</th>
<th>IADMM</th>
<th>ADMMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>1.62</td>
<td>6.24</td>
</tr>
<tr>
<td>50</td>
<td>15</td>
<td>5.87</td>
<td>94.02</td>
</tr>
<tr>
<td>100</td>
<td>25</td>
<td>9.19</td>
<td>1088.05</td>
</tr>
<tr>
<td>150</td>
<td>40</td>
<td>39.44</td>
<td>2510.14</td>
</tr>
<tr>
<td>200</td>
<td>50</td>
<td>74.89</td>
<td>6997.91</td>
</tr>
<tr>
<td>300</td>
<td>80</td>
<td>231.78</td>
<td>18246.87</td>
</tr>
</tbody>
</table>

When LO is employed after D-IADMM, we denote the whole process as LD-IADMM for convenience.

VI. SIMULATION RESULTS

In this section, we apply the proposed algorithms to several networks ranging from synthetic networks to real-life networks.

We first take ER network with \( N = 150, M = 60, \) and \( \langle k \rangle = \langle k \rangle^{\text{in}} = \langle k \rangle^{\text{out}} = 5 \) as an example to show the convergence of the proposed IADMM. The iterative process is shown in Fig. 2. From Fig. 2, it is seen that IADMM can converge within 100 steps when the residuals are set as \( \epsilon_{\text{pri}} = \epsilon_{\text{dual}} = 10^{-3} \). This validates Theorem 1 in this paper. Table I compares the simulation time between IADMM and ADMM with the same initialized \( b_0 \). Obviously, IADMM performs much faster than ADMM. This is because, for ADMM, the step in calculating \( b \)-minimization stated in Remark 3 is a significant time-consuming process.

Note that the effectiveness of proposed algorithms can be validated by the control cost associated with the selected key nodes. The lower the control cost, the better the algorithm performs. We first carry out experiments on ER and SF networks, where SF networks are generated by Barabási–Albert model. Fig. 3 presents the minimum-control cost with different \( M \) obtained by PGME, IADMM, and LD-IADMM, respectively.

In Fig. 4, the number of external control sources is fixed as \( M = 20 \) while the mean degree of ER and SF networks varies. Based on the results in Figs. 3 and 4, it is worthy of highlighting the following observations. First, it is noted that the performance of D-IADMM is slightly better than PGME. This is because D-IADMM has taken the degree information into account. Second, it is observed that the technique of LO can greatly improve the performance of D-IADMM. Third, since the simulations in Fig. 3 are carried on ER and SF networks with the same mean degree \( \langle k \rangle = 7.5 \) and the same size of network \( N = 100 \), it is obvious that the minimum control cost of ER network is lower than that of SF network. The results in Fig. 4 further demonstrate this finding. Finally, when the number of external control sources is fixed, the denser the network, the lower the control cost, as shown in Fig. 4.

Note that in Algorithm 2, we mainly take the information of in-degree of nodes into account. This is motivated by the process of maximum matching in directed networks [17]. Now, we explore the in-degree distribution for key nodes obtained...
TABLE II
PERFORMANCE COMPARISON IN ER AND SF NETWORKS

<table>
<thead>
<tr>
<th>Networks</th>
<th>N</th>
<th>M</th>
<th>\langle k \rangle</th>
<th>PGME</th>
<th>D-IADMM</th>
<th>LD-IADMM</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER</td>
<td>500</td>
<td>200</td>
<td>2</td>
<td>3.06E06</td>
<td>1.31E06</td>
<td>2.19E04</td>
<td>4.68E09</td>
</tr>
<tr>
<td>SF</td>
<td>500</td>
<td>200</td>
<td>2</td>
<td>7.00E04</td>
<td>6.10E04</td>
<td>4.04E04</td>
<td>2.04E05</td>
</tr>
<tr>
<td>SF</td>
<td>500</td>
<td>200</td>
<td>4</td>
<td>3.90E05</td>
<td>1.19E05</td>
<td>6.62E02</td>
<td>3.22E07</td>
</tr>
<tr>
<td>SF</td>
<td>500</td>
<td>200</td>
<td>4</td>
<td>5.69E04</td>
<td>5.18E04</td>
<td>2.26E03</td>
<td>3.71E05</td>
</tr>
</tbody>
</table>

Fig. 3. Control cost versus number of key nodes for directed (a) ER and (b) SF networks with \( N = 100 \) and \( \langle k \rangle = 7.5 \).

Fig. 4. Comparison between the proposed algorithms for directed networks with different mean degree. (a) Directed ER network with \( N = 50 \) and \( M = 20 \). (b) Directed SF network with \( N = 50 \) and \( M = 20 \).

by LD-IADMM, based on the results of ER and SF networks with \( N = 50 \) and \( M = 15 \) presented in Fig. 5 which are obtained by averaging over ten realizations. It is seen that the fraction of key nodes is significantly high among nodes with low in-degree and medium in-degree. Clearly, key nodes tend to avoid nodes with high in-degree.

To further validate the effectiveness of the algorithms, larger ER and SF networks with \( N = 500 \) are considered by comparing with the random allocation method (RAM) [30]. Specifically, for RAM, first we apply maximum matching to obtain \( N_D \) driver nodes for ensuring the controllability of dynamic systems. Second, we randomly choose the rest \( M - N_D \) nodes. Combining previously selected \( N_D \) driver nodes, we totally choose \( M \) nodes which are regarded as the “key node set.” Third, to reconstruct \( B' \), i.e., if node \( i \) is the \( k \)th node in the key node set, we set \( B'_{ki} = 1 \), otherwise, \( B'_{ki} = 0 \). The comparison between PGME, D-IADMM, LD-IADMM, and RAM is shown in Table II. It is obvious that performances of D-IADMM and LD-IADMM are much better than PGME and RAM.

We also apply our proposed algorithm on real-world networks, including electronic circuit network, food Web, and social networks. Four algorithms, PGME, D-IADMM, LD-IADMM, and RAM are compared. The obtained results
TABLE III
PERFORMANCE COMPARISON BETWEEN PROPOSED ALGORITHMS IN REAL-WORLD NETWORKS

<table>
<thead>
<tr>
<th>Data set</th>
<th>Networks</th>
<th>N</th>
<th>M</th>
<th>PGME</th>
<th>D-IADMM</th>
<th>LD-IADMM</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic circuit</td>
<td>circuit-208</td>
<td>122</td>
<td>70</td>
<td>7.48E03</td>
<td>6.01E03</td>
<td>5.78E02</td>
<td>9.16E05</td>
</tr>
<tr>
<td></td>
<td>circuit-420</td>
<td>252</td>
<td>100</td>
<td>2.62E06</td>
<td>1.13E06</td>
<td>8.72E03</td>
<td>2.15E08</td>
</tr>
<tr>
<td>Food web</td>
<td>Rhode</td>
<td>25</td>
<td>12</td>
<td>2.52E03</td>
<td>2.95E03</td>
<td>8.01E01</td>
<td>2.83E05</td>
</tr>
<tr>
<td></td>
<td>Maspalomas</td>
<td>30</td>
<td>10</td>
<td>3.83E05</td>
<td>2.50E05</td>
<td>1.28E02</td>
<td>1.18E08</td>
</tr>
<tr>
<td>Social</td>
<td>cons-free-rev</td>
<td>46</td>
<td>12</td>
<td>1.23E09</td>
<td>3.29E08</td>
<td>1.51E05</td>
<td>8.20E09</td>
</tr>
<tr>
<td></td>
<td>phys-friend-rev</td>
<td>228</td>
<td>90</td>
<td>2.02E07</td>
<td>4.67E06</td>
<td>1.30E03</td>
<td>1.07E09</td>
</tr>
</tbody>
</table>

Fig. 5. In-degree distribution of key nodes in directed (a) ER and (b) SF networks with \( N = 50 \), \( M = 15 \), and \( \langle k \rangle = 5 \).

are summarized in Table III. It can be noted that the control cost obtain by LD-IADMM is 1–6 orders of magnitude lower than the cost obtained by the other three algorithms.

VII. CONCLUSION

In this paper, we investigated the problem of locating key nodes in one-to-one connection with external control sources such that the control cost is minimum. We first formulate the problem in Boolean constraints. By relaxing them into their convex hull, an algorithm named IADMM is introduced to solve the convex relaxed problem. Through exploiting the nodes’ degree information, D-IADMM is proposed to obtain the solution of the original problem and pinpoint key node set. In addition, LD-IADMM is introduced by employing the technique of LO on D-IADMM to improve the performance. The effectiveness of the proposed algorithms is validated on ER network, SF network, and real-life networks. Several interesting findings are made, which provide a better understanding and deeper insight of controlling complex networks from the energy point of view.

APPENDIX A
PROOF OF PROPOSITION 1

For convenience, let \( P = e^{-At} \), and \( P_i(t) \) be its \( i \)th column. Then, \( \mathcal{E}(b) \) can be written as

\[
\mathcal{E}(b) = \text{tr} \left( \sum_{i=1}^{N} P_i(t)P_i^T(t) dt \right)^{-1}
\]

\[
= \text{tr} \left( \sum_{i=1}^{N} P_i(t)P_i^T(t) dt \right)^{-1}.
\]

To prove the result, we must show that

\[
\mathcal{E}(\alpha a + (1 - \alpha) b) \leq \alpha \mathcal{E}(a) + (1 - \alpha) \mathcal{E}(b)
\]

with equality if and only if \( a = b \) for all \( \alpha \in (0, 1) \).

Based on (34), we have

\[
\mathcal{E}(\alpha a + (1 - \alpha) b) = \text{tr} \left( \alpha \sum_{i=1}^{N} P_i(t)P_i^T(t) dt \right)
\]

\[
+ (1 - \alpha) \sum_{i=1}^{N} P_i(t)P_i^T(t) dt.
\]

For simplicity, let \( E_1 = \sum_{i=1}^{N} a_i \int_0^T P_i(t)P_i^T(t) dt \) and \( E_2 = \sum_{i=1}^{N} b_i \int_0^T P_i(t)P_i^T(t) dt \). With Assumption 1, it is obvious that \( E_1 \) and \( E_2 \) are positive definite. According to Lemma 3, we can write \( E_1 = SS^T \), \( E_2 = \Lambda S^T \) for some nonsingular matrix \( S \) and \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_N) \) with each \( \lambda_i > 0 \). Then it can be obtained that

\[
\mathcal{E}(\alpha a + (1 - \alpha) b) = \text{tr} \left( (\alpha E_1 + (1 - \alpha) E_2)^{-1} \right)
\]

\[
= \text{tr} \left( (\alpha SS^T + (1 - \alpha) \Lambda S^T)^{-1} \right)
\]
of the proximal operator of its indicator function [47], the update

\[ \Pi_{S_1} \left( \mathbf{b}^{k+1} - \mu \left( \nabla \mathcal{E}(\mathbf{b}^k) + \lambda^k + \rho \left( \mathbf{b}^k - z^k \right) \right) \right) \]

Let \( s_1, \ldots, s_n \) be the positive main diagonal entries of the positive definite matrix \( S^{-1} S^{-T} \). We have

\[ \mathcal{E}(\mathbf{a} + (1 - \alpha) \mathbf{b}) = \sum_{i=1}^{N} (\mathbf{a} + (1 - \alpha) \lambda_i) s_i \]

\[ \leq \sum_{i=1}^{N} (\lambda_i) s_i \]

\[ = \alpha \operatorname{tr} \left( S^{-1} S^{-T} \right) + (1 - \alpha) \operatorname{tr} \left( \Lambda^{-1} S^{-1} S^{-T} \right) \]

\[ = \alpha \operatorname{tr} \left( S S^{-T} \right)^{-1} + (1 - \alpha) \operatorname{tr} \left( S A S^{-T} \right)^{-1} \]

\[ = \alpha \operatorname{tr} (E_1)^{-1} + (1 - \alpha) \operatorname{tr} (E_2)^{-1} \]

\[ = \alpha \mathcal{E}(\mathbf{a}) + (1 - \alpha) \mathcal{E}(\mathbf{b}). \quad (37) \]

The equality holds if \( \lambda_i = 1 \), which implies \( \mathbf{a} = \mathbf{b} \).

Therefore, with the optimal condition, we have

\[ 0 \in \mu \partial \phi_1 (\mathbf{b}^{k+1}) + \mathbf{b}^{k+1} - \left( \mathbf{b}^k - \mu \left( \nabla \mathcal{E}(\mathbf{b}^k) + \lambda^k + \rho \left( \mathbf{b}^k - z^k \right) \right) \right). \]

Rearranging the terms of above equation yields the desired result.

**REFERENCES**


Jie Ding received the B.Eng. degree in automation from Harbin Engineering University, Harbin, China, in 2012 and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2018. She is currently with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. Her current research interests include machine learning, pattern recognition, convex optimization, and complex networks.

Changyun Wen (F’10) received the B.Eng. degree in automatic control from Xi’an Jiaotong University, Xi’an, China, in 1983 and the Ph.D. degree in automatic control from the University of Newcastle, Callaghan, NSW, Australia, in 1990.

From 1989 to 1991, he was a Post-Doctoral Fellow with the University of Adelaide, Adelaide, SA, Australia. Since 1991, he has been with the School of EEE, Nanyang Technological University, Singapore, where he is currently a Full Professor. His current research interests include control systems and applications, intelligent power management system, smart grids, and model-based online learning and system identification.

Dr. Wen was a recipient of the IES Prestigious Engineering Achievement Award from the Institution of Engineers, Singapore, in 2005. He is an Associate Editor of a number of journals, including Automatica, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and IEEE CONTROL SYSTEMS Magazine. He is the Executive Editor-in-Chief of the Journal of Control and Decision. He served for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL as an Associate Editor from 2000 to 2002. He has been actively involved in organizing international conferences playing the roles of the general chair, the general co-chair, the technical program committee chair, a program committee member, the general advisor, and the publicity chair. He is a member of IEEE Fellow Committee from 2011 to 2013 and a Distinguished Lecturer of IEEE Control Systems Society from 2010 to 2013.

Guoqi Li received the B.Eng. degree in automation from the Xi’an University of Technology, Xi’an, China, in 2004, the M.Eng. degree in control engineering from Xi’an Jiaotong University, Xi’an, in 2007, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2011.

He was a Scientist with the Data Storage Institute and Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore, from 2011 to 2014. Since 2014, he has been with the Department of Precision Instrument, Tsinghua University, Beijing, China, where he is currently an Associate Professor. He has published over 80 journal and conference papers. His current research interests include brain inspired computing, complex systems, machine learning, neuromorphic computing, and system identification.

Dr. Li is an Editorial-Board Member and a Guest Associate Editor for Frontiers in Neuroscience (Neuromorphic Engineering section). He services as a reviewer for a number of international journals and has also been actively involved in professional services, such as serving as an International Technical Program Committee Member and the Track Chair for international conferences.