Identification of Network Topology via Quadratic Optimization

Chuangchuang Sun and Ran Dai

Abstract—Identification of network topology is to estimate the topology of a controllable and observable network with given number of nodes such that the identified network will satisfy the response between specified input and observed output. This paper examines the network topology identification (NTI) problems to find the original graph Laplacian from input-output data. A ‘similar’ set of state-space matrices satisfying the input-output response is firstly constructed through system identification procedure. Based on the similarity relationship, we reformulate the NTI problems as general Quadratically Constrained Quadratic Programming (QCQP) problems. The QCQP problem is then transformed into semidefinite programming (SDP) problem with a rank one constraint. An iterative rank minimization method is proposed to gradually approach the optimal solution. Examples are presented to verify the convergence of the proposed method.

Index Terms—Network Identification; Quadratically Constrained Quadratic Programming; Semidefinite Programming; Nonconvex Optimization

I. INTRODUCTION

A priori knowledge of network structure/topology is essential in many types of multi-agent involved missions, where interaction among agents is determined by the underlying network structure. For example, successful control of leader-follower network toward desired states requires identifying node indices of leaders, connectivity between leaders and followers, and interaction relationship among followers [1]. In fact, obtaining network topology is a prerequisite for operations such as spacecraft formation, mobile robot rendezvous, unmanned aerial vehicle flocking [2]. In this paper, we address the problem of network topology identification (NTI) by tracking input and output data observed in discrete time sequence.

Extensive work has been developed in the area of identification for a linear time-invariant (LTI) system which is controllable and observable. The well-known Kung [3] and subspace methods [4] find ‘similar’ state space matrices that match the response between input and output. Such ‘black box’ model constructed by input-output data is applied in this paper for solving NTI problems. We assume the dynamics of each agent in a connected network is governed by the consensus protocol. Thus the linear transfer matrix of LTI is defined by the negative graph Laplacian that directly reflects the network topology. However, finding ‘similar’ state-space matrices is far away from the objective of identifying the exact network topology.

Previous work in [5] used the ‘black-box’ setup to establish the generating function of graph Laplacian by observing input-output data from selected network nodes. Work in [6], meanwhile, used an integrated spectral characterization of graphs and similarity transformation approach to find an approximate graph Laplacian. Reconstruction of tree-like networks and sparse networks can be found in recent work of [7]–[10]. However, mapping the exact graph Laplacian from ‘similar’ state-space matrices constructed by input-output data is very challenging. The major reason comes from the involvement of unknown binary variables representing the edge set of the network. For a network with given number of nodes, the number of network topology configurations is an exponential function about its number of nodes. It will be extremely time consuming to find the exact graph Laplacian that has the closest spectra to the ‘similar’ state transfer matrix while keeping the similarity properties.

In this paper, we reformulate the similarity transformation between graph Laplacian and ‘similar’ state space matrices as bilinear constraints. The NTI problem is then transformed as an optimization problem with bilinear constraints, as well as constraints on the Laplacian structure. We further generalize the NTI problem as a nonconvex quadratically constrained quadratic programming (QCQP) problem, classified as NP-hard.

Efforts on solving QCQP problems have been pursued in two directions, obtaining the bounds on the optimal value and finding a feasible solution [11], [12]. Extensive relaxation methods have been investigated to obtain the bounds on the optimal value of QCQP. The semidefinite programming (SDP) relaxation introduces a rank one matrix to replace the quadratic objective and constraints with linear matrices equalities/inequalities. However, the nonlinear rank one constraint on the unknown matrix is substituted by semidefinite relaxation. In general, the SDP relaxation reaches a tighter bound on the optimal value than one obtained from linear relaxation [13]. A detailed discussion of various relaxation approaches and the comparison of their relative accuracy is provided in [14].

However, finding a bound on the optimal value of QCQP does not imply generating an optimal solution, not even a feasible solution. One of the efforts for obtaining a feasible solution utilizes an iterative linearization approach to gradually improve the objective value [13]. However, this method does not provide any guarantee of global optimality and convergence. Another approach is to generate randomized samples and solve the QCQP on average of the distribution. However, the randomization approach does not apply to problems with equality constraints and the optimality is not guaranteed either. Branch and bound (BNB) method has been frequently utilized to search for the optimal solution of nonconvex problems [15], [16]. Although BNB can lead to global optimal solution, the searching procedure is time consuming, especially for large scale optimization problems.

Inspired by the SDP relaxation, we focus on finding the unknown rank one matrix. The focus is then transferred to how to reduce the rank of the unknown matrix to one. In our previous work of [17], an iterative rank minimization (IRM) approach has been proposed to gradually approach the rank one constraint on the unknown matrix when solving homogeneous QCQPs. Due to the nonhomogeneous nature of QCQP formulation in NTI problems, we explore the approach to solve inhomogeneous QCQPs. Furthermore, convergence analysis on the proposed IRM approach is provided.

The paper is organized as follows. In §II, we formulate the NTI problem and transform the original formulation into QCQP problem. We then introduce an iterative method in §III to solve inhomogeneous QCQP with convergence analysis. The simulation results for NTI with different nodes are presented in §IV. We conclude the paper in §V.
II. PROBLEM FORMULATION FOR NTI

A. LTI System Identification

We start with introducing the black-box approach of identifying a LTI system by observing input-output data. Consider a linear continuous system in state-space format

\[ \dot{x}(t) = Ax(t) + Bu(t) \]
\[ y(t) = Cx(t) \]  

(2.1)

where \( x \in \mathbb{R}^n \) is the state vector, \( u \in \mathbb{R}^r \) is the input vector, \( y \in \mathbb{R}^m \) is the output vector, \( A \in \mathbb{R}^{n \times n} \), \( B \in \mathbb{R}^{n \times r} \), and \( C \in \mathbb{R}^{m \times n} \) are the state-space matrices. Through discrete integration, the continuous linear differential equation in (2.1) can be integrated at discrete sampling time \( t_k \), \( k = 1, \ldots, K \), and reformulated as discrete LTI system in the format

\[ x(t_{k+1}) = A_d x(t_k) + B_d u(t_k) \]
\[ y(t_k) = C_d x(t_k) \]  

(2.2)

where \( A_d = e^{A T} \), \( B_d = (\int_0^T e^{A T} dT) B \), and \( C_d = C \). For a given \( A_d \), \( A = 1/T \log_{A_d} A_d \), where \( \log_{A_d} \) denotes the matrix logarithm. By propagation, the input-output expression for discrete LTI at sampling sequence \( p \) is expressed as

\[ y(p) = H(p) u(p) \]  

(2.3)

where \( H(p) = \Gamma(p) \Omega(p) \) denotes the Hankel matrix and matrices \( \Gamma(p) \) and \( \Omega(p) \) are defined by the \( A_d, B_d, C_d \) in (2.2) as \( \Gamma(p) = [ C_d \ A_d A_d \ldots \ A_d^{p-1} B_d ] \) and \( \Omega(p) = [ B_d \ A_d B_d \ldots \ A_d^{p-1} B_d ] \), respectively. By tracking a series of input and output data at discrete sampling time, the Hankel matrix can be constructed via impulse response parameters [3] or least-square estimation method [4]. There are many methods to reconstruct the extended controllability and observability matrices, \( \Gamma(p) \) and \( \Omega(p) \), from the Hankel matrix. For example, a commonly used one is the singular value decomposition method [3]. Results of this identification procedure leads to realization of 'similar' state-space matrices set \( (A_T, B_T, C_T) \) that represents the identified LTI system

\[ \dot{x}(t) = A_T x(t) + B_T u(t) \]
\[ y(t) = C_T x(t) \]  

(2.4)

while satisfying the input-output response. Furthermore, from Kalman’s theorem [18], the set \( (A, B, C) \) and \( (A_T, B_T, C_T) \) are linearly related by a nonsingular matrix \( T \in \mathbb{R}^{n \times n} \) such that

\[ A_T = T A T^{-1}, \quad B_T = T B, \quad C = C T^{-1} \]  

(2.5)

This system identification procedure can be applied to the NTI problem where the state-space transfer matrix is solely determined by the network topology.

B. Consensus Protocol of Dynamic Networks

We consider a undirected network \( G = (V, E) \) with vertex set \( V = \{1, 2, \ldots, n\} \) and edge set \( E \) consisting of two element subsets of \( V \). We use nodes or agents interchangeably with vertices. The connection among the nodes in the undirected network \( G = (V, E) \) is expressed by the entries of the adjacency matrix with \( [A(G)]_{ij} = 1 \) when \( v_i, v_j \in E \) and \( [A(G)]_{ij} = 0 \) otherwise. Since the adjacency matrix for a graph on \( n \) nodes, \( A(G) \), is symmetric, we use a set of binary variables comprised of \( n(n-1)/2 \) elements to determine off-diagonal entries of the \( A(G) \). We note that the diagonals are simply zeros. Using such a framework, we assign binary variable \( a_{ij} \) to represent the element \( [A(G)]_{ij} \) in \( A(G) \) with \( a_{ij} = a_{ji}, (i \neq j) \) and \( a_{ii} \) is set to be zero. The degree matrix \( \Delta(G) \) of the graph can also be expressed in terms of the binary variables \( a_{ij} \) as \( \Delta(G)_{ii} = \sum_{j=1}^{n} a_{ij} \) and \( \Delta(G)_{ij} = 0 \) \( (i \neq j) \). Therefore, the Laplacian \( L(G) = \Delta(G) - A(G) \) is completely determined by these binary variables.

Assuming nodes belong to the set \( I \in \mathcal{V} \) with cardinality \( r_I \) are selected to receive infused control signals \( u \). Meanwhile, nodes set \( O \in \mathcal{V} \) with cardinality \( r_O \) is used to observe output response \( y \). Recall that when \( x_i \) denotes the state of dynamic agent \( i \) in the connected network \( G \), the consensus protocol of the overall system with input \( u \) is represented by

\[ \dot{x}(t) = -L(G)x(t) + Bu(t), \]

(2.6)

which will drive each agent to the consensus set \( C = \{ x \in \mathbb{R}^n | x_i = x_j, \forall i, j \in V \} \) by exchanging state information with connected agents in the specified network \( G \). \( B \in \mathbb{R}^{n \times r_T} \) is the control matrix with element \( B_{ij} \) set as one if node \( i, i = 1, \ldots, n \), is selected as the \( j \)’th output, where \( j = 1, \ldots, r_T \). For the output, we have

\[ y(t) = Cx(t), \]

(2.7)

where \( C \in \mathbb{R}^{r_O \times n} \) is the observation matrix with element \( C_{ij} \) set as one if node \( i, i = 1, \ldots, n \), is selected as the \( j \)’th output, where \( j = 1, \ldots, r_O \). An illustrative example is demonstrated in Fig. 1.

![Fig. 1. Illustrative example of NTI concept](image-url)
by Laplacian structure. The above statement of NTI can be summarized as a constrained optimization problem, written as

\[
J = \min_{A,T} \| A - AT \|^2_F \\
\text{s.t.} \quad TA = AT = A^T, \quad B_T = TB, \quad C_T = C^T, \quad A_{i,j} = A_{j,i}, \quad V \forall i,j \in \{0, 1\}, \quad V v_i, v_j \in V, \quad i \neq j \quad A1 = 0,
\]

where \(A = -L(G)\). The above formulation including both binary variables in \(A\) and continuous variables in \(T\) (which is nonsingular as mentioned before) under nonlinear constraints is a mixed-integer nonlinear optimization problem. In addition, as \(AT\) is supposed to be similar to a negative laplacian matrix of a graph, it must have the same eigenstructure, namely, all eigenvalues should be negative except the one which is zero. This can be used to check the feasibility of (2.8) or the system identification procedures. For a given matrix \(\Phi \in \mathbb{R}^{n \times n}\), its Frobenius norm is determined by

\[
\| \Phi \|_F := \sqrt{\text{trace}(\Phi^T \Phi)} = \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \Phi_{i,j}^2 \right)^{\frac{1}{2}}.
\]

Furthermore, the binary variables \(A_{i,j}\) can be constrained via quadratic functions \(A_{i,j} = \min_{1 \leq q \leq 0, \quad V v_i, v_j \in V, \quad i \neq j} (A_{i,j} - A_T)^2\). With these transformations, problem in (2.8) can be transformed into a quadratic optimization problem with quadratic and linear constraints in the form of

\[
J = \min_{A,T} \sum_{i=1}^{n} \sum_{j=1}^{n} (A_{i,j} - A_T)^2 \\
\text{s.t.} \quad TA = AT = A^T, \quad B_T = TB, \quad C_T = C, \quad A_{i,j} = A_{j,i}, \quad A1 = 0.
\]

The first three constraints are from (2.5) and the following three are the properties of a graph Laplacian matrix. It's obvious that the quadratic constraints such as the binary variable constraints and bilinear constraints in (2.9) is nonconvex. Solving the NTI problem is then transformed as solving a nonconvex QCQP problem. An iterative approach is proposed in the following section to gradually approach the optimal solution.

III. SOLUTION FOR NONCONVEX QCQP PROBLEMS

A. Iterative Rank Minimization Algorithm

In general, the QCQP problem can be formulated as

\[
J = \min x^T Q_o x + q_o^T x \\
\text{s.t.} \quad x^T Q_j x + q_j^T x + c_j \leq 0, \quad V j = 1, \ldots, m,
\]

where \(x \in \mathbb{R}^n\), \(Q_j \in \mathbb{R}^{n \times n}\), \(q_j \in \mathbb{R}^n\), \(c_j \leq 0, \quad V j = 1, \ldots, m\). Here, \(\leq\) is used to represent \(\leq\) or \(=\). In particular, linear constraints can be included as well if the hessian matrix, \(Q_j\), is set as a zero matrix. For the cases where \(Q_j\) is negative semidefinite, indefinite, or the quadratic constraints have equality ones, the problem can be considered to be concave, which is NP-hard.

After introducing a rank one matrix \(X = xx^T\), the nonconvex QCQP problem in (3.10) is equivalent to [20]

\[
J = \min X \bullet Q_o + q_o^T x \\
\text{s.t.} \quad X \bullet Q_j + q_j^T x + c_j \leq 0, \quad V j = 1, \ldots, m \quad X = xx^T,
\]

where \(\bullet\) denotes the trace inner product. However, the rank one constraint \(X = xx^T\) is highly nonlinear. It’s necessary to reconsider the rank one constraint in other format.

**Proposition 3.1:** \(X = xx^T\) is equivalent to \(\text{rank}(\begin{bmatrix} x & x^T \end{bmatrix}) = 1\) and \(\begin{bmatrix} x & x^T \end{bmatrix} \geq 0\), where \(X \in \mathbb{R}^{n \times n}\) and \(x \in \mathbb{R}^n\).

\[
\text{Proof:} 1. \text{If } X = xx^T, \text{we get } \\
\begin{bmatrix} 1 & x^T \end{bmatrix} = \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} x & x^T \end{bmatrix}.
\]

which implies that \(\begin{bmatrix} 1 & x \end{bmatrix} \geq 0\) and \(\begin{bmatrix} 1 & x \end{bmatrix} x^T\) is a rank-1 matrix, denoted as \(\text{rank}(\begin{bmatrix} 1 & x \end{bmatrix}) = 1\).

2. If \(\text{rank}(\begin{bmatrix} 1 & x \end{bmatrix}) = 1\), and \(\begin{bmatrix} 1 & x \end{bmatrix} \geq 0\), we get
\[
\begin{bmatrix} 1 & x \end{bmatrix} x^T = LL^T, \text{ where } L \in \mathbb{R}^{n \times m}.
\]

As the element in the first row and first column of \(\begin{bmatrix} 1 & x \end{bmatrix} x^T\) is 1, it means that the first element of \(L\) must be \(\pm 1\), which implies \(L = [\pm 1, L^T]\), where \(L^T \in \mathbb{R}^n\). Then it can be seen that \(x = \pm L^T x = L^T L\). Hence, we can get \(X = xx^T\).

From proposition (3.1), we can transform problem in (3.11) into the following form

\[
J = \min X \bullet Q_o + q_o^T x \\
\text{s.t.} \quad X \bullet Q_j + q_j^T x + c_j \leq 0, \quad V j = 1, \ldots, m \quad \text{rank}(\begin{bmatrix} 1 & x \end{bmatrix} x^T) = 1.
\]

Satisfying rank one constraint for an unknown matrix is computationally complicated. The direct method is to examine the eigenvalues of the matrix. When only one eigenvalue is nonzero, we can claim that the matrix has rank one. However, for a unknown matrix \(X \in \mathbb{R}^{n \times n}\), there is no way to examine its eigenvalues before it is determined. Although heuristic search methods have been used to minimize the rank of symmetric or asymmetric matrix, they cannot guarantee that the rank of the final matrix is one [21, 22].

We focus on the fact that when a matrix rank is one, it has only one nonzero eigenvalue. Therefore, instead of making constraint on the rank, we focus on constraining the eigenvalues of \(X\) such that the \(n - 1\) smallest eigenvalues of \(X\) are all zero. The eigenvalue constraints on matrices have been used for graph design [23] and are applied here for rank minimization. Before addressing the detailed IRM approach, we first provide necessary observations that will be used subsequently in the approach.

**Proposition 3.2:** The second largest eigenvalue \(\lambda_{n-1}\) of matrix \(X \in \mathbb{R}^{n \times n}\) is less than \(r\) if and only if \(r I_{n-1} - V^T X V \geq 0\) where \(I_{n-1} \in \mathbb{R}^{(n-1) \times (n-1)}\) is the identity matrix, \(V \in \mathbb{R}^{n \times (n-1)}\) are the eigenvectors corresponding to the \(n - 1\) smallest eigenvalues of \(X\).

\[
\text{Proof:} \text{Assume the eigenvalues of } X \text{ is sorted in descending orders in the form of } [\lambda_n, \lambda_{n-1}, \ldots, \lambda_1]. \text{ Since the Rayleigh}
\]
quotient of an eigenvector is its associated eigenvalue, then

\[ V^T U V = \begin{bmatrix}
\lambda_{n-1} & 0 & \cdots & 0 \\
0 & \lambda_{n-2} & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \lambda_1
\end{bmatrix}, \]

Hence,

\[ r I_{n-1} - V^T U V = \begin{bmatrix}
r - \lambda_{n-1} & 0 & \cdots & 0 \\
0 & r - \lambda_{n-2} & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & r - \lambda_1
\end{bmatrix}. \]

Therefore \( r \geq \lambda_{n-1} \) if and only if \( r I_{n-1} - V^T X V \succeq 0 \).

**Corollary 3.3:** When \( r = 0 \) and \( X \) is nonzero positive semidefinite matrix, \( X \) is rank one if and only if \( r I_{n-1} - V^T X V \succeq 0 \) where \( V \in \mathbb{R}^{n \times (n-1)} \) are the eigenvectors corresponding to the \( n-1 \) smallest eigenvalues of \( X \).

From the above discussion, we will substitute the rank one constraint in (3.12) by the semidefinite constraint

\[ r I_{n-1} - V^T X V \succeq 0 \quad (3.14) \]

where \( r = 0 \) and \( V \in \mathbb{R}^{n \times (n-1)} \) are the eigenvectors corresponding to the \( n-1 \) smallest eigenvalues of \( X \). However, before we solve \( X \), we cannot obtain the exact \( V \) matrix, thus an iterative method is proposed to gradually minimize the rank of \( X \). At each step \( k \), we will solve the following semidefinite programming problem formulated as

\[
\begin{aligned}
J &= \min_{X_k} Q_j + q_j^T x_k + c_j \leq 0, \quad \forall j = 1, \ldots, m \\
& \quad \begin{bmatrix} 1 & x_k^T X_k \end{bmatrix} \succeq 0 \\
r_k I_n - V_{k-1}^T \begin{bmatrix} 1 & x_k^T X_k \end{bmatrix} V_{k-1} \succeq 0,
\end{aligned}
\]

where \( w > 1 \) is a weighting factor and \( w^k \) increases with iteration, \( V_{k-1} \) are the eigenvectors corresponding to the \( n \) smallest eigenvalues of \( \frac{1}{X_k} \) solved at previous iteration \( k - 1 \). At each step, we are trying to optimize the original objective function and at the same time minimize parameter \( r \) such that when \( r = 0 \), the rank one constraint on \( X \) is satisfied. The above approach is repeated until \( r \leq \delta \), where \( \delta \) is a small threshold for stopping criteria.

We summarize the IRM algorithm below.

**Algorithm: Iterative Rank Minimization**

**Input:** Problem parameters \( Q_j, q_j, c_j, w, j = 1, \ldots, m \)

**Output:** Unknown rank one matrix \( X \) and unknown state vector \( x \)

**begin**

1) **initialize** Set \( k = 0 \), solve the convex relaxation of problem (3.13) by dropping the last rank one constraint and obtain \( V_k \) from \( \begin{bmatrix} 1 & x_k^T X_k \end{bmatrix} \), set \( k = k + 1 \)

2) **while** \( r_k > \delta \)

3) Solve problem (3.15) and obtain \( x_k, X_k, r_k \)

4) Update \( V_k \) from \( \begin{bmatrix} 1 & x_k^T X_k \end{bmatrix} \)

5) \( k = k + 1 \)

6) **end while**

7) Find \( x \) and \( X \)

**end**

### B. Convergence of IRM Algorithm

In the following, we provide the convergence analysis of the proposed IRM method. Considering the following problem.

\[
J = \min \begin{bmatrix} x^T & t \end{bmatrix} \begin{bmatrix} Q_0 & q_0/2 \\
q_0/2 & 0 \end{bmatrix} \begin{bmatrix} x \\
t \end{bmatrix} \\
s.t. \begin{bmatrix} x^T & t \end{bmatrix} \begin{bmatrix} Q_j & q_j/2 \\
q_j/2 & 0 \end{bmatrix} \begin{bmatrix} x \\
t \end{bmatrix} + c_j \leq 0, \forall j = 1, \ldots, m, \\
t^2 = 1
\]

(3.16)

The above homogeneous QCQP is equivalent to the inhomogenous QCQP in (3.10) based on the fact that if \( \begin{bmatrix} x^*, t^* \end{bmatrix} \) is a solution pair of (3.16), then \( x^*/t^* \) is the solution of (3.10). Without loss of generality, we provide the convergence analysis in the content of homogenous QCQP formulation in the following.

At each IRM step, the subproblem considered for homogeneous QCQP is

\[
J = \min_{X_k} X_k \bullet C_0 + w^k r_k \\
s.t. \quad X_k \bullet C_i + c_i \leq 0, \quad \forall i \in I \\
X_k \bullet C_j + c_j = 0, \quad \forall j \in E
\]

(3.17)

where \( I \) and \( E \) represent the index sets of inequality and equality constraints, respectively.

**Proposition 3.4:** \( \lim_{k \to +\infty} r_k = 0 \) in the IRM algorithm for a feasible optimization problem formulated in (3.17).

**Proof:** A Lagrange dual function of (3.17) is constructed as

\[
\mathcal{L} = X_k \bullet C_0 + w^k r_k \\
+ \sum_{i \in I} \lambda_i (X_k \bullet C_i + c_i) + \sum_{j \in E} \mu_j (X_k \bullet C_j + c_j) \\
- \text{trace}(S_1 (r_k I_{n-1} - V_{k-1}^T X_k V_{k-1})) \\
- \text{trace}(S_2 X_k)
\]

where \( \lambda_i \in \mathbb{R} \geq 0, \mu_j \in \mathbb{R}, S_1, S_2 \in \mathbb{S}_+^{n-1} \) and \( S_2 \in \mathbb{S}_+^m \) are the Lagrange dual multipliers. The dual function is then expressed as

\[
g(\lambda, \mu, S_1, S_2) = \inf_{X_k, r_k} \mathcal{L}(X_k, r_k, \lambda, \mu, S_1, S_2).
\]

(3.19)

Based on the Lagrange dual in (3.18), the first order condition is
derived as
\[
\frac{\partial L}{\partial X_k} = C_0 + \sum_{i \in \mathcal{X}} \lambda_i C_i + \sum_{j \in \mathcal{X}} \mu_j C_j + V_{k-1} S_1 V_{k-1}^T - S_2
\]
\[
= 0
\]
\[
\frac{\partial L}{\partial r_k} = w_k - \text{trace}(S_1) = 0.
\]
\[
\text{(3.20)}
\]
Consequently, we build the dual problem as follows,
\[
J = \max_{\lambda, \mu, S_1, S_2} \sum_{i \in \mathcal{X}} \lambda_i C_i + \sum_{j \in \mathcal{X}} \mu_j C_j
\]
\[
s.t. \quad C_0 + \sum_{i \in \mathcal{X}} \lambda_i C_i + \sum_{j \in \mathcal{X}} \mu_j C_j + V_{k-1} S_1 V_{k-1}^T - S_2 = 0
\]
\[
\text{and } w_k - \text{trace}(S_1) = 0
\]
\[
\text{(3.21)}
\]
It is obvious that the problem described in (3.17) is convex. Moreover, it can be verified that the Slater’s constraints are satisfied. We come to the conclusion that the strong duality holds and thus at the optimal solution points of iteration \(k\) and \(k+1\), i.e., \((X_k^*, r_k^*, \mu_k^*, (S_k^1), (S_k^2))\) and \((X_{k+1}^*, r_{k+1}^*, \mu_{k+1}^*, (S_{k+1}^1), (S_{k+1}^2))\), we have that the objective value of the primal program is equal to that of the dual program
\[
C_0 \bullet X_k^* + w_k r_k^* = \sum_{i \in \mathcal{X}} (\lambda_i^*) C_i + \sum_{j \in \mathcal{X}} \mu_j C_j
\]
\[
\text{and } w_k \bullet r_k^* = \sum_{i \in \mathcal{X}} (\lambda_i^*) C_i + \sum_{j \in \mathcal{X}} \mu_j C_j
\]
\[
\text{(3.22)}
\]
Since the linear (matrix) programming in (3.17) is feasible, the dual problem in (3.21) is bounded. On the assumption that \(X_k\) is finite, subtract one from another of the above two equations, for \(w > 1\) we can get
\[
\lim_{k \to +\infty} (w r_{k+1}^* - r_k^*) = 0
\]
\[
\lim_{k \to +\infty} (w g_{k+1} - g_k) - C_0 \bullet (X_{k+1}^* - X_k^*) = 0.
\]
\[
\text{Hence,}
\]
\[
\lim_{k \to +\infty} r_{k+1}^* = \frac{1}{w} \lim_{k \to +\infty} r_k^*.
\]
\[
\text{Since } r^* = 0, r_{k+1}^* > 0, r_k^* > 0, \text{and } 0 < \frac{1}{w} < 1, \text{then}
\]
\[
\lim_{k \to +\infty} \frac{|r_{k+1}^* - r_k^*|}{|r_k^* - r_{k-1}^*|} = \frac{1}{w}.
\]
\[
\text{The above equation indicates that } r_k^* \text{ converges to } r^* \text{ linearly and } w_k r_k^* \text{ is non-increasing.}
\]
\[
\text{Proposition 3.5: } X_k \text{ converges to a local optimal solution } X^* \text{ in the IRM.}
\]

\[
\text{Proof: When } k \to \infty, r \to 0, \text{the third constraint in (3.17) will become}
\]
\[
\lim_{k \to \infty} -V_k^T X_{k+1} V_k \geq 0.
\]
\[
\text{(3.22)}
\]
For a positive semidefinite matrix \(X_{k+1} \geq 0\), it leads to
\[
\lim_{k \to \infty} V_k^T X_{k+1} V_k = 0.
\]
\[
\text{(3.23)}
\]
Since the above is a similarity transformation, and as known before, \(V \in \mathbb{R}^{n \times (n-1)}\) and \(X_{k+1} \in \mathbb{S}^n\), it implies that the rank of \(X_k\) is no more than one when \(k\) approaches infinity. Hence we have
\[
\lim_{k \to \infty} V_k^T X_k V_k = 0.
\]
\[
\text{(3.24)}
\]
Subtracting (3.24) from (3.23) yields
\[
\lim_{k \to \infty} V_k^T (X_{k+1} - X_k) V_k = 0.
\]
\[
\text{(3.25)}
\]
As \(X_k\) is a positive semidefinite matrix with a rank of no more than one, we can get
\[
\lim_{k \to \infty} X_{k+1} = \alpha X_k.
\]
\[
\text{(3.26)}
\]

The above relationship indicates that \(V_k\) is constant when \(k \to \infty\) since \(X_k\) and \(\alpha X_k\) share the same eigenvectors. Consequently, with fixed \(V_k\), the subproblem will converge to a local optimum.

IV. SIMULATION EXAMPLES
In this section, we provide two simulation examples to identify the topology of two graphs, composed of 5 and 6 nodes, respectively. In the six-node case, from system identification procedure introduced in §II-A, the ‘similar’ state-space matrices set \((A_T, B_T, C_T)\) is listed below, \(A_T = \)
\[
\begin{bmatrix}
-2.873 & -0.720 & 0.183 & -0.480 & 1.594 & 0.393 \\
-0.720 & -4.227 & -1.197 & -0.076 & -0.927 & -0.723 \\
0.183 & -1.197 & -5.17 & 0.139 & 0.368 & 0.351 \\
-0.479 & -0.077 & 0.139 & -4.625 & 0.726 & 0.678 \\
1.594 & -0.927 & 0.368 & 0.726 & -2.813 & 0.908 \\
0.398 & -0.723 & 0.351 & 0.678 & 0.908 & -2.325
\end{bmatrix}, \text{and } C_T = B_T^T. \text{ We define the nodes with indices } i = 1, 2, 3 \text{ as the selected input and output nodes. Under this assumption, matrices } B \text{ and } C \text{ are set as}
\]
\[
B = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \text{and } C = B^T, \text{ respectively.}
\]

By formulating the problem in the format described in (2.9) and applying the IRM method, the optimal transformation matrix \(T\) and the graph Laplacian are found as
\[
T_{opt} = \begin{bmatrix}
0.515 & -0.031 & -0.320 & -0.292 & -0.551 & -0.490 \\
-0.031 & 0.997 & -0.021 & -0.019 & -0.036 & -0.032 \\
-0.320 & -0.021 & 0.787 & -0.193 & -0.364 & -0.324 \\
-0.292 & -0.019 & -0.193 & 0.823 & -0.333 & -0.296 \\
-0.551 & -0.036 & -0.364 & -0.333 & 0.372 & -0.558 \\
-0.490 & -0.032 & -0.324 & -0.296 & -0.558 & 0.503
\end{bmatrix}
\]
\[
L_{opt} = \begin{bmatrix}
3 & -1 & -1 & 0 & -1 & 0 \\
-1 & 4 & 0 & -1 & -1 & 0 \\
0 & 1 & 0 & 4 & -1 & -1 \\
-1 & -1 & -1 & 4 & -1 & 0 \\
0 & 0 & 0 & 4 & -1 & 3
\end{bmatrix}
\]

Comparing with the original network demonstrated in Fig. 2(a), it is verified that the network topology obtained from the proposed identification procedure is identical to the original one. In addition, the history of the second smallest eigenvalue at each iteration of the IRM algorithm is shown in Fig. 2(b). It indicates that \(\lambda_{n-1}\), which is represented by \(r\) in (3.15), quickly reduces to zero within 18 steps, where the stopping threshold is set as \(\delta = 10^{-4}\). This fact verifies that we obtain a rank one matrix of \(X\) within a few iterative steps. Finding the network topology for five-node case is similar to the previous one. We define nodes with indices \(i = 1, 2\) as the input and output nodes. Following the same procedure, we can get the following results,
The similarity properties. The reformulated optimization problem is generalized as a nonconvex quadratically constrained quadratic programming (QCQP) problem. An iterative rank minimization method is proposed to gradually approach the QCQP solution. Convergence analysis of the proposed algorithm is provided. Examples are presented to demonstrate the feasibility and effectiveness of the algorithm in solving NTI problems.

**REFERENCES**


