Matrix Function Optimization under Weighted Boundary Constraints and Its Applications in Network Control

Guoqi Li*†, Pei Tang*, Chen Ma, Ran Wang, Gaoxi Xiao and Luping Shi†

***Abstract*—The matrix function optimization under weighted boundary constraints on the matrix variables is investigated in this work. An "index-notation-arrangement based chain rule" (*I-Chain rule*) is introduced to obtain the gradient of a matrix function. By doing this, we propose the weighted trace-constraint-based projected gradient method (WTPGM) and weighted orthornormal-constraint-based projected gradient method (WOPGM) to locate a point of minimum of an objective/cost function of matrix variables iteratively subject to weighted trace constraint and weighted orthonormal constraint, respectively. New techniques are implemented to establish the convergence property of both algorithms. In addition, compared with the existing scheme termed "orthornormal-constraint-based projected gradient method" (OPGM) that requires the gradient has to be represented by the multiplication of a symmetrical matrix and the matrix variable itself, such a condition has been relaxed in WOPGM. Simulation results show the effectiveness of our methods not only in network control but also in other learning problems. We believe that the results reveal interesting physical insights in the field of network control and allow extensive applications of matrix function optimization problems in science and engineering.**

Keywords: Matrix function optimization, Matrix variable, Weighted orthornormal constraint, Weighted trace constraint, Network control

I. Introduction

It is well known that the derivative is a fundamental tool in many science and engineering problems [1][2]. For a scalar function of a real variable, the derivative measures the sensitivity of function change with respect to such a variable, which has meaningful physical insights. For example, the derivative of the position of a moving object with respect to time is the object's velocity, and it measures how quickly the object position changes when time involves. However, finding the derivative of a function with respect to a real variable is not enough when one wants to describe a more complicated problem in which a function is determined by a set of variables. In such a case, the study of the derivative of a function with respect to a vector becomes necessary. Vector derivatives that take in vector variables are extremely important, where they arise throughout fluid mechanics [3], electricity and magnetism [4], elasticity [5], and many other areas of theoretical and applied physics [6]. Vector derivatives can be combined in different ways, such as divergence [7] and curl [8] operators, producing sets of identities that are also very important in physics.

A vector is a special form of a matrix in which all elements are organized in a line, and a matrix can always be stacked to a vector form. However, in various practical statistics and engineering problems, stacking a matrix into a vector will lose the physical meaning within each column. For example, in the problem of control of complex networks [9], we need to design an input matrix to achieve the control objective. The number of columns of the input matrix is the number of external control sources available, and stacking the input matrix into a vector makes the network become uncontrollable. In these cases, taking the derivative of a function with respect to a matrix variable becomes essential. To this end, we need to collect various partial derivatives of a single function with respect to many variables, and/or of a multivariate function with respect to a single variable, and obtain the total differential information. Thus, operations in finding a local maximum/minimum of a multivariate function solving differential equations can be significantly simplified via various gradient descent methods [10]. In this paper, optimization problems where vector variables and matrix variables are involved are termed by *vector function optimization* and *matrix function optimization* problems, respectively. The notation "vector" and "matrix" used here is commonly used in statistics and engineering, while the tensor index notation [11][12] is preferred in physics.

Motivated by an irresistible longing to understand the above issues, we moved from *vector function optimization* problems to *matrix function optimization* problems whose variables are matrices in our most recently work [13]. To accomplish this issue, we hope to know the gradient information of a matrix function. However, it is generally hard to obtain such information when matrices-by-matrices derivatives are involved. It should be noticed that the derivative of a vector function with respect to another matrix is a high-order tensor. For example, for a scalar cost function $E(B)$ where $B \in \mathbb{R}^{n\times m}$, it is a function of another matrix $Q \in \mathbb{R}^{p\times q}$. The derivative of cost function $E(B)$ with respect to B captures how the matrix variable B affects the cost function. To this end, we explore how the value of the element B_{kl} of the matrix B affects the

G.Li and P. Tang contributes equally to this work. G. Li, P. Tang, C. Ma and L. Shi are with the Department of Precision Instrument, Center for Brain Inspired Computing Research, Tsinghua University, Beijing, China, 100084 (email: liguoqi@mail.tsinghua.edu.cn (G. Li), tangp14@mails.tsinghua.edu.cn (P. Tang), macheng@mail.tsinghua.edu.cn(C. Ma) and lpshi@mail.tsinghua.edu.cn (L. Shi)). R. Wang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, (email: wangran@nuaa.edu.cn (R. Wang)). G. Xiao is with the School of EEE, Nanyang Technological University, Singapore, (email: EGXXiao@ntu.edu.sg, (G. Xiao)). † The corresponding authors.

value of element Q_{ij} of the matrix Q. Particularly, we need to differentiate $\frac{\partial Q_{ij}}{\partial B_{kl}}$ for all i,j and k,l. Such a differentiation results in a fourth order tensor [14][15]. In short it is an $m\times n$ matrix, and each of its entries is a $p\times q$ matrix.

Although vector derivative has been well established, matrix derivative is difficult. Currently there is no unified framework that can completely solve this problem [16]. Existing schemes mainly use two basic ways to deal with this issue, one is the Vec operator and Kronecker products arrangement [17][18], and the other is the index notation arrangement [19]. However, for implementations, there are a lot of intricacy and tedious calculation. The main difficulty here is keeping track of where things are put since a matrix variable may depend on numerous intermediate matrix variables. This situation becomes worse when $E(.)$ has a more complicated form. We find that index notation arrangement relatively simplifies the presentation and manipulation of differential geometry when doing matrix differentiation. Thus, we proposed an index-notation-arrangement based chain rule (*I-Chain rule*) in [13]. By obtaining the gradient of a matrix function using *I-Chain rule*, two iterative algorithms, namely, *trace-constraint-based projected gradient method* (TPGM) and *orthornormal-constraint-based projected gradient method* (OPGM) were presented to solve the matrix function optimization problems. Projection and normalization operators were utilized to establish the convergence of TPGM and OPGM. This work provided a unified framework which reveals important physical insights and deepens our understanding of various matrix function optimization problems, and inspires wide applications in science and engineering.

However, in the work [13], to guarantee the convergence of TPGM/OPGM, it is required that the gradient can be represented by the multiplication of a symmetrical matrix and the matrix variable itself. That is to say, the gradient of the cost function should be represented in the form of $\nabla E(B) = F(B)\cdot B$ where $F(B)\in\mathbb{R}^{N\times N}$ is symmetrical and $B_k\in\mathbb{R}^{N\times M}$. Although such an assumption holds in various applications, it indeed does not hold in some cases. For example, consider the case that $E(B) = tr((L-BX)^T(L-BX))$ where $tr(.)$ denotes the matrix trace function. In this case, the gradient $\nabla E(B)$ cannot be represented by $\nabla E(B_k) = F(B_k)\cdot B_k$. Therefore we investigate how to develop an algorithm and ensure its convergence in this work. We also find that the boundary constraints in [13] can be further relaxed. More particularly, by introducing an real symmetry positive definite weight matrix G, the trace constraint and the orthornormal constraint can be relaxed to the weighted trace constraint and weighted orthornormal constraint, respectively. The boundary constraints in [13] become special cases of this work in which G is an identity matrix.

The main problem we faced is how to deal with the non-symmetrical of $\nabla E(B)B^T$ for a formulated matrix objective function under more relaxed constraints. It brought this question up: a non symmetrical matrix diagonalized as its eigenvalues may not be all real values and therefore existing techniques cannot guarantee the convergence of the TPGM/OPGM algorithm in this case. To this end, we propose the weighted trace-constraint-based projected gradient method (WTPGM) and the weighted orthornormal-constraint-based projected gradient method (WOPGM) to locate a point of minimum of an objective/cost function of matrix variables iteratively subject to weighted trace boundary constraint condition and weighted orthornormal constraint condition, respectively. Our main idea is to replace $\nabla E(B)$ by $\nabla E(B)B^TGB$, which can be represented by $F(B)\cdot B$ such that $F(B) = \nabla E(B)B^TG$. The key technique in guaranteeing the convergence of WTPGM is to obtain the orthonormal basis of GB in the iteration process. While for WOPGM, the essential issue is to the establishment of the value condition of λ_k in the iteration process. Introducing the parameter λ_k is similar to the idea of Levenberg-Marquardt stabilization, also known as the damped least-squares (DLS) [20][21] method, which is generally used to solve non-linear least squares problems. In the next section, we shall prove that $E(B_k)$ is convergent to $E(B^*)$ as $k\to\infty$, with B^* having orthonormal columns, provided that the step length η is sufficiently small. Thus, both the assumption on the gradient of the cost function and requirement on the boundary constraint have been relaxed in this paper. This means that we are able to extend our method regarding the optimization of matrix functions to more extensive applications in science and engineering. Simulation results show the effectiveness of our framework.

To show the effectiveness of our method, various case studies including two in the area of network control are illustrated. In the first case study, we focus on how to identify nodes to which the external control sources are connected so as to minimize a pre-defined energy cost function of a control strategy. Different from the work in [13], a positive definite diagonal weight matrix G reflecting the restriction on each external control source is considered. The matrix function optimization model built in this work allows us to investigate how G can affect the control cost. By applying WTPGM and WOPGM, we uncover that the control cost is related to the condition number of G. This interesting observation may lead to heuristic algorithm design for the minimum cost control of large scale of real life complex networks, which deserves great attention for the future research. In the second case study, we consider controlling directed networks by only evolving the connection strengths on a fixed network structure. In this case, the topology matrix A becomes a matrix variable of the control cost function while the input matrix B is fixed. By this example, we also show that the proposed WTPGM and WOPGM are applicable when G becomes an identity matrix, which suggests that WTPGM and WOPGM are more general than TPGM and OPGM, respectively. In addition, we uncover that, when the control sources are evenly allocated, the system can be considered as a few identical subsystems and the control cost attains its minimum. This is meaningful when one want to explore how network topology evolution affects the cost of controlling these networks.

There are some literatures considering optimization problems where matrix variables are involved under specific constraints [13] [22][23][24][25][26][27] [28][29][22][30][31] [32][33][34] [35] [36] [37] [38]. However, the cost functions in these works are in relatively simple and specific forms [24][25]. For example, they are usually simple trace functions such as $tr(X^TAX)$ where X are the matrix variable and

A is a given symmetrical matrix [37] [38]. Regarding the constraints, applications of trace and orthornormal constraints can be found in many practical problems such as in machine learning problems [26][27], image processing [28][29], signal processing [22][30][31], modularity detection [32][33] and complex networks [34][36]. Existing schemes translate each of the above applications into some particular models that are manageable, so they fail to deal with the problems in a general way.

The remaining part of the paper is organized as follows. In Section 2, we illustrate how a matrix function optimization problem is formulated. The algorithm of WTPGM and WOPGM are presented in Section 3. In Section 4, how to obtain the gradient of a matrix function is discussed using *I-Chain rule*. The convergence of WTPGM and WOPGM are established in Section 5. In Section 6, three example in different areas involving the matrix function optimizations are illustrated to show the performance of WTPGM and WOPGM. Finally, this paper is concluded in Section 7.

II. Problem Formulation

Let $E(B) \geq 0$ be a general cost function of a matrix $B \in \mathbb{R}^{N\times M}$. Without losing of generality, we assume that $N \geq M$. By denoting a real symmetrical and positive definite (or semi-definite) matrix G as a weight matrix, the matrix function optimization problem is formulated as:

$$\begin{array}{ll} argmin_B & E(B) \\ s.t. & tr(B^TGB) = M \end{array} \tag{1}$$

under weighted trace constraint where $tr(\cdot)$ denotes the matrix trace function, or

$$\begin{array}{ll} argmin_B & E(B) \\ s.t. & B^TGB = I_M \end{array} \tag{2}$$

under weighted orthonormal constraint, where I_M denotes an identity matrix with a dimension M. Here $tr(B^TGB) = M$ and $B^TGB = I_M$ represent different boundary constraints on the energy profile of the matrix variable B but with different physical implications.

Based on Cholesky decomposition, $G = \mathcal{G}^T\mathcal{G}$, where $\mathcal{G}$ is a postive definite upper triangular matrix. Therefore $tr(B^TGB) = M$ is equivalent to $\|\mathcal{G}B\|_F^2 = M$ where $\|\cdot\|_F$ denotes the Frobenius norm, and this implies that the quadratic sum of all elements of matrix product $\mathcal{G}B$ is a fixed value M. $B^TGB = I_M$ means that all column of $\mathcal{G}B$ are orthonormal to each other. By defining feasible regions such that $S_1 = \{B|\ B^TGB = I_M\}$ and $S_2 = \{B|\ tr(B^TGB) = M\}$, it is easy to see that S_1 is a subset of S_2, i.e., $S_1 \subset S_2$.

Thus, in this paper, *trace constraint* $tr(B^TB) = M$ and *orthornormal constraint* $B^TB = I_M$ are relaxed to more general conditions by introducing a weight matrix G, which reflects variable boundary requirements on the energy profiles of the input matrix B. When G is an identity matrix, the optimization models in (1) and (2) in this work reduce to the models in [13].

III. Algorithms

Motivated by [9] [13] [39], we propose two iterative algorithms, namely, *weighted trace-constraint-based projected gradient method* (WTPGM) and *weighted orthonormal-constraint-based projected gradient method* (WOPGM), for solving the optimization problem formulated in (1) and (2), respectively. The detailed steps for each algorithm are summarized as follows.

A. WTPGM for solving the optimization model (1)

Let $\mathbb{O}_1^{N\times M} := \{B \in \mathbb{R}^{N\times M} : trace(B^TGB) = M\}$, and $\tilde{B} \in \mathbb{R}^{N\times M}$ be the orthonormal basis of $GB \in \mathbb{R}^{N\times M}$ where B is an arbitrary matrix. This implies that $\tilde{B}^T\tilde{B} = I_M$. The minimization problem of (1) is converted to minimization of $E(B)$ over $\mathbb{O}_1^{N\times M}$ that can be viewed as an embedded submanifold of the Euclidean space $\mathbb{R}^{N\times M}$. By defining a projection operator $\mathcal{T}_{\tilde{B}} = (I_N - \tilde{B}\tilde{B}^T)$, WTPGM for solving the optimization model (1) is presented as follows.

1) Step 1. Set $k = 0$ and initialize B as a random matrix $B_0 \in \mathbb{R}^{N\times M}$.
2) Step 2 (*Projected gradient descent step*). Calculate the gradient $\frac{\partial E(B)}{\partial B}$ at $B = B_k$ denoted as $\nabla E(B_k)$, update B_k to B_{k+1} as follows

$$\begin{aligned} \hat{B}_{k+1} &= B_k + \triangle\hat{B}_k \\ &= B_k + \eta\triangle B_k \\ &= B_k - \eta\cdot\left(I_N - \tilde{B}_k\tilde{B}_k^T\right)\nabla E(B_k) \\ &= B_k - \eta\cdot\mathcal{T}_{\tilde{B}_k}\nabla E(B_k) \end{aligned} \tag{3}$$

where η is a chosen step length, $\triangle\hat{B}_k = \eta\triangle B_k$, $\triangle B_k = -\mathcal{T}_{\tilde{B}_k}\nabla E(B_k)$ and $\mathcal{T}_{\tilde{B}_k} = \left(I_N - \tilde{B}_k\tilde{B}_k^T\right)$, and $\tilde{B}_k$ is the orthonormal basis of matrix product GB_k. As shown in Lemma 6, $\mathcal{T}_{\tilde{B}_k}$ is a projection operator which projects a matrix $Z \in \mathbb{O}_1^{N\times M}$ onto a space perpendicular to the space spanned by GB_k (denoted as $Span\{\tilde{B}_k\}$). Since $\tilde{B}_k$ is the orthonormal basis of GB_k, GB_k can be represented by $\tilde{B}_k$, i.e., $GB_k \in Span\{\tilde{B}_k\}$. Therefore, in Fig.1, $\triangle\hat{B}_k$ is perpendicular to GB_k.
3) Step 3 (*Normalization step*) Obtain B_{k+1} by normalizing $\hat{B}_{k+1}$ onto the surface $\mathbb{O}_1^{N\times M}$ by

$$B_{k+1} = \sqrt{\frac{M}{tr\left(\hat{B}_{k+1}^TG\hat{B}_{k+1}\right)}}\cdot\hat{B}_{k+1} \tag{4}$$

Denote that $\triangle B_k = B_{k+1} - B_k$ as the quantity of the variety at each iteration step.
4) Step 4. Calculate the angle θ_k between $-\nabla E(B_k)$ and $\triangle B_k$, based on the following Definition:

$$\theta_k = arccos\left(\frac{tr\left([\nabla E(B_k)]^T\mathcal{T}_{\tilde{B}_k}\nabla E(B_k)\right)}{\|\nabla E(B_k)\|_F\cdot\left\|\mathcal{T}_{\tilde{B}_k}\nabla E(B_k)\right\|_F}\right) \tag{5}$$

5) Step 5. If $\left|\theta_k - \frac{\pi}{2}\right| > \xi$, then update $k = k+1$ and go to Step 2; otherwise, stop the iteration and let $B^* = B_k$.

Note that ξ is a positive small constant which is a termination condition of the iteration process. The solution for the matrix B obtained by WTPGM is denoted as B^*. Fig.2 shows the illustration of WTPGM for solving the matrix optimization problem (1).

B. WOPGM for solving the optimization model (2)

Similarly, let $\mathbb{O}_2^{N\times M} := \{B \in \mathbb{R}^{N\times M} : B^T GB = I_M\}$. Viewing $\mathbb{O}_2^{N\times M}$ as an embedded submanifold of the Euclidean space $\mathbb{R}^{N\times M}$. Here we define a projection operator such that $\mathcal{T}_B\mathbb{O}_2^{N\times M} := \{X \in \mathbb{R}^{N\times M} : X = (I_N - GBB^T)Z,\ \forall B \in \mathbb{O}_2^{N\times M},\ Z \in \mathbb{R}^{N\times M}\}$. By Lemma 6, the space $\mathcal{T}_B\mathbb{O}_2^{N\times M}$ at any $Z \in \mathbb{R}^{N\times M}$ is perpendicular to $Span\{B\}$.

Assume that the analysis formula of $\nabla E(B_k)$, which is the gradient of the cost function can be written as $\nabla E(B_k) = F(B_k) \cdot B_k$ where $F(B_k) \in \mathbb{R}^{N\times N}$ and $B_k \in \mathbb{R}^{N\times M}$. Otherwise, we use $\widehat{F(B_k)} = \nabla E(B_k) \cdot B_k^T$ to replace $F(B_k)$, making $\nabla \widehat{E(B_k)} = \widehat{F(B_k)} \cdot G \cdot B_k$. For simplicity, hereafter we use $\nabla \widehat{E(B_k)} = \widehat{F(B_k)} \cdot G \cdot B_k$ no matter whether $\nabla E(B_k)$ can be written as $F(B_k)B_k$ or not.

Similar to WTPGM, we first randomly choose B_0 from $\mathbb{R}^{N\times M}$. Then, in the *projected gradient descent step*, calculate $\nabla E(B_k)$, and we denote that

$$\begin{aligned} \widehat{F(B_k)} &= \nabla E(B_k) \cdot B_k^T \\ \nabla \widehat{E(B_k)} &= \widehat{F(B_k)} \cdot G \cdot B_k \\ \widetilde{F(B_k)} &= \widehat{F(B_k)} + \lambda_k I \\ \nabla \widetilde{E(B_k)} &= \widetilde{F(B_k)} \cdot G \cdot B_k \end{aligned} \tag{6}$$

Let $a = trace(B_k^T \mathcal{T}_{B_k} G \mathcal{T}_{B_k} \widehat{F(B_k)} G B_k)$ and $b = trace(B_k^T \mathcal{T}_{B_k} G \mathcal{T}_{B_k} G B_k)$. The value of λ_k is chosen according to the following Tab.I. Then, we do the iteration with

TABLE I
THE VALUE CONDITION OF λ_k

$b > 0$	$b < 0$
$\lambda_k < -\frac{a}{b}$	$\lambda_k > -\frac{a}{b}$

$$\begin{aligned} \hat{B}_{k+1} &= B_k + \Delta\hat{B}_k \\ &= B_k + \eta \cdot \Delta B_k \\ &= B_k - \eta \cdot \left(I_N - GB_kB_k^T\right) \cdot \nabla \widetilde{E(B_k)} \\ &= B_k - \eta \cdot \mathcal{T}_{B_k} \cdot \nabla \widetilde{E(B_k)} \end{aligned} \tag{7}$$

where $\Delta\hat{B}_k = \eta \cdot \Delta B_k$ and $\Delta B_k = -\mathcal{T}_{B_k} \nabla \widetilde{E(B_k)}$ with $\mathcal{T}_{B_k} = I_N - GB_kB_k^T$. In the *Normalization step*, update B_{k+1} by

$$B_{k+1} = \sqrt{\frac{tr\left(\hat{B}_{k+1}^T G \hat{B}_{k+1}\right)}{tr\left(\hat{B}_{k+1}^T G \hat{B}_{k+1} \hat{B}_{k+1}^T G \hat{B}_{k+1}\right)}} \cdot \hat{B}_{k+1} \tag{8}$$

Fig.3 illustrates the iteration process of WOPGM for solving the optimization model (2). In implementations, the termination criteria is revised from (5) via replacing $\nabla E(B_k)$ by $\nabla \widetilde{E(B_k)}$:

$$\theta_k = arccos\left(\frac{tr\left(\left[\nabla \widetilde{E(B_k)}\right]^T \mathcal{T}_{B_k} \nabla \widetilde{E(B_k)}\right)}{\left\|\nabla \widetilde{E(B_k)}\right\|_F \cdot \left\|\mathcal{T}_{B_k} \nabla \widetilde{E(B_k)}\right\|_F}\right) \tag{9}$$

Remark 2. *The idea of updating B_{k+1} based on $\hat{B}_{k+1}$ in (8) is explained as follows. We first define a norm function as*

$$N(B) = tr\left(\left(B^TGB - I_M\right)^T \left(B^TGB - I_M\right)\right) \tag{10}$$

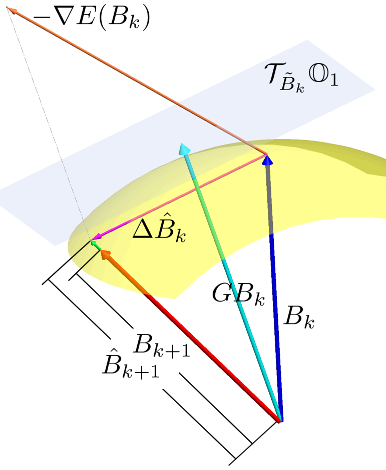


Fig. 1. **The projection and normalization operators in WTPGM.** The initial B_0 can be randomly assigned, and at the $k+1$th iteration, we calculate $\hat{B}_{k+1}$ first and then obtain B_{k+1}. The notations in this figure is shown in the algorithm and detailed description can be found in Theorem 1.

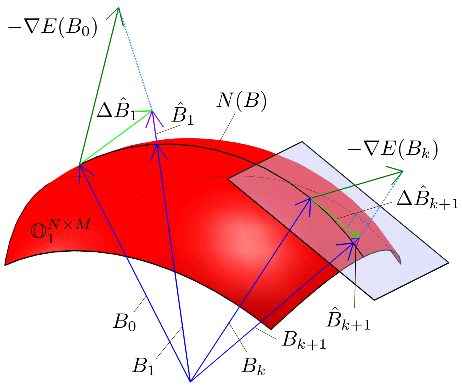


Fig. 2. **Iteration process of WTPGM for solving optimization problem (1).**

Note that $\forall B \in R^{N\times M}$, $B^TGB = I_M$ is equivalent to $N(B) = 0$. By setting $B_{k+1} = \rho_k \cdot \hat{B}_{k+1}$, it is seen that there is no exact solution ρ_k satisfying that $N(B_{k+1}) = 0$ since it is generally impossible to drag a tensor of B on the boundary constraints by just automatic scaling. In the iteration, we minimize $N(B_{k+1})$ by solving $\frac{\partial N(\rho_k \cdot \hat{B}_{k+1})}{\partial \rho_k} = 0$, which gives

$$\begin{aligned} N(\rho_k) = \rho_k^4 tr(\hat{B}_{k+1}^T G \hat{B}_{k+1} \hat{B}_{k+1}^T G \hat{B}_{k+1}) \\ - 2\rho_k^2 tr(\hat{B}_{k+1}^T G \hat{B}_{k+1}) + M \end{aligned} \tag{11}$$

Thus, it is obtained that $\rho_k = \sqrt{\frac{tr\left(\hat{B}_{k+1}^T G \hat{B}_{k+1}\right)}{tr\left(\hat{B}_{k+1}^T G \hat{B}_{k+1} \hat{B}_{k+1}^T G \hat{B}_{k+1}\right)}}$. □

IV. *I-Chain Rule* FOR MATRIX FUNCTION DIFFERENTIATION

As mentioned, obtaining $\nabla E(B_k)$ is of high importance for implementing both WTPGM and WOPGM. To achieve this, we investigate matrix differentiation in this section. Since generally a complicated matrix function involves combinations of some basic operators such as matrix product, matrix trace,

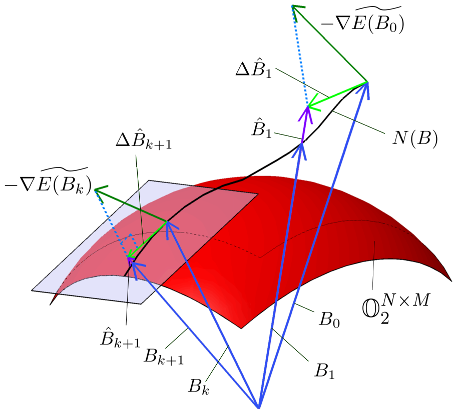


Fig. 3. **Iteration process of WOPGM for solving optimization problem (2).** The detailed information of the notations are shown in Theorem 2.

matrix determinant and matrix inverse, we show how to calculate $\nabla E(B_k)$ on these basic operators and some lemmas will be then introduced. Based on these lemmas, we claim that, the derivative of a unintuitive matrix function can be calculated by applying the *I-Chain Rule*.

A. Matrix product

Lemma 1. *[40] For a matrix X with no special structure (i.e., elements of X are independent), denote $[X]_{ij}$ as the ij-th element of matrix X and $\delta_{ij} = 1$ iff $i = j$ (otherwise, $\delta_{ij} = 0$), we have:*

1) $\frac{\partial [X]_{kl}}{\partial [X]_{ij}} = \delta_{ki} \cdot \delta_{lj}$
2) $\frac{\partial [X^T]_{kl}}{\partial [X]_{ij}} = \delta_{kj} \cdot \delta_{li}$
3) $[X]_{kl} = [X]^T_{lk}$
4) $\delta_{ki} \cdot \delta_{ij} = \delta_{kj}$
5) *For compatible matrices A, B, X such that $X = AB$, then $[X]_{ij} = \sum_k [A]_{ik}[B]_{kj}$ and it can be written as $[X]_{ij} = [A]_{ik} \cdot [B]_{kj}$) by introducing $[\cdot]$ operator and omitting the summation notation. 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 $[X]_{ij} = A_{ik} \cdot B_{kp} \cdot C_{pj}$.*

The method introduced in Lemma 1 is regarded as the "index notation arrangement" method [19]. Based on this method, it is convenient to simplify the notations in many applications. For example, we have $\sum_k A_{ik}\delta_{kj} = A_{ik} \cdot \delta_{kj} = A_{ij}$ for a $N \times M$ matrix $A \in \mathbb{R}^{N\times N}$.

B. Matrix trace differentiation operator

For a matrix $X \in \mathbb{R}^{N\times N}$, we have $\frac{\partial tr(X)}{\partial X} = I$ This can be easily derived from Lemma 1. As we have $\frac{\partial tr(X)}{\partial X_{kl}} = \frac{\partial \sum_i X_{ii}}{\partial X_{kl}} = \sum_i \delta_{ik}\delta_{il} = \delta_{kl}$, and $\delta_{kl} = 1$ if and only if $k = l$, otherwise $\delta_{kl} = 0$. Thus, we obtain that $\frac{\partial tr(X)}{\partial X} = I$. Also, high order of matrix trace differentiation can be derived. For example, as $\frac{\partial tr(X^2)}{\partial X_{kl}} = \frac{\partial \sum_i \sum_p X_{ip} X_{pi}}{\partial X_{kl}} = \sum_i \delta_{ki} \cdot \delta_{pl} \cdot X_{pi} + \sum_i X_{ip} \cdot \delta_{pk} \cdot \delta_{li} = 2X_{lk}$, we obtain that $\frac{\partial tr(X^2)}{\partial X} = 2X^T$.

C. Matrix inverse differentiation operator

Lemma 2. *[19] For an invertible matrix $Y \in \mathbb{R}^{N\times N}$ and a scalar x, we have*

$$\frac{\partial Y^{-1}}{\partial x} = -Y^{-1}\frac{\partial Y}{\partial x}Y^{-1} \tag{12}$$

Lemma 3. *[40] For an invertible matrix $X \in \mathbb{R}^{N\times N}$, we have*

$$\frac{\partial [X^{-1}]_{kl}}{\partial [X]_{ij}} = -[X^{-1}]_{ki}[X^{-1}]_{jl} \tag{13}$$

D. Matrix determinant differentiation operator

Lemma 4. *[40] For an invertible matrix $X \in \mathbb{R}^{N\times N}$, we have*

$$\frac{\partial det(X)}{\partial X_{ij}} = det(X)[X^{-T}]_{ij} \tag{14}$$

where $det(\cdot)$ denotes the matrix determinant operator.

E. Index-notation-arrangement based chain rule (I-Chain rule) for matrix function derivatives

As discussed in Introduction section, there is no unified framework for matrix function differentiation in the existing literatures. In this subsection, we propose index-notation-arrangement based chain rule to do the derivative of general matrix function by exploring Lemmas 1-4. Simulation results in Section 6 will show the effectiveness of our methods.

I-Chain rule: Suppose that matrix $U \in R^{N\times M}$ is a function of matrix B, i.e., $U = g(B)$, the derivative of the function $E(U) = E(g(B))$ with respect to B is given by the chain rule as follows:

$$\begin{aligned} \frac{\partial E(g(B))}{\partial B} &= \frac{\partial E(g(B))}{\partial B_{ij}} \\ &= \sum\nolimits_{m,n} \frac{\partial E(U)}{\partial U_{mn}} \frac{\partial U_{mn}}{\partial B_{ij}} \\ &= \frac{\partial E(U)}{\partial U_{mn}} \cdot \frac{\partial U_{mn}}{\partial B_{ij}} \end{aligned} \tag{15}$$

where B_{ij} is the $ij-$th element of matrix B, and the indexes $\sum_{m,n}$ ($1 \le m \le M$, $1 \le n \le N$) is omitted for the convenience of representation (Index-notation-arrangement). □

The above *I-Chain rule* has meaningful implications. Basically it says that when we do not know how to find the derivative of an expression using matrix calculus directly, we can always fall back on index notation and convert back to matrices in the end. This is the main idea of derivation steps and it reduces a potentially unintuitive matrix-valued problem into one involving scalars which we are used to. In addition, it is less painful to massage an expression into a familiar form and apply previously-derived identities. Fig.4 illustrates how to do the derivative of $E(U)$, in which U is function of B denoted as $U = g(B)$, with respect to B using the chain rule. The derivation steps are summarized as follows.

1) For a matrix function $E(B) : \mathbb{R}^{N\times M} \to \mathbb{R}$ where $B \in \mathbb{R}^{N\times M}$, the derivative of $E(B)$ with respect to B has the same dimension with matrix B. In order to take the derivative, we only need to take the derivative

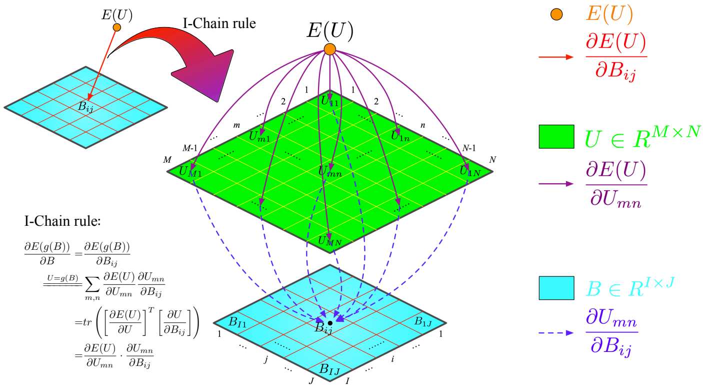

$E(U)$
I-Chain rule
B_{ij}
$E(U)$
U_{11}
U_{m1}
U_{1n}
U_{M1}
U_{mn}
U_{1N}
U_{MN}
1
2
m
M-1
M
1
2
n
N-1
N
I-Chain rule:
$\frac{\partial E(g(B))}{\partial B} = \frac{\partial E(g(B))}{\partial B_{ij}}$
$\overset{U=g(B)}{=\!=\!=} \sum_{m,n} \frac{\partial E(U)}{\partial U_{mn}} \frac{\partial U_{mn}}{\partial B_{ij}}$
$= tr\left(\left[\frac{\partial E(U)}{\partial U}\right]^T \left[\frac{\partial U}{\partial B_{ij}}\right]\right)$
$= \frac{\partial E(U)}{\partial U_{mn}} \cdot \frac{\partial U_{mn}}{\partial B_{ij}}$
B_{I1}
B_{ij}
B_{1J}
B_{IJ}
1
j
J
1
i
I
$E(U)$
$\frac{\partial E(U)}{\partial B_{ij}}$
$U \in R^{M\times N}$
$\frac{\partial E(U)}{\partial U_{mn}}$
$B \in R^{I\times J}$
$\frac{\partial U_{mn}}{\partial B_{ij}}$

Fig. 4. *I-chain rule* **for the matrix function differentiation.**

of $E(B)$ with respect to B_{ij}, denoted as $\frac{\partial E(B)}{\partial B_{ij}}$, which is the $ij-$th element of $\frac{\partial E(B)}{\partial B}$.

2) For the case that the matrix function contains basic operators such as matrix trace, matrix inverse, matrix determinate and so on, normally new variables will be introduced and chain rule can be applied to represent the derivation as a form such that the terms in the derivation chain can be interpreted as taking a sequence of derivatives of basic operators.

3) Sum all the derivatives of newly defined variables with respect to B_{ij} using the chain rule. To this end, switching the places of entities such that the derivation can be written as $[\bullet]_{ij}$ is necessary. Based on the obtained $\frac{\partial E(B)}{\partial B_{ij}}$ for all i and j and convert all the terms back to a matrix, finally we could obtain $\frac{\partial E(B)}{\partial B}$.

Now we take e^{At} for example to illustrate how to obtain the first order derivative of e^{At} with respect to A using *I-Chain rule*. Assume that matrix A has k eigenvalues $\lambda_1, \lambda_2, ..., \lambda_k$ where λ_1 is a $m-$th order repeated eigenvalue and $\lambda_2, \lambda_3, ..., \lambda_k$ $(k = N - m - 1)$ are the remaining $N - m$ eigenvalues which are assumed distinct. By Cayley-Hamilton theorem [41][42], it is obtained that

$$e^{At} = \alpha_0(t)I + \alpha_1(t)A + \alpha_2(t)A^2 + ... + \alpha_{N-1}(t)A^{N-1}$$

where $\alpha_{n_0}(t) = \alpha_{n_0}t^{n_0}$ for $i = 0, ..., N-1$ are obtained by solving matrix equations.

Note that the derivative of e^{At} with respective to A is a fourth tensor, to obtain the derivative of e^{At} with respect to A, we need to obtain the derivative of $[e^{At}]_{kl}$ with respect to A_{ij}.

By using the *I-Chain rule*, we have

$$\begin{aligned}\frac{\partial[A^{n_0}]_{kl}}{\partial A_{ij}} &= \sum_{k_0=0}^{n_0-1} \frac{\partial\left([A^{k_0}]_{kp}A_{pz}[A^{n_0-1-k_0}]_{zl}\right)]_{kl}}{\partial A_{ij}} \\ &= \sum_{k_0=0}^{n_0-1}\left([A^{k_0}]_{kp}\delta_{pi}\delta_{zj}[A^{n_0-1-k_0}]_{zl}\right) \qquad (16) \\ &= \sum_{k_0=0}^{n_0-1}\left([A^{k_0}]_{ki}[A^{n_0-1-k_0}]_{jl}\right)\end{aligned}$$

By Lemma 2, we obtain

$$\frac{\partial[e^{At}]_{kl}}{\partial A_{ij}} = \sum_{n_0=1}^{N-1}\alpha_{n_0}(t)\left(\sum_{k_0=0}^{n_0-1}\left([A^{k_0}]_{ki}[A^{n_0-1-k_0}]_{jl}\right)\right) \quad (17)$$

V. Convergence Proof

Here we prove the convergence of WTPGM and WOPGM in solving the optimization problems given in (1) and (2), respectively. Before that, we first introduce following lemmas.

Lemma 5. *For symmetric matrices X, $Y \in \mathbb{R}^{N\times N}$ and random matrices $U \in \mathbb{R}^{M\times N}$, $Z \in \mathbb{R}^{N\times M}$,*

$$tr(UXYZ) = tr(UYXZ),$$

as long as ZU is symmetric [40].

Lemma 6. *Define a Stiefel manifold $S := \{B \in \mathbb{R}^{N\times M} : B^TB = I_M\}$ and a space spanned by B as $Span\{B\} := \{X : X = B * Y, B \in S, Y \in \mathbb{R}^{M\times M}\}$ Then,*

$$\mathcal{T}_B = (I_N - BB^T) \quad (18)$$

is a projection operator which projects $Z \in \mathbb{R}^{N\times M}$ onto a space perpendicular to the space $Span\{B\}$.

Proof: Note that $\mathcal{T}_B^T\mathcal{T}_B = \mathcal{T}_B$. So $\mathcal{T}_B$ is an projection operator. By definition, for a matrix $X \in Span\{B\}$, there exist a $Y \in \mathbb{R}^{M\times M}\}$ such that $X = BY$. Then,

$$tr(X^T\mathcal{T}_BZ) = tr(Y^T(B^T - B^TBB^T)Z) = 0 \quad (19)$$

which implies that the angle between X and $\mathcal{T}_BZ$ is 90^0. Thus, $\mathcal{T}_B$ is a projection operator which projects an arbitrary $Z \in \mathbb{R}^{N\times M}$ onto the space perpendicular to $Span\{B\}$. ■

Remark 3. *In mathematics, Stiefel manifold [43], named after the Swiss mathematician Eduard Stiefel, is a set of orthonormal k-frames [44][45] in $\mathbb{R}^N$. A k-frames is an ordered set of k linearly independent vectors in a space with $k \le N$ being the dimension of the vector space. This implies the orthonormal constraint on the matrix variables, i.e., they are located on Stiefel manifold which is a submanifold of $\mathbb{R}^{N\times k}$. Such a constraint has its physical meaning in many practical applications [46] [47].*

Theorem 1. *The iteration process of WTPGM given in (3)-(4) in solving the optimization problem in (1) is convergent.*

Proof: Fig.1 shows $\mathbb{O}_1^{N\times M}$ and $\mathcal{T}_B\mathbb{O}_1^{N\times M}$, represented with a red surface and a translucent plane, respectively. At the start of every step, we have B_k obtained from previous step, which is represented with a black arrow in Fig. 1. Since $E(B_k) \ge 0$, we have to show that

$$E(B_{k+1}) - E(B_k) \le 0 \quad (20)$$

to establish the convergence property of WTPGM. To simplify representation, we use operator $\mathcal{T}_{\tilde{B}_k}$ and $\nabla E(B_k)$ to represent $(I_N - \tilde{B}_k\tilde{B}_k^T)$ and $\frac{\partial E(B)}{\partial B}$ at $B = B_k$, respectively.

Now we consider $tr\left(B_k^TG\mathcal{T}_{\tilde{B}_k}\nabla E(B_k)\right)$. Because $\tilde{B}_k$ is the orthonormal basis of GB_k, GB_k can be written as $\tilde{B}_k \cdot Y$, where Y is a $N \times N$ diagonal matrix. Hence, we have

$$\begin{aligned} & tr\left(B_k^T G \mathcal{T}_{\tilde{B}_k} \nabla E(B_k)\right) \\ & = tr\left((GB_k)^T \mathcal{T}_{\tilde{B}_k} \nabla E(B_k)\right) \\ & = tr\left((\tilde{B}_k \cdot Y)^T \cdot \mathcal{T}_{\tilde{B}_k} \nabla E(B_k)\right) \\ & = tr\left(Y^T(\tilde{B}_k^T - \tilde{B}_k^T \tilde{B}_k \tilde{B}_k^T) \nabla E(B_k)\right) \\ & = tr\left(Y^T(\tilde{B}_k^T - \tilde{B}_k^T) \nabla E(B_k)\right) \\ & = 0. \end{aligned} \tag{21}$$

based on Lemma 6. Note that the operator $\mathcal{T}_{\tilde{B}_k}$ projects a matrix $B_k \in \mathbb{O}^{N \times M}$ onto the tangent space $\mathcal{T}_{\tilde{B}} \mathbb{O}^{N \times M}$, so we have $tr(B_k^T G B_k) = M$ and $\mathcal{T}_{\tilde{B}_k}^T \mathcal{T}_{\tilde{B}_k} = \mathcal{T}_{\tilde{B}_k}$.

Now by substituting $\hat{B}_{k+1} = B_k - \eta \mathcal{T}_{B_k} \nabla E(B_k)$, we have

$$\begin{aligned} B_{k+1} & = \sqrt{\frac{M}{tr\left(\hat{B}_{k+1}^T G \hat{B}_{k+1}\right)}} \cdot \hat{B}_{k+1} \\ & = \sqrt{\frac{M}{tr\left(B_k^T G B_k\right) - 2\eta tr\left(B_k^T G \mathcal{T}_{B_k} \nabla E(B_k)\right) + o(\eta)}} \\ & \quad \left(B_k - \eta \cdot \mathcal{T}_{\tilde{B}_k} \cdot \nabla E(B_k)\right) \\ & = B_k - \eta \mathcal{T}_{\tilde{B}_k} \nabla E(B_k) + o(\eta) \end{aligned} \tag{22}$$

Through Taylor expansion and using (22), we have

$$\begin{aligned} & E(B_{k+1}) - E(B_k) \\ & = E(B_k) + tr\left([\nabla E(B_k)]^T \cdot (B_{k+1} - B_k)\right) \\ & \quad + o(B_{k+1} - B_k) - E(B_k) \\ & = tr\left([\nabla E(B_k)]^T \cdot (-\eta \mathcal{T}_{\tilde{B}_k} \nabla E(B_k))\right) + o(\eta) \\ & = -\eta tr\left([\nabla E(B_k)]^T \mathcal{T}_{\tilde{B}_k} \nabla E(B_k)\right) + o(\eta) \end{aligned} \tag{23}$$

Again, as $\mathcal{T}_{\tilde{B}_k}$ is an projection operator, we have $\mathcal{T}_{\tilde{B}_k}^T \mathcal{T}_{\tilde{B}_k} = \mathcal{T}_{\tilde{B}_k}$. Then,

$$[\nabla E(B_k)]^T \mathcal{T}_{\tilde{B}_k} \nabla E(B_k) = \left(\mathcal{T}_{\tilde{B}_k} E(B_k)\right)^T \mathcal{T}_{\tilde{B}_k} E(B_k). \tag{24}$$

Hence, $[\nabla E(B_k)]^T \mathcal{T}_{\tilde{B}_k} \nabla E(B_k)$ is a positive semi definite matrix, which means every eigenvalue λ_i of $[\nabla E(B_k)]^T \mathcal{T}_{\tilde{B}_k} \nabla E(B_k)$ is nonnegative, which gives

$$tr\left([\nabla E(B_k)]^T \mathcal{T}_{\tilde{B}_k} \nabla E(B_k)\right) = \textstyle\sum_i \lambda_i \geq 0. \tag{25}$$

Thus, we have $E(B_{k+1}) \leq E(B_k)$ and we can draw the conclusion that the iteration is convergent, as long as η is sufficiently small. The iteration stops when the optimality condition $tr\left([\nabla E(B_k)]^T \mathcal{T}_{\tilde{B}_k} \nabla E(B_k)\right) = 0$ (weighted trace constraint) is satisfied. ■

Theorem 2. *Suppose λ_k is selected as illustrated in Table 1. For a randomly choosing $B_0 \in R^{N \times M}$ and a sufficiently small η, the proposed WOPGM in (6)-(8) ensures that $E(B_k)$ converges to $E(B^*)$ where B^* is a matrix such that the weighted orthornormal boundary constraint condition is satisfied, i.e., ${B^*}^T G B^* = I_M$.*

Proof: For the representation of simplicity, we use $\mathcal{T}_{B_k}$ and $\nabla N(B_k)$ to replace $(I_N - GB_kB_k^T)$ and $\frac{\partial N(B_k)}{\partial B_k}$ respectively. Our gradient descent method is to minimize $E(B)$ and $N(B)$ simultaneously. Since $E(B_k) \geq 0$, we next need to to prove two parts as follows:

$$E(B_{k+1}) - E(B_k) \leq 0, \tag{26}$$

$$N(B_{k+1}) - N(B_k) \leq 0,\ and\ N(B_k) \geq 0 \tag{27}$$

It is apparent that $N(B_k) \geq 0$.

Now we consider $N(\hat{B}_{k+1}) - N(B_k)$. Firstly, denote that

$$\nabla N(B_k) = 4G(B_kB_k^TG - I_N)B_k. \tag{28}$$

As $B_kB_k^T$ is symmetric, from Lemmas 5-6, and by using Taylor expansion to expand $N(B)$[48], we have

$$\begin{aligned} & N(\hat{B}_{k+1}) - N(B_k) \\ = & N(B_k + \eta \Delta B_k) - N(B_k) \\ = & N(B_k) + \eta tr([\nabla N(B_k)]^T \cdot \Delta B_k) - N(B_k) \\ = & -\eta tr\left(\left(4G(B_kB_k^TG - I_N)B_k\right)^T \mathcal{T}_{B_k} \widetilde{F(B_k)} G B_k\right) \\ = & 4\eta \cdot tr\left(B_k^T \mathcal{T}_{B_k} G \mathcal{T}_{B_k} \cdot \widetilde{F(B_k)} G B_k\right) \\ = & 4\eta \cdot tr\left(B_k^T \mathcal{T}_{B_k} G \mathcal{T}_{B_k} \cdot (\widehat{F(B_k)} + \lambda_k) G B_k\right) \\ = & 4\eta \cdot tr\left(B_k^T \mathcal{T}_{B_k} G \mathcal{T}_{B_k} \cdot \widehat{F(B_k)} G B_k\right) \\ & + 4\eta\lambda_k \cdot tr\left(B_k^T \mathcal{T}_{B_k} G \mathcal{T}_{B_k} \cdot G B_k\right) \\ = & 4\eta(a + b\lambda_k). \end{aligned} \tag{29}$$

From the definition of λ_k in Tab.I, we have

$$4\eta \cdot (a + b\lambda_k) < 0. \tag{30}$$

which gives that

$$N(\hat{B}_{k+1}) - N(B_k) = 4\eta \cdot (a + b\lambda_k) < 0. \tag{31}$$

Now we consider $N(B_{k+1}) - N(\hat{B}_{k+1})$. From 10 and 8, we have

$$N(B_{k+1}) = M - \frac{tr^2\left(\hat{B}_{k+1}^T G \hat{B}_{k+1}\right)}{tr\left(\hat{B}_{k+1}^T G \hat{B}_{k+1} \hat{B}_{k+1}^T G \hat{B}_{k+1}\right)}. \tag{32}$$

So we have

$$\begin{aligned} & N(B_{k+1}) - N(\hat{B}_{k+1}) \\ & = -\frac{\left(tr(\hat{B}_{k+1}^T G \hat{B}_{k+1} \hat{B}_{k+1}^T G \hat{B}_{k+1}) - tr(\hat{B}_{k+1}^T G \hat{B}_{k+1})\right)^2}{tr(\hat{B}_{k+1}^T G \hat{B}_{k+1} \hat{B}_{k+1}^T G \hat{B}_{k+1})} \\ & \leq 0. \end{aligned} \tag{33}$$

From (31) and (33), we obtain

$$\begin{aligned} & N(B_{k+1}) - N(B_k) \\ = & N(B_{k+1}) - N(\hat{B}_{k+1}) + N(\hat{B}_{k+1}) - N(B_k) \leq 0, \end{aligned} \tag{34}$$

and the iteration ends when $\nabla N(B_k) = 0$, i.e., $B_k^T G B_k = I_M$. This implies that $\forall \epsilon > 0$, $\exists K \in \mathbb{N}$, $\forall k > K$, $|N(B_k)| < \epsilon$, which can also be written as $N(B_k) = o(\eta)$, or

$$B_k^T G B_k = I_M + o(\eta) \tag{35}$$

To prove $E(B_{k+1}) - E(B_k) \leq 0$ when $N(B_k) = o(\eta)$, we aim to use B_k to express B_{k+1} directly. Firstly, we have

$$\begin{aligned} & tr(B_{k+1}^T G B_{k+1} B_{k+1}^T G B_{k+1}) \\ = & \rho_k^4 tr(\hat{B}_{k+1}^T G \hat{B}_{k+1} \hat{B}_{k+1}^T G \hat{B}_{k+1}) \end{aligned}$$

$$
\begin{aligned}
&=\rho_k^2 tr(\hat{B}_{k+1}^T G\hat{B}_{k+1})\\
&=tr(B_{k+1}^T GB_{k+1})
\end{aligned} \tag{36}
$$

Hence, it is straightforward to prove that $\rho_k = 1 + o(\eta)$, and we have

$$B_{k+1} = B_k + \eta \cdot \Delta B_k + o(\eta). \tag{37}$$

Secondly, by applying Taylor expansion to expand $E(B)$[48] by substituting $\widetilde{\nabla E(B_k)} = \widetilde{F(B_k)}GB_k$, we obtain

$$
\begin{aligned}
&E(B_{k+1}) - E(B_k)\\
&= E(B_k) + \eta tr\left(\nabla E^T(B_k)\Delta B_k\right) - E(B_k) + o(\eta)\\
&= -\eta tr(\nabla E^T(B_k) \cdot \mathcal{T}_{B_k}\widetilde{F(B_k)}GB_k) + o(\eta)\\
&= -\eta tr\left(\nabla E^T(B_k) \cdot \mathcal{T}_{B_k}\left(\nabla E(B_k)B_k^T + \lambda_k I\right) GB_k\right) + o(\eta)\\
&= -\eta tr\left(\nabla E^T(B_k)\mathcal{T}_{B_k}\nabla E(B_k)\right)\\
&\quad - \eta\lambda_k tr(\nabla E^T(B_k)\mathcal{T}_{B_k}GB_k) + o(\eta)
\end{aligned} \tag{38}
$$

Note that

$$
\begin{aligned}
&tr(\nabla E^T(B_k)\mathcal{T}_{B_k}GB_k)\\
&= tr(\nabla E^T(B_k)(I_N - GB_kB_k^T)GB_k)\\
&= tr(\nabla E^T(B_k)(GB_k - GB_kB_k^TGB_k))\\
&= tr(\nabla E^T(B_k)(GB_k - GB_k(I_M + o(\eta))))\\
&= o(\eta).
\end{aligned} \tag{39}
$$

when k is sufficiently large. Then,

$$
\begin{aligned}
&E(B_{k+1}) - E(B_k)\\
&= -\eta tr\left(\nabla E^T(B_k)\mathcal{T}_{B_k}\nabla E(B_k)\right) + o(\eta)
\end{aligned} \tag{40}
$$

Since

$$
\begin{aligned}
&tr\left(\nabla E(B_k)^T\mathcal{T}_{B_k}\nabla E(B_k)\right) + o(\eta)\\
=&tr\left(\left(\nabla E(B_k)\mathcal{T}_{B_k}\right)^T\left(\mathcal{T}_{B_k}\nabla E(B_k)\right)\right) + o(\eta)\\
\geq&o(\eta)
\end{aligned} \tag{41}
$$

we have $\exists K \in \mathbb{N},\ \forall k > K,\ E(B_{k+1}) \leq E(B_k)$ and the convergence of $E(B_k)$ is now proven for a sufficiently small η. By combining the results shown in (35), $E(B_k)$ converges to $E(B^*)$ with $B^{*T}GB^* = I_M$. ■

VI. Case studies

In order to show that the proposed WTPGM and WOPGM are applicable to various practical problems, in this section, four case studies including one numerical example and two applications in the field of network control as well as one example in dimension reduction are illustrated. Simulation results show that our method can be not only applied in network control but also in other learning problems.

A. An example of matrix trace function optimization

In this subsection, we consider a general model of minimizing a matrix trace function given by

$$
\begin{aligned}
&argmin_B \qquad E(B) = tr(UB^TWBV + B^TP + Q)\\
&s.t. \qquad\quad tr(B^TGB) = M \ or \ B^TGB = I
\end{aligned}
$$

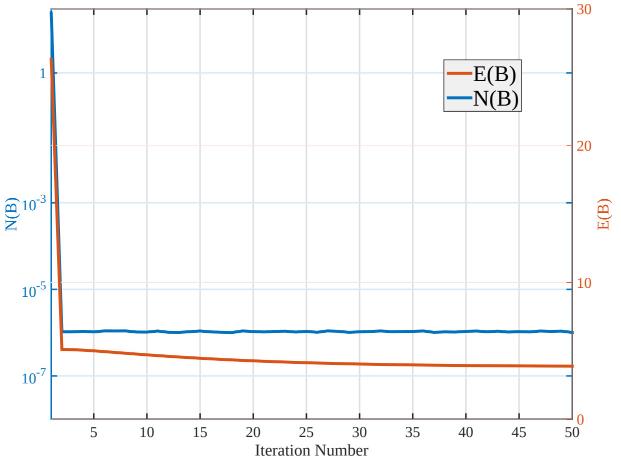


Fig. 5. **WTPGM under constraint $tr(B^TGB) = M$.**

where $B \in \mathbb{R}^{n\times m}$ and U, V and $Q \in \mathbb{R}^{m\times m}$ with $W \in \mathbb{R}^{n\times n}$, $P \in \mathbb{R}^{n\times m}$, and G is a real symmetry positive definite matrix. By *I-Chain* rule, we obtain the gradient of $E(B)$ given by

$$\nabla E(B) = WBVU + W^TBU^TV^T + P$$

Note that WTPGM is applicable when the weighted trace constraint is considered. In order to address the weighted orthonormal constraint, as we observe that $\nabla E(B)$ cannot be represented by $\nabla E(B) = F(B)B$, we reconstruct $\widetilde{\nabla E(B)}$ and $\widetilde{F(B)}$ based on (6). Thus, the above matrix trace function optimization problem can be converted to model (2) and it can be solved by applying WOPGM.

For illustration, we set U, W, V, P, Q and G randomly as $N = 5, M = 3$. The convergence of both $N(B)$ and $E(B)$ under different constraints are shown in Fig.5 and Fig.6. It is seen that $N(B)$ rapidly converges to a small value below 10^{-5} which can be regarded as zero. After $N(B)$ converges to zero, $E(B)$ decreases monotonically and converges to an optimum value finally.

Note that the step length η in the WTPGM and WOPGM algorithms should be set appropriately to guarantee the convergence. It can be seen in the proofs of Theorems 1-2 that, when approximating the equations using Taylor series expansion under the condition that η is sufficiently small, some terms which are mainly matrices or the trace of matrices are ignored. Although theoretically η should be sufficiently small to ensure the convergence, it can be empirically selected when doing experiments. We find that if we select η smaller than the reciprocal of the largest element in matrix B for more than three orders of magnitude, the convergence can be always guaranteed.

For WOPGM, it is known that the matrix $(\mathcal{T}_{B_k}\nabla E(B_k))^T(\mathcal{T}_{B_k}\nabla E(B_k))$ is non-negative in (41), all its eigenvalues can be denoted as $\lambda^1 \geq 0, ..., \lambda^M \geq 0$. Let $\lambda^* = \sum_i \lambda^i$. Then we have $E(B_k) - E(B_{k+1}) = \eta\lambda^*$.

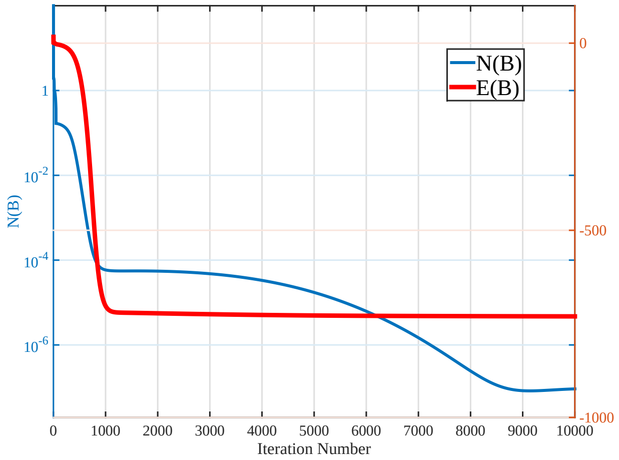


Fig. 6. **WOPGM under constraint** $B^TGB = I_M$**.**

$E(B_k) - E(B^*) = E(B_{k+1}) - E(B^*) + \eta\lambda^*$ which gives

$$\frac{E(B_{k+1}) - E(B^*)}{E(B_k) - E(B^*)} = 1 - \frac{\eta\lambda^*}{E(B_k) - E(B^*)} = \nu_k \qquad (42)$$

Therefore, we have $\nu_k < 1$ until the algorithm converges, and $lim_{k\to+\infty}\nu_k = 1$. For WTPGM, similar conclusion holds. Therefore algorithms WTPGM and WOPGM converge linearly to $E(B^*)$.

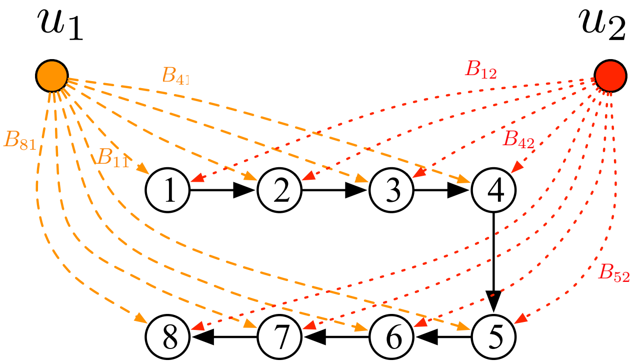
B. Applications in minimum cost control of complex networks with selectable input matrices

The *minimum cost control* of complex networks [36] subjected to linear dynamics $\dot{x}(t) = Ax(t)+Bu(t)$ has been a hot topic recently with wide applications. The $N \times N$ matrix A is the given network's adjacency matrix which describes the connection and interaction strength between the network nodes. The $N \times M$ matrix B is the selectable input matrix where B_{im} is nonzero if the $m-$th external control source is connected to node i and zero otherwise. $x(t) = [x_1(t), ..., x_N(t)]^T$ is the state vector of N nodes at time t with the initial state being x_0, and $u(t) = [u_1(t), ..., u_M(t)]^T$ is the time-dependent external control input vector with M ($M \leq N$) being the number of inputs, where the same input $u_i(t)$ may drive multiple nodes. The objective is to design the input matrix B and $u(t)$ such that the system states can be driven to the origin at $t = t_f$, i.e., the final state $x_f = [0, ..., 0]^T$, subject to the condition that the average cost $\mathbb{E}\left[\int_0^{t_f} \|u(t)\|_2^2 dt\right]$ is minimized. Here $\mathbb{E}[.]$ takes the expectation of the argument over all realizations of the random initial state.

By assuming that each element of the initial state $x_0 = [x_{01}, ..., x_{0N}]^T$ is an identical independent distributed (i.i.d) variable with zero mean and variance 1, we have $\mathbb{E}\left[x_0 x_0^T\right] = X_0 = I_N$ and $X_f = \mathbb{E}[x_f x_f^T] = e^{At_f} X_0 e^{A^T t_f}$. Different from the work in [13], we consider a positive definite diagonal weight matrix

$$G = diag\{g_1, ...g_i, ..., g_M\}$$

subject to $\sum_i g_i = M$, which reflects the restriction on each external control source. Therefore, the optimization problem in [13] can be converted to find an optimal input matrix B^* that minimizes the cost function $E(B)$ defined by $E(B) \triangleq \text{tr}[W_B^{-1} X_f]$ where W_B is the Gramian matrix such that $W_B = \int_0^{t_f} e^{At} BB^T e^{A^T t} dt$ [36] under a weighted trace constraint:



$$\dot{x} = Ax + Bu \qquad N = 8 \qquad M = 2 \qquad G = \begin{bmatrix} 1.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$A = \begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \end{bmatrix} \qquad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \\ B_{41} & B_{42} \\ B_{51} & B_{52} \\ B_{61} & B_{62} \\ B_{71} & B_{72} \\ B_{81} & B_{82} \end{bmatrix} = ?$$

Fig. 7. **An example of the minimum cost control problem.** The stem network has 8 nodes and 2 external drivers. The objective is to determine B such that the control cost can be minimized while $tr(BGB^T) \leq M$.

$$\begin{aligned} &argmin_B \quad E(B) = tr\left[\left(\int_0^{t_f} e^{At} BB^T e^{A^T t} dt\right)^{-1} X_f\right] \\ &\text{s.t.} \qquad tr(BGB^T) \leq M \end{aligned} \qquad (43)$$

Suppose that the input matrix B is represented as $B = [b_1, ..., b_j, ..., b_M]$ where $b_i = [B_{1j}, ..., B_{ij}, ..., B_{Nj}]^T \in \mathbb{R}^{N\times 1}$ describes all the weights between node i and the $j-$th external control source. It is seen that, there is an energy capacity restriction on each external control source j as long as $g_j \neq 0$. This implies that the corresponding norm of the weight vector $\|b_j\|_2 = \sqrt{\sum_{i=1}^{8} B_{ij}^2}$ has an upper bound due to the bound condition $tr(BGB^T) \leq M$. A simple example is illustrated in Fig.7 with $M = 2$. In this case, the constraint $tr(BGB^T) \leq M$ is represented as

$$g_1 b_1^T b_1 + g_2 b_2^T b_2 = 2$$

When $g_1 = g_2 = 1$, the constraint reduces to the trace constraint in [13], which implies that each external control source has equal restriction. Suppose that (A, B) is controllable at initial time. Then, by applying WTPGM, $N(B) = (tr(BGB^T) - M)^2$ converges to zero very fast while $E(B)$ reduces relative slowly and converges to a local minimum

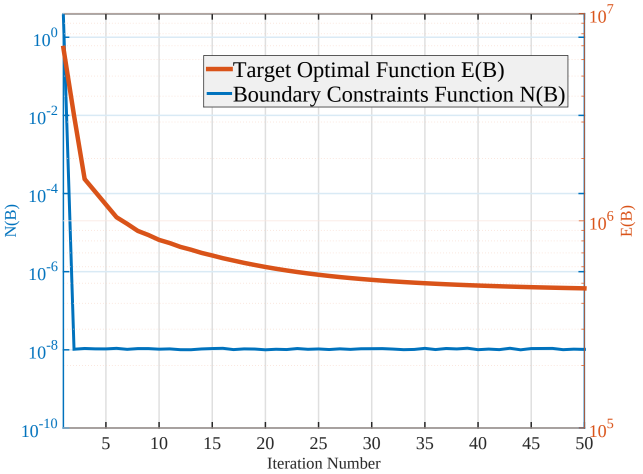


Fig. 8. **WTPGM for the minimum cost control problem.** $N(B)$ converges to zero very fast and $E(B)$ reduces continuously and converge to a local minimum value eventually.

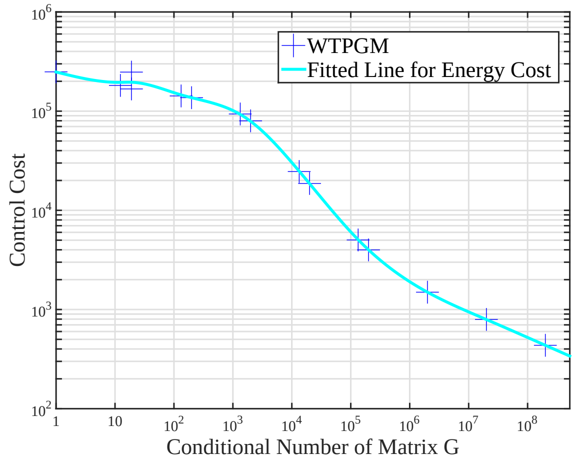


Fig. 9. The control cost with respect to $Cond(A)$.

value eventually. In Fig. 8, it is seen that $N(B)$ reduces fast and converges to around 10^{-8}, which is small enough to be treated as zero, and WTPGM would suppress $N(B)$ being below 10^{-8} (this precision is affected by η, as when a smaller η is chosen, $N(B)$ would converge to smaller value). After $N(B)$ has converged to zero rapidly, $E(B)$ reduces slowly and continuously, and it converges to a local minimum value eventually.

We would like to note that two important works [49] [50] were most recently published, which show network control is becoming a hot topic. Both of these two works consider network control theories and their potential applications in real physical systems, and they are similar to the case we consider here. A significant difference of this work is that we set a weight matrix G to take into account the constraints on the external control inputs in practical applications.

We further investigate how does the diagonal weight matrix G affect the obtained control cost $E(B^*)$. By denoting the condition number of G as $cond(G) = \|G^{-1}\|_F^2\|G\|_F^2$, it is found that there is a negative correlation between $cond(G)$ and $E(B^*)$. The result is presented in Fig.9. The reason is explained as follows. With the increase of $cond(G)$, the constriction on the control source corresponding to the smaller value of $\{g_1, g_2\}$ becomes smaller and smaller. In an ideal case, we consider $g_1 = 0$ and $g_2 = 2$, which implies that there is no more energy capacity restriction on the first external control source, i.e., the norm of the weight vector $\|b_1\|_2 = \sqrt{\sum_{i=1}^{8} B_{i1}^2}$ can be positive infinity. So $\|b_1\|_2$ can be sufficiently large and $u_1(t)$ can be sufficiently small while $b_{i1}u_1(t)$ for all i could be still finite. Thus $\int_0^{t_f} u_1^2(t)dt$ can be reduced by increasing $\|b_1\|_2$.

In [13], it is pointed out that the absolute value of link weight B_{ij} actually evaluates the importance of node i for the jth controller, and an *importance index* vector is defined as $r = \frac{[r_1\ \dots\ r_i\ \dots\ r_N]}{\max(r_1,\dots,r_i,\dots,r_N)}$ where $r_i = \sum_j |B_{ij}^*|$ is the sum of the absolute values of the $i-$th row of B^* for $i = 1, ..., N$. Therefore the importance index of node i reflects the relative importance of this node in achieving the minimum cost objective. The key node set, i.e., the top M nodes with the highest importance index values can be identified by WTPGM. Here we also investigate how the weight matrix G affect the distributions of key nodes on an elementary stem in Fig.10. As mentioned, when G is an identity matrix, the model is reduced to the case in our previous work [13], the probability that the set $\{node\ 1, node\ 5\}$ which divides the stem averagely to be selected as key node set is almost 1. This is consistent with our main finding in [13]. This observation is of great importance, as it may lead to heuristic algorithm design for the minimum-cost control of large scale of real life complex networks, which definitely deserve great attention for the future research.

However, as $cond(G)$ becomes larger and larger, the probability of selecting $\{node\ 1, node\ 5\}$ as the key node set slightly decreases while the probability of selecting $\{node\ 1, node\ 6\}$ gradually increases. This indicates that the key node is gradually moving to the tail of the stem as $node\ 1$ has to be selected for ensuring the controllability of the network. For the case $g_1 = 1$ and $g_2 = 1$, each of the two control source has equal importance and $\{node\ 1, node\ 5\}$ is selected as key nodes with a probability of almost one, and with each control source driving equal number of nodes. While as the conditional number $cond(G)$ increases, the control source with larger $\|b_i\|_2$ affects/drives more nodes and the other source only affects/drives the remaining nodes. This leads to the fact of the key node set slightly moving to the tail of the stem. In an ideal case of $g_1 = 0$ and $g_2 = 2$, i.e., there is no more energy constriction on the first control source, the norm of the weight $\|b_1\|_2 = \sqrt{\sum_{i=1}^{8} B_{i1}^2}$ could be infinity. And in this case, if we only connect $u_1(t)$ to $node$ 1, the control cost can be still keep in a low level.

As discussed in [13], we would also like to point out that for the non-convex optimization problem (1) or (2), the proposed WTPGM method, like any optimization method with a reasonably low complexity, can only guarantee converging to a local minimum. It is well known that through multiple rounds of experiments, a suboptimal solution can always be easily

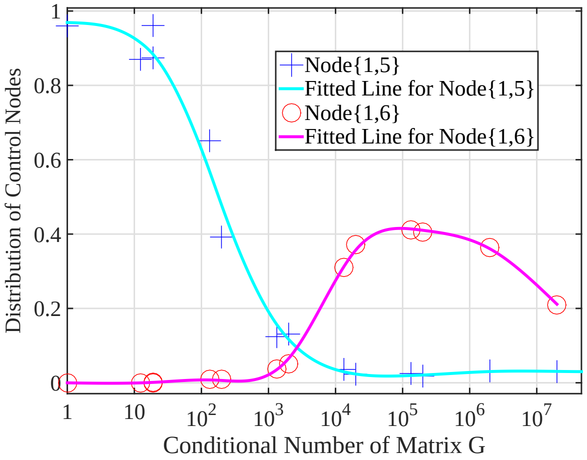


Fig. 10. Illustration how does the $Cond(A)$ affect the distribution of the key nodes.

obtained, and such a suboptimal solution steadily approaches the global optimum as the number of experimental implementations increases. However, an interesting observation based on our extensive implementations of WTPGM in this example starting with different initial B_0 matrices is that the solutions of different rounds of implementations typically lead to nearly the same control cost. Whether it means that the solution found by a single round implementation of WTPGM is steadily close to the global optimum requests further careful studies.

C. Applications in controlling directed networks by only evolving the connection strengths on a fixed network structure

In last subsection, we consider the *minimum cost control* of complex networks subjected to linear dynamics $\dot{x}(t) = Ax(t) + Bu(t)$ with selectable input matrix [13]. This implies that B is a matrix variable while the topology matrix A is fixed, Here we consider another case, the topology matrix A becomes a matrix variable while the input matrix B is fixed. In this case, this above problem can be formulated as the following matrix function optimization problem:

$$\begin{aligned} &argmin_A \qquad E(A) = tr\left[\left(\int_0^{t_f} e^{At}BB^Te^{A^Tt}dt\right)^{-1} X_f\right] \qquad (44)\\ &\text{s.t.} \quad\; tr(A^TA) = M \end{aligned}$$

where X_f is a constant matrix given by

$$X_f = \mathbb{E}[x_f x_f^T] = e^{At_f}X_0e^{A^Tt_f} \qquad (45)$$

The problem can be converted to find an optimal A^* to minimize the cost function $E(A)$ defined by $E(A) \triangleq \text{tr}[W_A^{-1}X_f]$ where $W_A = \int_0^{t_f} e^{At}BB^Te^{A^Tt}dt$ is the Gramian matrix. In practice, it is interesting and also useful to investigate how the connection strengths of a network evolve on a fixed network structure. For a network with topology matrix A in which A_{ij} describes the connection strength between node i and node j, we build a corresponding network structure matrix $\mathcal{A}$ as follow: $\mathcal{A}_{ij} = 1$ if $|A_{ij}| \ll \omega$ for a sufficiently small positive ω, otherwise, $\mathcal{A}_{ij} = 0$. Similar to the previous case, suppose that $(\mathcal{A}, B)$ is controllable at initial time. However, to guarantee that the network structure matrix $\mathcal{A}$ remains fixed in the iterative process of WTPGM, the gradient $\frac{\partial E(A)}{\partial A}$ needs to be replaced by $\frac{\partial E(A)}{\partial A} \circ \mathcal{A}$, where the '$\circ$' denotes the Hadamard (element-wise) product of two matrices. Note that Theorem 1 still holds when implementing WTPGM in this way as there is no requirement on specific form of the gradient.

Our experiment is on controlling a circle topology with $N = 6$ nodes and $M = 2$ external control sources in Fig.11(a). The initial topology matrix A is built based Fig.11(a) by setting $A_{ij} = 1$ if there is an edge from i to j, otherwise $A_{ij} = 0$. In this example, the above model is a special case of model (44) with G being an identity matrix. Obviously, WTPGM can be applied. Therefore, we implement WTPGM, with the replacement of the gradient $\frac{\partial E(A)}{\partial A}$ by $\frac{\partial E(A)}{\partial A} \circ \mathcal{A}$ at each iteration, in order to find out how the connection strength evolves when the control cost attains its minimum. Based on Lemma 1 and by introducing the *I-Chain Rule*, the derivative of $\frac{\partial E(A)}{\partial A}$ can be expressed as

$$\begin{aligned} &\frac{\partial tr(W_A^{-1}e^{At_f}X_0e^{A^Tt_f})}{\partial A_{ij}}\\ &= -\,\delta_{mn}\cdot\frac{\partial[W_A^{-1}]_{mu}}{\partial A_{ij}}\cdot[e^{At_f}]_{uz}\cdot[X_0]_{zc}\cdot[e^{A^Tt_f}]_{cn}\\ &\quad+\delta_{mn}\cdot[W_A^{-1}]_{mu}\cdot\frac{\partial[e^{At_f}]_{uz}}{\partial A_{ij}}\cdot[X_0]_{zc}\cdot[e^{A^Tt_f}]_{cn}\\ &\quad+\delta_{mn}\cdot[W_A^{-1}]_{mu}\cdot[e^{At_f}]_{uz}\cdot[X_0]_{zc}\cdot\frac{\partial[e^{A^Tt_f}]_{cn}}{\partial A_{ij}} \end{aligned} \qquad (46)$$

We have

$$\begin{aligned} &\tfrac{\partial[W_A^{-1}]_{mu}}{\partial A_{ij}}\\ &{=}{-}[W_A^{-1}]_{mf}\cdot[W_A^{-1}]_{ru}\cdot\tfrac{\partial\left[\int_0^{t_f} e^{At}BB^Te^{A^Tt}dt\right]_{fr}}{\partial A_{ij}}\\ &{=}{-}[W_A^{-1}]_{mf}\cdot[W_A^{-1}]_{ru}\cdot\textstyle\int_0^{t_f}\tfrac{\partial[e^{At}]_{fs}}{\partial A_{ij}}[BB^T]_{sd}[e^{A^Tt}]_{dr}dt\\ &\quad-[W_A^{-1}]_{mf}\cdot[W_A^{-1}]_{ru}\cdot\textstyle\int_0^{t_f}[e^{At}]_{fs}[BB^T]_{sd}\tfrac{\partial[e^{A^Tt}]_{dr}}{\partial A_{ij}}dt \end{aligned} \qquad (47)$$

From the example given in the end of Section 4, we have

$$\frac{\partial[e^{At}]_{kl}}{\partial A_{ij}} = \sum_{n_0=1}^{N-1}\alpha_{n_0}(t)\left(\sum_{k_0=0}^{n_0-1}\left([A^{k_0}]_{ki}[A^{n_0-1-k_0}]_{jl}\right)\right) \qquad (48)$$

and

$$\frac{\partial[e^{A^Tt}]_{kl}}{\partial A_{ij}} = \sum_{n_0=1}^{N-1}\alpha_{n_0}(t)\left(\sum_{k_0=0}^{n_0-1}\left([A^{k_0}]_{li}[A^{n_0-1-k_0}]_{jk}\right)\right) \qquad (49)$$

Substituting (48) and (49) into (47), we obtain

$$\begin{aligned} &\frac{\partial[W_A^{-1}]_{mu}}{\partial A_{ij}}\\ &= -\int_0^{t_f}\sum_{n_0=1}^{N-1}\alpha_{n_0}(t)\Bigg(\sum_{k_0=0}^{n_0-1}\bigg([W_A^{-1}A^{k_0}]_{mi}[A^{n_0-1-k_0}BB^Te^{A^Tt}W_A^{-1}]_{ju}\bigg)\\ &\quad-\int_0^{t_f}\sum_{n_0=1}^{N-1}\alpha_{n_0}(t)\Bigg(\sum_{k_0=0}^{n_0-1}\bigg(\,\left[W_A^{-1}e^{At}BB^T(A^{n_0-1-k_0})^T\right]_{mj} \end{aligned}$$

$$\cdot [(A^{k_0})^T W_A^{-1}]_{iu} \Big) \Big) dt$$

Combining (47)-(49), we rewrite (46) in the form of $[\bullet]_{ij}$ by rearranging the orders of the terms based on *I-Chain Rule*, we finally obtain that

$$\begin{aligned} & \frac{\partial tr(W_A^{-1} e^{At_f} X_0 e^{A^T t_f})}{\partial A} \\ = & -2 \int_0^{t_f} \sum_{n_0=1}^{N-1} \alpha_{n_0}(t) \sum_{k_0=0}^{n_0-1} (A^T)^{k_0} W_A^{-1} e^{At_f} X_0 e^{A^T t_f} \\ & \cdot W_A^{-1} e^{At} B B^T (A^T)^{n_0-1-k_0} dt \\ & + 2 \sum_{n_0=1}^{N-1} \alpha_{n_0}(t_f) \sum_{k_0=0}^{n_0-1} (A^T)^{k_0} W_A^{-1} e^{At_f} X_0 (A^T)^{n_0-1-k_0} \end{aligned}$$

From our experiment, it is observed that the link weights evolve depending on the locations of the control sources. Technically, there are 4 different ways to place these two control sources. Fig.11 (b) and (c) illustrate two examples. In Fig.11 (b), nodes 1 and 5 are connected with the control sources (stars) respectively. The optimal topology indicates that the strength of edge $\{4 \rightarrow 5\}$ and $\{1 \rightarrow 6\}$ are very small (dash lines), which implies two elementary stems are formed when ignoring the extremely small connection strength. Fig.11 (c) shows similar results. Moreover, by applying WTPGM, the control cost of in Fig.11 (c) is the smallest among the 4 different control sources placements. It can be concluded that when the control sources are evenly allocated, the system can be considered as two identical subsystems and the control cost attains its minimum. This finding also coincides with the main finding in [13] for the case of selectable input matrix.

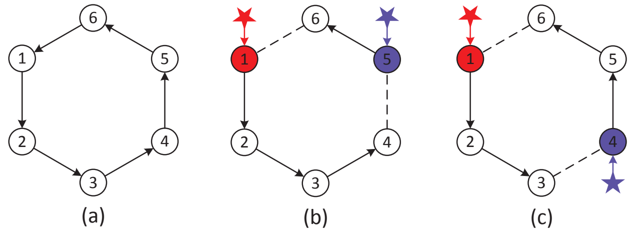


Fig. 11. **The evolution of an elementary cycle topology with $N = 6$, $M = 2$ when the control sources are pre-allocated.** (a) The cycle topology. (b)When nodes $\{1, 5\}$ are connected to external control sources (represented by stars), it is observed that for the obtained topology A^*, the link weights pointing to nodes $\{1\}$ and $\{5\}$ are extremely small (represented by dash lines). (c) When nodes $\{1, 4\}$ is connected to external control sources, the control cost attains its minimum.

D. Dimension reduction

Besides the applications in network control, we also show the effectiveness of our methods in dimension reduction which is commonly applied in various learning problems. Recent decades have seen the successful application of dimension reduction in widespread areas sufferring from dimension curse including engineering, biology, sensing and economics[51]. To manipulate high-dimensional data like the writing, sounds, images and videos, several algorithms have been developed which can be divided into linear and nonlinear types. The linear type can be regarded as a projective operation transforming high-dimensional data into low-dimensional space. Principal Component Analysis (PCA) has proven its remarkable ability to reduce data dimension without losing most data information[52]. Linear discriminant analysis (LDA) projects data into a low-dimensional space which has the minimum-in-cluster-distance and maximum-between-cluster-distance[53]. The second class do the nonlinear dimension reduction (NLDR) by providing a mapping from high dimension to low dimension[54] including ISOMAP[55], Locally linear embedding (LLE)[56], Laplacian eigenmaps[57] and other methods.

Generally, many algorithms in dimension reduction are under the framework called *Graph Embedding*[58], which aim to find a transformation matrix $W \in \mathbb{R}^{N \times M}$ that transforms the high-dimensional data vector $x \in \mathbb{R}^N$ into the low-dimensional data $y \in \mathbb{R}^M$ $(M < N)$ as $y = W^T x$.

The graph embedding is a framework to generalize most algorithms which are proposed to find such W. The graph embedding defines an undirected weighted graph $\{X, S\}$ with vertex set X representing the original data $\{x_i | x_i \in \mathbb{R}^N\}_{i=1}^{\mathbb{N}}$ and similarity matrix $S \in \mathbb{R}^{\mathbb{N} \times \mathbb{N}}$. The element s_{ij} of S measures the similarity of the vertex pair, i.e. data pair x_i and x_j, which indicates S is a real symmetric matrix. Then the dimension reduction problem can be transformed into finding a y^* such that

$$y^* = arg \min_{y^T G y = c} y^T L y,$$

where c is a constant, G is a constraint matrix which is similar to the weight matrix in this paper, $L = D - S$ is the Laplacian matrix and D is a diagonal matrix with $D_{ii} = \sum_{j \neq i} s_{ij}$. Along with the linearization of the graph embedding, this objective function can be further converted to find a vector w^* such that

$$w^* = arg \min_{w^T X G X^T w = c} w^T X L X^T w. \tag{50}$$

Under this general framework, many algorithms of dimension reduction can be essentially regarded as solving a trace ratio optimization problem such as LDA and MFA (Marginal Fisher Analysis). Take LDA[59] as an example. Rather than finding a projection matrix W projecting data x onto a subspace which can describe x best, LDA aims to discriminate x among classes in a subspace just as its name implies. For this purpose, LDA defines two matrices to measure the metric among classes. One is the within-class (also named as intra-class) scatter matrix

$$S_{\mathbf{W}} = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (x_i^j - \mu_j)(x_i^j - \mu_j)^T,$$

where c is the number of classes, N_j is the number of data samples in class j, μ_j is the mean of class j and x_i^j is the ith sample in class j. Another is the between-class (also named as inter-class) scatter matrix

$$S_{\mathcal{B}} = \sum_{j=1}^{c} (\mu_j - \mu)(\mu_j - \mu)^T,$$

where μ is the mean of all classes. As the goal of LDA is to search for a subspace where the distance of data among different classes is furthest and the distance of data in the same class is nearest, $S_{\mathcal{W}}$ should be minimized and $S_{\mathcal{B}}$ should be maximized. A straightforward way is to maximize the ratio of $\frac{||S_{\mathbf{B}}||}{||S_{\mathcal{W}}||}$. Thus, equation (50) is converted to

$$w^* = arg \min_w \frac{w^T S_{\mathcal{W}} w}{w^T S_{\mathcal{B}} w}, \tag{51}$$

which can be further formulated in trace ratio form

$$W = arg \max_{W^T G W = I_M} \frac{tr(W^T S_p W)}{tr(W^T S_l W)}$$

where S_p and S_l are defined in [60], and G is a positive semi-definite matrix. Thus, the dimension reduction problem can be rewritten as

$$\begin{array}{ll} argmin_B & E(B) \\ \text{s.t.} & B^T G B = I_M \end{array} \tag{52}$$

by setting $E(W) = -\frac{tr(W^T S_p W)}{tr(W^T S_l W)}$ and replacing W with B.

To solve the above trace-ratio problem, we can use the proposed WOPGM based on constructed randomly semi definite matrices $\mathcal{A}$ and $\mathcal{C}$, and positive definite matrix G. After applying the algorithm, the result is shown in Fig. 12. The blue line represents the value of the boundary condition function $N(B)$ which rapidly decreases to the magnitude of 10^{-5} as small as the step lengthη. Limited to the computational accuracy in MATLAB, $N(B)$ is oscillating around 10^{-5} and never larger than 10^{-4}. Compared with the step length η, $N(B)$ becomes sufficiently small and can be considered as zero. After the constraint $N(B)$ approaches zero, the cost function $E(B)$ starts to strictly decrease as the red line showing, which is consistent with the theoretical proof. We would also like to point out that the motivation of the proposed method compared method is fundamentally different from existing methods to deal with the dimension reduction problems. For example in [60], the iterative algorithm for the Trace Ratio optimization problem (abbreviated to ITR) requires eigenvalue decomposition method to compute the eigenvalues and corresponding eigenvectors of intermediate symmetric matrix iteratively). It is known that derive the eigenpair of real symmetric matrices have $O(N^3)$ time complexity at each iteration, which indicates that ITR has $O(N^3)$ complexity as the iteration number is independent of N. For WOPGM, the iteration only contains the multiplication of the matrices and also has $O(N^3)$ complexity. However, when both the WOPGM and the ITR are applied to the trace ratio problem generalized from dimension reduction in (52), it is observed that the iteration number of WOPGM is smaller than that of ITR. This is easily explicable as the iteration in WOPGM is more intuitive and simpler. When we fix the iteration number, WOPGM usually obtain lower values than ITR.

Last but not lease, we discuss the selection of λ_k for WOPGM. In the proof of the theorem 2, $N(\hat{B}_{k+1}) - N(B_k) = 4\eta(a + b\lambda_k) \leq 0$ requires that λ_k should meet the inequations in Tab. I at least. As we generalize the optimization problem from $B^T B = I_M$ to $B^T G B = I_M$, λ_k plays a key role in the convergency of $N(B)$ and $E(B)$.

For λ_k, when $N(B)$ is far from zero, λ_k has a great effect on the decrease of $N(B)$. As $N(B)$ converges to zero, the update iteration formula itself of this line search method would ensure that $N(B)$ is always approaching and near zero with deviation which depends on how small η is. When $N(B)$ has converged to zero, Theorem 2 guarantees that $E(B)$ converges to a local optimized value. Thus, the selection of λ_k mainly influences the convergency of $N(B)$. For example, if we set

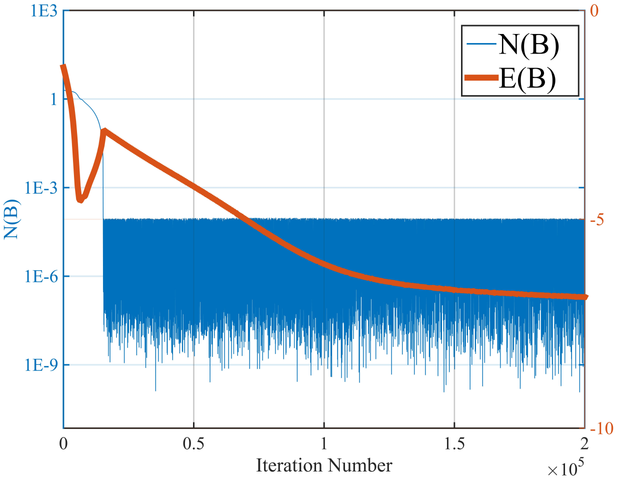


Fig. 12. **WOPGM for the dimension reduction trace-ratio problem.** When $N(B)$ is far away from zero, $E(B)$ may fluctuate some time while after $N(B)$ approaches to zero, $E(B)$ reduces strictly and converge to a local minimum value eventually.

$$\lambda_k = \frac{-1-a}{b} - |\frac{-1-a}{b}|, b > 0$$

or

$$\lambda_k = \frac{-1-a}{b} + |\frac{-1-a}{b}|, b < 0$$

$N(B)$ will converge to zero faster than that if we set

$$\lambda_k = \frac{-1-a}{b} - |\frac{-1-a}{2b}|, b > 0$$

or

$$\lambda_k = \frac{-1-a}{b} + |\frac{-1-a}{2b}|, b < 0$$

However, as $N(B)$ is near zero and smaller than η, λ_k should be selected more wisely to restrain the increase of $N(B)$, e.g.

$$\lambda_k = -\frac{1}{4\eta}, b > 0 \ or \ \lambda_k = -\frac{1}{4\eta}, b < 0.$$

Generally, the constraints in Tab. I are the necessary requirements of λ_k to ensure the convergency of $N(B)$. And the proper selection of λ_k can accelerate the convergency of $N(B)$, suppress the increase of $N(B)$ and finally result in the better convergence of $E(B)$.

VII. Discussions

It is well known that matrix function optimization problems are much more general than vector function optimization problems. Because in many science and engineering problems, the variables that affect the objective function are described by matrices instead of vectors, where each column of the matrix variable has its physical meaning and cannot be flatten into a vector form. In this work, we investigated matrix optimization problems under weighted boundary constraints, which has meaningful physical insights since many real life applications are under such constraints. By introducing *I-Chain rule* to obtain the gradient of the objective matrix function, two algorithms (WTPGM and WOPGM) are advanced to solve the problems. The convergence of both WTPGM and WOPGM can be guaranteed. Our method has also been illustrated and validated in different applications not only in network control but also in other learning problems, with the simulation results showing its effectiveness. We believe that our work opens the door of matrix function optimization problem and its wide and extensive applications in science and engineering.

References

[1] D. Ward, "Directional derivative calculus and optimality conditions in nonsmooth mathematical programming," *Journal of Information and Optimization Sciences*, vol. 10, no. 1, pp. 81–96, 1989.

[2] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *icip*. IEEE, 1995, p. 3444.

[3] R. Aris, *Vectors, tensors and the basic equations of fluid mechanics*. Courier Corporation, 2012.

[4] R. E. Pepper, S. V. Chasteen, S. J. Pollock, and K. K. Perkins, "Observations on student difficulties with mathematics in upper-division electricity and magnetism," *Physical Review Special Topics-Physics Education Research*, vol. 8, no. 1, p. 010111, 2012.

[5] J. E. Marsden and T. J. Hughes, *Mathematical foundations of elasticity*. Courier Corporation, 1994.

[6] J. P. Perdew, R. G. Parr, M. Levy, and J. L. Balduz Jr, "Density-functional theory for fractional particle number: derivative discontinuities of the energy," *Physical Review Letters*, vol. 49, no. 23, p. 1691, 1982.

[7] G. Arfken and H. J. Weber, "Mathematical methods for physicists academic," *New York*, vol. 19852, p. 309, 1985.

[8] E. W. Weisstein, "Curl," 2002.

[9] G. Li, W. Hu, G. Xiao, L. Deng, P. Tang, J. Pei, and L. Shi, "Minimum-cost control of complex networks," *New Journal of Physics*, vol. 18, no. 1, p. 013012, 2015.

[10] J. Snyman, "Practical mathematical optimization: Basic theory and gradient-based algorithms," 2005.

[11] J. A. Schouten, *Tensor analysis for physicists*. Courier Corporation, 1954.

[12] G. H. Katzin, J. Levine, and W. R. Davis, "Curvature collineations: A fundamental symmetry property of the space-times of general relativity defined by the vanishing lie derivative of the riemann curvature tensor," *Journal of Mathematical Physics*, vol. 10, no. 4, pp. 617–629, 1969.

[13] G. Li, P. Tang, C. Wen, and Z. Meng, "Boundary constraints for minimum cost control of directed networks," *IEEE transactions on cybernetics*, dOI: 10.1109/TCYB.2016.2602358, 2017.

[14] E. W. Grafarend and R.-J. You, "Fourth order taylor–kármán structured covariance tensor for gravity gradient predictions by means of the hankel transformation," *GEM-International Journal on Geomathematics*, vol. 6, no. 2, pp. 319–342, 2015.

[15] S. Ge, M. Han, and X. Hong, "A fully automatic ocular artifact removal from eeg based on fourth-order tensor method," *Biomedical Engineering Letters*, vol. 4, no. 1, pp. 55–63, 2014.

[16] K. M. Abadir and J. R. Magnus, *Matrix algebra*. Cambridge University Press, 2005, vol. 1.

[17] J. R. Magnus, H. Neudecker *et al.*, "Matrix differential calculus with applications in statistics and econometrics," 1995.

[18] A. Hjørungnes, *Complex-valued matrix derivatives: with applications in signal processing and communications*. Cambridge University Press, 2011.

[19] R. J. Barnes, "Matrix differentiation," *Springs Journal*, 2006.

[20] C. L. Lawson, "Hanson rj solving least squares problems," 1974.

[21] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

[22] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 635–650, 2002.

[23] H. A. Kiers, "Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems," *Computational statistics & data analysis*, vol. 41, no. 1, pp. 157–170, 2002.

[24] D. Xu, S. Yan, D. Tao, S. Lin, and H.-J. Zhang, "Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval," *Image Processing, IEEE Transactions on*, vol. 16, no. 11, pp. 2811–2821, 2007.

[25] J. Yu, X. Gao, D. Tao, X. Li, and K. Zhang, "A unified learning framework for single image super-resolution," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 4, pp. 780–792, 2014.

[26] Y. Liu, F. Shang, L. Jiao, J. Cheng, and H. Cheng, "Trace norm regularized candecomp/parafac decomposition with missing data," *IEEE transactions on cybernetics*, vol. 45, no. 11, pp. 2437–2448, 2015.

[27] L. Yang, X. Cao, D. Jin, X. Wang, and D. Meng, "A unified semi-supervised community detection framework using latent space graph regularization," *IEEE transactions on cybernetics*, vol. 45, no. 11, pp. 2585–2598, 2015.

[28] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[29] A. C. Bovik, *Handbook of image and video processing*. Academic press, 2010.

[30] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi, "Iterative water-filling for gaussian vector multiple-access channels," *Information Theory, IEEE Transactions on*, vol. 50, no. 1, pp. 145–152, 2004.

[31] S. Ye and R. S. Blum, "Optimized signaling for mimo interference systems with feedback," *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2839–2848, 2003.

[32] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[33] S. Zhang and H. Zhao, "Normalized modularity optimization method for community identification with degree adjustment," *Physical Review E*, vol. 88, no. 5, p. 052802, 2013.

[34] C. Cai, Z. Wang, J. Xu, X. Liu, and F. E. Alsaadi, "An integrated approach to global synchronization and state estimation for nonlinear singularly perturbed complex networks," *IEEE transactions on cybernetics*, vol. 45, no. 8, pp. 1597–1609, 2015.

[35] Z. Meng, G. Shi, K. H. Johansson, M. Cao, and Y. Hong, "Behaviors of networks with antagonistic interactions and switching topologies," *Automatica*, vol. 73, pp. 110–116, 2016.

[36] G. Yan, J. Ren, Y.-C. Lai, C.-H. Lai, and B. Li, "Controlling complex networks: How much energy is needed?" *Physical review letters*, vol. 108, no. 21, p. 218703, 2012.

[37] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 9, pp. 1299–1313, 2009.

[38] E. Y. Chan and D.-Y. Yeung, "A convex formulation of modularity maximization for community detection," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI), Barcelona, Spain*, 2011.

[39] G. Li, P. Tang, Z. Meng, C. Wen, P. Jing, and L. Shi, "Matrix function optimization and its applications in learning problems," *IEEE Transactions on Neural Networks and Leaning Systems*.

[40] K. Petersen and M. Pedersen, "The matrix cookbook. technical university of denmark," *Technical Manual*, 2008.

[41] P. Lancaster and M. Tismenetsky, *The theory of matrices: with applications*. Elsevier, 1985.

[42] J. Gilbert and L. Gilbert, *Linear Algebra and Matrix Theory*. Academic Press, 2014.

[43] A. M. Bloch, P. E. Crouch, and A. K. Sanyal, "A variational problem on stiefel manifolds," *Nonlinearity*, vol. 19, no. 10, p. 2247, 2006.

[44] M. Harada, "Extremal type i-codes and-frames of odd unimodular lattices," *Information Theory, IEEE Transactions on*, vol. 61, no. 1, pp. 72–81, 2015.

[45] P. Găvruţa, "On the duality of fusion frames," *Journal of Mathematical Analysis and Applications*, vol. 333, no. 2, pp. 871–879, 2007.

[46] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[47] R.-A. Pitaval, W. Dai, and O. Tirkkonen, "Convergence of gradient descent for low-rank matrix approximation," *Information Theory, IEEE Transactions on*, vol. 61, no. 8, pp. 4451–4457, 2015.

[48] J. Dattorro, *Convex optimization & Euclidean distance geometry*. Lulu. com, 2010.

[49] G. Yan, P. E. Vértes, E. K. Towlson, Y. L. Chew, D. S. Walker, W. R. Schafer, and A.-L. Barabási, "Network control principles predict neuron function in the caenorhabditis elegans connectome," *Nature*, vol. 550, no. 7677, p. 519, 2017.

[50] A. Li, S. P. Cornelius, Y.-Y. Liu, L. Wang, and A.-L. Barabási, "The fundamental advantages of temporal networks," *Science*, vol. 358, no. 6366, pp. 1042–1046, 2017.

[51] I. K. Fodor, "A survey of dimension reduction techniques," 2002.

[52] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, 2006.

[53] B. Scholkopft and K.-R. Mullert, "Fisher discriminant analysis with kernels," *Neural networks for signal processing IX*, vol. 1, no. 1, p. 1, 1999.

[54] J. A. Lee and M. Verleysen, *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[55] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[56] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[57] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *NIPS*, vol. 14, 2001, pp. 585–591.

[58] S. Yan, D. Xu, B. Zhang, and H.-J. Zhang, "Graph embedding: A general framework for dimensionality reduction," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. Ieee, 2005, pp. 830–837.

[59] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[60] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang, "Trace ratio vs. ratio trace for dimensionality reduction," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.